

## **NOTA INFORMATIVA EN RELACIÓN CON LA PUBLICACIÓN DE LOS TEMARIOS DE LOS PROCESOS SELECTIVOS CONVOCADOS POR LA XUNTA DE GALICIA EN EL DIARIO OFICIAL DE GALICIA Nº 142 DE 26 /07/11**

El Diario Oficial de Galicia nº 142, de 26 de julio de 2011, publica distintas órdenes de la Consellería de Facenda por las que se convocan diferentes procesos selectivos para el ingreso en la Administración autonómica de Galicia.

Cumpliendo con el compromiso adquirido, la EGAP, continúa con la publicación de los temarios correspondientes a los distintos procesos selectivos convocados formalmente.

Teniendo en cuenta, por una parte, el volumen y complejidad en la elaboración de un material didáctico que sirva de referencia básica y, por otra, el interés de la EGAP para que los posibles usuarios dispongan a la mayor brevedad posible de dicho material, la publicación del mismo en la página web de la Escuela (<http://egap.xunta.es>), se irá produciendo de la siguiente manera:

- 1) **Legislación**, actualizada y consolidada a la fecha de publicación en el DOG del nombramiento del tribunal del proceso (Base II.1 de la convocatoria<sup>1</sup>), correspondiente a los procesos selectivos para el ingreso en el cuerpo superior de la Administración de la Xunta de Galicia, subgrupo A1; cuerpo de gestión de la Administración de la Xunta de Galicia, subgrupo A2; cuerpo superior de la Administración de la Xunta de Galicia, subgrupo A1, escala de sistemas y tecnología de la información; cuerpo de gestión de la Administración de la Xunta de Galicia, subgrupo A2, escala de gestión y sistemas de información y cuerpo auxiliar de la Xunta de Galicia, subgrupo C2.

La fecha prevista para dicha publicación es de 5 de agosto de 2011.

- 2) **Temarios específicos**, se irán publicando en la página web de la Escuela, a medida que los procesos de elaboración y revisión vayan concluyendo.

Sin perjuicio de su publicación en las dos lenguas oficiales y a fin de facilitar a la mayor brevedad posible este material a los usuarios, los temas se irán publicando en la web en el idioma originalmente utilizado por cada uno de los autores.

Para mayor información se pueden poner en contacto con el servicio de Estudios y Publicaciones a través del correo electrónico [temarios.egap@xunta.es](mailto:temarios.egap@xunta.es), y teléfono 881 997 251.

La Escuela reitera que los temarios por ella facilitados no tienen carácter oficial, por lo que en ningún caso vincularán a los opositores o a los tribunales; sino que se trata de instrumentos complementarios que servirán de apoyo y ayuda como textos de referencia pero nunca de forma exclusiva y excluyente.

Santiago de Compostela, 4 de agosto de 2011

---

<sup>1</sup> II. Proceso selectivo.

II.1. Procedimiento de oposición.

(...) Se tendrán en cuenta las normas de derecho positivo relacionadas con el contenido del programa que en el momento de publicación en el DOG del nombramiento del tribunal del proceso cuenten con publicación oficial en el boletín o diario correspondiente.

# **GESTIÓN DE LAS TIC**

**10. DIRECCIÓN Y GESTIÓN DE  
PROYECTOS. GESTIÓN DE LA  
INTEGRACIÓN. EL PLAN  
GENERAL DEL PROYECTO.  
GESTIÓN DEL ALCANCE.  
GESTIÓN DEL COSTE.  
PRESUPUESTOS. GESTIÓN DEL  
TIEMPO. TÉCNICAS DE  
PLANIFICACIÓN. GESTIÓN DE  
LA CALIDAD. PLAN DE  
CALIDAD. CAPACIDADES DEL  
JEFE DE PROYECTO. GESTIÓN  
DE LAS COMUNICACIONES.  
GESTIÓN DEL RIESGO.  
CONTINGENCIAS. GESTIÓN DE  
LA SUBCONTRATACIÓN Y  
ADQUISICIONES.**

**Tema 10. Dirección y gestión de proyectos. Gestión de la integración. El plan general del proyecto. Gestión del alcance. Gestión del costo. Presupuestos. Gestión del tiempo. Técnicas de planificación. Gestión de la calidad. Plan de calidad. Capacidades del jefe de proyecto. Gestión de las comunicaciones. Gestión del riesgo. Contingencias. Gestión de la subcontratación y adquisiciones.**

## **INDICE**

- 10.1 Dirección y gestión de proyectos
  - 10.1.1 Introducción
  - 10.1.2 Project Management Institute (PMI®)
  - 10.1.3 El Project Management Body of Knowledge (PMBOK®)
    - 10.1.3.1 Introducción
    - 10.1.3.2 Relación del PMBOK® con la dirección de las TIC
    - 10.1.3.3 Estructura del PMBOK®
- 10.2 Gestión de la integración del proyecto
  - 10.2.1 El plan general del proyecto
- 10.3 Gestión del alcance del proyecto
- 10.4 Gestión del tiempo del proyecto
  - 10.4.1 Técnicas de planificación
- 10.5 Gestión de los costos del proyecto
  - 10.5.1 Presupuestos
- 10.6 Gestión de la calidad del proyecto
  - 10.6.1 Plan de calidad
- 10.7 Capacidades del jefe de proyecto.
  - 10.7.1 Gestión de los Recursos Humanos del Proyecto
  - 10.7.2 Capacidades del jefe de proyecto
- 10.8 Gestión de las comunicaciones del proyecto
- 10.9 Gestión de los riesgos del proyecto
  - 10.9.1 Contingencias
- 10.10 Gestión de las adquisiciones del proyecto



## **10.1 DIRECCIÓN Y GESTIÓN DE PROYECTOS**

### **10.1.1 INTRODUCCIÓN**

Un proyecto es una actividad temporal diseñada para producir un único producto, servicio o resultado. Un proyecto tiene un comienzo y un final, así como un alcance y recursos definidos.

Un proyecto es único en el sentido de que no es una operación repetitiva, sino un conjunto específico de operaciones diseñadas para lograr un objetivo. El equipo de proyecto incluye gente que normalmente no trabaja junta, y en muchos casos pertenecen a distintas organizaciones y países.

El desarrollo de software, la construcción de un puente o edificio, el esfuerzo de reconstrucción después de una catástrofe natural, todo son proyectos, y deben ser gestionados de forma experta para ser completados a tiempo, según lo presupuestado, y con la calidad exigida por las organizaciones.

La gestión de proyectos es la aplicación de conocimientos, habilidades y técnicas para ejecutar los proyectos de forma eficaz y eficiente. Es algo estratégico para las organizaciones, capacitándolas para alinear los resultados del proyecto a los objetivos del negocio - y por ello siendo más competitivas en sus mercados - .

### **10.1.2 PROJECT MANAGEMENT INSTITUTE (PMI®)**

PMI es una de las asociaciones profesionales más grandes del mundo, con medio millón de miembros en más de 185 países. Es una organización sin ánimo de lucro que intenta mejorar la profesión de la gestión de proyectos por medio de estándares y certificaciones reconocidas

globalmente, comunidades colaborativas, un extenso programa de investigación, y oportunidades de desarrollo profesional.

### Antecedentes

PMI Internacional se fundó en 1969 con socios voluntarios. Durante los años setenta PMI se desarrolló principalmente en el campo de la ingeniería, mientras el mundo de los negocios desarrollaba sus proyectos a través de especialistas de la misma empresa y formaban grupos de trabajo llamados “Task Force”. En los años ochenta, el mundo de los negocios comenzó gradualmente a dirigir sus esfuerzos por proyectos.

Durante este tiempo el PMI, a través del comité de estándares y colaboradores (entre ellos empresas, universidades, asociaciones de profesionales, especialistas y consultores en proyectos) realizó el estudio, evaluación y revisión de los estándares generalmente aceptados a nivel internacional, dando como resultado los estándares que representan el cuerpo de conocimientos de la Dirección de Proyectos, cuyo título original es “Project Management Body of Knowledge” (PMBOK). En 1987 se publicó su primera edición.

### Estándares

Desde 1987, el PMI se ha encargado de investigar, recopilar y publicar las buenas prácticas generalmente aceptadas para la mayoría de los proyectos. Desde entonces, ha publicado 14 libros de estándares. Uno de ellos, el PMBOK, tiene en circulación más de 2.000.000 de ejemplares. Se tiene acceso a estos estándares como socio del PMI®.

### Misión

La misión del PMI es servir a su comunidad de asociados y profesionales interesados, desarrollando el arte de dirigir y llevar a la práctica la Dirección de Proyectos como disciplina profesional.

### Objetivos

Los principales objetivos del PMI son:

- Promover la dirección de proyectos.
- Compartir la experiencia internacional a través del desarrollo de profesionales.
- Desarrollar calidad en los recursos humanos para la dirección de proyectos.
- Compartir los conocimientos generalmente aceptados que dan reconocimiento a la profesión.
- Consolidar estándares internacionales
- Certificación de profesionales en proyectos reconocidos a nivel mundial.

### **10.1.3 EL PROJECT MANAGEMENT BODY OF KNOWLEDGE (PMBOK)**

#### **10.1.3.1 INTRODUCCIÓN**

El Project Management Body of Knowledge (PMBOK) es un término que describe la suma de los conocimientos involucrados en la profesión de la administración de proyectos.

El conocimiento y las prácticas descritas en el PMBOK son aplicables a la mayoría de los proyectos. Sin embargo, el equipo administrador del proyecto es siempre el responsable de determinar lo que es apropiado para cada proyecto.

PMBOK provee la terminología común de la administración de proyectos.

PMBOK describe los métodos y prácticas que deben tenerse en consideración desde que se inicia un proyecto hasta su finalización. La aplicación de éstas prácticas permitirá llevar una buena gestión del

proyecto y mantener un mayor control, permitiendo al Project Manager y a su equipo realizar proyectos de manera eficaz y eficiente (en alcance, tiempo, coste), así como asegurar la calidad y transparencia a lo largo de toda la vida del proyecto

La guía del PMBOK es integrada mediante un proceso de desarrollo de normas por consenso voluntario. Este proceso reúne a voluntarios y/o trata de obtener las opiniones de personas que tienen interés en el tema cubierto por esta publicación. Aunque PMI administra el proceso y establece reglas para promover la equidad en el desarrollo del consenso, no redacta el documento y no prueba, ni evalúa, ni verifica de manera independiente la exactitud o integridad de ninguna información ni la solidez de ningún juicio contenido en ella.

### Finalidad del PMBOK

La finalidad principal del PMBOK es identificar, concentrar y publicar las mejores prácticas generalmente aceptadas en la Dirección de Proyectos.

Generalmente aceptadas se refiere a que los conocimientos y las prácticas descritos son aplicables a la mayoría de los proyectos, la mayor parte del tiempo, y que existe un amplio consenso sobre su valor y utilidad.

Mejores prácticas se refiere a que existe un acuerdo general en que la correcta aplicación de estas habilidades, herramientas y técnicas puede aumentar las posibilidades de éxito de una amplia variedad de proyectos diferentes.

### ¿Cómo se organiza el PMBOK?

La base del PMBOK son los procesos, los cuales se clasifican por grupos y áreas de conocimiento:

- Grupos de Procesos: modo lógico de agrupar los procesos de dirección de proyectos, necesarios para cualquier proyecto, con dependencias entre ellos, y que se llevan a cabo en la misma secuencia siempre. Son los siguientes: Iniciación, Planificación, Ejecución, Seguimiento y Control y Cierre
- Áreas de Conocimiento: categoría que agrupa elementos en común. Son 9 en total: Integración, Alcance, Tiempo, Coste, Calidad, Comunicaciones, Recursos Humanos, Riesgo y Adquisiciones

### ¿A quién está dirigido el PMBOK?

Los conocimientos contenidos en el PMBOK proporcionan una referencia internacional para cualquiera que esté interesado en la profesión de la Dirección de Proyectos. Entre ellos se pueden mencionar: Altos ejecutivos, Gerentes de programa y gerentes de directores de proyectos, Directores del proyecto y otros miembros del equipo del proyecto, Miembros de una oficina de gestión de proyectos, Clientes y otros interesados, Consultores Formadores, Empresas , etc.

### Fortalezas de PMBOK

- ✓ La guía de PMBOK es un marco y un estándar.
- ✓ Está orientada a procesos.
- ✓ Indica el conocimiento necesario para manejar el ciclo vital de cualquier proyecto, programa y portafolio a través de sus procesos.
- ✓ Define para cada proceso sus entradas, herramientas, técnicas y reportes necesarios.
- ✓ Define un cuerpo de conocimiento en el que cualquier industria pueda construir las mejores prácticas específicas para su área de aplicación

### Limitaciones de PMBOK

- ✓ Complejo para los pequeños proyectos
- ✓ Tiene que ser adaptado a la industria del área de aplicación, tamaño y alcance del proyecto, el tiempo y el presupuesto y las exigencias de calidad.

### ¿Qué necesidades satisface PMBOK?

Por lo que respecta a las TI, PMBOK aporta principalmente elementos para satisfacer las necesidades de administración de proyectos de TI.

Los proyectos son una forma de organizar las actividades que no pueden ser tratadas dentro de los límites operativos de la organización. Así mismo, los proyectos se utilizan para lograr los objetivos definidos en el plan estratégico de la empresa, independientemente si el proyecto es administrado por la propia organización, o si corresponde a un proveedor de servicios externo.

Por lo anterior, las mejores prácticas integradas en PMBOK ayudan al departamento de TI a lograr los resultados esperados de los proyectos informáticos de forma más efectiva, garantizando que éstos sirven como inductores para la consecución de los objetivos definidos en el plan estratégico.

### **10.1.3.2 RELACIÓN DEL PMBOK CON LA DIRECCIÓN DE LAS TIC**

En los siguientes apartados se analiza la relación del PMBOK con las diversas normas, modelos, marcos, prácticas, etc. que se pueden aplicar al ámbito de la dirección de las TIC.

### **BSC (Balanced Scorecard)**

Contempla un sistema de administración del desempeño que permite a las empresas conducir su estrategia acorde a lo planeado mediante el monitoreo continuo, complementando los indicadores financieros tradicionales con criterios de medición de desempeño orientados a: “Clientes”, “Procesos Internos” y “Aprendizaje y Crecimiento”. Las mejores prácticas definidas en PMBOK sirven como referencia para la administración de los proyectos que sustentan la consecución de la estrategia de la empresa, la cual puede ser definida e implementada mediante Balanced Scorecard.

### **COBIT (Control Objectives for Information and Related Technology)**

Es un compendio de objetivos de control para la Tecnología de Información que incluye herramientas de soporte que permiten a la administración cubrir la brecha entre los requerimientos de control, los aspectos tecnológicos y los riesgos de negocio.

Las mejores prácticas definidas en PMBOK están relacionadas con los objetivos de control “Administrar las Inversiones de TI”, “Administrar la Calidad”, “Evaluar y Administrar los Riesgos de TI”, “Administrar los Proyectos de TI” y “Aprovisionamiento de los Recursos de TI” definidos en COBIT.

### **IT Governance**

Es un conjunto de mecanismos utilizados por la administración de una organización para dirigir y controlar su desarrollo tecnológico, asegurando que las metas del negocio sean alcanzadas de forma efectiva mediante la detección y control de los riesgos asociados. IT Governance puede tomar como referencia las áreas de conocimiento definidas en PMBOK para determinar las habilidades y conocimientos necesarios para la administración efectiva de los proyectos dentro de un ambiente de TI.

### **CMMI (Capability Maturity Model Integration)**

Es un modelo de mejora de procesos de construcción de software que puede tomar como referencia PMBOK para administrar los proyectos orientados a mejorar la capacidad y madurez de los procesos involucrados en la construcción de software.

### **ITIL (Information Technology Infrastructure Library)**

Es el marco de referencia para la Gestión de Servicios de TI más aceptado y utilizado en el mundo. Proporciona un conjunto de mejores prácticas en materia de administración de TI extraídas de organismos del sector público y privado que están a la vanguardia tecnológica a nivel internacional.

### **ISO 9000**

Es el estándar para establecer sistemas de gestión de calidad más reconocido y adoptado en el mundo, debido a los beneficios que brinda el uso de sus normas definidas para establecer, documentar, controlar, medir y mejorar los procesos y productos dentro de la organización.

#### **10.1.3.3 ESTRUCTURA DEL PMBOK**

La Guía del PMBOK está dividida en tres secciones:

**Sección I. Marco Conceptual de la Dirección de Proyectos:** proporciona una estructura básica para entender la Dirección de Proyectos.

Se definen los términos clave y proporciona una descripción general del resto de la Guía del PMBOK. También se describe el Ciclo de Vida del Proyecto y la Organización.

**Sección II. Norma para la Dirección de Proyectos aplicable a un proyecto:** especifica todos los procesos de Dirección de Proyectos que usa el equipo del proyecto para gestionar el mismo.



Se describen los cinco Grupos de Procesos de Dirección de Proyectos aplicables a cualquier proyecto y los procesos de Dirección de Proyectos que componen tales grupos.

Un grupo de procesos es un modo lógico de agrupar los procesos de dirección de proyectos, necesarios para cualquier proyecto, con dependencias entre ellos, y que se llevan a cabo en la misma secuencia siempre. Son los siguientes:

Procesos de inicio: procesos mediante los cuales se lleva a cabo la autorización formal para comenzar un proyecto.

Procesos de Planificación: procesos que deberán refinar los objetivos planteados durante el grupo de procesos de Inicio y planificar el curso de acción requerido para lograr los objetivos y el alcance pretendido del proyecto.

Procesos de Ejecución: procesos que se despliegan para completar el trabajo definido en el grupo de procesos de Planificación con objeto de cumplir los requisitos del proyecto.

Procesos de Control: procesos realizados para medir y supervisar regularmente el avance del proyecto, de manera que se puedan identificar las variaciones respecto a la planificación y adoptar, cuando sea necesario, las acciones correctivas, preventivas y de control de cambios para cumplir con los objetivos del proyecto.

Procesos de Cierre: procesos requeridos para cerrar formalmente un proyecto y que aseguran que las experiencias adquiridas durante el proyecto queden registradas y a disposición de futuros usos.

**Sección III: Áreas de Conocimiento de la Dirección de Proyectos:** organiza los 42 procesos de Dirección de Proyectos en nueve Áreas de Conocimiento. La introducción de la Sección III describe la leyenda de los

diagramas de flujo de procesos que se usan en cada capítulo de Área de Conocimiento y en la introducción de todas las Áreas de conocimiento.

Un equipo de proyectos funciona en 9 áreas de conocimiento con un número de procesos básicos según el resumen que presentamos a continuación:

- **Integración.** Desarrollo de la carta del proyecto, la declaración del alcance y el plan. Dirección, supervisión y control del proyecto.
- **Alcance.** Planificación, definición, creación, verificación y control de la estructura de desglose de trabajo (EDT).
- **Tiempo.** Definición, secuenciación, estimación de recursos necesarios y de la duración, desarrollo y control del cronograma.
- **Costo.** Planificación de recursos, costos estimados, presupuesto y control.
- **Calidad.** Planificación de la calidad, aseguramiento de calidad y control de calidad.
- **Recursos Humanos.** Planificación, contratación, desarrollo y administración de los Recursos Humanos.
- **Comunicaciones.** Planificación de comunicaciones, distribución de la información, difusión del desempeño, Gestión de *stakeholders* (*interesados*).
- **Riesgos.** Planificación e identificación de riesgos, Análisis de riesgos (cualitativa y cuantitativa), planificación de la respuesta ante riesgos (acción), y supervisión y control del riesgo.
- **Adquisiciones.** Plan de contrataciones y adquisiciones, selección e incentivos de los vendedores, administración y cierre de contratos

En los siguientes apartados se comenta cada una de dichas áreas de conocimiento, describiendo cada uno de los procesos que las componen.

### **3.1 GESTIÓN DE LA INTEGRACIÓN DEL PROYECTO**

Describe los procesos y actividades que forman parte de los diversos elementos de la Dirección de Proyectos, que se identifican, definen, combinan, unen y coordinan dentro de los Grupos de Procesos de Dirección de Proyectos. Se compone de los procesos:

- Desarrollar el **Acta de Constitución del Proyecto**: Se autoriza formalmente el proyecto o una fase del mismo, documentándose los requisitos iniciales que satisfagan las necesidades y expectativas de los interesados.
- Desarrollar el **Plan para la Dirección del Proyecto**: Se documentan las acciones necesarias para definir, preparar, integrar y coordinar todos los planes subsidiarios.
- **Dirigir y Gestionar** la Ejecución del Proyecto: Se lleva a cabo el trabajo definido en el plan para la dirección del proyecto para así cumplir con los objetivos del mismo.
- **Monitorear y Controlar** el Trabajo del Proyecto: Se monitorea, revisa y regula el avance para ver las posibles desviaciones respecto a las líneas base definidas en el plan para la dirección del proyecto.
- Realizar el **Control Integrado de Cambios**: Se revisan todas las solicitudes de cambio, además de aprobar y gestionar los cambios.
- **Cerrar** el Proyecto o una Fase: Se finalizan todas las actividades, dando por completado formalmente el proyecto o una fase del mismo.

### **10.2.1 EL PLAN GENERAL DEL PROYECTO**

Desarrollar el Plan para la Dirección del Proyecto es el proceso que consiste en documentar todas las acciones necesarias para definir, preparar, integrar y coordinar todos los planes subsidiarios. El plan para la dirección del proyecto define el modo en que el proyecto se ejecuta, se monitorea, se controla y se cierra. En función del área de aplicación y de la

complejidad del proyecto, el contenido del plan para la dirección del proyecto podrá variar. El plan para la dirección del proyecto se desarrolla a través de una serie de procesos integrados hasta llegar al cierre del proyecto.

Este proceso da lugar a un plan para la dirección del proyecto que se elabora de forma gradual por medio de actualizaciones, y se controla y se aprueba a través del proceso Realizar el Control Integrado de Cambios

### **10.3 GESTIÓN DEL ALCANCE DEL PROYECTO**

Describe los procesos necesarios para asegurarse que el proyecto incluya todo el trabajo requerido para completarse satisfactoriamente. Se compone de los procesos:

- **Recopilar Requisitos:** Se documentan las necesidades de los interesados para convertirlas en requisitos del proyecto.
- **Definir** el Alcance: Se desarrolla el enunciado del alcance detallado, el qué.
- Crear la **Estructura de Desglose del trabajo** o EDT: Descomponer el proyecto en partes más pequeñas.
- **Verificar** el Alcance: Conseguir la aceptación formal del alcance por parte del cliente o patrocinador.
- **Controlar** el Alcance: Gestionar los cambios en el alcance.

En el contexto del proyecto, el término alcance puede referirse a:

- **Alcance del producto:** Las características y funciones que definen un producto, servicio o resultado.
- **Alcance del proyecto:** El trabajo que debe realizarse para entregar un producto, servicio o resultado con las características y funciones especificadas.

## **10.4 GESTIÓN DEL TIEMPO DEL PROYECTO**

Describe los procesos relativos a la puntualidad en la conclusión del proyecto. Se compone de los procesos:

- **Definir** las Actividades: Se identifica cada una de las actividades que se deben realizar para lograr un proyecto exitoso.
- **Secuenciar** las Actividades: Se analiza qué tipo de dependencias existe entre las distintas actividades.
- Estimar los **Recursos** de las Actividades: Se determina cuáles son los recursos necesarios y disponibles para llevar a cabo cada actividad.
- Estimar la **Duración** de las Actividades: Se estima el tiempo necesario para completar las actividades.
- Desarrollar el **Cronograma**: Se analiza la integración existente entre la secuencia, los recursos necesarios, las restricciones y la duración de cada actividad.
- **Controlar el Cronograma**: Se administran los cambios en el cronograma

### **10.4.1 TÉCNICAS DE PLANIFICACIÓN**

Método de la Ruta Crítica: El método de la ruta crítica calcula las fechas teóricas de inicio y finalización tempranas y tardías para cada una de las actividades, sin tener en cuenta las limitaciones de recursos, realizando un análisis que recorre hacia adelante y hacia atrás la red del cronograma.

Método de la Cadena Crítica: Consiste en modificar el cronograma del proyecto teniendo en cuenta la restricción de recursos.

Nivelación de recursos: Se modifica la programación del proyecto para mejorar la eficiencia en cuanto a asignación de recursos.

Análisis “que pasa si”: Se realizan simulaciones de cómo cambiaría el cronograma del proyecto si cambiásemos alguna de las variables que lo afectan

Aplicación de Adelantos y Retrasos: Los adelantos y retrasos son una técnica de refinamiento que se aplica durante el análisis de la red para desarrollar un cronograma viable.

Compresión del cronograma: Consiste en acortar el cronograma del proyecto sin modificar el alcance. Dos de las técnicas más utilizadas son:

- Compresión (Intensificación o Crashing): se agregan más recursos al proyecto para acortar la duración. Por lo general, esta técnica implicará mayores costos.
- Ejecución rápida (fast-tracking): se realizan actividades en paralelo para acelerar el proyecto, agregando riesgos al mismo.

## **10.5 GESTIÓN DE LOS COSTOS DEL PROYECTO**

Describe los procesos involucrados en la planificación, estimación, presupuesto y control de costes de forma que el proyecto se complete dentro del presupuesto aprobado. Se compone de los procesos:

- **Estimar** los Costos: Se calculan los costos de cada recurso para completar las actividades del proyecto.
- Determinar el **Presupuesto**: Se suman los costos de todas las actividades del proyecto a través del tiempo.
- **Controlar** los Costos: Se influye sobre las variaciones de costos y se administran los cambios del presupuesto

Existen diversos tipos de costos, a continuación mencionaremos los principales.

- **Costos variables:** dependen del volumen de producción.
- **Costos fijos:** no cambian con el volumen de producción.
- **Costos directos:** se pueden atribuir directamente al proyecto.
- **Costos indirectos:** benefician a varios proyectos y generalmente no se puede identificar con exactitud la proporción que corresponde a cada uno
- **Costo de oportunidad:** el costo de oportunidad de un recurso es su mejor alternativa dejada de lado.
- **Costos hundidos o enterrados:** costos que ya fueron devengados y no cambiarán con la decisión de hacer o no hacer el proyecto

### **10.5.1 PRESUPUESTOS**

Para realizar los presupuestos se cuenta con las siguientes herramientas y técnicas:

#### **Suma de Costos**

Las estimaciones de costos se suman por paquetes de trabajo, de acuerdo con la EDT; a continuación se van sumando los niveles superiores de componentes de la EDT, tales como las cuentas de control, y finalmente el proyecto completo.

#### **Análisis de Reserva**

El análisis de reserva del presupuesto puede establecer tanto las reservas para contingencias como las de gestión del proyecto. Las reservas para contingencias son asignaciones para aquellos cambios que no han sido planificados previamente, pero que son potencialmente necesarios. Las reservas de gestión son presupuestos reservados para cambios no planificados al alcance y al costo del proyecto. Las reservas no forman parte de la línea base de costo, pero pueden incluirse en el presupuesto total del proyecto.

### **Juicio de Expertos**

Es aquel que se obtiene en base a la experiencia en un área de aplicación, un área de conocimiento, una disciplina, una industria, etc., según resulte apropiado para la actividad que se está desarrollando, y que debe utilizarse para determinar el presupuesto. El juicio de expertos puede provenir de diversas fuentes, entre otras:

- otras unidades dentro de la organización ejecutante
- consultores
- interesados, incluyendo clientes
- asociaciones profesionales y técnicas
- grupos industriales

### **Relaciones Históricas**

Cualquier relación histórica que dé como resultado estimaciones paramétricas o análogas implica el uso de características (parámetros) del proyecto para desarrollar modelos matemáticos que permitan predecir los costos totales del proyecto. Estos modelos pueden ser simples (p.ej., construir una vivienda costará una cierta cantidad por metro cuadrado de espacio útil) o complejas (p.ej., un modelo de costo de desarrollo de software utiliza varios factores de ajuste separados, con múltiples criterios por cada uno de esos factores).

## **10.6 GESTIÓN DE LA CALIDAD DEL PROYECTO**

Describe los procesos necesarios para asegurarse de que el proyecto cumpla con los objetivos para los cuales ha sido emprendido. Se compone de los procesos:

- **Planificar la Calidad:** Se establece qué normas son relevantes y cómo se van a satisfacer.
- **Asegurar la Calidad:** Se utilizan los procesos necesarios para cumplir con los requisitos del proyecto. Dicho de otro modo, se asegura que se estén utilizando los planes para la gestión de calidad.



- **Controlar la Calidad:** Se supervisa que el proyecto esté dentro de los límites pre-establecidos.

En todo proyecto es sumamente importante dedicar tiempo a la gestión de calidad para:

- Prevenir errores y defectos
- Evitar realizar de nuevo el trabajo, lo que implica ahorrar tiempo y dinero
- Tener un cliente satisfecho

La gestión de la calidad implica que el proyecto satisfaga las necesidades por las cuales se emprendió. Para ello será necesario lo siguiente:

- Convertir las necesidades y expectativas de los interesados en requisitos del proyecto
- Lograr la satisfacción del cliente cuando el proyecto produzca lo planificado y el producto cubra las necesidades reales
- Realizar acciones de prevención sobre la inspección
- Buscar de forma permanente la perfección: mejora continua

La gestión moderna de la calidad complementa la dirección de proyectos. Ambas disciplinas reconocen la importancia de:

- **La satisfacción del cliente.** Entender, evaluar, definir y gestionar las expectativas, de modo que se cumplan los requisitos del cliente.
- **La prevención antes que la inspección.** Uno de los preceptos fundamentales de la gestión moderna de la calidad establece que la calidad se planifica, se diseña y se integra (y no se inspecciona).

- **La mejora continua.** El ciclo planificar-hacer-revisar-actuar es la base para la mejora de la calidad, según la definición de Shewhart, modificada por Deming.
- **La responsabilidad de la dirección.** El éxito requiere la participación de todos los miembros del equipo del proyecto, pero proporcionar los recursos necesarios para lograr dicho éxito sigue siendo responsabilidad de la dirección.

### **10.6.1 PLAN DE CALIDAD**

El plan de gestión de calidad describe cómo el equipo de dirección del proyecto implementará la política de calidad de la organización ejecutante. Es un componente o un plan subsidiario del plan para la dirección del proyecto. El plan de gestión de calidad proporciona entradas al plan general para la dirección del proyecto y comprende: el control de calidad, el aseguramiento de la calidad y métodos de mejora continua de los procesos del proyecto.

El plan de gestión de calidad puede ser redactado de modo formal o informal, muy detallado o formulado de manera general. El formato y el grado de detalle se determinan según los requisitos del proyecto. El plan de gestión de calidad debe revisarse en las etapas iniciales del proyecto, para asegurarse que las decisiones estén basadas en informaciones precisas. Los beneficios de esta revisión pueden incluir la reducción del costo y sobrecostos en el cronograma ocasionados por el reproceso.

## **10.7 CAPACIDADES DEL JEFE DE PROYECTO**

### **10.7.1 GESTIÓN DE LOS RECURSOS HUMANOS DEL PROYECTO**

La Gestión de los Recursos Humanos del Proyecto incluye los procesos que organizan, gestionan y conducen el equipo del proyecto. El equipo del proyecto está conformado por aquellas personas a las que se les

han asignado roles y responsabilidades para completar el proyecto. Se compone de los procesos:

- Desarrollar el **Plan de Recursos Humanos**: Se definen los roles, responsabilidades y habilidades de los miembros del equipo, así como las relaciones de comunicación.
- **Adquirir** el equipo: Se obtienen los recursos humanos necesarios para llevar a cabo las actividades del proyecto.
- **Desarrollar** el equipo: Se mejoran las competencias y las habilidades de interacción entre los miembros del equipo.
- **Gestionar** el equipo: Se monitorea el desempeño individual y de grupo de cada persona y se resuelven los conflictos que suelen ocurrir entre los miembros del equipo.

### **10.7.2 CAPACIDADES DEL JEFE DE PROYECTO**

Los directores de proyecto usan una combinación de habilidades técnicas, humanas y conceptuales para analizar las situaciones e interactuar de manera apropiada con los miembros del equipo. Mediante dichas habilidades, los directores de proyecto podrán sacar provecho de los puntos fuertes de los miembros del equipo.

Algunas de las habilidades interpersonales usadas con mayor frecuencia por los directores del proyecto se describen brevemente a continuación.

**Liderazgo.** Los proyectos exitosos requieren fuertes habilidades de liderazgo. El liderazgo es importante en todas las fases del ciclo de vida del proyecto. Es fundamental comunicar la visión e inspirar al equipo del proyecto a fin de lograr un alto desempeño.

Existen distintos estilos de **liderazgo** como por ejemplo:

- Directivo: decir qué hay que hacer

- Consultivo (Coaching): dar instrucciones
- Participativo (Supporting): ofrecer asistencia
- Delegativo (Empowerment): el empleado decide por sí solo
- Facilitador: coordina a los demás
- Autocrático: tomar decisiones sin consultar
- Consenso: resolución de problemas grupales

**Influencia.** Dado que a menudo la autoridad directa de los directores del proyecto sobre los miembros de su equipo es escasa o nula en una organización de tipo matricial, su capacidad de influir en los interesados resulta vital para el éxito del proyecto. Entre las habilidades clave de influencia se encuentran:

- Tener la habilidad para persuadir y expresar con claridad los puntos de vista y las posiciones asumidas.
- Contar con gran habilidad para escuchar de manera activa y eficaz.
- Tener en cuenta las diversas perspectivas en cualquier situación.
- Recopilar información relevante y crítica a fin de abordar los asuntos importantes y lograr acuerdos.

**Toma de decisiones eficaz.** Esto implica tener la habilidad de negociar e influir sobre la organización y el equipo de dirección del proyecto. Algunas pautas en materia de toma de decisiones incluyen:

- centrarse en los objetivos perseguidos
- seguir un proceso de toma de decisiones
- desarrollar las cualidades personales de los miembros del equipo
- fomentar la creatividad del equipo
- gestionar las oportunidades y los riesgos

## **10.8 GESTIÓN DE LAS COMUNICACIONES DEL PROYECTO**

Describe los procesos relacionados con la generación, distribución, almacenamiento y destino final de la información del proyecto en tiempo y forma. Se compone de los procesos:

- **Identificar** a los Interesados: Se identifica a todas las personas u organizaciones que de alguna manera se verán afectadas por el proyecto.
- **Planificar** las Comunicaciones: Se determinan cuáles serán las necesidades de información del proyecto.
- **Distribuir** la Información: Se coloca la información a disposición de los interesados.
- **Gestionar las Expectativas** de los Interesados: Se satisfacen los requisitos de los interesados y se resuelven los conflictos entre los recursos humanos.
- **Informar** el Desempeño: Se comunica el estado de avance del proyecto.

En el momento de distribuir la información hay que tener en cuenta las distintas dimensiones de la comunicación:

- Interna: entre las personas que forman parte del proyecto
- Externa: hacia los interesados externos del proyecto
- Vertical: entre jefe-empleado y viceversa
- Horizontal: entre colegas del proyecto
- Escrita formal: planes, solicitud, etc.
- Escrita informal: memos, e-mails, notas
- Oral formal: presentaciones
- Oral informal: reuniones, conversaciones

La mayoría de las habilidades de comunicación son comunes a la dirección en general y a la dirección de proyectos. Entre estas habilidades, se incluye:

- escuchar de manera activa y eficaz
- formular preguntas, sondear ideas y situaciones para garantizar una mejor comprensión
- educar para aumentar el conocimiento del equipo a fin de que sea más eficaz
- investigar para identificar o confirmar la información
- identificar y gestionar expectativas
- persuadir a una persona u organización para llevar a cabo una acción
- negociar a fin de lograr acuerdos entre partes que resulten mutuamente aceptables
- resolver conflictos para prevenir impactos negativos

## **10.9 GESTIÓN DE LOS RIESGOS DEL PROYECTO**

No se debería comenzar con la ejecución del proyecto sin un análisis de riesgo. La planificación de los riesgos es un área integradora del resto de las áreas del conocimiento. Por ejemplo, no podemos afirmar que tenemos un cronograma y presupuesto realista si todavía no hemos finalizado el análisis de riesgo. Con el análisis de riesgo se determinarán las reservas para contingencia de plazos y costos que deben incluirse en el plan para la dirección del proyecto. Se compone de los procesos:

- **Planificar** la Gestión de Riesgos: Se definen las actividades de gestión de los riesgos para un proyecto.
- **Identificar** los Riesgos: Una vez realizado el plan de gestión de riesgos, es necesario comenzar con la identificación de los eventos con riesgo que, si ocurriesen, afectarían el resultado del proyecto ya sea para bien o para mal.

- Realizar el **Análisis Cualitativo de Riesgos**: Se priorizan los riesgos para realizar otros análisis o acciones posteriores, evaluando y combinando la probabilidad de ocurrencia y el impacto de dichos riesgos. Para ello se pueden usar las siguientes herramientas y técnicas:
  - **Evaluación de probabilidad e impacto**: se estima cuál es la probabilidad de ocurrencia y el impacto de cada riesgo identificado.
  - **Matriz de probabilidad e impacto**: tabla de doble entrada donde se combina la probabilidad y el impacto para poder hacer una priorización de los riesgos.
  - **Categorización de los riesgos**: se agrupan los riesgos por causas comunes
- Realizar el **Análisis Cuantitativo de Riesgos**: Se analiza numéricamente el efecto de los riesgos identificados sobre los objetivos generales del proyecto. Para ello se pueden usar las siguientes herramientas y técnicas:
  - **Distribuciones de probabilidad**: uniforme, triangular, beta, normal, log normal, poisson, F, Chi-cuadrada, etc.
  - **Valor monetario esperado**: se obtiene de multiplicar la probabilidad de ocurrencia por el impacto en valor monetario
  - **Árbol de decisión**: diagrama que describe las implicaciones de elegir una u otra alternativa entre todas las disponibles.
- **Planificar la Respuesta** a los Riesgos: Se desarrollan las acciones para mejorar las oportunidades y reducir las amenazas a los objetivos del proyecto. Para los **riesgos negativos** se suelen utilizar las siguientes estrategias o herramientas: evitar, transferir, mitigar o aceptar. Por su parte, para los **riesgos positivos** se suelen utilizar las siguientes estrategias o herramientas: explotar, compartir, mejorar, aceptar.

- **Monitorear y Controlar** los Riesgos: Se implementan planes de respuesta a los riesgos, se rastrean los riesgos identificados, se monitorean los riesgos residuales, se identifican nuevos riesgos y se evalúa la efectividad del proceso contra riesgos a través del proyecto.

### **10.9.1 CONTINGENCIAS**

Algunas estrategias están diseñadas para ser usadas únicamente si se presentan determinados eventos. Para algunos riesgos, resulta apropiado para el equipo del proyecto elaborar un plan de respuesta que sólo se ejecutará bajo determinadas condiciones predefinidas, si se cree que habrá suficientes señales de advertencia para implementar el plan. Los eventos que disparan la respuesta para contingencias, tales como no cumplir con hitos intermedios u obtener una prioridad más alta con un proveedor, deben definirse y rastrearse.

#### **Reservas para contingencias**

Para aquellos riesgos conocidos, identificados y cuantificados, se puede estimar una reserva monetaria para contingencias, que no forma parte de la línea base de costo del proyecto. Por su parte, los riesgos desconocidos no se pueden gestionar de manera proactiva y podrían considerarse asignando una reserva de gestión general al proyecto, que no forma parte de la línea base de costo, pero sí se incluye en el presupuesto total del proyecto

### **3.10 GESTIÓN DE LAS ADQUISICIONES DEL PROYECTO**

Describe los procesos para comprar o adquirir productos, servicios o resultados, así como para contratar procesos de dirección. Se compone de los procesos:



- **Planificar** las Adquisiciones: Se documentan las decisiones de compra para el proyecto, especificando cómo hacerlo e identificando a los posibles vendedores.
- **Efectuar** las Adquisiciones: Se obtienen las respuestas de los vendedores, seleccionando los más ventajosos para el proyecto y adjudicando los contratos.
- **Administrar** las Adquisiciones: Se gestionan las relaciones de adquisiciones, se monitoriza la ejecución de los contratos, y se efectúan cambios y correcciones según sea necesario.
- **Cerrar** las Adquisiciones: Se completa cada adquisición para el proyecto.

Los procesos de Gestión de las Adquisiciones del Proyecto implican la realización de contratos, que son documentos legales que se establecen entre un comprador y un vendedor. Dicho documento representa un acuerdo vinculante para las partes en virtud del cual el vendedor está obligado a proveer los productos, servicios o resultados especificados, y el comprador debe proporcionar dinero o cualquier otra contraprestación válida. Un contrato de adquisición incluye términos y condiciones, y puede incorporar otros aspectos especificados por el comprador para establecer lo que el vendedor debe realizar o proporcionar. Es responsabilidad del equipo de dirección del proyecto asegurar que todas las adquisiciones cumplen las necesidades del proyecto, a la vez que se respetan las políticas de la organización en cuanto a adquisiciones se refiere.

Bibliografía:

Guía de los fundamentos para la dirección de proyectos (Guía del PMBOK)  
4ª Edición.

Sitios web:

<http://www.pmi.org>      Project Management Institute®

Autor:

Ramón Seoane Freijido

Director de Sistemas de Información Molduras del Noroeste

Colegiado del CPEIG

# **11. SISTEMAS DE GESTIÓN DE CALIDAD. NORMALIZACIÓN Y CERTIFICACIÓN. EFQM. SERIE ISSO 9000. CMMI.**

## **Tema 11. Sistemas de Gestión de Calidad. Normalización y certificación. EFQM. Serie ISO 9000. CMMI**

### **INDICE**

#### **11.1 Normalización y certificación**

##### **11.1.1 Normalización. Conceptos básicos**

##### **11.1.2 Certificación. Pasos**

#### **11.2 EFQM. Modelo EFQM de excelencia**

##### **11.2.1 Introducción al modelo**

##### **11.2.2 ¿Qué es la EFQM?**

##### **11.2.3 Historia de la EFQM**

##### **11.2.4 ¿Qué es el modelo EFQM?**

##### **11.2.5 ¿Para qué sirve el modelo EFQM?**

##### **11.2.6 Composición del modelo EFQM**

##### **11.2.7 Ventajas de adoptar el modelo EFQM**

##### **11.2.8 Cambios que origina la excelencia en gestión**

#### **11.3 Serie ISO 9000**

##### **11.3.1 Introducción**

##### **11.3.2 La familia de normas ISO 9000**

##### **11.3.3 Principios de gestión**

##### **11.3.4 Versiones específicas de la norma ISO 9000**

##### **11.3.5 Costos y beneficios de establecer un sistema de gestión de la calidad**

##### **11.3.6 Implementación de un sistema de gestión de la calidad**

#### **11.4 CMMI**

### **11.1 NORMALIZACIÓN Y CERTIFICACIÓN.**

#### **11.1.1 NORMALIZACIÓN. CONCEPTOS BÁSICOS.**

¿Qué se entiende por normalización?

La normalización es una actividad colectiva encaminada a establecer soluciones a situaciones repetitivas.

En particular, esta actividad consiste en la elaboración, difusión y aplicación de normas.

La normalización ofrece importantes beneficios, como consecuencia de adaptar los productos, procesos y servicios a los fines a los que se destinan, proteger la salud y el medio ambiente, prevenir los obstáculos al comercio y facilitar la cooperación tecnológica.

### ¿Qué es una norma?

Las normas son documentos técnicos con las siguientes características:

- Contienen especificaciones técnicas de aplicación voluntaria.
- Son elaborados por consenso de las partes interesadas:
- Están basados en los resultados de la experiencia y el desarrollo tecnológico.
- Son aprobados por un Organismo Nacional/Regional/Internacional de Normalización reconocido.
- Están disponibles al público.

Las normas ofrecen un lenguaje común de comunicación entre las empresas, la Administración y los usuarios y consumidores, establecen un equilibrio socioeconómico entre los distintos agentes que participan en las transacciones comerciales, base de cualquier economía de mercado, y son un patrón necesario de confianza entre cliente y proveedor.

### ¿Qué ventajas ofrece la normalización?

a) Para los consumidores:

- Establece niveles de calidad y seguridad de los productos y servicios.
- Informa de las características del producto.
- Facilita la comparación entre diferentes ofertas.

b) Para los fabricantes:

- Racionaliza variedades y tipos de productos.
- Disminuye el volumen de existencias en almacén y los costes de producción.
- Mejora la gestión y el diseño.
- Agiliza el tratamiento de los pedidos.
- Facilita la comercialización de los productos y su exportación.
- Simplifica la gestión de compras.

c) Para la Administración:

- Simplifica la elaboración de textos legales.
- Establece políticas de calidad, medioambientales y de seguridad.
- Ayuda al desarrollo económico.
- Agiliza el comercio.

¿Qué se puede normalizar?

El campo de actividad de las normas es tan amplio como la propia diversidad de productos o servicios, incluidos sus procesos de elaboración.

Así, se normalizan los Materiales (plásticos, acero, papel, etc.), los Elementos y Productos (tornillos, televisores, herramientas, tuberías, etc.), las Máquinas y Conjuntos (motores, ascensores, electrodomésticos, etc.), Métodos de Ensayo, Temas Generales (medio ambiente, calidad del agua, reglas de seguridad, estadística, unidades de medida, etc.), Gestión y Aseguramiento de la Calidad, Gestión Medioambiental (gestión, auditoría, análisis del ciclo de vida, etc.), Gestión de prevención de riesgos en el trabajo (gestión y auditoría), etc.

### ¿Qué clases de normas existen?

Los documentos normativos pueden ser de diferentes tipos dependiendo del organismo que los haya elaborado.

En la clasificación tradicional de normas se distingue entre:

- Normas nacionales. Son elaboradas, sometidas a un período de información pública y sancionadas por un organismo reconocido legalmente para desarrollar actividades de normalización en un ámbito nacional. En España estas normas son las normas UNE, aprobadas por AENOR, que es el organismo reconocido por la Administración Pública española para desarrollar las actividades de normalización en nuestro país (Real Decreto 2000/1995).
- Normas regionales. Son elaboradas en el marco de un organismo de normalización regional, normalmente de ámbito continental, que agrupa a un determinado número de Organismos Nacionales de Normalización. Las más conocidas, aunque no las únicas, son las normas europeas elaboradas por los Organismos Europeos de Normalización (CEN, CENELEC, ETSI), y preparadas con la participación de representantes acreditados de todos los países miembros. AENOR es el organismo nacional de normalización español miembro de CEN y CENELEC y, por lo tanto, la organización a través de la cual se canalizan los intereses y la participación de los agentes socioeconómicos de nuestro país en la normalización europea.
- Normas internacionales. Tienen características similares a las normas regionales en cuanto a su elaboración, pero se distinguen de ellas en que su ámbito es mundial. Las más representativas por su campo de actividad son las normas CEI/IEC (Comité Electrotécnico Internacional) para el área eléctrica, las UIT/ITU (Unión Internacional de Telecomunicaciones) para el sector de las telecomunicaciones y las normas ISO (Organización Internacional de Normalización) para el resto. AENOR es el organismo nacional de normalización español miembro de ISO y CEI y, por lo tanto, la

organización a través de la cual se canalizan los intereses y la participación de los agentes socioeconómicos de nuestro país en la normalización internacional.

### ¿Qué es una norma UNE?

Una norma UNE es una especificación técnica de aplicación repetitiva o continuada cuya observancia no es obligatoria, establecida con participación de todas las partes interesadas, que aprueba AENOR, organismo reconocido a nivel nacional e internacional por su actividad normativa (Ley 21/1992, de 16 de julio, de Industria).

### ¿Cómo se elabora una norma UNE?

La elaboración de una norma UNE, incluida la adopción de normas europeas, se lleva a cabo en el seno de los Comités Técnicos de Normalización (CTN) a través de las siguientes fases:

- Trabajos preliminares (recopilación de documentación, discusión sobre el contenido...) previos a la toma en consideración de una nueva iniciativa;
- Elaboración del proyecto de norma; incluye todas aquellas actividades que se desarrollan por el Comité hasta la aprobación de un documento como proyecto de norma, buscando siempre el consenso de todas las partes;
- Información pública en el BOE; anuncio de la existencia del proyecto de norma, tanto nacional como europea, para que cualquier persona, física o jurídica, pueda remitir las observaciones al mismo que estime oportunas;
- Elaboración de la propuesta de norma; una vez superada la fase anterior, y recibidas en AENOR las posibles observaciones al proyecto, el CTN



procede al estudio de las mismas y aprobación de la propuesta de norma final, para su consideración y adopción por AENOR;

- Registro, edición y difusión de la norma UNE; publicación de la norma UNE por AENOR, notificación a BOE, promoción y comercialización, a través de los servicios comerciales de AENOR.

### **11.1.2 CERTIFICACIÓN. PASOS.**

El proceso de certificarse con base en ISO 9001, y de mantener este status una vez conseguido, se presenta en los pasos siguientes:

#### 1. Cómo seleccionar un organismo de certificación

Las organizaciones que deseen obtener un certificado, deben presentar una solicitud al organismo de certificación de su elección. Los aspectos a considerar al seleccionar el organismo de certificación incluyen:

- Si la naturaleza de la acreditación del organismo de certificación es aceptable en el mercado al cual la organización desea exportar.
- La imagen del organismo de certificación en el mercado.
- Cotizaciones de las tarifas de certificación y auditorías, etc.

#### 2. Preparación para la evaluación

De acuerdo con la ISO 9001, el primer requisito es definir los procesos de la organización que afectan a la calidad, de manera que el primer paso es que el auditor del organismo de certificación se reúna con la alta dirección de la organización, con el fin de que aquél obtenga una comprensión clara acerca de los procesos de la organización.

#### 3. Auditoría

Los auditores recogen evidencia de conformidad o no conformidad mediante la observación de actividades, el examen de procedimientos/registros, observaciones de las condiciones de manejo de la

empresa, a través de entrevistas con los directores y personal involucrado de la organización, etc. La información recolectada mediante las entrevistas es verificada o ensayada por los auditores mediante la recolección de la misma información de otras fuentes, tales como observaciones físicas o mediciones realizadas en el producto y sus registros relacionados. Los auditores visitan y verifican la conformidad con el SGC en todos los departamentos y funciones dentro del alcance del SGC.

#### 4. No conformidades

La evidencia recogida por los auditores es comparada con los criterios de la auditoría (políticas y objetivos de la compañía, manuales, procedimientos, instrucciones, contratos, reglamentaciones, etc.) y los hallazgos de las auditorías, incluidas las no conformidades, si las hay, son aclaradas y reportadas a la alta dirección al final de auditoría en el sitio, en una reunión formal con la alta dirección, llamada “Reunión de Cierre”. Las no conformidades (NC) son clasificadas por los auditores como “mayores” o “menores”. Las “observaciones” también se registran.

#### 5. Otorgamiento del certificado ISO 9000

Con base a las recomendaciones del auditor y después de la revisión independiente de estas recomendaciones por el organismo certificador, éste expide un certificado a la organización. El certificado se expide para el alcance específico del negocio y para los productos o servicios para los cuales la organización ha implementado un SGC.

#### 6. Auditorías de seguimiento

El certificado se otorga inicialmente por un período de tres años. Durante este tiempo, el organismo de certificación realiza auditorías de seguimiento periódicas (una o dos veces al año), en fechas acordadas mutuamente. El organismo de certificación informa previamente un plan de auditoría de tres años, en el que se indique el alcance de cada auditoría de seguimiento. Estas auditorías se planifican de manera que todos los

aspectos del SGC se auditen en un período de tres años. Después de los tres años se lleva a cabo una auditoría de re-certificación usando los pasos 2 y 5 anteriores.

## **11.2 EFQM. MODELO EFQM DE EXCELENCIA**

### **11.2.1 INTRODUCCIÓN AL MODELO**

El Modelo EFQM de Excelencia es un marco de trabajo no-prescriptivo que tiene nueve criterios. Los criterios que hacen referencia a un Agente Facilitador tratan sobre lo que la organización hace. Los criterios que hacen referencia a los Resultados tratan sobre lo que la organización logra. Los Resultados son consecuencia de los Agentes Facilitadores.

El Modelo, que reconoce que la Excelencia en todo lo referente a resultados y rendimiento de una organización se puede lograr de manera sostenida mediante distintos enfoques, se fundamenta en que:

"Los resultados excelentes con respecto al Rendimiento de la Organización, a los Clientes, las Personas y la Sociedad se logran mediante un Liderazgo que dirija e impulse la Política y Estrategia, las Personas de la organización, las Alianzas y Recursos, y los Procesos."

### **11.2.2 ¿QUÉ ES LA EFQM?**

EFQM (European Foundation for Quality Management o Fundación Europea para la Gestión de la Calidad).

MISSION: Ser la fuerza que impulsa la Excelencia en las organizaciones europeas de manera sostenida.

VISION: Un mundo en el que las organizaciones europeas sobresalgan por su Excelencia

EFQM es una organización sin ánimo de lucro cuyo ámbito es Europa y su sede está en Bruselas.

EFQM es el creador y el gestor del premio a la Excelencia EEA (EFQM Excellence Award) que reconoce la Excelencia en Gestión en las organizaciones.

EFQM es la propietaria del Modelo de Excelencia EFQM y es la encargada de actualizarla con las buenas prácticas que se están llevando en las organizaciones punteras en el tema de la Excelencia en Gestión.

### **11.2.3 HISTORIA DE LA EFQM**

1988 Fue creada la Fundación Europea para la Gestión de la Calidad (EFQM) siendo una organización sin ánimo de lucro formada por 14 organizaciones europeas (Bosch, BT, Bull, Ciba-Geigy, Dassault, Electrolux, Fiat, KLM, Nestlé, Olivetti, Philips, Renault, Sulzer y Volkswagen)

1989 Fue establecida la misión, visión y objetivos del EFQM y se comienzan los trabajos de desarrollo del Modelo Europeo de Calidad. Además, se añadieron otras 53 empresas.

1991 Nace el Modelo de Excelencia EFQM y se lanza el primer Premio Europeo de Calidad para empresas

1992 Se presenta el Premio Europeo de Calidad

1995 Se adapta el Modelo y lanza el Premio Europeo para el sector público

1996 Se simplifica el Modelo y lanza el Premio Europeo para pymes y unidades operativas

2003 Se actualiza el Modelo de Excelencia

2005 Se lanza el sistema 2005+ para la presentación de memorias y evaluación para el Premio EFQM a la Excelencia (EEA)

### **11.2.4 ¿QUÉ ES EL MODELO EFQM?**

El Modelo EFQM de Excelencia es un instrumento práctico que ayuda a las organizaciones a establecer un sistema de gestión apropiado, midiendo en qué punto se encuentran dentro del camino hacia la

excelencia, identificando posibles carencias de la organización y definiendo acciones de mejora.

### **11.2.5 ¿PARA QUÉ SIRVE EL MODELO EFQM?**

Es un marco que las organizaciones pueden utilizar para ayudarse a desarrollar su visión y las metas para el futuro de una manera tangible.

Es un instrumento que las organizaciones pueden utilizar para identificar y entender la naturaleza de su negocio, es decir, las relaciones entre los distintos agentes presentes en la actividad, y las relaciones causa-efecto.

Es una herramienta que permite establecer un mismo lenguaje y modo de pensar en toda la organización.

Es una herramienta de diagnóstico para determinar la salud actual de la organización, detectando puntos de mejora e implantando acciones que le ayuden a mejorar.

Es la base para la concesión del Premio EFQM a la Excelencia, esto es, un proceso de evaluación que permite a Europa reconocer a sus organizaciones mejor gestionadas y promoverlas como modelos de excelencia del que las demás organizaciones puedan aprender.

### **11.2.6 COMPOSICIÓN DEL MODELO EFQM**

El Modelo de Excelencia EFQM es un marco no preceptivo basado en nueve criterios.

Cinco de estos son “Agentes Facilitadores” (Lo que la organización hace. Incluye 24 subcriterios) y cuatro son “Resultados” (Lo que la organización logra. Incluye 8 subcriterios). Total: 9 CRITERIOS, 32 subcriterios y 298 áreas a contemplar.

#### **CRITERIO 1: LIDERAZGO**

Cómo los líderes desarrollan y facilitan la consecución de la misión y la visión, desarrollan los valores necesarios para alcanzar el éxito a largo plazo e implantan todo ello en la organización mediante las acciones y los comportamientos adecuados, estando implicados personalmente en asegurar que el sistema de gestión de la organización se desarrolla e implanta.

#### Subcriterios

- 1a. Desarrollo de la misión, visión y valores por parte de los líderes, que actúan como modelo de referencia dentro de una cultura de Excelencia.
- 1b. Implicación personal de los líderes para garantizar el desarrollo, implantación y mejora continua del sistema de gestión de la organización.
- 1c. Implicación de los líderes con clientes, partners y representantes de la sociedad.
- 1d. Refuerzo por parte de los líderes de una cultura de Excelencia entre las personas de la Organización.
- 1e. Los cambios en la organización son definidos e impulsados por los líderes.

#### CRITERIO 2: POLÍTICA Y ESTRATEGIA

Cómo implanta la organización su misión y visión mediante una estrategia claramente centrada en todos los grupos de interés y apoyada por políticas, planes, objetivos, metas y procesos relevantes.

#### Subcriterios

- 2a. Las necesidades y expectativas actuales y futuras de los grupos de interés son el fundamento de la política y estrategia
- 2b. La información procedente de las actividades relacionadas con la medición del rendimiento, investigación, aprendizaje y creatividad son el fundamento de la política y estrategia
- 2c. Desarrollo, revisión y actualización de la política y estrategia

2d. Comunicación y despliegue de la política y estrategia a través de un esquema de procesos clave

### CRITERIO 3: PERSONAS

Cómo gestiona, desarrolla y aprovecha la organización el conocimiento y todo el potencial de las personas que la componen, tanto a nivel individual, como de equipos o de la organización en su conjunto; y cómo planifica estas actividades en apoyo de su política y estrategia y del eficaz funcionamiento de sus procesos

#### Subcriterios

3a. Planificación, gestión y mejora de los recursos humanos

3b. Identificación, desarrollo y mantenimiento del conocimiento y la capacidad de las personas de la organización

3c. Implicación y asunción de responsabilidades por parte de las personas de la organización

3d. Existencia de un diálogo entre las personas de la organización

3e. Recompensa, reconocimiento y atención a las personas de la organización

### CRITERIO 4: ALIANZAS Y RECURSOS

Cómo planifica y gestiona la organización sus alianzas externas y sus recursos internos en apoyo de su política y estrategia y del eficaz funcionamiento de sus procesos

#### Subcriterios

4a. Gestión de las alianzas externas

4b. Gestión de los recursos económicos y financieros

4c. Gestión de los edificios, equipos y materiales

4d. Gestión de la tecnología

4e. Gestión de la información y del conocimiento

## CRITERIO 5: PROCESOS

Cómo diseña, gestiona y mejora la organización sus procesos para apoyar su política y estrategia y para satisfacer plenamente, generando cada vez mayor valor, a sus clientes y otros grupos de interés.

### Subcriterios

5a. Diseño y gestión sistemática de los procesos

5b. Introducción de las mejoras necesarias en los procesos mediante la innovación, a fin de satisfacer plenamente a clientes y otros grupos de interés, generando cada vez mayor valor

5c. Diseño y desarrollo de los productos y servicios basándose en las necesidades y expectativas de los clientes

5d. Producción, distribución y servicio de atención, de los productos y servicios

5e. Gestión y mejora de las relaciones con los clientes

## CRITERIO 6: RESULTADOS EN LOS CLIENTES

Qué logros está alcanzando la organización en relación con sus clientes externos.

### Subcriterios

6a. Medidas de percepción

Se refieren a la percepción que tienen los clientes de la organización, y se obtienen, por ejemplo, de las encuestas a clientes, grupos focales, clasificaciones de proveedores existentes en el mercado, felicitaciones y reclamaciones.

6b. Indicadores de rendimiento

Son medidas internas que utiliza la organización para supervisar, entender, predecir y mejorar el rendimiento, así como para anticipar la percepción de sus clientes externos.

## CRITERIO 7: RESULTADOS EN LAS PERSONAS



Qué logros está alcanzando la organización en relación con las personas que la integran.

#### Subcriterios

##### 7a. Medidas de percepción

Se refieren a la percepción de la organización por parte de las personas que la integran, y se obtienen, por ejemplo, de encuestas, grupos focales, entrevistas y evaluaciones de rendimiento estructuradas.

##### 7b. Indicadores de rendimiento

Son medidas internas que utiliza la organización para supervisar, entender, predecir y mejorar el rendimiento de las personas que la integran, así como para anticipar sus percepciones.

#### CRITERIO 8: RESULTADOS EN LA SOCIEDAD

Qué logros está alcanzando la organización en la sociedad.

#### Subcriterios

##### 8a. Medidas de percepción

Se refieren a la percepción de la organización por parte de la sociedad, y se obtienen, por ejemplo, de encuestas, informes, reuniones públicas, representantes sociales y autoridades gubernativas.

##### 8b. Indicadores de rendimiento

Son medidas internas que utiliza la organización para supervisar, entender, predecir y mejorar su rendimiento, así como para anticipar las percepciones de la sociedad.

#### CRITERIO 9: RESULTADOS CLAVE

Qué logros está alcanzando la organización con relación al rendimiento planificado

#### Subcriterios

##### 9a. Resultados Clave del Rendimiento de la Organización

Estas medidas son los resultados clave planificados por la organización y, dependiendo del objeto y de los objetivos de la misma, pueden hacer referencia a:

- Resultados económicos y financieros
- Resultados no económicos

#### 9b. Indicadores Clave del Rendimiento de la Organización

Son las medidas operativas que utiliza la organización para supervisar, entender, predecir y mejorar los probables resultados clave del rendimiento de la misma.

### **11.2.7 VENTAJAS DE ADOPTAR EL MODELO EFQM**

Aumentar la competitividad de la organización:

- Siendo más rentables
- Logrando un buen clima de trabajo
- Ofreciendo una excelente calidad de servicio, teniendo en cuenta tanto los requisitos legales como las necesidades y expectativas de los clientes.

### **11.2.8 CAMBIOS QUE ORIGINA LA EXCELENCIA EN GESTIÓN**

Concepto tradicional

- Desconocimiento del cliente
- Los empleados buscan satisfacer a los jefes
- La calidad se refiere a la producción y a las materias primas
- El departamento de calidad es el que asegura la calidad
- Existe una reticencia hacia el cambio
- La organización está dividida en departamentos
- No hay involucración entre departamentos
- La participación y la involucración no es prioritario e incluso es sancionada

- Los jefes son los que deciden
- Gestión cualitativa

### Concepto Excelente

- El cliente es el que manda
- Toda la organización busca satisfacer a los clientes
- La calidad concierne a todas las personas de la organización
- Cada empleado garantiza la calidad
- El entorno es cambiante por lo tanto el cambio es natural en las empresas
- La organización está integrada y cohesionada
- Se estimula y se premia la participación y la involucración
- Los líderes delegan
- Gestión con datos, los indicadores señalan oportunidades de mejora

## **11.3 SERIE ISO 9000.**

### **11.3.1 INTRODUCCIÓN.**

La Norma Internacional UNE EN ISO 9001 es un método de trabajo considerado como el mejor para la mejora de la calidad y de la satisfacción del cliente. En su última revisión, ISO 9001:2008 se clarifican algunos aspectos de su anterior revisión (ISO 9001:2000), manteniendo la esencia de la misma, sin ampliar su especificación.

El Estándar ISO 9000 está basado en un modelo de gestión por procesos que desarrolla los ocho principios de la Gestión de la Calidad.

La nueva versión de la norma ISO 9001:2008 fue publicada en 2008, fruto del trabajo realizado por el Comité ISO TC/176/SC2.

La norma ISO 9001:2008 mantiene de forma general la filosofía del enfoque a procesos y los ocho principios de gestión de la calidad, a la vez que seguirá siendo genérica y aplicable a cualquier organización independientemente de su actividad, tamaño o su carácter público o privado.

Si bien los cambios abarcan prácticamente la totalidad de los apartados de la norma, éstos no suponen un impacto para los sistemas de gestión de la calidad de las organizaciones basados en la ISO 9001:2000, ya que fundamentalmente están enfocados a mejorar o enfatizar aspectos como:

- Importancia relevante del cumplimiento legal y reglamentario.
- Alineación con los elementos comunes de los sistemas ISO 14001
- Mayor coherencia con otras normas de la familia ISO 9000
- Mejora del control de los procesos subcontratados.
- Aumento de comprensión en la interpretación y entendimiento de los elementos de la norma para facilitar su uso.

### **11.3.2 LA FAMILIA DE NORMAS ISO 9000**

*ISO 9000, Quality management systems – Fundamentals and vocabulary (Sistemas de gestión de la calidad – Fundamentos y vocabulario)*

Esta norma describe los conceptos de un Sistema de Gestión de la Calidad (SGC) y define los términos fundamentales usados en la familia ISO 9000. La norma también incluye los ocho principios de gestión de la calidad que se usaron para desarrollar la ISO 9001 y la ISO 9004.

*ISO 9001, Quality management systems - Requirements (Sistemas de gestión de la calidad – Requisitos)*

Esta norma especifica los requisitos de un SGC, con el cual una organización busca evaluar y demostrar su capacidad para suministrar productos que cumplan con los requisitos de los clientes y los

reglamentarios aplicables, y con ello aumentar la satisfacción de sus clientes.

*ISO 9004, Quality management systems – Guidelines for performance improvements (Sistemas de gestión de la calidad – Directrices para la mejora del desempeño)*

Esta norma proporciona orientación para la mejora continua y se puede usar para mejorar el desempeño de una organización. Mientras que la ISO 9001 busca brindar aseguramiento de la calidad a los procesos de fabricación de productos y aumentar la satisfacción de los clientes, la ISO 9004 asume una perspectiva más amplia de gestión de la calidad y brinda orientación para mejoras futuras. Las directrices para autoevaluación se han incluido en el Anexo A de la ISO 9004. Este anexo brinda un enfoque sencillo y de fácil uso para determinar el grado relativo de madurez del SGC de una organización e identificar las principales áreas de mejora.

La ISO 9000 es un punto de partida para entender las normas, ya que define los términos fundamentales usados en la “familia” ISO 9000, o en el grupo de normas relativas a gestión de la calidad. La ISO 9001 especifica los requisitos para un sistema de gestión de la calidad con el cual se pueda demostrar la capacidad de suministrar productos que cumplan los requisitos de los clientes, al igual que los requisitos aplicables; también busca incrementar la satisfacción de los clientes. La ISO 9004 le brinda orientación sobre la mejora continua de su sistema de gestión de la calidad, de manera que se cumplan las necesidades y expectativas de todas las partes interesadas. Dentro de las partes interesadas se incluyen los clientes y los usuarios finales; los directores y personal de la organización; los propietarios e inversionistas; los proveedores y socios, y la sociedad en general.

La ISO 9001 y la ISO 9004 son un “par coherente” de normas que relacionan la gestión de la calidad moderna con los procesos y actividades

de una organización, y enfatizan en la promoción de la mejora continua y el logro de la satisfacción del cliente. La ISO 9001, que se enfoca en la eficacia del sistema de gestión de la calidad para cumplir los requisitos de los clientes, se usa para certificación o para acuerdos contractuales entre proveedores y compradores. Por otra lado, la ISO 9004 no se puede usar para certificación, ya que no establece requisitos sino que proporciona orientación sobre la mejora continua del desempeño de una organización. La ISO 9001 se enfoca en la “eficacia”, es decir, en hacer lo correcto, mientras que la ISO 9004 hace énfasis tanto en la “eficacia” como en la “eficiencia”, es decir, en hacer lo correcto en la forma correcta.

### **11.3.3 PRINCIPIOS DE GESTIÓN**

La ISO 9000 se basa en los 8 principios de gestión:

- Enfoque al cliente, que da como resultado el cumplimiento de los requisitos de los clientes y el esforzarse por excederlos.
- Liderazgo, que apunta a crear un ambiente interno en el cual las personas estén totalmente involucradas.
- Participación del personal, que es la esencia de una organización.
- Enfoque basado en procesos, que da como resultado la mejora de la eficiencia para obtener los resultados deseados.
- Enfoque de sistema para la gestión, que conduce a la mejora de la eficiencia y la eficacia por medio de la identificación, comprensión y gestión de procesos interrelacionados.
- Mejora continua, que se convierte en un objetivo permanente de la organización.
- Enfoque basado en hechos para la toma de decisiones, basado en el análisis de datos e información, y
- Relaciones mutuamente beneficiosas con el proveedor, basado en la comprensión de su interdependencia

Para el manejo de una organización la ISO 9000 estimula la adopción del enfoque basado en procesos. Para el modelo de procesos revisado en la ISO 9000 se consideran cinco áreas principales:

- Sistema de gestión de la calidad
- Responsabilidad de la alta dirección
- Gestión de recursos
- Realización del producto
- Medición, análisis y mejora

El modelo de proceso usado en las normas es completamente compatible con el bien conocido ciclo de PLANEAR, HACER, VERIFICAR, ACTUAR.

La gestión de calidad debe incluir los procesos requeridos para lograr calidad, y resaltar la interacción entre ellos. La alta gerencia debe asumir la responsabilidad por el liderazgo, compromiso y participación activa para desarrollar y mantener el sistema de calidad. La alta dirección debería suministrar los recursos adecuados, de manera que los clientes obtengan lo que se acordó mutuamente. Es necesario contar con procesos bien definidos, tanto operacionales como de soporte, para poder realizar el producto. La satisfacción de los clientes se debe medir y analizar de manera que la organización pueda mejorar continuamente.

#### **11.3.4 VERSIONES ESPECÍFICAS DE LA NORMA ISO 9000**

Las normas para “sectores específicos” son normas de gestión de la calidad destinadas a una industria específica, un producto o grupo de productos. Por ejemplo, existen normas de gestión de calidad específicas para la industria automotriz, la industria de alimentos y bebidas, la industria de las telecomunicaciones, etc.

La familia de normas ISO 9000, genérica por naturaleza, es aplicable a cualquier tipo de producto o servicio y puede ser implementada por cualquier industria. Por tanto, la ISO (Organización Internacional de

Normalización), busca limitar la proliferación de normas en el campo de la gestión de la calidad. El comité técnico ISO 176 (ISO/TC 176), responsable del desarrollo de la familia de normas ISO 9000, apoya el desarrollo de normas para sectores específicos, una vez se haya establecido que hay necesidad de ellas.

### **11.3.5 COSTOS Y BENEFICIOS DE ESTABLECER UN SISTEMA DE GESTIÓN DE LA CALIDAD**

#### 1. Costos...

La implementación de costos en que incurren las compañías se puede pormenorizar en costos directos e indirectos.

Los costos directos incluyen, entre otros, los siguientes:

- Contratación de formadores o consultores externos, si se requieren.
- Envío de personal para recibir formación externa.
- Adquisición de las normas nacionales e internacionales pertinentes de la familia ISO 9000, y los libros y publicaciones relacionadas, y
- Adquisición de equipos adicionales, instrumentos y otros recursos que identifique la compañía.

Los costos indirectos incluyen, entre otros, los siguientes:

- Tiempo empleado por la dirección y demás personal, para el desarrollo del sistema.
- Reorganización de los procesos, incluidas las mejoras en el manejo de la empresa, si se requieren.
- Costos de calibración externa de los equipos, con el fin de asegurar la trazabilidad de las mediciones comparado con patrones de medición trazables a patrones de medición nacionales o internacionales.
- Organización de la formación interna.
- Tiempo gastado por los auditores internos para las auditorías internas periódicas.



- Acciones correctivas, incluida la actualización de manuales y procedimientos, si se requiere.
- Gastos en digitalización de documentos, papelería y otros artículos de consumo requeridos para la preparación de manuales y documentación de procesos, etc.

Algunos factores que pueden ayudar a reducir los costos anteriores incluyen:

- Hacer que el personal de la compañía se familiarice con los requisitos del SGC.
- Contar con actividades documentadas relacionadas con el sistema, como por ejemplo instrucciones de trabajo, planes de calidad, procedimientos, etc., ya implementadas.
- La contratación de consultores únicamente para actividades específicas tales como "análisis de brechas", formación de auditores, auditorías de preevaluación, etc., y contar con personal interno para supervisar las actividades restantes.

De otra parte, hay factores que pueden significar costos de implementación mayores para la compañía. Por ejemplo, si su compañía realiza actividades en diferentes lugares, o está involucrada en el diseño y desarrollo de productos, esto puede aumentar los costos.

## 2. ... y beneficios de obtener una certificación con base en ISO 9000

La implementación de un sistema de gestión de calidad genera beneficios internos a la mayoría de organizaciones, al igual que oportunidades con relación al mundo exterior.

Los beneficios internos para la compañía incluyen:

- Enfoque mejorado hacia el cliente y orientación a los procesos dentro de la compañía.
- Mayor compromiso de la dirección y mejor toma de decisiones.
- Condiciones de trabajo mejoradas para los empleados.

- Aumento de motivación por parte de los empleados.
- Costo reducido de fallas internas (menores tarifas de reprocesos, rechazo, etc.) y fallas externas (menos devoluciones de los clientes, reemplazos, etc.), y último, aunque no el menos importante,
- La mejora continua del sistema de gestión de la calidad.

Se generan los siguientes beneficios externos:

- Los clientes tienen más confianza en que recibirán productos conformes a sus requisitos, lo que a su vez redunda en mayor satisfacción del cliente.
- Una mejor imagen de la compañía.
- Publicidad más agresiva, ya que los clientes pueden estar informados de los beneficios de realizar negocios con una compañía que maneja la calidad de sus productos.
- Más confianza en que los productos de la compañía cumplen los requisitos reglamentarios pertinentes.
- Mejor evidencia objetiva para defenderse contra demandas por obligación civil, si los clientes llegaran a entablar alguna.

### **11.3.6 IMPLEMENTACIÓN DE UN SISTEMA DE GESTIÓN DE LA CALIDAD**

Un sistema de gestión de calidad con base en ISO 9000 se puede implementar en los siguientes pasos:

#### 1. Evaluar la necesidad y metas de la organización con relación a la implementación de un SGC

La necesidad puede surgir a raíz de quejas repetidas de los clientes, devoluciones frecuentes por garantía, entregas retrasadas, altos inventarios, retrasos frecuentes en la producción, un alto nivel de reprocesos, o rechazo de productos o servicios. En esta etapa, identifique las metas que quisiera alcanzar a través de un SGC, tales como la

satisfacción de sus clientes, una mayor participación en el mercado, mejores comunicaciones y moral de la organización, una mayor eficiencia y rentabilidad, etc.

Otro objetivo de implementar un SGC puede ser la demostración de conformidad por medio de una certificación por tercera parte, que puede solicitar un cliente importante, o que se exige para poder registrarse como proveedor de grandes compañías, por ejemplo, los fabricantes de equipos originales (OEMs).

## 2. Obtener información acerca de la familia ISO 9000

Las personas identificadas para iniciar el desarrollo de un SGC con base en ISO 9000 necesitan entender los requisitos de la ISO 9001, conjuntamente con la ISO 9000 y la ISO 9004.

La información de soporte, como por ejemplo los principios de gestión de calidad, preguntas frecuentes (FAQ), orientación sobre el numeral 1.2 (aplicación) de la ISO 9001, orientación sobre los requisitos de documentación de la ISO 9001 y otros folletos, se encuentran disponibles en la página web de ISO: <http://www.iso.org>

## 3. Nombrar un consultor, si es necesario

Si dentro de la organización no se cuenta con la competencia adecuada para desarrollar un SGC, se puede contratar un consultor. Antes de hacerlo, es conveniente verificar sus conocimientos y experiencia; el conocimiento de éste acerca de los procesos de realización del producto de su organización, y su experiencia en ayudar a otras organizaciones a alcanzar sus metas establecidas, incluida la certificación.

## 4. Toma de conciencia y formación

Hay que despertar la conciencia acerca de los requisitos del SGC entre todo el personal que realiza actividades que afectan a la calidad. También planificar y brindar formación específica acerca de cómo desarrollar Manuales de Calidad, cómo planear un SGC, cómo identificar e

implementar procesos de mejora, y sobre cómo auditar la conformidad con el SGC.

#### 5. Realizar el análisis de brechas (Gap analysis)

Se deben evaluar las brechas que hay entre el sistema de gestión de la calidad existente y los requisitos de ISO 9001 para el SGC, y preparar la manera de cerrar estas brechas, incluida la planificación de los recursos adicionales requeridos. El análisis de estas brechas se puede llevar a cabo mediante una autoevaluación o un consultor externo.

#### 6. Procesos de realización del producto

Examinar el numeral 7 de la ISO 9001 relativo a "realización del producto", para determinar cómo los requisitos se aplican o no al SGC de la compañía. Los procesos cobijados por este numeral incluyen:

- Procesos relacionados con el cliente.
- Diseño y desarrollo.
- Compras.
- Producción y suministro del servicio.
- Control de dispositivos de medición y seguimiento

#### 7. Suministrar el personal

Decidir sobre las responsabilidades de las personas que estarán involucradas en el desarrollo y documentación de su SGC, incluido el nombramiento de un representante de la dirección, quien supervisará la implementación del SGC. La creación de un Comité Director del proyecto también puede ser útil para supervisar el progreso y suministrar los recursos cuando estos se requieran.

#### 8. Elaborar el cronograma

Preparar un plan completo para cerrar las brechas identificadas en el Paso 5 para desarrollar los procesos del SGC. En este plan incluir las actividades por realizar, los recursos requeridos, las responsabilidades y un

tiempo de finalización estimado para cada actividad. Los numerales 4.1 y 7.1 de la ISO 9001 brindan información que se debería usar al desarrollar el plan. El tiempo total requerido para cada fase (planificación, documentación, implementación y evaluación) depende de la extensión de las brechas en su SGC existente.

## 9. Redactar el Manual de Calidad

En el Manual de Calidad:

- Incluir cómo se aplica el SGC a los productos, procesos, instalaciones y departamentos de la organización.
- Excluir cualquier requisito que se haya decidido en el paso 6, con su respectiva justificación.
- Hacer referencia o incluir procedimientos documentados para su SGC.
- Describir la interacción entre los procesos del SGC, por ejemplo, la interacción entre los procesos de realización del producto y otros procesos de gestión, medición y mejora, y
- Redactar la política de calidad y los objetivos de calidad de la organización.

El personal involucrado en la organización debería revisar el Manual de Calidad y los procedimientos documentados, de manera que sus comentarios y sugerencias puedan ser tenidos en cuenta antes de que el Manual de Calidad y los procedimientos sean aprobados para publicación y uso. También se debería llegar a una decisión acerca de la fecha de implementación.

## 10. Realización de auditorías internas

Durante la fase de implementación, de aproximadamente tres a seis meses después de que se escribe la documentación, los auditores entrenados deberían llevar a cabo una o dos auditorías internas que cubran todas las actividades del SGC, y la dirección involucrada debería

emprender sin demora las acciones correctivas sobre los hallazgos de auditoría. Cuando se requiera, actualizar los manuales, los procedimientos y los objetivos. Después de cada auditoría interna, la alta dirección debería revisar la eficacia del sistema y suministrar los recursos necesarios para las acciones correctivas y mejoras.

### 11. Solicitud de la certificación

Una vez finalizado satisfactoriamente el Paso 10, y si la compañía decide obtener una certificación por tercera parte, se puede solicitar una certificación a un organismo de certificación acreditado.

### 12. Realización de evaluaciones periódicas

Después de la certificación, la organización debería realizar periódicamente auditorías internas para revisar la eficacia del SGC y ver cómo se puede “mejorar continuamente”. La organización debería evaluar periódicamente si el propósito y metas (ver el Paso 1) para los cuales se desarrolló el SGC se están logrando, incluida su mejora continua.

## **11.4 CMMI**

El Modelo de Madurez de la Capacidad del Software **SW-CMM** (*Software Capability Maturity Model*) fue definido por Paulk, Curtis, Chrisis y Weber en 1.993 como un modelo que establece los niveles por los cuales las organizaciones de software hacen evolucionar sus definiciones, implementaciones, mediciones, controles y mejoras de sus procesos de software. El modelo CMM permite definir el grado de madurez de las prácticas de gestión y reingeniería de software de dichas organizaciones y determinar cuales son las acciones de mejora prioritarias para sus procesos de software.

El modelo CMM se compone de **cinco niveles de madurez** de acuerdo con la capacidad del proceso de software, definidos por los objetivos de los

procesos que, cuando son satisfechos, permiten evolucionar al próximo nivel, ya que uno o más componentes importantes del proceso de software han sido estabilizados.

En cada nivel, se definen un conjunto de **áreas clave del proceso** que describen las funciones de ingeniería del software que deben llevarse a cabo para el desarrollo de una buena práctica. Mediante un amplio conjunto de métricas se determina la calidad de cada una de las áreas clave, obteniéndose una visión precisa del rigor, la eficacia y la eficiencia de la metodología de desarrollo de una organización productora de software.

Cada una de las áreas está organizada en cinco secciones, denominadas **características comunes**. Estas son las siguientes:

- **Compromiso.** Es el conjunto de acciones que la organización debe realizar para poder asegurar que el proceso es repetible y duradero. Normalmente está relacionado con las políticas de la organización y el liderazgo de la dirección.
- **Capacidad.** Describe las precondiciones que deben darse en un proyecto o en la organización para implantar de forma efectiva los procesos de software. Habitualmente afecta a los recursos, a la estructura y a la formación.
- **Actividades.** Describen los roles y los procedimientos necesarios para implantar un área clave de proceso. Habitualmente incluyen procedimientos relacionados con la planificación y el seguimiento del trabajo, así como las acciones correctivas necesarias.
- **Medidas y análisis.** Describen la necesidad de realizar mediciones de los procesos y analizan dichas medidas. Suelen incluir ejemplos de las medidas tomadas para determinar el estado y la eficacia de las actividades realizadas.
- **Verificación.** Describe los pasos que deben llevarse a cabo para asegurar que las actividades se realizan según los procesos

establecidos. Habitualmente incluye revisiones y auditorías por parte de la dirección y del grupo de aseguramiento de la calidad.

A su vez, en cada característica común se especifican unas **prácticas clave**, que son normas, procedimientos y actividades cuya realización lleva a la consecución de los objetivos del área. En algunos casos se detallan subprácticas más específicas, guías e interpretaciones de la práctica y, cuando procede, ejemplos y referencias cruzadas a otras prácticas. Asimismo, el modelo define **indicadores clave**, que son aquellas prácticas clave o componentes de prácticas clave que ofrecen una visión mayor de la consecución de los objetivos de un área clave de proceso.

Los cinco niveles de madurez del modelo son:

- 1- **Inicial**: el proceso de software es un proceso improvisado y caótico. No se han definido procesos metodológicos, o se han definido pero no se siguen. El éxito que se pueda obtener depende de las habilidades, conocimientos y motivaciones del personal. No existen calendarios ni estimados de costos y las funcionalidades y calidad del producto son impredecibles, ni un ambiente estable para el desarrollo y mantenimiento del software. El proceso del software es impredecible por el continuo cambio o modificación a medida que avanza el trabajo.
- 2- **Repetible**: se establecen políticas y procedimientos de administración e implantación del proceso básico para determinar costos, calendarios y funcionalidades. La madurez metodológica de la organización permite estimar fiablemente el tamaño funcional o físico del sistema, así como recursos, esfuerzo, costes y calendario. Se han sentado las bases para repetir éxitos anteriores en proyectos con aplicaciones similares. La organización muestra problemas de calidad y carece de una adecuada estructura para mejorarla. En este nivel,



las áreas clave, cuyo estado se puede conocer mediante diversas métricas, son las siguientes:

- Gestión de requisitos.
- Planificación del proyecto software.
- Seguimiento y control del proyecto.
- Gestión de la subcontratación del software.
- Aseguramiento de la calidad del software.
- Gestión de la configuración del software.

3- **Definido:** el proceso de software para las actividades administrativas y técnicas está documentado, homogeneizado e integrado en un proceso de software estándar dentro de la organización, que ayudará a obtener un desempeño más efectivo. El grupo que trabaja en el proceso enfoca y guía sus esfuerzos al mejoramiento de su desarrollo, facilita la introducción de técnicas y métodos e informa a la administración del estado del proceso. La capacidad del proceso está basada en una amplia comprensión común dentro de la organización de las actividades, roles y responsabilidades definidas en el desarrollo de software. Las áreas claves, definidas para este nivel, son las siguientes:

- Desarrollo y mejora de los procesos de la organización.
- Definición de los procesos de la organización.
- Programa de formación.
- Gestión integrada del software.
- Ingeniería de producto software.
- Coordinación intergrupos.
- Revisión conjunta.

4- **Gestionado:** se recolectan medidas detalladas del proceso de software y de la calidad del producto. Ambos son cuantitativamente entendidos y controlados. Este nivel de capacidad permite a la

organización predecir las tendencias en la calidad del producto dentro de los límites establecidos y tomar las acciones necesarias en caso que sean excedidos. Se puede predecir que los productos de dicha categoría son de alta calidad. Las áreas clave definidas para este nivel son las siguientes:

- Gestión cuantitativa del proyecto.
- Gestión de calidad del software.

5- **Optimizado**: el mejoramiento continuo del proceso es garantizado por la retroalimentación cuantitativa y desde las pruebas de técnicas y herramientas innovadoras. La organización tiene los medios para identificar los puntos débiles y conocer como fortalecerlos. Su actividad clave es el análisis de las causas de defectos y su prevención. Las áreas clave definidas para este nivel son:

- Prevención de defectos.
- Gestión de cambios tecnológicos
- Gestión de cambios en los procesos.

La estructura del CMM brinda un camino progresivo recomendado para las organizaciones dedicadas al desarrollo de software que quieren mejorar la capacidad de su proceso de software. De forma general se identifican los siguientes **usos fundamentales**:

- Equipos de asesores, que usan el modelo para identificar los puntos fuertes y débiles en la organización.
- Equipos de evaluación, que utilizan CMM para identificar el riesgo de seleccionar entre diferentes contratos de negocio y para monitorearlos.
- Personal técnico y de dirección, que usa CMM para entender las actividades necesarias que ayuden a planear e implementar el programa de mejoramiento del proceso de software de la organización.

- Grupo de mejoramiento del proceso, que lo emplean como guía para ayudarse a definir y mejorar el proceso de software en la organización.

Además del Modelo de Madurez de la Capacidad del Software existen el Modelo de Madurez de la Capacidad en la Adquisición de Software (SA-CMM), y el Modelo de Madurez de la Capacidad de las Personas (P-CMM), etc.

Desde el año 1991, el modelo CMM se fue adaptando a múltiples disciplinas: Ingeniería de sistemas, Ingeniería del Software, compras, desarrollo de procesos y productos integrados, etc., derivando modelos diferentes. Las organizaciones que deseaban mejorar sus procesos en todas estas disciplinas se encontraban con que el modelo presentaba grandes diferencias de arquitectura, enfoque, contenido y aplicación. Este hecho provocaba un gran incremento del coste de la implantación de CMM en términos de formación, evaluaciones y actividades de mejora, ya que no existía una integración de todos estos modelos. El **proyecto de integración de CMM o CMMI** (Capability Maturity Mode of Software Integration) surgió como solución a los problemas de falta de integración, y fue puesto en marcha para desarrollar un marco de trabajo simple de mejora de procesos, para organizaciones que persiguen la mejora en todos los ámbitos y niveles de la empresa.

**CMMI** contiene un conjunto de productos que, además de numerosos modelos adaptables a las diferentes áreas de conocimiento, contiene métodos de evaluación según cada modelo así como material de formación. El objetivo inicial de CMM: *“Obtener productos de calidad dentro de los márgenes temporales previstos con el mínimo coste”* no ha cambiado en CMMI. En cambio, CMMI basa la aplicación de todos los principios de CMM a lo largo de todo el ciclo de vida de ingeniería, no únicamente al ciclo de vida del desarrollo del producto software. Además, el conjunto de productos CMMI ha sido diseñado para mantener compatibilidad y correspondencia con el modelo **SPICE**.

En resumen, CMMI puede ser considerado como una extensión del CMM-SW. **Las diferencias principales son:**

- Se han añadido nuevas áreas de proceso.
- Se han añadido mejores y más modernas prácticas clave.
- Se ha añadido un objetivo genérico para cada área de proceso.

Si se analizan estas diferencias en función del Nivel de madurez en que se hallan, se pueden encontrar las siguientes áreas de proceso en el modelo CMMI que no se encuentran en el modelo CMM:

- **Nivel 2.** Medición y análisis. Han sido aisladas de CMM todas las prácticas relacionadas con este objetivo y han sido agrupadas dentro de esta nueva área de proceso.
- **Nivel 3.** El área Ingeniería del producto software de CMM ha sido reemplazada en CMMI por múltiples y más detalladas áreas de proceso:
  - o Desarrollo de requisitos,
  - o Soluciones técnicas,
  - o Integración del producto,
  - o Verificación y Validación.

En el área de Gestión integrada del proyecto de CMM se contemplaba la Gestión de riesgos, pero ahora ha sido considerada como un área de proceso independiente. Finalmente, a este Nivel se ha añadido una nueva área denominada Análisis de decisiones y resolución, que no se encontraba en CMM.

- **Nivel 4.** Este Nivel ha sufrido una reestructuración y las áreas de Gestión cuantitativa de procesos y Gestión de la calidad de software han sido convertidas a Gestión cuantitativa del proyecto y Rendimiento o realización del proceso organizacional, respectivamente.

- **Nivel 5.** Tampoco ha habido grandes cambios en este Nivel, simplemente una fusión de las áreas Gestión de los cambios tecnológicos y Gestión del cambio en los procesos en una única área de proceso: Innovación organizacional y despliegue. El área de Prevención de defectos ha sido reestructurada y renombrada a Análisis causal y resolución.

CMMI v1.1 tiene cuatro disciplinas disponibles:

- Ingeniería de Software: CMMI-SW (Software Engineering)
- Ingeniería de sistemas: CMMI-SE (Systems Engineering)
- Desarrollo integrado del producto y del proceso: CMMI-IPPD (Integrated Product and Process Development)
- Proveedores externos: CMMI-SS (Supplier Sourcing)

CMMI presenta dos representaciones del modelo:

- **Continua:** capacidad de cada área de proceso. En esta representación, se enfoca la capacidad de cada área de proceso para establecer una línea a partir de la cual se puede medir la mejora individual en cada área. Al igual que el modelo por etapas, el modelo continuo tiene áreas de proceso que contienen prácticas, pero éstas se organizan de manera que soportan el crecimiento y la mejora de un área de proceso individual. Hay seis niveles de capacidad, del 0 al 5. La representación continua se centra en seleccionar una cierta área de procesos que mejorar y en fijar el nivel de capacidad deseado para esa área
- **Por etapas:** madurez organizacional. En esta representación se da un mapa predefinido dividido en etapas (los niveles de madurez), para la mejora organizacional y que se basa en procesos probados, agrupados y ordenados y sus relaciones asociadas. Cada nivel de madurez tiene un conjunto de áreas de proceso que indican donde una organización debería enfocar la mejora de su proceso. Cada área

de proceso se describe en términos de prácticas que contribuyen a satisfacer sus objetivos. Las prácticas describen las actividades que más contribuyen a la implementación eficiente de un área de proceso; se aumenta el “nivel de madurez” cuando se satisfacen los objetivos de todas las áreas de proceso de un determinado nivel de madurez.

Ambas representaciones incluyen Metas (Genéricas y Específicas, definiciones de resultados a obtener por la implementación efectiva de los grupos de prácticas) y Prácticas (Genéricas y Específicas, acciones a realizar para cumplir objetivos de área de proceso).

### Bibliografía:

- Ingeniería del Software. Un enfoque práctico. ROGER S. PRESSMAN. Ed. McGraw Hill
- <http://modelosdegestiondelacalidad.blogspot.com/>
- Apuntes y papeles de trabajo de Ingeniería de Sistemas de Información. RAMÓN ORTIGOSA.
- Temario de las pruebas selectivas para ingreso en el Cuerpo Superior de Sistemas y Tecnologías de la Información de la Administración del Estado. ASTIC.
- Enginyeria del Software III. Antonia Mas Pichaco. Universitat de les illes Balears.

### Sitios web:

<a href="http://www.iso.org">http://www.iso.org</a>	International Organization for Standardization
<a href="http://www.aenor.es">http://www.aenor.es</a>	Asociación Española de Normalización y Certificación
<a href="http://www.efqm.org">http://www.efqm.org</a>	European Foundation por Quality Management

**Autor: Ramón Seoane Freijido**

**Director de Sistemas de Información Molduras del Noroeste**

**Colegiado del CPEIG**

**Autor: Hernán Vila Pérez**

**Jefe del Servicio de Informática. Instituto Galego de Vivenda e Solo**

**Vicepresidente del CPEIG**





**12. LA BIBLIOTECA DE  
INFRAESTRUCTURA TI (ITIL).  
SOPORTE AL SERVICIO.  
ENTREGA DE SERVICIOS. ISO  
20000-1:2005, ISO 20000-2:2005.  
OBJETIVOS DE LA NORMA.  
MAPA Y DESCRIPCIÓN DE LOS  
PROCESOS.**

## **Tema 12. La biblioteca de infraestructura TI (ITIL). Soporte al servicio. Entrega de servicios. ISO 20.000-1:2005, ISO 20.000-2:2005. Objetivos de la norma. Mapa y descripción de los procesos.**

### **INDICE**

#### **12.1 La biblioteca de infraestructura TI (ITIL)**

##### **12.1.1 Introducción**

##### **12.1.2 Antecedentes**

##### **12.1.3 ¿Qué es ITIL?**

###### **12.1.3.1 Objetivos**

###### **12.1.3.2 Beneficios**

###### **12.1.3.3 Estructura**

##### **12.1.4 Libros de ITIL V3**

##### **12.1.5 Conclusiones**

#### **12.2 Soporte al servicio. Entrega de servicios.**

##### **12.2.1 Soporte al servicio**

###### **12.2.1.1 Procesos**

##### **12.2.2 Entrega de servicios**

###### **12.2.2.1 Procesos**

#### **12.3 ISO 20.000. Objetivos de la norma**

##### **12.3.1 ¿Qué es ISO 20.000?**

##### **12.3.2 Utilidades del certificado ISO 20.000**

##### **12.3.3 ISO 20.000 e ITIL**

##### **12.3.4 ¿Qué representa exactamente ser conforme a ISO 20.000?**

#### **12.4 Diseño del mapa de procesos ITIL**

##### **12.4.1 Utilidad del modelo de referencia de ITIL**

### **12.1 LA BIBLIOTECA DE INFRAESTRUCTURA TI (ITIL)**

#### **12.1.1 INTRODUCCIÓN**

En la actualidad la dependencia por las TI es un factor crítico en el desarrollo de las organizaciones. El depender de las TI solamente podrá ser vista como positiva siempre y cuando exista una manera de operar que permita aprovechar las TI y las convierta en una ventaja que aporte funcionalidad y flexibilidad institucional. Para lograr lo anteriormente dicho, es indispensable que existan estándares internacionales que orienten a las organizaciones respecto a cómo es posible organizar y estructurar de la mejor manera y a los mejores costos, todos los servicios de TI que giran en torno a la organización, además de lograr que se comuniquen los principales actores que intervienen en el desarrollo de la estrategia del negocio.

Es por ello que actualmente existen una serie de estándares o lineamientos definidos por diversas instituciones de reconocido prestigio, los cuales pretender ofrecer a las organizaciones un marco de trabajo que les permita adoptar nuevas políticas interinstitucionales para la administración organizada y estructurada de los Servicios de TI. Entre los estándares comúnmente reconocidos a nivel mundial se encuentra ISO 9000, COBIT, BS-15000, ITIL, ISO 20000, etc.

ITIL (Information Technology Infrastructure Library) considerado como un modelo que permite adoptar “mejores prácticas” en las organizaciones, es el resultado de la unión de varias librerías estandarizadas dedicadas a una práctica en particular dentro de la Gestión de Servicios de TI; ofrece herramientas que facilitan la administración y optimización de las TI en las organizaciones. Es una estrategia que puede ser aplicable en todo tipo de organizaciones pues su propósito final es implementar buenas prácticas en la prestación de servicios de TI.

### **12.1.2 ANTECEDENTES**

El incremento en la dependencia de las Tecnologías de Información (TI), así como la adopción de estándares propios para gestionar la información, incurriendo algunas veces en la duplicidad de esfuerzos en los proyectos o en mayores costos y la calidad de los servicios TI que ofrecía el gobierno británico, conllevaron a que en el año 1980 la Central Computer and Telecommunications Agency (CCTA) desarrollara unas primeras recomendaciones que facilitarían la administración y optimización de las TI, recomendaciones que actualmente se conocen como ITIL (Information Technology Infraestructura Library). Esta iniciativa se convirtió en un marco de trabajo conformado por numerosos volúmenes, que probó su utilidad no solamente en organizaciones gubernamentales sino en todos los sectores, convirtiéndose en la base para todo tipo de empresas, grandes ó pequeñas, que tuvieran la disposición de implementar Mejores Prácticas.

Inicialmente, CCTA se enfocó a recopilar información tendiente a verificar cómo las organizaciones dirigían la administración de sus servicios, logrando analizar y filtrar los diferentes problemas que se generaban; luego, comprobaron la utilidad y los beneficios de sus recomendaciones. En la década de los 90s, muchas empresas del gobierno europeo adoptaron este marco de trabajo, convirtiéndose en una buena práctica para la administración de los servicios de TI.

En el año 2001, la CCTA y todas las actividades que estaban bajo su control pasaron a formar parte de la OGC (Office of Government Commerce), oficina del Ministerio de Hacienda Británico, convirtiéndose de esta manera en la nueva entidad propietaria de ITIL, cuya finalidad era ayudar a modernizar la provisión de TI del gobierno británico, a través del uso de buenas prácticas, logrando de esta manera el mejor valor monetario en sus relaciones comerciales. Posteriormente, la OGC publicaría nuevas versiones de librerías de buenas prácticas, escritas por expertos internacionales de diversas organizaciones del sector público y privado.

En 1991, se crea en el Reino Unido la red mundial de grupos de usuarios de las TI que ofrecen mejores prácticas y guías basadas en estándares para la Gestión de Servicios de TI; esta red es denominada itSMF (Information Technology Service Management Forum); es el único grupo dedicado exclusivamente a este tipo de gestión siendo reconocido internacionalmente.

Está presente en varios países de Europa y en algunos de América Latina, trabajando en asociación con la OGC, BSI (British Standard Institute (BSI), la Information Systems Examination Board (ISEB) y Examination Institute of the Netherlands (EXIN) y contribuyendo de esta manera a la industria de las Mejores Prácticas. Los capítulos desarrollados por itSMF, fomentan el intercambio de información y experiencias vividas, orientando a las organizaciones de TI en la implementación y mejoras en los servicios que ofrecen.

### **12.1.3 ¿QUÉ ES ITIL?**

ITIL (Information Technology Infrastructure Library) se considera una colección de guías especializadas en temas organizacionales enfocadas a la planeación, el suministro y soporte de servicios de TI. Se reconoce como un estándar global que resume las mejores prácticas para el área de la Gerencia de Servicios de TI, enfocadas específicamente a describir qué funciones o procesos son los que se recomienda desarrollar, mas no en el cómo desarrollarlos; para este último, es responsabilidad de la organización definir las estrategias y métodos necesarios para implementarla, siempre y cuando se adapten al tamaño, a la cultura y a las necesidades internas de la organización.

ITIL ofrece un marco de trabajo para las actividades del área de TI, proporcionando una descripción de los roles, tareas, procedimientos y responsabilidades que pueden ser adaptados en organizaciones de TI cuya

finalidad sea mejorar la Gestión de sus Servicios; gracias a la cantidad de temas que cubre, se considera un elemento de referencia útil para las organizaciones, ya que permite fijar nuevos objetivos de mejora para la organización de TI, basándose en la calidad del servicio y en el desarrollo de los procesos de una manera eficaz y eficiente.

En varias ocasiones, las mejores prácticas se han considerado como procesos que cubren las actividades más importantes a tener en cuenta dentro de las organizaciones de servicios de TI, por lo que se puede afirmar que ITIL es una colección coherente de las mejores prácticas desarrolladas en la industria y no solamente puede ser adaptado al sector público sino también al privado.

Las publicaciones de ITIL describen cómo pueden ser optimizados y coordinados de una mejor manera todos aquellos procesos que han sido previamente identificados y que intervienen en la administración y operación de la infraestructura de TI, tales como el desarrollo del servicio, la gestión de la infraestructura y la provisión y soporte de los servicios; de igual manera, revelan cómo estos pueden ser formalizados dentro de una organización, brindando un marco de trabajo que facilita unificar la terminología relevante dentro de la organización y permite a las personas hablar un lenguaje común, permitiendo así definir objetivos claros e identificar los recursos y el esfuerzo necesarios para su cumplimiento.

#### **12.1.3.1 OBJETIVOS**

Específicamente ITIL se concentra en ofrecer servicios de alta calidad, dando especial importancia a las relaciones establecidas con los Clientes, para lo cual el departamento de TI debe proveer y cumplir con todos los acuerdos de servicios previamente definidos con ellos, y para lograrlo es necesario que exista una fuerte relación entre estos dos; es por esto que algunos de los objetivos de ITIL están relacionados con:

- Promover la visión de IT como proveedor de servicios.
- Generar mejoras en la calidad del suministro de los servicios de TI.
- Fomentar la reducción de costos de los servicios de TI.
- Alinear la prestación de los servicios de TI con las necesidades actuales y futuras del negocio de las organizaciones, además de mejorar la relación con los Clientes.
- Estandarizar los procesos que forman parte de la Gestión de Servicios de TI.
- Promover el uso de un lenguaje común por parte de las personas para mejorar la comunicación dentro de las organizaciones.
- Servir de base para la certificación de las personas y de las organizaciones que desean adoptar este estándar.
- Mejorar la eficiencia, aumentando la efectividad.
- Reducir los posibles riesgos que se pueden presentar.

### **12.1.3.2 BENEFICIOS**

ITIL centra sus esfuerzos en la satisfacción de los requerimientos organizacionales con la mejor relación costo/beneficio, a través de la descripción de un enfoque sistémico y profesional de la Gerencia de Servicios de TI. Algunos de los beneficios que se consiguen con la adopción de las mejores prácticas manejadas en ITIL están relacionados directamente con el Cliente y con la organización; principalmente tienen que ver con:

- El suministro de los servicios de TI se orienta especialmente al Cliente y los acuerdos sobre la calidad del servicio mejoran la relación entre el departamento de TI y el Cliente.
- La mejora en los niveles de satisfacción de los Clientes por medio de medidas objetivas y eficacia en la disponibilidad y desempeño de la calidad de los servicios de TI.
- Implantación de estándares que permitan realizar el control, la administración y operación de los recursos de la organización.

- Los servicios ofrecidos son descritos en un lenguaje más cómodo y con mayores detalles para los Clientes.
- Se gestionan de una mejor manera la calidad, disponibilidad, fiabilidad y coste de los servicios ofrecidos en la organización.
- Mejoras en la comunicación con el departamento de TI en el momento de acordar los puntos de contacto.
- El departamento de TI genera una estructura clara, centrada en los objetivos corporativos de una manera eficaz.
- Soporte a los procesos de negocio y las actividades de los decisores de TI.
- El departamento de TI cuenta con un mayor control sobre la infraestructura y los servicios que tiene a cargo, obteniéndose una visión clara de la capacidad del departamento; además, permite administrar los cambios de una manera sencilla y fácil de manejar.
- La definición de funciones, roles y responsabilidades en el área de los servicios.
- Es posible identificar, a través de una estructura procedimental, la externalización de algunos de los elementos de los servicios de TI.
- Suministro de servicios de TI que satisfagan las necesidades del negocio de la organización.
- Incremento y mejoras en la productividad y eficiencia organizacional a través de las experiencias vividas y los conocimientos adquiridos.
- Genera un cambio cultural hacia la provisión de servicios y sustenta la introducción de un sistema de gestión de calidad.
- Establece un marco de trabajo coherente para las comunicaciones tanto internas como externas, permitiendo contar con la identificación y estandarización de los procedimientos a seguir.
- Mejoras en la satisfacción del personal de la organización reduciendo notablemente su rotación.
- Mejoras en la comunicación entre el personal de TI y sus clientes.
- Genera el intercambio de experiencias obtenidas con su adopción.



### **12.1.3.3 ESTRUCTURA**

El marco de trabajo de ITIL está conformado por cinco (5) elementos principales, los cuales tienen directa relación entre sí, ya que el éxito de cada uno de ellos depende de su interacción y coordinación con los demás. Estos elementos son:

- The Business Perspective (La Perspectiva del Negocio)
- Managing Applications (Administración de Aplicaciones)
- Deliver IT Services (Entrega de Servicios de TI)
- Support IT Services (Soporte a los Servicios de TI)
- Manage the Infrastructure (Gestión de la Infraestructura)

Cada una de las publicaciones de ITIL se enfoca en documentar uno a uno los elementos del marco de trabajo, se realiza una descripción general de lo que se requiere para estructurar la Gestión de Servicios de TI y se definen los objetivos, las actividades, los roles, los flujos de comunicación necesarios y las entradas y salidas de cada uno de los procesos que son indispensables en una organización de TI.

Se han realizado tres publicaciones de las mejores prácticas de ITIL, la primera versión (V1) fue inicialmente desarrollada en la década de 1980, estaba conformada por diez (10) libros básicos que se enfocaban a cubrir la Gestión del Servicio, específicamente en sus dos áreas principales: (i) la entrega del servicio de TI y (ii) el soporte a dichos servicios, siendo soportados posteriormente por treinta (30) libros complementarios, que abarcaban diversos temas, desde el Cableado, hasta la Gestión de Continuidad del Negocio; debido a la cantidad de información existente, surge la segunda versión (V2), la cual empezó a ser reestructurada entre 1999 a 2001, cuando ITIL se convierte en la piedra angular para la Gestión del Servicio, reorganizándose de una manera más sencilla, donde la mayoría de la información relacionada con la entrega del servicio y el soporte de los servicios se convierte en la base del marco de trabajo y se agrupa en dos grandes volúmenes, eliminando de esta forma la duplicidad

en la información existente en la primera versión, de ahí que esta versión quedara reorganizada aproximadamente en diez (10) libros; en la tercera versión (V3) liberada en mayo de 2007, se redujeron las publicaciones a cinco (5) volúmenes articulados, los cuales principalmente se enfocan en el concepto y desarrollo del ciclo de vida del Servicio de TI. Ese ciclo se inicia con una definición de la estrategia del servicio, luego se concentra en realizar el diseño del servicio, posteriormente inicia un periodo de transición donde se busca realizar el desarrollo y la implantación del servicio, en seguida se realiza la operación del servicio y finalmente se concentra en proveer una mejora continua del servicio, la cual está relacionada permanentemente con las demás etapas del ciclo de vida.

#### **12.1.4 LIBROS DE ITIL V3**

##### Service Strategy

Se encarga de asegurar que la estrategia del servicio sea definida, se mantenga y se implemente; se introducen nuevos conceptos, tales como la consecución del valor, la definición del mercado y el espacio de solución; se centra en el desarrollo de prácticas que permitan la toma de decisiones, basado en la comprensión del servicio activo, las estructuras y los servicios de la economía con el objetivo final de incrementar la vida económica de los servicios; busca obtener el alineamiento entre las TI y el negocio, no como anteriormente se venía trabajando, donde solamente las TI eran las que debían adaptarse al negocio.

Algunos de los conceptos generales que se abordan en este libro tienen que ver con la definición del servicio, la estrategia del Service Management y la planificación del valor, la identificación de la dirección y del gobierno de los servicios de las TI, la correspondencia existente entre los planes de negocio y las estrategias de los servicios de TI, algunos arquetipos de servicios y tipos de proveedores de servicios, y lo más importante, qué debe formularse, implantarse y revisarse como estrategia del negocio.

### Service Design

Este libro se concentra en definir cómo se diseñará el servicio identificado previamente en la estrategia, a través del desarrollo de planes que la conviertan en realidad; para el diseño de servicios de TI adecuados e innovadores, es necesario establecer e implantar políticas de TI, arquitecturas y alguna documentación pertinente. Dentro de los aspectos abordados en el nuevo proceso de Gestión de Proveedores que forma parte del diseño del servicio, se encuentran el aprovechamiento de la disponibilidad, la capacidad, la continuidad y administración del SLA, así como los conceptos de garantía del servicio y utilidad, los cuales son considerados como aspecto fundamental por los Clientes.

Otros conceptos que son trabajados en este volumen están asociados con el ciclo de vida del servicio, objetivos y elementos en el diseño de los servicios, selección de un modelo de servicios apropiado, identificación de servicios, personas, procesos, herramientas, etc., modelo de costos, un análisis de riesgos y beneficios, y la implementación del diseño del servicio.

### Service Transition

Tiene como objetivo minimizar de una manera eficaz la brecha existente entre los proyectos y las operaciones; se concentra en las acciones que intervienen una vez el servicio diseñado debe ponerse en producción, enfocándose especialmente al papel que desempeña el Change Management y explicando las prácticas existentes para un correcto Release Management, de una manera amplia y con visión a largo plazo, permitiendo que se consideren todos los factores que participan, tales como mecanismos de entrega, riesgos, beneficios y facilidad en la posterior operación continua del servicio diseñado.

La transición del servicio tiene que ver con la calidad y el control de la entrega de las operaciones y proporciona modelos para apoyar la transición orientando la forma para reducir variaciones en la entrega.

### Service Operation

El ciclo de vida de cualquier servicio culmina con su operación, la cual debe ser tan robusta y efectiva que permita obtener una estabilidad en la gestión del servicio en todo momento y de extremo a extremo. En este volumen se explican las actividades necesarias para garantizar operatividad en el día a día, abarca muchas de las disciplinas y conceptos definidas en la V2 de ITIL, específicamente se concentra en los libros de Service Support y Service Delivery. A través de los conocimientos que se adquieren con la prestación real del servicio, puede llegar a influir en la estrategia del servicio, el diseño, la transición y la mejora continua del servicio.

### Continual Service Improvement

Este libro abarca la calidad del servicio en el contexto de mejora continua, además se centra también en ofrecer mejora continua del servicio aún cuando este se encuentre cercano a ser retirado. Uno de los mayores beneficios de este libro es que indica explícitamente las acciones que se deben realizar para la revisión y mejora de los procesos, información que en la V2 no era tan clara.

Algunos de los conceptos que contempla este libro están relacionados con los principios de la mejora continua del servicio, la implantación de la mejora de servicios, algunos elementos del negocio y de la tecnología que pueden conllevar a la mejora continua del servicio y los beneficios generados que favorecen el negocio, la organización y el aspecto financiero.

## **12.1.5 CONCLUSIONES**

Las mejores prácticas de ITIL ofrecen un marco de trabajo que permite a las organizaciones mejorar el nivel de calidad en los servicios de TI ofrecidos, es por esto que su adopción es un paso fundamental y trascendental que debe ser tomado, ya que los beneficios que se van a obtener, no solamente en el mediano sino en el largo plazo, van a permitir que se perciba una mejora continua a nivel institucional.

La adopción de un estándar como el de ITIL implica el desarrollo disciplinado de un proceso enfocado en el ciclo de vida del servicio de TI, en el cual intervienen varios actores de la organización, tanto internos como externos y donde el aporte y contribución de cada uno de ellos con el cumplimiento de las políticas y actividades definidas para todos los procesos, favorecen de manera significativa el éxito de su implantación. La estructura de las publicaciones que han sido liberadas de esta metodología en su V3 permite a las organizaciones adaptarlo de una mejor manera, ya que ofrece las herramientas necesarias que se deben tener en cuenta durante la definición e implantación de los procesos de TI; como el objetivo central es el Ciclo de Vida de los Servicios de TI, presenta de una manera organizada y centralizada todas las actividades que deben ser consideradas durante su implantación, desde la definición de la estrategia del servicio, el diseño del servicio, el periodo de transición por el que debe pasar, la operación del servicio y hasta llegar a obtener una mejora continua del servicio.

Cada uno de los procesos que se han identificado para ITIL tienen estrecha relación entre ellos mismos y de su interacción y comunicación depende en gran parte el éxito de la implantación de las mejores prácticas; el simple hecho de que algunos de los elementos necesarios en los procesos no cumpla con los lineamientos establecidos en el estándar, incrementa las posibilidades de que la adopción se convierta en un fracaso.

Cada vez son más las organizaciones que entran a formar parte de la familia de ITIL, lo que ha ocasionado que ahora los Gerentes empiecen a

preocuparse por la identificación de la manera de cómo deben planear, implementar y administrar exitosamente estas mejores prácticas, convirtiéndose en uno de sus retos laborales y personales. También son varias las organizaciones que ya tienen implementadas exitosamente las mejores prácticas de ITIL y gracias a las experiencias que cada uno de ellos han vivido con su adopción se ha podido enriquecer este modelo, ofreciendo valiosos tips a las nuevas organizaciones que la están implementando.

## **12.2 SOPORTE AL SERVICIO. ENTREGA DE SERVICIOS.**

La Gestión de los Servicios de TI está conformada por dos grandes áreas: Entrega del Servicio (Service Delivery) y Soporte al Servicio (Service Support). En libros independientes se trabaja todo lo relacionado con la perspectiva del negocio, la gestión de la infraestructura, la planeación que se requiere para implementar la gestión del servicio, la gestión de la seguridad y la gestión de las aplicaciones.

### **12.2.1 SOPORTE AL SERVICIO**

Es considerado como uno de los ejes principales de la Gestión del Servicio de TI; el contenido del libro se centra en describir los procesos necesarios para mantener las operaciones funcionando en el día a día; explica cómo el Service Desk es el responsable y soporta la Gestión de Incidentes, proporcionando una base para el soporte a las solicitudes y problemas que se le pueden presentar a los usuarios en una organización. Así mismo, se encarga de explicar cómo la Gestión de Problemas necesita ser, en cierto modo, proactiva y reactiva, además de exponer los beneficios que se pueden obtener cuando se realiza un análisis efectivo de las causas fundamentales que ocasionan los problemas, ofreciendo una amplia visión en la reducción del impacto que se genera cuando existe una suspensión en el servicio a los usuarios.

En resumen, se encarga de describir la manera en que los clientes y usuarios pueden acceder a los servicios que les permitan apoyar el desarrollo de sus actividades diarias y las del negocio, así como la forma en que dichos servicios deben ser soportados; además, se centra en todos los aspectos que intervienen para garantizar que el servicio ofrecido a los usuarios sea un servicio continuo, que esté disponible y que sea de calidad.

### **12.2.1.1 PROCESOS**

#### Service Desk

Es el único punto de contacto entre el cliente y los usuarios con los proveedores de servicios de TI para todo lo relacionado con el suministro de servicios de TI; también es el punto de partida encargado de informar sobre los Incidentes y la toma de solicitudes de servicio realizadas por los usuarios. Su obligación es mantener informados a los usuarios del servicio sobre los eventos, acciones y oportunidades que pueden llegar a afectar de alguna manera la disponibilidad del servicio y por consiguiente la continuidad en el suministro del servicio en el día a día. Todo lo anterior es posible a través del registro, resolución y monitoreo de problemas.

Dentro del marco de trabajo de las mejores prácticas de ITIL, el Service Desk no fue concebido como un proceso sino como una función por desarrollar dentro de la organización del servicio; algunas de las tareas a su cargo incluyen: ser el punto primario de contacto (SPOC) con los clientes y usuarios; recibir y atender todas las solicitudes, consultas e inquietudes de los clientes y usuarios relacionadas con el suministro de los servicios de TI; documentar, priorizar y realizar un seguimiento adecuado a las solicitudes de modificaciones o cambios realizadas por los usuarios; atender todos los procesos de la Gerencia de Servicios definidos por ITIL; mantener informados sobre el estado y progreso de las solicitudes a los usuarios que las realizaron; clasificar las solicitudes recibidas e iniciar su

proceso según los acuerdos del nivel de servicio (SLA) y procedimientos establecidos; cuando se requiera de un soporte de segundo nivel, deberá encargarse de realizar la coordinación del soporte y el suministro del servicio, al igual que la de los proveedores o participación de terceros; gestionar la restauración de los servicios con el mínimo impacto en el negocio, según los SLA's y prioridades del negocio establecidas; cerrar las solicitudes de servicio realizadas por los usuarios y aplicando la evaluación de satisfacción del cliente; realizar seguimiento a los SLA's definidos tomando las acciones necesarias en caso de presentarse incumplimientos; suministrar la información solicitada por la Gerencia de TI para mantener y mejorar la calidad en los servicios ofrecidos.

De acuerdo con las mejores prácticas de ITIL, el Service Desk se clasifica en tres tipos: Call Center (Centro de Llamadas), Help Desk (Mesa de Ayuda) y Service Desk (Centro de Soporte); por su parte, el Help Desk esta categorizado en: Service Desk Local, a través del cual se busca canalizar localmente todas las necesidades del negocio, tornándose práctico en varios sitios que requieren servicios de soporte, pero siendo grandes los costos en los que se puede llegar a incurrir, ya que el servicio es ofrecido en diferentes lugares, lo cual implica la definición de un estándar operacional; el Service Desk Central, busca que todos los requerimientos del servicio sean registrados en una ubicación central, minimizando costos operacionales ya que existe solamente una mesa de ayuda a nivel organizacional que atiende todos los requerimientos; y el Service Desk Virtual cuya finalidad es ofrecer el servicio en cualquier parte del mundo a través de la red, no importa su ubicación física ya que el servicio se encuentra disponible en todo momento, y su éxito se da siempre y cuando todos los usuarios de la organización cuenten con infraestructura tecnológica para poder acceder a ella. En conclusión, la implementación exitosa y ejecución del proceso de Service Desk generará mayores beneficios en la organización, representado en la satisfacción de



los clientes, minimización de costos, compromiso personal y profesionalidad.

### Configuration Management

Conocida también como Gestión de la Configuración, es parte integral de todos los demás procesos de la Gestión del Servicio; tiene por objetivo controlar los activos y elementos de configuración que forman parte de la infraestructura de TI, por lo cual se encarga de todos los procesos, herramientas y técnicas necesarias para lograrlo; también es su responsabilidad proporcionar información confiable y actualizada no solamente de los elementos específicos de la infraestructura (Elementos de Configuración ó CIs) necesarios para ejecutar los procesos del negocio, sino también sobre las relaciones entre ellos mismos, asegurando la integración con las demás disciplinas de la Gestión del Servicio; permite el desarrollo de los servicios informáticos de mejor calidad de una manera viable económicamente y suministra información importante para el cálculo de los costos y la facturación de los servicios ejecutados. Las solicitudes de cambio sobre los CIs, se registran en una base de datos creada para la Gestión de la Configuración denominada Configuration Management Database (CMDB); en esta base de datos se encuentran registrados todos los datos de los CIs requeridos para la prestación del servicio, desde su descripción e interconexión, hasta un nivel de detalle que incluye la categoría, las relaciones, los atributos y los posibles estados en los cuales puede estar en determinado momento; es necesario actualizar la CMDB cada vez que se realiza un cambio en la infraestructura y dicho cambio esté relacionado con la gestión de la configuración.

### Incident Management

El objetivo de este proceso es resolver cualquier incidente que genere una interrupción en la prestación del servicio, restaurándolo nuevamente de la manera más rápida y efectiva posible; este proceso no se detiene en encontrar, evaluar y analizar cuales fueron las causas que

desencadenaron la ocurrencia de dicho incidente, el cual generó una interrupción en el servicio, sino simplemente se limita a solucionarlo temporalmente y a restaurar el servicio de cualquier manera, lo que probablemente puede llegar a generar nuevas interrupciones al servicio por el mismo incidente; los incidentes se reportan sobre los CIs.

Una de las mayores contribuciones que se le atribuyen a las mejores prácticas de ITIL fue la de establecer la diferencia que existe entre los incidentes y los problemas, donde se distingue entre la restauración rápida del servicio (Incident Management) y la identificación y corrección total de la causa que ocasionó el incidente (Problem Management), pero se destaca la articulación que debe existir entre estos dos procesos, al igual que con Change Management (Gestión de Cambios) y el Service Desk; se recomienda que todas las modificaciones realizadas a los incidentes sean relacionados en la misma CMDB lo mismo que los registros de problemas, errores conocidos y cambios, pues de esta manera será más fácil identificar si la ocurrencia del incidente puede convertirse posteriormente en un problema, lo que permite analizar y buscar su solución definitiva o corrección total, además de que se evita la ocurrencia de nuevos incidentes como consecuencia de los cambios implementados. Algunas de las tareas a cargo de este proceso tienen que ver con: identificación y documentación de todos los reportes que se hayan realizado de los incidentes ocurridos, incluyendo informes e investigaciones; priorizar y categorizar los reportes de los incidentes que se han generado; proporcionar un análisis inicial del incidente ofreciendo un soporte de primer nivel; escalar, cuando sea necesario, la ejecución de soporte de segundo y tercer nivel; cuando se encuentre en riesgo el cumplimiento en los SLAs, es necesario incrementar la asignación de recursos que trabajan en la solución del incidente; resolver la situación en el menor tiempo posible, restaurando el servicio; cerrar y documentar los incidentes ocurridos; realizar seguimiento exhaustivo a cada uno de los incidentes que se presentan (monitoreo, revisión y comunicación del progreso); realizar una evaluación de los incidentes

reportados, analizarlos y generar informes sobre posibles mejoras al servicio.

### Problem Management

Este proceso se dedica a identificar las causas (origen) que ocasionan los problemas que se presentan en la infraestructura de TI y su solución definitiva para evitar nuevas ocurrencias; cuando existe un incidente que se repite más de una vez, es posible que posteriormente se pueda convertir en un problema, aunque la idea es evitar que esto suceda siendo proactivos y previniendo nuevas ocurrencias cuando sea posible; de ahí que se hable de la gestión de problemas proactiva, en donde los incidentes son detectados con demasiada anterioridad de tal manera que permita tomar las acciones preventivas necesarias garantizando que el servicio permanezca disponible y no se vea afectado en ningún momento.

Como resultado de la identificación temprana de los incidentes, las acciones preventivas que se recomienda realizar se traducen en la ejecución de cambios en los CIs interactuando de esta forma con la Gestión de Cambios; este proceso también se relaciona con la Gestión de Incidentes, ya que requiere de un registro preciso y completo de todos los incidentes con el fin de identificar eficiente y eficazmente su causa y las tendencias en su ocurrencia; además, una vez se encuentre solución a un problema, los incidentes que previamente habían sido reportados y que tenían directa relación con las causas del problema podrán pasar a un estado Cerrado o Datos de Baja.

### Change Management

Este proceso tiene una estrecha relación con el proceso de Configuration Management, ya que de la exactitud de los datos de los elementos de la infraestructura (CMDB) es posible garantizar que el análisis del impacto es realizado y conocido, logrando tramitar de esta manera los cambios necesarios a través de procesos y procedimientos estandarizados

y consistentes; se encarga de dirigir la aprobación para realizar cualquier cambio, así como de controlar la implementación de los cambios de la infraestructura de TI.

Dentro de sus objetivos se encuentran: realizar una valoración de los cambios y garantizar que pueden ejecutarse ocasionando el mínimo impacto en la prestación de los servicios de TI y en la infraestructura actual o nueva, y asegurar de manera simultánea la trazabilidad de los cambios; implementar los cambios autorizados y requeridos para el cumplimiento de los SLAs de manera eficiente, efectiva, económica y oportuna; minimizar los cambios, evitando que se implementen cambios no autorizados.

### Release Management

Con la implementación de los cambios, se puede generar como resultado la instalación de nuevo hardware, la instalación de nuevas versiones de software ó simplemente la actualización o generación de nueva documentación; es por ello que todas estas acciones deben ser controladas y distribuidas de una manera organizada, como parte de un nuevo paquete o versión.

Este proceso está asociado con la correcta implantación de todas las versiones de los CIs requeridas para la prestación de un SLA, proporcionando un marco de trabajo para la coordinación, el control y la introducción física de un cambio; se encarga de llevar el control de todos los cambios y nuevas versiones que se han generado como resultado de la implementación de un cambio o de una nueva adquisición (ej. Nuevo software instalado en una máquina); es importante tener claridad en las relaciones que existen entre los CIs para que cuando se realice un cambio de versión se conozca con certeza qué acciones o consideraciones es necesario tener en cuenta y a qué otros CIs se está afectando, además de mantener los registros actualizados en la CMDB.

### **12.2.2 ENTREGA DE SERVICIOS.**

Considerado otro de los ejes fundamentales de la Gestión del Servicio de TI, el Service Delivery se concentra en describir todos los aspectos que deben tenerse en cuenta para realizar una planificación y mejora continua del servicio de TI a largo plazo y en todos los procesos que intervienen para que la prestación del servicio se mantenga y se suministre de tal manera que satisfaga las necesidades actuales y futuras del negocio; algunos de los aspectos que describe están relacionados con la gestión de los niveles del servicio, los niveles de seguridad requeridos, la viabilidad financiera de los servicios, su capacidad, continuidad y disponibilidad, entre otros.

#### **12.2.2.1 PROCESOS**

##### Availability Management

Este proceso se encarga de garantizar que los servicios de TI puedan ser accedidos de una manera confiable y se encuentren disponibles y funcionando correctamente cada vez que los Clientes o usuarios así lo requieran, enmarcados en los SLAs que se hayan definido para la prestación del servicio.

Dentro de los objetivos definidos para este proceso se encuentra realizar el diseño de los servicios de TI con el nivel de disponibilidad requerido por el negocio, garantizar que exista una disponibilidad no solamente en los servicios de TI sino también en su infraestructura, de tal forma que cumpla con los SLAs establecidos, generar reportes de disponibilidad que demuestren la confiabilidad y mantenibilidad del sistema y minimizar la frecuencia y duración que tarda la solución de un incidente.

##### Capacity Management

El objetivo de este proceso es asegurar la existencia de cierta capacidad a nivel de infraestructura de TI, la cual debe encontrarse disponible constantemente para satisfacer los requerimientos del negocio a

nivel del volumen de transacciones, el tiempo de proceso, el tiempo de respuesta y ante todo contemplando su viabilidad cuantitativa y económica para no incurrir en costos desproporcionados.

Como su nombre indica, la gestión de la capacidad se concentra en verificar y garantizar que todos los servicios de TI estén soportados con la suficiente capacidad de proceso y almacenamiento y que además esté dimensionada de tal manera que no implique costos innecesarios para la organización, pero que tampoco genere insatisfacción en los Clientes o usuarios debido a la escasa calidad en la prestación del servicio.

#### Financial Management for IT Services

Se concentra en realizar un adecuado manejo del recurso financiero (ingresos y gastos) asociado a lo que implica la prestación de los servicios de TI, siempre enfocándose en el cumplimiento de los SLAs definidos; determina cuál es el manejo financiero asociado para cada uno de los recursos que participan en el suministro de un servicio, buscando mantener un balance permanente. Mantiene una estrecha relación con la gestión de la capacidad, la gestión de la configuración y la gestión de niveles de servicio, y a través de la información que cada uno de ellos provee, es posible determinar exactamente cuál es el costo real de un servicio.

#### IT Service Continuity Management

La función principal de este proceso es evitar que una grave e imprevista interrupción en el servicio atente contra la continuidad del negocio, por lo que se concentra en la preparación y planificación de las medidas que se deben tomar para recuperar el servicio en caso de que algún desastre ocurra.

Busca asegurar la disponibilidad del servicio a través de la toma de medidas preventivas que se orienten a reducir la probabilidad de fallas y en el evento de que ocurra algún fenómeno considerado como catastrófico o

desastre, el servicio pueda ser restablecido en el menor tiempo posible y con las menores pérdidas de información para la organización.

### Service Level Management (SLM)

Se encarga de definir los servicios de TI ofrecidos, formalizándolos a través de Acuerdos de Niveles de Servicio (SLA) y Acuerdos de Nivel Operativo (SLO); realiza una evaluación del impacto que ocasionan los cambios sobre la calidad del servicio y los SLAs una vez estos cambios hayan sido propuestos e implementados, asegurando de esta manera que cualquier impacto negativo sobre la calidad en los servicios de TI sea relativamente bajo; también se encarga de la creación de planes y emisión de informes respecto a la calidad del servicio que se está ofreciendo.

Explica la importancia de establecer una buena relación con los Clientes y de esta manera asegurar que las necesidades de las empresas sean entendidas; es por esto que el Service Level Management se enfoca también en conocer las necesidades de los Clientes, definir correctamente los servicios que serán ofrecidos y monitorear la calidad de los servicios ofrecidos por medio de los SLAs definidos.

Algunos de los aspectos más importantes que se deben considerar en la definición de los SLAs están relacionados con la descripción del servicio y sus características de funcionamiento, con la disponibilidad del servicio, es decir, cuál es el tiempo que la organización se compromete a mantener el servicio disponible a sus Clientes, para lo cual también es indispensable que sean acordados tiempos de reacción (mínimos y máximos) en la resolución de incidentes, de ahí que los SLAs dependan de la solución de los incidentes en los tiempos acordados. Otro aspecto a tener en cuenta tiene que ver con los objetivos de disponibilidad, seguridad y continuidad del servicio, las obligaciones que recaen tanto en los Clientes como en los Proveedores y las horas críticas del negocio, entre otros.

## **12.3 ISO 20.000. OBJETIVOS DE LA NORMA**

### **12.3.1 ¿QUÉ ES ISO 20.000?**

ITIL muestra todo lo que se debe hacer para que los usuarios ofrezcan servicios de TI adecuados cumpliendo los procesos de su empresa. Para personas individuales es posible obtener certificaciones de ITIL pero hasta el momento no ha sido posible para una organización de TI presentar pruebas de que trabaja según las recomendaciones de ITIL.

Las normas ISO fueron concebidas para llenar este vacío. En base a ITIL las organizaciones itSMF y BSI (British Standard Institute) elaboraron una normativa que define los requisitos de la gestión de servicios a organizaciones de TI.

En la actualidad, la normativa del BSI se conoce internacionalmente como normativa ISO 20000 y une los enfoques de ITIL y COBIT. ISO 20000 abre las puertas a las organizaciones de TI para que puedan obtener por primera vez una certificación.

Aquellas organizaciones que aspiren a lograr una certificación según ISO 20000 deben cumplir los requisitos formulados en la normativa —UNE-ISO/IEC 20000, Parte 1: Especificaciones, en los que se fijan los requisitos obligatorios que debe cumplir toda organización que desee una certificación según esta normativa.

Los requisitos centrales de la normativa ISO 20000 a una organización de TI son:

- o El alineamiento de los procesos de TI según las normas de ISO 20000, que corresponden esencialmente a las recomendaciones de la Gestión del Servicio de ITIL (en especial tras la introducción de ITIL V3).



- o El uso de un método de gestión en la organización de TI según las normas ISO 9001, basado en los principios de la gestión de procesos y dirigido a una mejora continua de la calidad.

La norma ISO/IEC 20000 está estructurada en dos documentos:

- ISO/IEC 20000-1. Este documento de la norma incluye el conjunto de los “requisitos obligatorios” que debe cumplir el proveedor de servicios TI, para realizar una gestión eficaz de los servicios que responda a las necesidades de las empresas y sus clientes.
- ISO/IEC 20000-2. Esta parte contiene un código de prácticas para la gestión de servicios (“Code of Practice for Service Management”) que trata cada uno de los elementos contemplados en la parte 1 analizando y aclarando su contenido. En síntesis este documento pretende ayudar a las organizaciones a establecer los procesos de forma que cumplan con los objetivos de la parte 1.

### **12.3.2 UTILIDADES DEL CERTIFICADO ISO 20.000**

Un certificado ISO 20000 demuestra que una organización de TI:

- está orientada a las necesidades de los clientes,
- está en condiciones de prestar servicios que cumplen con los objetivos de calidad fijados,
- utiliza sus recursos de forma económica.

Este certificado supone siempre una ventaja sobre la competencia. Cada vez hay más clientes que esperan una certificación ISO 20000 de su proveedor de TI, con lo cual el certificado se convierte en una condición imprescindible para ganar cuota de mercado.

Pero también para la empresa misma, trabajar según los principios de ISO 20000 (e ITIL) conlleva una serie de beneficios. La normativa tiene como objetivo proveer a los negocios con los servicios de TI que realmente necesite y ocuparse siempre de que esto suceda de forma eficiente.

Empezar una iniciativa ISO 20000 es una buena forma de impulsar la introducción de mejores prácticas en la organización de TI y mantener a la larga la motivación para su implementación.

### **12.3.3 ISO 20.000 E ITIL**

ISO 20000 fija requisitos a los procesos sin ocuparse de cómo deben ser conformados tales procesos de forma concreta.

Ahí es donde aparece ITIL en escena: ITIL (y más especialmente la nueva versión 3) se orienta a la normativa ISO 20000 y presenta un gran abanico de recomendaciones de mejores prácticas, lo que supone una base de partida bien fundamentada para diseñar procesos conforme a ISO 20000.

La introducción de ITIL es por lo tanto la mejor forma de prepararse para una certificación ISO 20000.

### **12.3.4 ¿QUÉ REPRESENTA EXACTAMENTE SER CONFORME A ISO 20000?**

Obtener una certificación ISO 20000 es un proyecto laborioso. La condición más importante al iniciar un proyecto así es determinar qué objetivo se persigue. Concretamente debe responderse a la pregunta: ¿Cómo debe ser la organización de TI al final del proyecto?

La normativa deja abierta sin embargo esta cuestión, ciñéndose sólo a nombrar los requisitos sin especificar cómo deben cumplirse. Por eso no existe una respuesta válida a la pregunta sobre lo que representa “ser conforme a ISO 20000”

Así las cosas, no es de extrañar que al empezar una iniciativa ISO 20000 se haga evidente un gran problema: No queda claro cómo debe estructurarse de forma concreta la labor de una organización de TI para cumplir los requerimientos de la normativa ISO y por lo tanto no es fácil determinar los cambios necesarios para ello.

Aquí es donde ITIL puede prestar una ayuda decisiva ya que ISO 20000 está orientada a ITIL.

Los conocimientos sobre ITIL se adquieren normalmente mediante libros o, de forma alternativa, con el Mapa de Procesos ITIL® V3 en combinación con el ITIL - ISO 20000 Bridge

## **12.4 DISEÑO DEL MAPA DE PROCESOS ITIL**

El Mapa de Procesos ITIL® V3 es un modelo de referencia íntegro de ITIL. Contiene la descripción completa de forma gráfica de todos los procesos estándar en la Gestión de Servicios de TI según ITIL V3. El modelo de procesos muestra cómo funciona ITIL en la práctica: le ahorra trabajo a la hora de convertir en procesos implementables las múltiples normas recogidas en la bibliografía sobre ITIL.

El Mapa de Procesos ITIL® V3 ha sido creado para organizaciones de TI y proveedores de servicios de TI que

- tengan previsto introducir por primera vez, parcial o totalmente, la Gestión de Servicios de TI según ITIL V3,
- deseen revalorar los procesos ya introducidos de ITIL en base a ITIL V3

- quieran orientarse según ISO 20000 y/o
- se estén preparando para una certificación según ISO 20000.

#### **12.4.1 UTILIDAD DEL MODELO DE REFERENCIA DE ITIL**

El Mapa de Procesos ITIL® V3 está organizado para prestar el apoyo óptimo en todos los pasos de cualquier proyecto ITIL o ISO 20000, desde la primera planificación hasta una organización de TI que funcione según los principios de las mejores prácticas. Su uso ofrece ventajas decisivas:

- El tratamiento gráfico y navegable de los contenidos de ITIL facilita la comprensión de los procesos ITIL y de sus interrelaciones. Con el modelo de procesos de ITIL puede aclarar ITIL a todos los colaboradores de su organización de TI de una forma muy efectiva y económica.
- Los modelos de procesos, claramente estructurados, y las guías complementarias sirven de hilo conductor a la hora de incorporar su proyecto a la implementación de ITIL y llevarlo a cabo. En la definición y documentación de procesos se reducirá su trabajo, ya que adaptará los procesos de referencia existentes a las necesidades de su organización sin tener que empezar con una hoja en blanco.
- El Mapa de Procesos es una documentación de procesos profesional con la que la gestión de TI estará en la ventajosa situación de poder demostrar a los clientes que la organización de TI realiza su labor de forma planificada, orientada al cliente y de calidad.

Bibliografía:

Sitios web:

<http://www.itiil-officialsite.com/>

<http://www.best-management-practice.com/>

<http://iso20000enespanol.com/>

Autor:

Ramón Seoane Freijido

Director de Sistemas de Información Molduras del Noroeste

Colegiado del CPEIG

# **ASPECTOS LEGALES**

**13. LEY DE ACCESO  
ELECTRÓNICO DE LOS  
CIUDADANOS A LOS SERVICIOS  
PÚBLICOS. DECRETO 198/2010  
POR EL QUE SE REGULA EL  
DESARROLLO DE LA  
ADMINISTRACIÓN ELECTRÓNICA  
EN LA XUNTA DE GALICIA Y EM  
LAS ENTIDADES DEPENDIENTES.  
REAL DECRETO 3/2010 POR EL  
QUE SE REGULA EL ESQUEMA  
NACIONAL DE SEGURIDAD EN EL  
ÁMBITO DE LA ADMINISTRACIÓN  
ELECTRÓNICA. REAL DECRETO  
4/2010 POR EL QUE SE REGULA  
EL ESQUEMA NACIONAL DE  
INTEROPERABILIDAD EN EL  
ÁMBITO DE LA ADMINISTRACIÓN  
ELECTRÓNICA.**

**Tema 13.- Ley de acceso electrónico de los ciudadanos a los servicios públicos. Decreto 198/2010 por el que se regula el desarrollo de la administración electrónica en la Xunta de Galicia y sus entidades dependientes. Real decreto 3/2010 por el que se regula el esquema nacional de seguridad en el ámbito de la administración electrónica. Real decreto 4/2010 por el que se regula el esquema nacional de interoperabilidad en el ámbito de la administración electrónica.**

## **INDICE**

### **13.1 Ley de acceso electrónico de los ciudadanos a los servicios publicos**

#### **13.1.1 Introducción**

#### **13.1.2 Objeto de la ley**

#### **13.1.3 Ámbito de aplicación**

#### **13.1.4 Finalidades**

#### **13.1.5 Principios.**

### **13.2 Decreto 198/2010 por el que se regula el desarrollo de la administración electrónica en la Xunta de Galicia y sus entidades dependientes.**

#### **13.2.1 Introducción**

#### **13.2.2 Objeto**

#### **13.2.3 Estructura**

### **13.3 Real decreto 3/2010 por el que se regula el esquema nacional de seguridad en el ámbito de la administración electrónica**

#### **13.3.1 Introducción**

#### **13.3.2 Principios**

#### **13.3.3 Objetivos**

#### **13.3.4 Ámbito de aplicación.**

### **13.4 Real decreto 4/2010 por el que se regula el esquema nacional de interoperabilidad en el ámbito de la administración electrónica.**

#### **13.4.1 Introducción**

#### **13.4.2 Objetivos**



### 13.4.3 Ámbito aplicación y análisis

## **1.- LA LEY 11/2007 DE ACCESO ELECTRÓNICO DE LOS CIUDADANOS A LOS SERVICIOS PÚBLICOS. LA CALIDAD DE LOS SERVICIOS PÚBLICOS Y DE ATENCIÓN AL CIUDADANO.**

### 1.1 Introducción.

Durante los últimos años se han producido numerosos cambios muy importantes en las relaciones entre Administración y Gobierno y ciudadanos, el progreso social, económico y tecnológico fomentaron el deseo de cambio y presionaron a la Administración para adaptarse a los nuevos problemas, las nuevas competencias y las necesidades ciudadanas.

La Administración Pública en cumplimiento de su deber de servir con objetividad a los intereses generales y actuar de acuerdo con los principios de eficacia, jerarquía, descentralización, desconcentración y coordinación con sometimiento pleno a la ley y el Derecho (art.103.1 CE) -, debe ser aquí una pieza fundamental para la implementación de las políticas de modernización de las administraciones públicas. Ha de ser capaz de adaptarse a las nuevas realidades para formar parte del proceso de desarrollo económico y social de las sociedades occidentales.

La mejora, y consecuentemente, la modernización de las administraciones públicas, debe ser un proceso continuo, dinámico y constante, en el que participen todos los que forman parte del sector público

En España, el punto de partida lo supuso el Acuerdo de Consejo de Ministros de 15 de noviembre de 1991, ya que en 1992 se aprobaría el Plan de Modernización de la Administración del Estado, compuesto por una serie de medidas que tenían como objetivo mejorar y modernizar la Administración Pública para responder a las necesidades cambiantes de los ciudadanos.

En la actualidad, las políticas de modernización están estrechamente ligadas con el desarrollo de la administración electrónica,

Siguiendo la definición dada por la Comisión Europea: la administración electrónica no es sino “el uso de las tecnologías en las Administraciones Públicas, combinado con cambios organizativos y nuevas aptitudes, con el fin de mejorar los servicios públicos y los procesos democráticos y reforzar el apoyo a las políticas públicas”.

Actualmente un elemento vertebrador del desarrollo de las tecnologías en la Administración lo constituye el denominado PLAN AVANZA II 2011-2015 (Aprobado por el Consejo de Ministros del 16 de julio de 2010)

Tomando como punto de partida el Plan Avanza aprobado en el año 2005, así como el marco europeo en el que se encuadran este tipo de iniciativas, se han identificado 34 retos concretos que debe abordar España en el ámbito de las TIC. En este contexto, la Estrategia 2011-2015 del Plan Avanza 2 va a centrar sus esfuerzos en la consecución de los siguientes 10 objetivos que facilitarán la superación de los retos definidos:

1. Promover procesos innovadores TIC( tecnologías información y comunicación) en las AAPP
2. Extender las TIC en la sanidad y el bienestar social

3. Potenciar la aplicación de las TIC al sistema educativo y formativo
4. Mejorar la capacidad y la extensión de las redes de telecomunicaciones
5. Extender la cultura de la seguridad entre la ciudadanía y las empresas
6. Incrementar el uso avanzado de servicios digitales por la ciudadanía
7. Extender el uso de soluciones TIC de negocio en la empresa
8. Desarrollar las capacidades tecnológicas del sector TIC
9. Fortalecer el sector de contenidos digitales garantizando la mejor protección de la propiedad intelectual en el actual contexto tecnológico y dentro del marco jurídico español y europeo.
10. Desarrollar las TIC verdes

Dentro de ese campo de actuación juega un papel decisivo en el desarrollo del mismo la Ley 11/2007, de 22 de junio, de acceso electrónico de los ciudadanos a los servicios públicos

La Ley 30/1992, de 26 de Régimen Jurídico de las Administraciones Públicas y del Procedimiento Administrativo Común (LRJAP-PC) en su primera versión recogió, en su artículo 45, el impulso al empleo y aplicación de las técnicas y medios electrónicos, informáticos y telemáticos, por la Administración para el desarrollo de su actividad y el ejercicio de sus competencias y que permitió a los ciudadanos relacionarse con las Administraciones cuando fuese compatible con los “medios técnicos de que dispongan”. Además su artículo 38, y posteriormente la Ley 24/2001, pasaron de la informatización de los registros y archivos a los registros telemáticos como forma de relacionarse con la administración, siempre que el interesado señalase este medio como preferente.

Se pasa de una declaración de impulso de la Administración electrónica a la obligación de emplear medios telemáticos, ya que la Ley 11/2007, reconoce el derecho de los ciudadanos a relacionarse con la Administración a través de estos medios.

El servicio al ciudadano exige consagrar su derecho a comunicarse con las Administraciones por medios electrónicos, ya que estas están obligadas a hacerlo mediante el reconocimiento de la Ley del derecho de los ciudadanos a establecer relaciones electrónicas. De ahí que cada Administración (Estatal, Autonómica y local) deba facilitar al ciudadano, entre otros:

- El acceso a la información y servicios de su competencia.
- Presentar solicitudes y recursos.
- Los medios para dirigirse a las demás Administraciones, lo cual conlleva la colaboración entre administraciones.
- Efectuar pagos.
- Acceder a las notificaciones y comunicaciones que les remita la Administración.
- Encontrar información en un punto de acceso único sobre los servicios multicanal o aquellos que se ofrezcan por más de un medio o plataforma.

Los puntos más destacables de la Ley son:

- Los ciudadanos verán reconocidos nuevos derechos en sus relaciones con las Administraciones Públicas.
- La creación de la figura del Defensor del Usuario.
- Las Administraciones tendrán la obligación de hacer estos derechos efectivos a partir de 2009.
- Los trámites y gestiones podrán hacerse desde cualquier lugar, en cualquier momento.
- La Administración será más fácil, más ágil y más eficaz.
- Los ciudadanos pasan a tomar el mando en sus relaciones con la administración.

- Es una Ley de consenso. En su elaboración han participado todas las administraciones, de ciudadanos, de partidos, de empresas y asociaciones.

## 1.2 Objeto de la Ley

- Reconocer el derecho de los ciudadanos a relacionarse con las Administraciones Públicas por medios electrónicos.

- Regular los aspectos básicos de la utilización de las tecnologías de la información en la actividad administrativa, en las relaciones entre las Administraciones Públicas, así como en las relaciones de los ciudadanos con las mismas con la finalidad de garantizar sus derechos, un tratamiento común ante ellas y la validez y eficacia de la actividad administrativa en condiciones de seguridad jurídica.

- Utilizar, por parte de las AA.PP., las tecnologías de la información de acuerdo con lo dispuesto en la Ley, asegurando la disponibilidad, el acceso, la integridad, la autenticidad, la confidencialidad y la conservación de los datos, informaciones y servicios que gestionen en el ejercicio de sus competencias.

## 1.3 Ámbito de aplicación (Disposición final primera).

- Las Administraciones Públicas, entendiendo por tales la Administración General del Estado, las Administraciones de las Comunidades Autónomas y las Entidades que integran la Administración Local, así como las entidades de derecho público vinculadas o dependientes de las mismas.

- Los ciudadanos en sus relaciones con las Administraciones Públicas.

- Las relaciones entre las distintas Administraciones Públicas.

La Ley no será de aplicación a las Administraciones Públicas en las actividades que desarrollen en régimen de derecho privado.

#### 1.4 Finalidades de la Ley

- Facilitar el ejercicio de derechos y el cumplimiento de deberes por medios electrónicos.
- Facilitar el acceso por medios electrónicos de los ciudadanos a la información y al procedimiento administrativo, con especial atención a la eliminación de las barreras que limiten dicho acceso.
- Crear las condiciones de confianza en el uso de los medios electrónicos, estableciendo las medidas necesarias para la preservación de la integridad de los derechos fundamentales, y en especial los relacionados con la intimidad y la protección de datos de carácter personal, por medio de la garantía de la seguridad de los sistemas, los datos, las comunicaciones, y los servicios electrónicos.
- Promover la proximidad con el ciudadano y la transparencia administrativa, así como la mejora continuada en la consecución del interés general.
- Contribuir a la mejora del funcionamiento interno de las Administraciones Públicas, incrementando la eficacia y la eficiencia de las mismas mediante el uso de las tecnologías de la información, con las debidas garantías legales en la realización de sus funciones.
- Simplificar los procedimientos administrativos y proporcionar oportunidades de participación y mayor transparencia, con las debidas garantías legales.
- Contribuir al desarrollo de la sociedad de la información en el ámbito de las Administraciones Públicas y en la sociedad en general.

#### 1.5 Principios generales

- Limitaciones de la utilización de las tecnologías de la información:
  - Las establecidas por la Constitución.
  - El resto del ordenamiento jurídico.

- Principios:

1. El respeto al derecho a la protección de datos de carácter personal en los términos establecidos por la Ley Orgánica 15/1999, de Protección de los Datos de Carácter Personal, en las demás leyes específicas que regulan el tratamiento de la información y en sus normas de desarrollo, así como a los derechos al honor y a la intimidad personal y familiar.

2. Principio de igualdad con objeto de que en ningún caso el uso de medios electrónicos pueda implicar la existencia de restricciones o discriminaciones para los ciudadanos que se relacionen con las Administraciones Públicas por medios no electrónicos, tanto respecto al acceso a la prestación de servicios públicos como respecto a cualquier actuación o procedimiento administrativo sin perjuicio de las medidas dirigidas a incentivar la utilización de los medios electrónicos.

3. Principio de accesibilidad a la información y a los servicios por medios electrónicos en los términos establecidos por la normativa vigente en esta materia, a través de sistemas que permitan obtenerlos de manera segura y comprensible, garantizando especialmente la accesibilidad universal y el diseño para todos de los soportes, canales y entornos con objeto de que todas las personas puedan ejercer sus derechos en igualdad de condiciones, incorporando las características necesarias para garantizar la accesibilidad de aquellos colectivos que lo requieran.

4. Principio de legalidad en cuanto al mantenimiento de la integridad de las garantías jurídicas de los ciudadanos ante las Administraciones Públicas establecidas en la Ley 30/1992, de Régimen Jurídico de las Administraciones Públicas y del Procedimiento Administrativo Común.

5. Principio de cooperación en la utilización de medios electrónicos por las Administraciones Públicas al objeto de garantizar tanto la interoperabilidad de los sistemas y soluciones adoptados por cada una de ellas como, en su caso, la prestación conjunta de servicios

a los ciudadanos. En particular, se garantizará el reconocimiento mutuo de los documentos electrónicos y de los medios de identificación y autenticación que se ajusten a lo dispuesto en la presente Ley.

6. Principio de seguridad en la implantación y utilización de los medios electrónicos por las Administraciones Públicas, en cuya virtud se exigirá al menos el mismo nivel de garantías y seguridad que se requiere para la utilización de medios no electrónicos en la actividad administrativa.

7. Principio de proporcionalidad en cuya virtud sólo se exigirán las garantías y medidas de seguridad adecuadas a la naturaleza y circunstancias de los distintos trámites y actuaciones. Asimismo sólo se requerirán a los ciudadanos aquellos datos que sean estrictamente necesarios en atención a la finalidad para la que se soliciten.

## **2. DECRETO 198/2010 POR EL QUE SE REGULA EL DESARROLLO DE LA ADMINISTRACIÓN ELECTRÓNICA EN LA XUNTA DE GALICIA Y EN SUS ENTIDADES DEPENDIENTES:**

### **2.1 Introducción.**

La administración de la CCAA de Galicia no puede permanecer ajena a los incesantes y cada vez mas frecuentes cambios en el seno de las relaciones con al Administración desde el punto de vista tecnológico, es en este ámbito en donde, tal y como se recoge en la exposición de motivos del Decreto, se pretende conseguir una administración diferente, que tendrá a la electrónica como elemento central en su modernización donde sus efectos reales sobre la población irán encaminados a la utilización de medios y formas que reduzcan la brecha tecnológica creando las condiciones de confianza precisas para el uso de las tecnologías de la información y de la comunicación.



## 2.1 Objeto

Tiene por objeto regular el derecho de los ciudadanos a relacionarse con las administraciones públicas por medios electrónicos, la tramitación de los procedimientos administrativos incorporados a la tramitación telemática, la creación y regulación de la sede electrónica, la creación de la edición electrónica del Diario Oficial de Galicia y del Registro Electrónico, el impulso y desarrollo de los servicios electrónicos y el establecimiento de infraestructuras y servicios de interoperabilidad.

## 2.3 Estructura

El Decreto consta de 40 artículos, agrupados en nueve capítulos, con tres disposiciones adicionales, tres transitorias, una derogatoria y cuatro finales.

En el capítulo I de la norma recoge su objeto, el de regular el derecho de los ciudadanos a relacionarse con las administraciones públicas por medios electrónicos, la tramitación de los procedimientos administrativos incorporados a la tramitación telemática, la creación y regulación de la sede electrónica, la creación de la edición electrónica del Diario Oficial de Galicia y del Registro Electrónico, el impulso y desarrollo de los servicios electrónicos y el establecimiento de infraestructuras y servicios de interoperabilidad.

Establece como medidas de carácter general

- Ordenar e impulsar la Administración electrónica, a fin de mejorar la eficiencia interna, las relaciones intra e inter administrativas y las relaciones con los ciudadanos.
- Garantizar el derecho de los ciudadanos a relacionarse por medios electrónicos con la Administración pública autonómica.

- Contribuir al desarrollo de la sociedad de la información en el ámbito de las administraciones públicas de Galicia.
- Preservar la integridad de los derechos fundamentales relacionados con la intimidad de las personas, para la garantía de la seguridad de los datos y de las comunicaciones y para la protección de los servicios prestados en soporte electrónico.
- Facilitar el acceso de los ciudadanos a los servicios de la Administración electrónica en las oficinas telemáticas integradas de atención a los ciudadanos, basadas en la cooperación interadministrativa, ofreciendo servicios a los ciudadanos en oficinas públicas, con independencia de cual sea la Administración competente para conocer el asunto.
- Posibilitar la intermediación entre administraciones públicas para la resolución de trámites administrativos solicitados a los ciudadanos cuando sean de competencia de la Xunta de Galicia.

El capítulo II establece que la sede electrónica es la dirección electrónica, a través de la cual los ciudadanos acceden a la información, servicios y trámites electrónicos, que representa una fuente de información auténtica en la que el organismo titular identificado con la sede garantiza responsablemente la integridad, veracidad y actualización de la información y los servicios a los que se pueda acceder a través de esta.

La dirección electrónica de referencia de la sede electrónica de la Xunta de Galicia será <https://sede.xunta.es> que será accesible directamente, así como a través del portal [www.xunta.es](http://www.xunta.es), configurándose como un conjunto de páginas web que asegurará:

- La calidad de la información y la coherencia en la navegación.
- La identificación y comunicación segura, mediante los correspondientes certificados electrónicos admitidos por la Xunta de Galicia.
- El acceso al Registro Electrónico, a las comunicaciones y notificaciones y a los formularios para iniciar los procedimientos administrativos o solicitar la prestación de servicios.
- Los principios de accesibilidad de acuerdo con las normas establecidas, estándares abiertos y, en su caso, aquellos otros que sean de uso general por los ciudadanos

El capítulo III regula la creación del Diario Oficial de Galicia en su edición electrónica que tendrá una consideración de publicación única, dotándola de validez jurídica, que sustituirá a la edición impresa.

El capítulo IV trata sobre los mecanismos de identificación y autenticación, estableciendo que los ciudadanos podrán utilizar los siguientes instrumentos de identificación para relacionarse con la Xunta de Galicia y las entidades incluidas en el ámbito de aplicación de este Decreto:

a. En todo caso, los sistemas de firma electrónica incorporados al documento nacional de identidad, para personas físicas.

b. Sistemas de firma electrónica avanzada, incluyendo los basados en certificado electrónico reconocido, admitidos por las administraciones públicas que tengan validez para la Xunta de Galicia y que se especifiquen en la sede electrónica.

c. Sistemas de firma electrónica, como la utilización de claves concertadas en un registro previo como persona usuaria inscrita en el registro de funcionarios habilitados por la Xunta de Galicia.

d. Otros sistemas de identificación que resulten proporcionales y seguros para la identificación de las personas interesadas

El capítulo V regula la tramitación de procedimientos administrativos en el ámbito de la administración electrónica, estableciendo que la gestión electrónica de la actividad administrativa respetará el ejercicio y la titularidad del órgano o entidad que tenga atribuidas sus competencias, así como la obligatoriedad de impulso de la administración electrónica,

Regula la iniciación y tramitación del procedimiento por medios electrónicos, reconociendo que cualquier persona interesada podrá iniciar y tramitar un procedimiento administrativo por medios electrónicos, ante y en relación con la Xunta de Galicia o las entidades incluidas en el ámbito de aplicación de este Decreto, conforme a las previsiones de estas y sin otras limitaciones que las establecidas en las normas y protocolos de aplicación en atención a razones tecnológicas.

Los capítulos VI y VII regulan los aspectos de la gestión y tramitación de los procedimientos administrativos, tanto en el ámbito interno, relativo a comunicaciones y notificaciones, establece que as entidades incluidas en el ámbito de aplicación del presente Decreto utilizarán un sistema de notificación electrónica que acredite la fecha y hora de puesta la disposición de la persona interesada del acto objeto de notificación, así como de la fecha y hora de acceso de ésta a su contenido mediante sistemas de sellado de tiempo, como en el ámbito externo relacionado, en concreto con las copias y documentos

electrónicos, define el documento electrónico este en los términos que se recogen en el anexo de la ley 11/2007, de administración electrónica “ Información de cualquier naturaleza en forma electrónica, archivada en un soporte electrónico según un formato determinado y susceptible de identificación y tratamiento diferenciado.”

El capítulo VIII trata sobre la interoperabilidad, tiene por objeto fomentar la cooperación interadministrativa, figura clave en el mismo es el denominado protocolo de interoperabilidad que es el documento que determinará el procedimiento para incorporar y consumir la información en soporte electrónico de las entidades incluidas en el ámbito de aplicación del presente Decreto.

El capítulo IX concreta las funciones que el órgano de dirección con competencias generales en materia de desarrollo de la administración electrónica lleva a cabo en relación con este Decreto, en desarrollo de las competencias y funciones que le atribuye el Decreto 325/2009, de 18 de junio, de estructura Orgánica de los órganos superiores dependientes de la Presidencia de la Xunta de Galicia para el impulso, gestión y coordinación de la Administración electrónica, como elemento indispensable para la modernización de la Administración pública, la dirección y gestión de todas las actuaciones de la Xunta en materia de tecnologías de la información y las comunicaciones y el establecimiento de directrices tecnológicas que deben seguir todos los órganos de la Xunta de Galicia.

### **3. REAL DECRETO 3/2010 POR EL QUE SE REGULA EL ESQUEMA NACIONAL DE SEGURIDAD EN EL AMBITO DE LA ADMINSTRACION ELECTRONICA.**

#### **3.1 Introducción**

Tiene por objeto regular el Esquema Nacional de Seguridad establecido en el artículo 42 de la Ley 11/2007, de 22 de junio, y determinar la política de seguridad que se ha de aplicar en la utilización de los medios electrónicos a los que se refiere la citada ley y recoge y regula los principios básicos y requisitos mínimos que permitan una protección adecuada de la información.

La finalidad del Esquema Nacional de Seguridad es crear las condiciones necesarias para la confianza en el uso de los medios electrónicos, a través de medidas para garantizar la seguridad de los sistemas, los datos, las comunicaciones, y los servicios electrónicos, que permita el ejercicio de derechos y el cumplimiento de deberes a través de estos medios. Persigue fundamentar la confianza en que los sistemas de información prestarán sus servicios y custodiarán la información de acuerdo con sus especificaciones funcionales, sin interrupciones o modificaciones fuera de control y sin que la información pueda llegar al conocimiento de personas no autorizadas.

Al objeto de crear estas condiciones, el Esquema Nacional de Seguridad introduce los elementos comunes que han de guiar la actuación de las Administraciones Públicas en materia de seguridad de las tecnologías de la información. En particular, introduce los siguientes elementos principales:

Los principios básicos a ser tenidos en cuenta en las decisiones en materia de seguridad.

Los requisitos mínimos que permitan una protección adecuada de la información.

El mecanismo para lograr el cumplimiento de los principios básicos y requisitos mínimos mediante la adopción de medidas de

seguridad proporcionadas a la naturaleza de la información, el sistema y los servicios a proteger.

Tiene en cuenta las recomendaciones de la Unión Europea, la situación tecnológica de las diferentes Administraciones Públicas, así como los servicios electrónicos ya existentes, y la utilización de estándares abiertos así como, en su caso y de forma complementaria, estándares que sean de uso generalizado por los ciudadanos.

En su elaboración se han manejado, entre otros, referentes en materia de seguridad tales como directrices y guías de la OCDE, recomendaciones de la Unión Europea, normalización nacional e internacional, normativa sobre administración electrónica, protección de datos de carácter personal, firma electrónica y Documento Nacional de Identidad Electrónico, así como a referentes de otros países.

Se ha realizado en un proceso coordinado por el Ministerio de la Presidencia con el apoyo del Centro Criptológico Nacional (CCN), con la participación de todas las Administraciones Públicas. A lo largo de los últimos tres años más de un centenar de expertos de las Administraciones Públicas ha colaborado en su elaboración; a los que hay que sumar los numerosos expertos que también han aportado su opinión a través de las asociaciones profesionales del sector TIC; todo ello a la luz del estado del arte y de los principales referentes en materia de seguridad de la información.

### 3.2 Principios básicos del Esquema Nacional de Seguridad.

El objeto último de la seguridad de la información es asegurar que una organización administrativa podrá cumplir sus objetivos utilizando sistemas de información. En las decisiones en materia de seguridad deberán tenerse en cuenta los siguientes principios básicos:

- a) Seguridad integral.
- b) Gestión de riesgos.
- c) Prevención, reacción y recuperación.
- d) Líneas de defensa.
- e) Reevaluación periódica.
- f) Función diferenciada.

Recoge el Real Decreto los requisitos mínimos que deberán tener los sistemas de seguridad, y así el artículo 11 establece que “Todos los órganos superiores de las Administraciones públicas deberán disponer formalmente de su política de seguridad, que será aprobada por el titular del órgano superior correspondiente. Esta política de seguridad, se establecerá en base a los principios básicos indicados y se desarrollará aplicando los siguientes requisitos mínimos:

- a) Organización e implantación del proceso de seguridad.
- b) Análisis y gestión de los riesgos.
- c) Gestión de personal.
- d) Profesionalidad.
- e) Autorización y control de los accesos.
- f) Protección de las instalaciones.
- g) Adquisición de productos.
- h) Seguridad por defecto.
- i) Integridad y actualización del sistema.
- j) Protección de la información almacenada y en tránsito.
- k) Prevención ante otros sistemas de información interconectados.
- l) Registro de actividad.
- m) Incidentes de seguridad.
- n) Continuidad de la actividad.
- o) Mejora continua del proceso de seguridad.



### 3.3 Objetivos.

Crear las condiciones necesarias de confianza en el uso de los medios electrónicos, a través de medidas para garantizar la seguridad de los sistemas, los datos, las comunicaciones, y los servicios electrónicos, que permita a los ciudadanos y a las administraciones públicas, el ejercicio de derechos y el cumplimiento de deberes a través de estos medios.

Establecer la política de seguridad en la utilización de medios electrónicos en el ámbito de la Ley 11/2007, que estará constituida por los principios básicos y los requisitos mínimos para una protección adecuada de la información.

Introducir los elementos comunes que han de guiar la actuación de las administraciones públicas en materia de seguridad de las tecnologías de la información.

Aportar un lenguaje común para facilitar la interacción de las administraciones públicas, así como la comunicación de los requisitos de seguridad de la información a la industria.

En el Esquema Nacional de Seguridad se concibe la seguridad como una actividad integral, en la que no caben actuaciones puntuales o tratamientos coyunturales, debido a que la debilidad de un sistema la determina su punto más frágil y, a menudo, este punto es la coordinación entre medidas individualmente adecuadas pero deficientemente ensambladas.

Dada la naturaleza de la seguridad, la consecución de estos objetivos requiere un desarrollo que tenga en cuenta la complejidad técnica, la obsolescencia de la tecnología subyacente y el importante cambio que supone en la operativa de la administración la aplicación de la Ley 11/2007.

### 3.4 Ámbito de aplicación

Su ámbito de aplicación es el establecido en el artículo 2 de la Ley 11/2007, de 22 de junio, es decir tanto las Administraciones Públicas, entendiendo por tales la Administración General del Estado, las Administraciones de las Comunidades Autónomas y las Entidades que integran la Administración Local, así como las entidades de derecho público vinculadas o dependientes de las mismas, a los ciudadanos en sus relaciones con las Administraciones Públicas y a las relaciones entre las distintas Administraciones Públicas.

Estarán excluidos los sistemas que tratan información clasificada regulada por Ley 9/1968 de 5 de abril, de Secretos Oficiales, modificada por Ley 48/1978, de 7 de octubre y normas de desarrollo.

## **4. REAL DECRETO 4/2010 POR EL QUE SE REGULA EL ESQUEMA NACIONAL DE INTEROPERABILIDAD EN EL AMBITO DE LA ADMINSTRACION ELECTRONICA.**

### 4.1 Introducción.

El Esquema Nacional de Interoperabilidad persigue la creación de las condiciones necesarias para garantizar el adecuado nivel de interoperabilidad técnica, semántica y organizativa de los sistemas y aplicaciones empleados por las Administraciones Públicas, que permita el ejercicio de derechos y el cumplimiento de deberes a través del acceso electrónico a los servicios públicos, a la vez que redunda en beneficio de la eficacia y la eficiencia.

Al objeto de crear estas condiciones, el Esquema Nacional de Interoperabilidad introduce los elementos comunes que han de guiar la actuación de las Administraciones Públicas en materia de

interoperabilidad. En particular, introduce los siguientes elementos principales:

- Se enuncian los principios específicos de la interoperabilidad.
- Se contemplan las dimensiones de la interoperabilidad organizativa, semántica y técnica a las que se refiere el artículo 41 de la Ley 11/2007, de 22 de junio.
- Se tratan las infraestructuras y los servicios comunes, elementos reconocidos de dinamización, simplificación y propagación de la interoperabilidad, a la vez que facilitadores de la relación multilateral.
- Se trata la reutilización, aplicada a las aplicaciones de las administraciones públicas, de la documentación asociada y de otros objetos de información, dado que la voz ‘compartir’ se encuentra presente en la definición de interoperabilidad recogida en la Ley 11/2007, de 22 de junio, y junto con la voz ‘reutilizar’, ambas son relevantes para la interoperabilidad y se encuentran entroncadas con las políticas de la Unión Europea en relación con la idea de compartir, reutilizar y colaborar.
- Se trata la interoperabilidad de la firma electrónica y de los certificados.
- Se atiende a la recuperación y conservación del documento electrónico, según lo establecido en la citada Ley 11/2007, de 22 de junio, como manifestación de la interoperabilidad a lo largo del tiempo, y que afecta de forma singular al documento electrónico.
- Por último, se crean las normas técnicas de interoperabilidad y los instrumentos para la interoperabilidad, para facilitar la aplicación del Esquema.

Tiene en cuenta las recomendaciones de la Unión Europea, la situación tecnológica de las diferentes Administraciones Públicas, así como los servicios electrónicos ya existentes, y la utilización de

estándares abiertos así como, en su caso y de forma complementaria, estándares que sean de uso generalizado por los ciudadanos.

En su elaboración se han manejado, entre otros, referentes en materia de desarrollo de la administración electrónica y, en particular, de interoperabilidad provenientes del ámbito de la Unión Europea, de actuaciones similares en otros países, de la normalización nacional e internacional; así como la normativa sobre administración electrónica, protección de datos de carácter personal, firma electrónica y Documento Nacional de Identidad Electrónico, entre otros.

Se ha realizado en un proceso coordinado por el Ministerio de la Presidencia, con la participación de todas las Administraciones Públicas. Se ha elaborado con la participación de todas las Administraciones Públicas. A lo largo de los últimos tres años ha colaborado en su elaboración más de un centenar de expertos de las Administraciones Públicas; a los que hay que sumar los numerosos expertos que también han aportado su opinión a través de las asociaciones profesionales del sector TIC; todo ello a la luz del estado del arte y de los principales referentes en materia de interoperabilidad.

#### 4.2 Objetivos

Sus objetivos son los siguientes:

- Comprender los criterios y recomendaciones que deberán ser tenidos en cuenta por las administraciones públicas para la toma de decisiones tecnológicas que garanticen la interoperabilidad y que eviten la discriminación a los ciudadanos por razón de su elección tecnológica.

- Introducir los elementos comunes que han de guiar la actuación de las administraciones públicas en materia de interoperabilidad.

- Aportar un lenguaje común para facilitar la interacción de las administraciones públicas, así como la comunicación de los requisitos de interoperabilidad a la industria.

La interoperabilidad se concibe en consecuencia desde una perspectiva integral, de manera que no caben actuaciones puntuales o tratamientos coyunturales, debido a que la debilidad de un sistema la determina su punto más frágil y, a menudo, este punto es la coordinación entre medidas individualmente adecuadas pero deficientemente ensambladas.

Dada la naturaleza de la interoperabilidad, la consecución de estos objetivos requiere un desarrollo que tenga en cuenta la complejidad técnica, la obsolescencia de la tecnología subyacente y el importante cambio que supone en la operativa de la administración la aplicación de la Ley 11/2007.

#### 4.3 Ámbito de aplicación y Análisis

Su ámbito de aplicación es el establecido en el artículo 42 de la Ley 11/2007, de 22 de junio, de acceso electrónico de los ciudadanos a los Servicios Públicos, es decir, a las Administraciones Públicas, entendiendo por tales la Administración General del Estado, las Administraciones de las Comunidades Autónomas y las Entidades que integran la Administración Local, así como las entidades de derecho público vinculadas o dependientes de las mismas.

A los ciudadanos en sus relaciones con las Administraciones Públicas.

A las relaciones entre las distintas Administraciones Públicas.

No será de aplicación a las Administraciones Públicas en las actividades que desarrollen en régimen de derecho privado

Como aspectos mas importantes a destacar se encuentran:

1.- Regula de forma clara los derechos de los ciudadanos en relación con la utilización de los medios electrónicos en la actividad administrativa, entre ellos

- A elegir, entre aquellos que en cada momento se encuentren disponibles, el canal a través del cual relacionarse por medios electrónicos con las Administraciones Públicas.
- A no aportar los datos y documentos que obren en poder de las Administraciones Públicas, las cuales utilizarán medios electrónicos para recabar dicha información siempre que, en el caso de datos de carácter personal, se cuente con el consentimiento de los interesados en los términos establecidos por la Ley Orgánica 15/1999, de Protección de Datos de Carácter Personal, o una norma con rango de Ley así lo determine, salvo que existan restricciones conforme a la normativa de aplicación a los datos y documentos recabados. El citado consentimiento podrá emitirse y recabarse por medios electrónicos.
- A la igualdad en el acceso electrónico a los servicios de las Administraciones Públicas.
- A conocer por medios electrónicos el estado de tramitación de los procedimientos en los que sean interesados, salvo en los supuestos en que la normativa de aplicación establezca restricciones al acceso a la información sobre aquéllos.

- A obtener copias electrónicas de los documentos electrónicos que formen parte de procedimientos en los que tengan la condición de interesado.
- A la conservación en formato electrónico por las Administraciones Públicas de los documentos electrónicos que formen parte de un expediente.
- A obtener los medios de identificación electrónica necesarios, pudiendo las personas físicas utilizar en todo caso los sistemas de firma electrónica del Documento Nacional de Identidad para cualquier trámite electrónico con cualquier Administración Pública.
- A la utilización de otros sistemas de firma electrónica admitidos en el ámbito de las Administraciones Públicas.
- A la garantía de la seguridad y confidencialidad de los datos que figuren en los ficheros, sistemas y aplicaciones de las Administraciones Públicas.
- A la calidad de los servicios públicos prestados por medios electrónicos.
- A elegir las aplicaciones o sistemas para relacionarse con las Administraciones Públicas siempre y cuando utilicen estándares abiertos o, en su caso, aquellos otros que sean de uso generalizado por los ciudadanos.

Regula el régimen jurídico de la Administración Electrónica, define la sede electrónica como aquella dirección electrónica disponible para los ciudadanos a través de redes de

telecomunicaciones cuya titularidad, gestión y administración corresponde a una Administración Pública, órgano o entidad administrativa en el ejercicio de sus competencias.

Regula la identificación y autenticación, disponiendo que las Administraciones Públicas admitirán, en sus relaciones por medios electrónicos, sistemas de firma electrónica que sean conformes a lo establecido en la Ley 59/2003, de 19 de diciembre, de Firma Electrónica y resulten adecuados para garantizar la identificación de los participantes y, en su caso, la autenticidad e integridad de los documentos electrónicos.

Los ciudadanos podrán utilizar los siguientes sistemas de firma electrónica para relacionarse con las Administraciones Públicas, de acuerdo con lo que cada Administración determine:

- En todo caso, los sistemas de firma electrónica incorporados al Documento Nacional de Identidad, para personas físicas.
- Sistemas de firma electrónica avanzada, incluyendo los basados en certificado electrónico reconocido, admitidos por las Administraciones Públicas.
- Otros sistemas de firma electrónica, como la utilización de claves concertadas en un registro previo como usuario, la aportación de información conocida por ambas partes u otros sistemas no criptográficos, en los términos y condiciones que en cada caso se determinen.

Con relación a los registros, comunicaciones y notificaciones dispone que las Administraciones Públicas crearan registros



electrónicos para la recepción y remisión de solicitudes, escritos y comunicaciones.

Los registros electrónicos podrán admitir: Documentos electrónicos normalizados correspondientes a los servicios, procedimientos y trámites que se especifiquen conforme a lo dispuesto en la norma de creación del registro, cumplimentados de acuerdo con formatos preestablecidos, y además cualquier solicitud, escrito o comunicación distinta de los mencionados en el apartado anterior dirigido a cualquier órgano o entidad del ámbito de la administración titular del registro.

Con respecto a las comunicaciones, los ciudadanos podrán elegir en todo momento la manera de comunicarse con las Administraciones Públicas, sea o no por medios electrónicos, excepto en aquellos casos en los que de una norma con rango de Ley se establezca o infiera la utilización de un medio no electrónico. La opción de comunicarse por unos u otros medios no vincula al ciudadano, que podrá, en cualquier momento, optar por un medio distinto del inicialmente elegido.

Con respecto a documentos y copias dispone que las Administraciones Públicas podrán emitir validamente por medios electrónicos los documentos administrativos a los que se refiere el artículo 46 de la Ley 30/1992, de Régimen Jurídico de las Administraciones Públicas y del Procedimiento Administrativo Común, siempre que incorporen una o varias firmas electrónicas.

Los documentos administrativos incluirán referencia temporal, que se garantizará a través de medios electrónicos cuando la naturaleza del documento así lo requiera.

Las copias realizadas por medios electrónicos de documentos electrónicos emitidos por el propio interesado o por las Administraciones Públicas, manteniéndose o no el formato original,

tendrán inmediatamente la consideración de copias auténticas con la eficacia prevista en el artículo 46 de la Ley 30/1992, de Régimen Jurídico de las Administraciones Públicas y del Procedimiento Administrativo Común, siempre que el documento electrónico original se encuentre en poder de la Administración, y que la información de firma electrónica y, en su caso, de sellado de tiempo permitan comprobar la coincidencia con dicho documento.

Regula la gestión electrónica de procedimientos, disponiendo que la gestión electrónica de la actividad administrativa respetará la titularidad y el ejercicio de la competencia por la Administración Pública, órgano o entidad que la tenga atribuida y el cumplimiento de los requisitos formales y materiales establecidos en las normas que regulen la correspondiente actividad. A estos efectos, y en todo caso bajo criterios de simplificación administrativa, se impulsará la aplicación de medios electrónicos a los procesos de trabajo y la gestión de los procedimientos y de la actuación administrativa

Por último establece el R D un marco institucional de colaboración entre Administraciones disponiendo II Comité Sectorial de administración electrónica, dependiente de la Conferencia Sectorial de Administración Pública, es el órgano técnico de cooperación de la Administración General del Estado, de las administraciones de las Comunidades Autónomas y de las entidades que integran la Administración Local en materia de administración electrónica.

El Comité Sectorial de la administración electrónica velará por el cumplimiento de los fines y principios establecidos en esta Ley, y en particular desarrollará las siguientes funciones:

- Asegurar la compatibilidad e interoperabilidad de los sistemas y aplicaciones empleados por las Administraciones Públicas.
- Preparar planes programas conjuntos de actuación para impulsar el desarrollo de la administración electrónica en España.
- Asegurar la cooperación entre las administraciones públicas para proporcionar al ciudadano información administrativa clara, actualizada e inequívoca.

Autor:

Alfonso García Magariños

Director Asesoría Jurídica Municipal Concello de A Coruña

**14. FACTURA ELECTRÓNICA.  
LEY 59/2003, DE 19 DE  
DICIEMBRE, DE FIRMA  
ELECTRÓNICA. LEY DE  
SERVICIOS DE LA SOCIEDAD  
DE LA INFORMACIÓN Y  
COMERCIO ELECTRÓNICO.  
ACCESIBILIDAD. DECRETO  
3/2010, DE 8 DE ENERO, POR EL  
QUE SE REGULA EL SISTEMA  
DE FACTURACIÓN  
ELECTRÓNICA DE LA XUNTA  
DE GALICIA.**

**Tema 14. Factura electrónica. Ley 59/2003, de 19 de diciembre, de firma electrónica. Ley de servicios de la sociedad de la información y comercio electrónico. Accesibilidad. Decreto 3/2010 de 8 de enero por el que se regula el sistema de facturación electrónica de la Xunta de Galicia**

## **INDICE**

- 14.1 Factura electrónica.
- 14.2 Ley de firma electrónica
  - 14.2.1 Introducción
  - 14.2.2 Análisis
  - 14.2.3 Efectos jurídicos y análisis
- 14.3 Ley de servicios de la sociedad de la información y comercio electrónico
  - 14.3.1 Introducción
  - 14.3.2 Ámbito aplicación
  - 14.3.3 Principios y requisitos de actuación
- 14.4 Accesibilidad
- 14.5 Decreto 3 /2010 de 8 de enero por el que se regula el sistema de facturación electrónica de la Xunta de Galicia
  - 14.5.1 Introducción
  - 14.5.2 Estructura

### **14.1.- FACTURA ELECTRÓNICA**

Con carácter general se puede definir la factura como un documento que refleja la entrega de un producto o la provisión de un servicio, junto a la fecha de devengo, además de indicar la cantidad a pagar como contraprestación.

En la factura se encuentran los datos del expedidor y del destinatario, el detalle de los productos y servicios suministrados, los precios unitarios, los precios totales, los descuentos y los impuestos.

La facturación electrónica consiste en la transmisión de las facturas o documentos análogos entre emisor y receptor por medios electrónicos (ficheros informáticos) y telemáticos (de un ordenador a otro), firmados digitalmente con certificados reconocidos (o cualificados), con la misma validez legal que las facturas emitidas en papel.

Si buscamos una definición específica, podemos recurrir al artículo 1 de la Ley 56/2007: “La factura electrónica es un documento electrónico que cumple con los requisitos legal y reglamentariamente exigibles a las facturas y que, además, garantiza la autenticidad de su origen y la integridad de su contenido”.

Aunque existen varios mecanismos para garantizar la autenticidad del origen, la integridad del contenido y la legibilidad de una factura, ya sea en papel o en formato electrónico, desde el momento de su expedición hasta el final del período de conservación de la factura, en el caso de la factura electrónica, el uso de la firma electrónica es el más generalizado en España.

En este sentido, en el texto consensuado de la futura modificación de la Directiva 112/2006, se recoge, en lo que se refiere a su artículo 233:

*“1. Se garantizará la autenticidad del origen, la integridad del contenido y la legibilidad de una factura, ya sea en papel o en formato electrónico, desde el momento de su expedición hasta el final del período de conservación de la factura. Cada sujeto pasivo determinará el modo de garantizar la autenticidad del origen, la integridad del contenido y la legibilidad de las facturas. Podrá realizarse mediante controles de gestión que creen un vínculo fiable de auditoría entre la factura y la entrega de bienes o la prestación de servicios.*

*Se entenderá por “autenticidad del origen”, la garantía de la identidad del proveedor de bienes o prestador de servicios o del emisor de la factura.*

*Por “integridad del contenido” se entenderá que el contenido requerido con arreglo a lo dispuesto en la presente Directiva no ha sido modificado.*

*2. Además de los tipos de control de la gestión contemplados por el segundo párrafo del apartado 1, otros ejemplos de tecnologías que garantizan la autenticidad del origen y la integridad del contenido de una factura electrónica son:*

- La firma electrónica avanzada en el sentido del punto 2 del artículo 2 de la Directiva 1999/93/CE del Parlamento Europeo y del Consejo, de 13 de diciembre de 1999, por la que se establece un marco comunitario para la firma electrónica, basada en un certificado reconocido y creada mediante un dispositivo seguro de creación de firma, en el sentido de los puntos 6 y 10 de la Directiva 1999/93/CE.*

- El intercambio electrónico de datos (IED), tal como se define en el artículo 2 de la Recomendación 1994/820/CE de la Comisión, de 19 de octubre de 1994, relativa a los aspectos jurídicos del intercambio electrónico de datos, si el acuerdo relativo al intercambio contempla el uso*

*de procedimientos que garanticen la autenticidad del origen y la integridad de los datos.”*

Esto significa que tras el período de transposición de la Directiva (1 de enero de 2013), la legislación española reflejará la posibilidad de que puedan enviarse facturas electrónicas entre empresas sin ningún requisito formal, si bien probablemente se mantengan los mismos requisitos que existen en la actualidad cuando el destinatario sea una administración pública.

En España, la adopción de la firma electrónica como mecanismo generalizado para garantizar la autenticidad e integridad de las facturas electrónicas se ha visto favorecido por la extensión del DNI electrónico y la amplia disponibilidad de certificados electrónicos de múltiples prestadores de servicios de certificación, así como por la disponibilidad de software gratuito que permite la generación y firma electrónica de las facturas electrónicas que se envían, así como su verificación en el caso de la recepción de facturas.

El proceso de facturación es un proceso importante para cualquier empresa, y culmina el proceso de compra y venta. Aunque tradicionalmente la relación entre empresas se ha basado en el intercambio de documentos en papel, esto implica el empleo de grandes cantidades de recursos y la realización de muchas tareas de forma manual. En un contexto de universalización de Internet, cada vez más las empresas se plantean la optimización de sus procesos para ganar eficiencia y ahorrar costes.

Y por ello se ha avanzado en la adopción de la facturación electrónica, que en España está regulada en el Reglamento de facturación publicado en Real Decreto 1496/2003 y modificado por el Real Decreto 87/2005.



Las denominaciones factura electrónica, factura telemática y factura digital son equivalentes, si bien la denominación generalmente utilizada en la normativa es remisión electrónica o remisión por medios electrónicos de factura. Frecuentemente se distingue con la denominación factura digital a la modalidad de factura electrónica que utiliza la firma digital para garantizar la autenticidad e integridad de la factura.

Las facturas electrónicas se pueden emitir en diferentes formatos (EDIFACT, XML, PDF, html, doc, xls, gif, jpeg o txt, entre otros) siempre que se respete el contenido legal exigible a cualquier factura y que se cumplan los requisitos de autenticidad e integridad, por ejemplo con la incorporación de la firma electrónica reconocida (qualified Electronic signature, en inglés ).

Sin embargo, tras la publicación de la Orden PRE/2971/2007, en se definió el uso obligatorio del formato XML facturade, cuando el destinatario sea una administración de la AGE (Administración General del Estado) y sus organismos públicos.

## **14.2. LEY DE FIRMA ELECTRÓNICA.**

### **14.2.1 Introducción**

Sin olvidar las referencias que sobre el tema de utilización de medios electrónicos establecía la ley 30/92 de 20 de noviembre ( LRJPAC ) es el Real Decreto-ley 14/1999, de 17 de septiembre, sobre firma electrónica, la norma pionera a la hora de fomentar la rápida incorporación de las nuevas tecnologías de seguridad de las comunicaciones electrónicas en la actividad de las empresas, los ciudadanos y las Administraciones públicas, actualmente le ley 59/2003 de firma electrónica, cuya ultima modificación tuvo lugar por la ley 56/2007 de medidas de impulso de la sociedad de la información

#### 14.2.2 Estructura y análisis.

Desde el punto de vista de su estructura, la ley 59/2003 de firma electrónica consta de 36 artículos agrupados en seis títulos, 11 disposiciones adicionales, dos disposiciones transitorias, una disposición derogatoria y tres disposiciones finales.

El título I contiene los principios generales que delimitan los ámbitos subjetivo y objetivo de aplicación de la ley, los efectos de la firma electrónica y el régimen de empleo ante las Administraciones públicas y de acceso a la actividad de prestación de servicios de certificación.

El régimen aplicable a los certificados electrónicos se contiene en el título II, que dedica su primer capítulo a determinar quiénes pueden ser sus titulares y a regular las vicisitudes que afectan a su vigencia. El capítulo II regula los certificados reconocidos y el tercero el documento nacional de identidad electrónico.

El título III regula la actividad de prestación de servicios de certificación estableciendo las obligaciones a que están sujetos los prestadores -distinguiendo con nitidez las que solamente afectan a los que expiden certificados reconocidos-, y el régimen de responsabilidad aplicable.

El título IV establece los requisitos que deben reunir los dispositivos de verificación y creación de firma electrónica y el procedimiento que ha de seguirse para obtener sellos de calidad en la actividad de prestación de servicios de certificación.

Los títulos V y VI dedican su contenido, respectivamente, a fijar los regímenes de supervisión y sanción de los prestadores de servicios de certificación.

Por último, cierran el texto las disposiciones adicionales -que aluden a los regímenes especiales que resultan de aplicación preferente-, las disposiciones transitorias -que incorporan seguridad jurídica a la actividad desplegada al amparo de la normativa anterior-, la disposición derogatoria y las disposiciones finales relativas al fundamento constitucional, la habilitación para el desarrollo reglamentario y la entrada en vigor.

-  
Supone la incorporación al ordenamiento interno de la regulación contenida en la Directiva 1999/93/ CE, de 13 de diciembre de 1999, del Parlamento Europeo y del Consejo, esta Ley parece tener presente en la elaboración de su contenido la Ley Modelo para las firmas electrónicas de la Comisión de las Naciones Unidas para el Derecho Mercantil Internacional (CNUDMI/ UNCITRAL), aprobada, junto a su Guía, el 5 de julio de 2001.

Tiene como principal finalidad reforzar el marco jurídico existente, incorporando a su texto «algunas novedades respecto del Real Decreto 14/1999, que contribuirán a dinamizar el mercado de la prestación de servicios de certificación, confiriendo seguridad a las comunicaciones a través de Internet, y configurando la firma electrónica como instrumento capaz de generar confianza en las transacciones telemáticas, además de agilizar el comercio electrónico. Se permitirá, en consecuencia, una comprobación de la procedencia y de la integridad de los mensajes intercambiados a través de redes de telecomunicaciones, ofreciendo las bases para evitar el repudio, si se adoptan las medidas oportunas basándose en fechas electrónicas»

Constituye su objeto, conforme dispone el art. 1, tanto la regulación de la firma electrónica, como elemento de seguridad de las comunicaciones en sus diversos aspectos, y su eficacia jurídica, como la prestación de

servicios de certificación en sus diversos aspectos (objetivo: certificados, y subjetivo: prestadores de servicio de certificación)

El apartado 1 del art. 3 de la LFE define de forma general la firma electrónica como «el conjunto de datos en forma electrónica, consignados junto a otros o asociados con ellos, que pueden ser utilizados como medio de identificación del firmante».

Se trata de una definición amplia que puede englobar a todo conjunto de firmas electrónicas, desde aquellas más complejas, como la firma digital basada en sistemas biométricos como el iris, la propia palma de la mano, la huella dactilar, etc. hasta la más simples, como un nombre u otro elemento identificativos (por ejemplo, la firma manual digitalizada, o un password o contraseña), incluido al final del mensaje electrónico, o la existencia de una pregunta-respuesta, y un pin de acceso, lo que se denomina tecnología de secreto compartido, de tan escasa seguridad que plantean la cuestión de su valor probatorio a efectos de autenticación o identificación del autor

Asimismo, de este concepto amplio y tecnológicamente indefinido de firma que nos ofrece el citado precepto podemos resaltar las siguientes características de la firma electrónica:

- La firma electrónica es un conjunto de datos y no un símbolo, sello o grafía electrónica que sirve para identificar al firmante de un mensaje y para acreditar la identificación del mismo, como la integridad del contenido del mensaje.
- Se trata de una técnica para identificar al firmante de un documento electrónico.

- Los datos de firma electrónica puede forma parte del documento o ir asociados funcionalmente con ellos o, lo que es lo mismo, pueden aparecer como un conjunto independiente. El modo concreto en que en cada momento se manifieste la firma electrónica dependerá del sistema técnico que se elija y de las aplicaciones prácticas que ofrezca cada modalidad

Define la ley que se considera documento electrónico, este es la información de cualquier naturaleza en forma electrónica, archivada en un soporte electrónico según un formato determinado y susceptible de identificación y tratamiento diferenciado.

Para que un documento electrónico tenga la naturaleza de documento público o de documento administrativo deberá cumplirse, o bien, estar firmados electrónicamente por funcionarios que tengan legalmente atribuida la facultad de dar fe pública, judicial, notarial o administrativa, siempre que actúen en el ámbito de sus competencias con los requisitos exigidos por la ley en cada caso, o bien tratarse de documento expedidos y firmados electrónicamente por funcionarios o empleados públicos en el ejercicio de sus funciones públicas, conforme a su legislación específica.

Al lado de la firma electrónica se encuentra la firma electrónica avanzada, que según la ley, artículo 3.2 “«La firma electrónica avanzada es la firma electrónica que permite identificar al firmante y detectar cualquier cambio ulterior de los datos firmados, que está vinculada al firmante de manera única y a los datos a que se refiere y que ha sido creada por medios que el firmante puede mantener bajo su exclusivo control»

La ley de firma electrónica frente a la regulación contenido tanto en el RDL 14/99 como en la Directiva, introduce un tercer tipo de firma, de mayor calidad y seguridad, como es la firma electrónica reconocida, definida en el art. 3.3 como «la firma electrónica avanzada basada en un certificado reconocido y generada mediante un dispositivo seguro de creación de firma».

Supone, como señala la Exposición de Motivos de la Ley, «la creación de un concepto nuevo demandado por el sector, sin que ello implique modificación alguna de los requisitos sustantivos que tanto la Directiva 1999/93/CE como el propio Real Decreto Ley 14/1999 venían exigiendo.

El Art. 24 de la LFE, bajo el título «Dispositivo de firma electrónica», define los datos de creación de firma como «los datos únicos, como códigos o claves criptográficas privadas, que el firmante utiliza para crear la firma electrónica»

Como elemento característico de los mismos viene a establecerse el que éstos deben ser únicos

Por su parte, el Art. 25 de la LFE, bajo el título «Dispositivos de verificación de firma electrónica», define los «datos de verificación de firma» como «los datos, códigos o claves criptográficas públicas, que se utilizan para verificar la firma electrónica».

Por tanto, bajo el Título «Dispositivos de firma electrónica» se regula como nociones previas tanto para la aplicación del dispositivo de creación como para el de verificación los datos de creación y verificación de firma, respectivamente, que desde un punto de vista técnico constituyen además elementos que posibilitan la creación de una firma o su verificación

#### 14.2.3 Efectos jurídicos de la firma electrónica

A la validez y eficacia de la firma electrónica dedica la LFE los apartados 4, 8, 9 y 10 del Art. 3, que coincide sustancialmente con lo dispuesto en el Art. 5 de la Directiva comunitaria, y en los que se equipara la firma electrónica reconocida a la firma manuscrita, se determina las consecuencias de la impugnación de la autenticidad de la firma electrónica reconocida por la otra parte no firmante, se reconoce valor jurídico a la autonomía de la voluntad de las partes para dotar de eficacia a la firma electrónica y se especifica la admisibilidad de los datos firmados electrónicamente como prueba documental en juicio;

El Art 3.4 de la Ley establece la regla del equivalente funcional entre la firma electrónica reconocida y la firma manuscrita, al disponer que «la firma electrónica reconocida tendrá respecto de los datos consignados en forma electrónica, el mismo valor que la firma manuscrita en relación con los consignados en papel».

En consecuencia, sobre lo expuesto, para la plena operatividad de la regla de la equivalencia funcional de la firma electrónica reconocida con la firma manuscrita, que dispone el art. 3.4 de la LFE, en una interpretación conjunta con el art. 3.3 de esta misma Ley, es necesario el cumplimiento de los siguientes requisitos:

1.º Debe tratarse de una firma electrónica avanzada (art. 3.2 de la LFE).

2.º Dicha firma electrónica avanzada ha de estar basada en un certificado reconocido, es decir, aquel que cumple los requisitos de los arts. 11, 12 y 13 de la LFE, y que haya sido expedido por un prestador de servicios de certificación que cumpla con los requisitos previstos en el Art. 20 de la LFE.

3.º Dicha firma electrónica avanzada, además, debe haber sido producida por un dispositivo seguro de creación de firma que cumpla con los requisitos del apartado 3 del Art. 24 de la LFE

Regula también la ley la figura del DNI electrónico, este es el documento nacional de identidad que acredita electrónicamente la identidad personal de su titular y permite la firma electrónica de documentos, se impone la obligación de que todas las personas físicas o jurídicas, públicas o privadas, reconocerán la eficacia del documento nacional de identidad electrónico para acreditar la identidad y los demás datos personales del titular que consten en el mismo, y para acreditar la identidad del firmante y la integridad de los documentos firmados con los dispositivos de firma electrónica en él incluidos.

La ley regula la figura de los prestadores de servicios de certificación, siendo estos “la persona física o jurídica que expide certificados electrónicos o presta otros servicios en relación con la firma electrónica “

Para la expedición de certificados electrónicos al público, los prestadores de servicios de certificación únicamente podrán recabar datos personales directamente de los firmantes o previo consentimiento expreso de éstos, debiendo cumplir por tanto lo dispuesto en la Ley Orgánica 15/1999, de 13 de diciembre, de Protección de Datos de Carácter Personal y en sus normas de desarrollo.

Los prestadores de servicios de certificación que expidan certificados electrónicos deberán cumplir las siguientes obligaciones:

- No almacenar ni copiar los datos de creación de firma de la persona a la que hayan prestado sus servicios.
- Proporcionar al solicitante antes de la expedición del certificado la siguiente información mínima, que deberá transmitirse de forma gratuita, por escrito o por vía electrónica:



- Las obligaciones del firmante, la forma en que han de custodiarse los datos de creación de firma, el procedimiento que haya de seguirse para comunicar la pérdida o posible utilización indebida de dichos datos y determinados dispositivos de creación y de verificación de firma electrónica que sean compatibles con los datos de firma y con el certificado expedido.
- Los mecanismos para garantizar la fiabilidad de la firma electrónica de un documento a lo largo del tiempo.
- El método utilizado por el prestador para comprobar la identidad del firmante u otros datos que figuren en el certificado.
- Las condiciones precisas de utilización del certificado, sus posibles límites de uso y la forma en que el prestador garantiza su responsabilidad patrimonial.
- Las certificaciones que haya obtenido, en su caso, el prestador de servicios de certificación y los procedimientos aplicables para la resolución extrajudicial de los conflictos que pudieran surgir por el ejercicio de su actividad.
- Las demás informaciones contenidas en la declaración de prácticas de certificación.
- La información citada anteriormente que sea relevante para terceros afectados por los certificados deberá estar disponible a instancia de éstos.

- Mantener un directorio actualizado de certificados en el que se indicarán los certificados expedidos y si están vigentes o si su vigencia ha sido suspendida o extinguida. La integridad del directorio se protegerá mediante la utilización de los mecanismos de seguridad adecuados.
- Garantizar la disponibilidad de un servicio de consulta sobre la vigencia de los certificados rápido y seguro

Regula la ley aspectos relacionados con la supervisión y control de la actividad de los prestadores de servicios, y así el Ministerio de Ciencia y Tecnología controlará el cumplimiento por los prestadores de servicios de certificación que expidan al público certificados electrónicos de las obligaciones establecidas en esta Ley y en sus disposiciones de desarrollo. Asimismo, supervisará el funcionamiento del sistema y de los organismos de certificación de dispositivos seguros de creación de firma electrónica.

El Ministerio de Ciencia y Tecnología realizará las actuaciones inspectoras que sean precisas para el ejercicio de su función de control, los funcionarios adscritos al Ministerio de Ciencia y Tecnología que realicen la inspección a que se refiere el apartado anterior tendrán la consideración de autoridad pública en el desempeño de sus cometidos.

Los prestadores de servicios de certificación, la entidad independiente de acreditación y los organismos de certificación tienen la obligación de facilitar al Ministerio de Ciencia y Tecnología toda la información y colaboración precisas para el ejercicio de sus funciones.

#### **14.3.- LEY DE SERVICIOS DE SOCIEDAD DE INFORMACIÓN Y COMERCIO ELECTRÓNICO**

### 14.3.1 Introducción.

La ley 34/2002 de 22 de julio tiene como objeto la incorporación al ordenamiento jurídico español de la Directiva 2000/31/CE, del Parlamento Europeo y del Consejo, de 8 de junio, relativa a determinados aspectos de los servicios de la sociedad de la información, en particular, el comercio electrónico en el mercado Interior (Directiva sobre el comercio electrónico). Asimismo, incorpora parcialmente la Directiva 98/27/CE, del Parlamento Europeo y del Consejo, de 19 de mayo, relativa a las acciones de cesación en materia de protección de los intereses de los consumidores, al regular, de conformidad con lo establecido en ella, una acción de cesación contra las conductas que contravengan lo dispuesto en esta Ley.

Lo que la Directiva 2000/31/CE denomina sociedad de la información viene determinado por la extraordinaria expansión de las redes de telecomunicaciones y, en especial, de Internet como vehículo de transmisión e intercambio de todo tipo de información. Su incorporación a la vida económica y social ofrece innumerables ventajas, como la mejora de la eficiencia empresarial, el incremento de las posibilidades de elección de los usuarios y la aparición de nuevas fuentes de empleo. Pero la implantación de Internet y las nuevas tecnologías tropieza con algunas incertidumbres jurídicas, que es preciso aclarar con el establecimiento de un marco jurídico adecuado, que genere en todos los actores intervinientes la confianza necesaria para el empleo de este nuevo medio.

Es objeto de la Ley la regulación del régimen jurídico de los servicios de la sociedad de la información y de la contratación por vía electrónica, en lo referente a las obligaciones de los prestadores de servicios incluidos los que actúen como intermediarios en la transmisión de contenidos por las redes de telecomunicaciones, las comunicaciones comerciales por vía

electrónica, la información previa y posterior a la celebración de contratos electrónicos, las condiciones relativas a su validez y eficacia y el régimen sancionador aplicable a los prestadores de servicios de la sociedad de la información.

#### 14.3.2 Ámbito aplicación

Se aplicará a los prestadores de servicios de la sociedad de la información establecidos en otro Estado miembro de la Unión Europea o del Espacio Económico Europeo cuando el destinatario de los servicios radique en España y los servicios afecten a las materias siguientes:

- a. Derechos de propiedad intelectual o industrial.
- b. Emisión de publicidad por instituciones de inversión colectiva.
- c. Actividad de seguro directo realizada en régimen de derecho de establecimiento o en régimen de libre prestación de servicios.
- d. Obligaciones nacidas de los contratos celebrados por personas físicas que tengan la condición de consumidores.
- e. Régimen de elección por las partes contratantes de la legislación aplicable a su contrato.
- f. Licitud de las comunicaciones comerciales por correo electrónico u otro medio de comunicación electrónica equivalente no solicitadas

La prestación de servicios de la sociedad de la información no estará sujeta a autorización previa

En caso de que un determinado servicio de la sociedad de la información atente o pueda atentar contra los principios que se expresan a continuación, los órganos competentes para su protección, en ejercicio de las funciones que tengan legalmente atribuidas, podrán adoptar las

medidas necesarias para que se interrumpa su prestación o para retirar los datos que los vulneran.

#### 14.4.3 Principios y requisitos de actuación

Los principios de aplicación son los siguientes:

- a. La salvaguarda del orden público, la investigación penal, la seguridad pública y la defensa nacional.
- b. La protección de la salud pública o de las personas físicas o jurídicas que tengan la condición de consumidores o usuarios, incluso cuando actúen como inversores.
- c. El respeto a la dignidad de la persona y al principio de no discriminación por motivos de raza, sexo, religión, opinión, nacionalidad, discapacidad o cualquier otra circunstancia personal o social, y
- d. La protección de la juventud y de la infancia.
- e. La salvaguarda de los derechos de propiedad intelectual

El prestador de servicios de la sociedad de la información estará obligado a disponer de los medios que permitan, tanto a los destinatarios del servicio como a los órganos competentes, acceder por medios electrónicos, de forma permanente, fácil, directa y gratuita, a la siguiente información:

- a. Su nombre o denominación social; su residencia o domicilio o, en su defecto, la dirección de uno de sus establecimientos permanentes en España; su dirección de correo electrónico y cualquier otro dato que permita establecer con él una comunicación directa y efectiva.
- b. Los datos de su inscripción en el Registro Mercantil en el que, en su caso, se encuentren inscritos o de aquel otro registro

público en el que lo estuvieran para la adquisición de personalidad jurídica o a los solos efectos de publicidad.

c. En el caso de que su actividad estuviese sujeta a un régimen de autorización administrativa previa, los datos relativos a dicha autorización y los identificativos del órgano competente encargado de su supervisión.

d. Si ejerce una profesión regulada deberá indicar:

1. Los datos del Colegio profesional al que, en su caso, pertenezca y número de colegiado.

2. El título académico oficial o profesional con el que cuente.

3. El Estado de la Unión Europea o del Espacio Económico Europeo en el que se expidió dicho título y, en su caso, la correspondiente homologación o reconocimiento.

4. Las normas profesionales aplicables al ejercicio de su profesión y los medios a través de los cuales se puedan conocer, incluidos los electrónicos.

e. El número de identificación fiscal que le corresponda.

f. Cuando el servicio de la sociedad de la información haga referencia a precios, se facilitará información clara y exacta sobre el precio del producto o servicio, indicando si incluye o no los impuestos aplicables y, en su caso, sobre los gastos de envío o en su caso aquello que dispongan las normas de las Comunidades Autónomas con competencias en la materia.

g. Los códigos de conducta a los que, en su caso, esté adherido y la manera de consultarlos electrónicamente

Los prestadores de servicios de la sociedad de la información estén sujetos a la responsabilidad civil, penal y administrativa establecida con carácter general en el ordenamiento jurídico, sin perjuicio de lo dispuesto en esta Ley

Los prestadores de servicios de la sociedad de la información que faciliten enlaces a otros contenidos o incluyan en los suyos directorios o instrumentos de búsqueda de contenidos no serán responsables por la información a la que dirijan a los destinatarios de sus servicios, siempre que:

- a. No tengan conocimiento efectivo de que la actividad o la información a la que remiten o recomiendan es ilícita o de que lesiona bienes o derechos de un tercero susceptibles de indemnización, o
- b. Si lo tienen, actúen con diligencia para suprimir o inutilizar el enlace correspondiente.

Las Administraciones públicas impulsarán, a través de la coordinación y el asesoramiento, la elaboración y aplicación de códigos de conducta voluntarios, por parte de las corporaciones, asociaciones u organizaciones comerciales, profesionales y de consumidores, en las materias reguladas en esta Ley. La Administración General del Estado fomentará, en especial, la elaboración de códigos de conducta de ámbito comunitario o internacional.

Los códigos de conducta podrán tratar, en particular, sobre los procedimientos para la detección y retirada de contenidos ilícitos y la protección de los destinatarios frente al envío por vía electrónica de comunicaciones comerciales no solicitadas, así como sobre los procedimientos extrajudiciales para la resolución de los conflictos que surjan por la prestación de los servicios de la sociedad de la información.

Los contratos celebrados por vía electrónica producirán todos los efectos previstos por el ordenamiento jurídico, cuando concurren el consentimiento y los demás requisitos necesarios para su validez

La prueba de la celebración de un contrato por vía electrónica y la de las obligaciones que tienen su origen en él se sujetará a las reglas generales del ordenamiento jurídico.

Cuando los contratos celebrados por vía electrónica estén firmados electrónicamente se estará a lo establecido en el artículo 3 de la Ley 59/2003, de 19 de diciembre, de firma electrónica

Los contratos celebrados por vía electrónica en los que intervenga como parte un consumidor se presumirán celebrados en el lugar en que éste tenga su residencia habitual

El prestador y el destinatario de servicios de la sociedad de la información podrán someter sus conflictos a los arbitrajes previstos en la legislación de arbitraje y de defensa de los consumidores y usuarios, y a los procedimientos de resolución extrajudicial de conflictos que se instauren por medio de códigos de conducta u otros instrumentos de autorregulación

#### **14.4.-ACCESIBILIDAD.**

La disposición adicional quinta de la ley de servicios de la sociedad de la información y del comercio electrónico ( ley 34/2002 de 11 de julio ) establece una norma relativa a accesibilidad para las personas con discapacidad y de edad avanzada a la información proporcionada por medios electrónicos y establece

*“Las Administraciones públicas adoptarán las medidas necesarias para que la información disponible en sus respectivas páginas de Internet pueda ser accesible a personas con discapacidad y de edad avanzada, de acuerdo con los criterios de accesibilidad al contenido generalmente reconocidos, antes del 31 de diciembre de 2005.*



*A partir del 31 de diciembre de 2008, las páginas de Internet de las Administraciones Públicas satisfarán, como mínimo, el nivel medio de los criterios de accesibilidad al contenido generalmente reconocidos. Excepcionalmente, esta obligación no será aplicable cuando una funcionalidad o servicio no disponga de una solución tecnológica que permita su accesibilidad.*

*Las Administraciones Públicas exigirán que tanto las páginas de Internet cuyo diseño o mantenimiento financien total o parcialmente como las páginas de Internet de entidades y empresas que se encarguen de gestionar servicios públicos apliquen los criterios de accesibilidad antes mencionados. En particular, será obligatorio lo expresado en este apartado para las páginas de Internet y sus contenidos de los Centros públicos educativos, de formación y universitarios, así como, de los Centros privados que obtengan financiación pública.*

*Las páginas de Internet de las Administraciones Públicas deberán ofrecer al usuario información sobre su nivel de accesibilidad y facilitar un sistema de contacto para que puedan transmitir las dificultades de acceso al contenido de las páginas de Internet o formular cualquier queja, consulta o sugerencia de mejora.*

*Igualmente, se promoverá la adopción de normas de accesibilidad por los prestadores de servicios y los fabricantes de equipos y software, para facilitar el acceso de las personas con discapacidad o de edad avanzada a los contenidos digitales.*

*Las Administraciones Públicas promoverán medidas de sensibilización, educación y formación sobre accesibilidad con objeto de promover que los titulares de otras páginas de Internet incorporen progresivamente los criterios de accesibilidad.*

*Los incumplimientos de las obligaciones de accesibilidad establecidas en esta Disposición adicional estarán sometidos al régimen de infracciones y sanciones vigente en materia de igualdad de oportunidades, no discriminación y accesibilidad universal de las personas con discapacidad.*

*Las páginas de Internet de las empresas que presten servicios al público en general de especial trascendencia económica, sometidas a la obligación establecida en el artículo 2 de la Ley 56/2007, de medidas de impulso de la sociedad de la información, deberán satisfacer a partir del 31 de diciembre de 2008, como mínimo, el nivel medio de los criterios de accesibilidad al contenido generalmente reconocidos. Excepcionalmente, esta obligación no será aplicable cuando una funcionalidad o servicio no disponga de una solución tecnológica que permita su accesibilidad.*

El "Diseño Web Accesible" pretende establecer las técnicas y mecanismos para permitir que un sitio Web pueda ser utilizado por cualquier persona, independientemente de su discapacidad.

En diciembre de 2007 se aprobó la Ley 56/2007, de Medidas de Impulso de la Sociedad de la Información (BOE 29-12-2008). Por un lado, esta ley indica claramente que la accesibilidad de las páginas web de la administración pública está sujeta al régimen de infracciones y sanciones (Ley 49/2007). Por otro lado, la obligación de hacer sitios web accesibles se amplía al sector privado, en concreto a las empresas que presten servicios al público en general de especial trascendencia económica.

En diciembre de 2007 se aprobó la Ley 49/2007, por la que se establece el régimen de infracciones y sanciones en materia de igualdad de oportunidades, no discriminación y accesibilidad universal de las personas con discapacidad (BOE 27-12-2007). En esta ley se definen sanciones que pueden llegar hasta 1.000.000 de euros.

En noviembre de 2007 se aprobó el Real Decreto 1494/2007 por el que se aprueba el Reglamento sobre las condiciones básicas para el acceso de las personas con discapacidad a las tecnologías, productos y servicios relacionados con la sociedad de la información y medios de comunicación social (BOE 21-11-2007). Este Reglamento obliga a las administraciones

públicas a que sus páginas web sean accesibles de acuerdo con el nivel 2 de la norma española UNE 139803:2004 antes del 2009.

La Ley IONDAU (Ley 51/2003, de 2 de diciembre, de Igualdad de Oportunidades, No Discriminación y Accesibilidad Universal de las Personas con Discapacidad. BOE 3-12-2003), fija varias fases: 1º a primeros de 2006 el Gobierno deberá haber establecido los criterios básicos de accesibilidad para las Tecnologías de la Sociedad de la Información; 2º en 2010 todos los nuevos productos y servicios de la Sociedad de la Información deberán ser accesibles; 3º en 2014 todos los productos y servicios de la Sociedad de la Información deberán ser accesibles.

La Ley SSICE (Ley 34/2002, de 11 de julio, de Servicios de la Sociedad de la Información y de Comercio Electrónico. BOE 12-7-2002), en su disposición adicional 5ª (que hace referencia a la accesibilidad para las personas con discapacidad y de edad avanzada a la información proporcionada por medios electrónicos), obligaba a las Administraciones Públicas a tener sus Webs accesibles para todos antes de 2006.

En el ámbito Europeo existen numerosas normas e iniciativas en este sentido (como las iniciativas eEurope (an Information Society for All) 2002 y 2005 y la estrategia i2010), puesto que se habla de un 20% de la población europea afectada por algún tipo de discapacidad.

#### **14.5.- DECRETO 3/2010 DE 8 DE ENERO POR EL QUE SE REGULA EL SISTEMA DE FACTURACIÓN ELECTRÓNICA DE LA XUNTA DE GALICIA**

##### **14.5.1 Introducción**

La implantación de los medios electrónicos en el ámbito de la facturación y la contratación pública se enmarca en las políticas corporativas comunes del Gobierno gallego para el desarrollo de la administración electrónica y se integrará armónica y complementariamente con el resto de las aplicaciones comunes para toda la Xunta de Galicia.

El Decreto regula, por un lado, las líneas generales de empleo de los medios electrónicos, informáticos y telemáticos en los procedimientos de contratación y establece las condiciones de utilización de los citados medios en el marco del desarrollo de la administración electrónica de la Xunta de Galicia. Así mismo, configura las bases de un sistema que se asienta sobre varios ejes que se complementan estructurando todas las relaciones telemáticas entre los actores que intervienen en los procesos.

Así, en términos de servicios a los licitadores, ordena y estructura nuevos canales de información y participación del empresariado en los procesos de contratación creando la Plataforma de Contratos Públicos de Galicia y un Portal de Contratación Pública de la Comunidad Autónoma que potenciará los servicios en línea existentes e incorporará prestaciones relacionadas con la licitación electrónica. Por otro lado, en términos de servicios entre administraciones se prevén mecanismos de interoperatividad que, en primer lugar, refuerzan la accesibilidad a las plataformas y sistemas y aplicaciones existentes o futuras y, en segundo lugar, facilitan la cooperación y colaboración intra e interadministrativas a los efectos de tráfico de información.

#### 14.5.2 Estructura

Respecto a la estructura, el Decreto consta de 18 artículos, agrupados en 5 capítulos.

El capítulo I regula el objeto del Decreto y su ámbito de aplicación.

El capítulo II regula la Plataforma de Contratos Públicos de Galicia, creada al amparo de lo dispuesto en el artículo 309.5 de la Ley 30/2007, de 30 de octubre, como servicio de información de las licitaciones del sector público gallego a través de internet.

El capítulo III regula el Sistema de Licitación Electrónica que permitirá la presentación de ofertas y proposiciones por vía telemática.

El capítulo IV regula determinados sistemas de tramitación y gestión electrónica de importante incidencia en la contratación y la facturación electrónica.

Finalmente, en el capítulo V se crea un Portal de Contratación Pública de la Comunidad Autónoma como punto central de acceso y entrada para los interesados a todas las aplicaciones y servicios que en materia de contratación pública y por vía electrónica, se puedan realizar a través de internet.

El Decreto modifica también, en la disposición final, el Decreto 262/2001, de 20 septiembre, por el que se refunde la normativa reguladora del Registro General de Contratistas de la Comunidad Autónoma de Galicia al introducir la tramitación por medios electrónicos, informáticos y telemáticos (EIT) de los procesos de alta, modificación y baja del registro y un sistema de notificaciones electrónicas con las empresas registradas

Autor:

Alfonso García Magariños

Director Asesoría Jurídica Municipal Concello de A Coruña

# **15. NORMATIVA EN EL ÁMBITO DE LA PROPIEDAD INTELECTUAL. LA PROTECCIÓN JURÍDICA DE LOS PROGRAMAS DE ORDENADOR. TIPOS DE LICENCIAS. SOFTWARE DE FUENTES ABIERTAS (FLOSS).**

## **Tema 15.- Normativa en el ámbito de la propiedad intelectual. La protección jurídica de los programas de ordenador. Tipos de licencias: Software de fuentes abiertas ( FLOSS )**

### **INDICE**

#### 15.1 Normativa en el ámbito de la propiedad intelectual.

##### 15.1.1 Estatal

##### 15.1.2 Comunitaria

##### 15.1.3 Internacional

#### 15.2 Protección jurídica programas ordenador

##### 15.2.1 Introducción

##### 15.2.2 Normativa Estatal

#### 15.3 Tipos de licencias

#### 15.4 Software de fuentes abiertas FLOSS

### **15.1- NORMATIVA EN EL ÁMBITO DE LA PROPIEDAD INTELECTUAL.**

Según la Organización Mundial de la Propiedad Intelectual, la propiedad intelectual (P.I.) tiene que ver con las creaciones de la mente: las invenciones, las obras literarias y artísticas, los símbolos, los nombres, las imágenes y los dibujos y modelos utilizados en el comercio.

La propiedad intelectual se divide en dos categorías: la propiedad industrial, que incluye las invenciones, patentes, marcas, dibujos y modelos industriales e indicaciones geográficas de procedencia; y el derecho de autor, que abarca las obras literarias y artísticas, tales como las novelas, los poemas y las obras de teatro, las películas, las obras musicales, las obras de arte, tales como los dibujos, pinturas, fotografías y esculturas, y los diseños arquitectónicos. Los derechos relacionados con el derecho de autor son los derechos de los artistas intérpretes y ejecutantes sobre sus interpretaciones y ejecuciones, los derechos de los productores de fonogramas sobre sus grabaciones y los derechos de los organismos de radiodifusión sobre sus programas de radio y de televisión

Por lo que respecta a los derechos que conforman la propiedad intelectual se distinguen los derechos morales y los derechos patrimoniales:

Derechos morales:

Frente a los sistemas de corte anglosajón, la legislación española es claramente defensora de los derechos morales, reconocidos para los autores y para los artistas intérpretes o ejecutantes. Estos derechos son irrenunciables e inalienables, acompañan al autor o al artista intérprete o ejecutante durante toda su vida y a sus herederos o causahabientes al



fallecimiento de aquellos. Entre ellos destaca el derecho al reconocimiento de la condición de autor de la obra o del reconocimiento del nombre del artista sobre sus interpretaciones o ejecuciones, y el de exigir el respeto a la integridad de la obra o actuación y la no alteración de las mismas.

Derechos de carácter patrimonial:

Hay que distinguir entre:

Derechos relacionados con la explotación de la obra o prestación protegida, que a su vez se subdividen en derechos exclusivos y en derechos de remuneración:

1.-Los derechos exclusivos son aquellos que permiten a su titular autorizar o prohibir los actos de explotación de su obra o prestación protegida por el usuario, y a exigir de este una retribución a cambio de la autorización que le conceda.

2.-Los derechos de remuneración, a diferencia de los derechos exclusivos, no facultan a su titular a autorizar o prohibir los actos de explotación de su obra o prestación protegida por el usuario, aunque si obligan a este al pago de una cantidad dineraria por los actos de explotación que realice, cantidad esta que es determinada, bien por la ley o en su defecto por las tarifas generales de las entidades de gestión.

3.-Derechos compensatorios, como el derecho por copia privada que compensa los derechos de propiedad intelectual dejados de percibir por razón de las reproducciones de las obras o prestaciones protegidas para uso exclusivamente privado del copista

### **15.1.1 Normativa Estatal**

La propiedad intelectual es el conjunto de derechos que corresponden a los autores y a otros titulares (artistas, productores, organismos de radiodifusión...) respecto de las obras y prestaciones fruto de su creación.

La propiedad intelectual, tal y como establece el Código Civil en sus artículos 428 y 429, forma parte de las llamadas propiedades especiales, y viene a constituir una forma especial de ejercer el derecho de propiedad sobre determinados objetos jurídicos que, por su cualidad, especializan el dominio.

Como propiedad especial, el Código Civil remite su regulación a una ley especial, y declara la aplicación supletoria de las reglas generales establecidas en el mismo sobre la propiedad para lo no específicamente previsto en dicha ley especial. Esta ley es la Ley de Propiedad Intelectual (LPI), cuyo Texto Refundido fue aprobado por el Real Decreto Legislativo 1/1996, de 12 de abril.

El citado Texto fue objeto de sucesivas modificaciones entre las que debemos destacar la operada por la ley 5/1998, de 6 de marzo, y la ley 23/2006 de 7 de julio, (responde esta a la necesidad de incorporar al derecho español una de las últimas directivas aprobadas en materia de propiedad intelectual, la Directiva 2001/29/CE del Parlamento Europeo y del Consejo, de 22 de mayo de 2001, relativa a la armonización de determinados aspectos de los derechos de autor y derechos afines a los derechos de autor en la sociedad de la información, con la que la Unión Europea, a su vez, ha querido cumplir los Tratados de la Organización Mundial de la Propiedad Intelectual (OMPI) de 1996 sobre Derecho de Autor y sobre Interpretación o Ejecución y Fonogramas )

Cabe también destacar la ley 19/2006, de 5 de junio, por la que se amplían los medios de tutela de los derechos de propiedad intelectual e industrial y

se establecen normas procesales para facilitar la aplicación de diversos reglamentos comunitarios.

### **15.1.2 Normativa comunitaria:**

1.-Directiva 92/100/CEE, de 19 de noviembre de 1992, sobre derechos de alquiler y préstamo y otros derechos afines a los derechos de autor en el ámbito de la propiedad intelectual. Esta Directiva, entre otros aspectos, reconoce el derecho de autorizar o prohibir el alquiler y préstamo de originales y copias de obras protegidas.

2.-Directiva 93/98/CEE, de 29 de octubre de 1993, relativa a la armonización del plazo de protección del derecho de autor y de determinados derechos afines. En esta Directiva se armoniza el plazo de protección del derecho de autor, fijándolo en un período de setenta años tras la muerte del autor o desde el momento de la primera difusión lícita entre el público, y por lo que se refiere a los derechos afines, en cincuenta años desde que se produce el hecho generador.

3.-Directiva 96/9/CE, de 11 de marzo de 1996, sobre la protección jurídica de las bases de datos. Reconoce un derecho de autor al creador de la estructura de la base de datos, y un derecho "sui generis" al fabricante de la misma, entendiendo por tal la persona física o jurídica que ha realizado una inversión sustancial para la fabricación de las bases de datos.

4.-Directiva 2001/29/CE, de 22 de mayo de 2001, relativa a la armonización de determinados aspectos de los derechos de autor y derechos afines en la sociedad de la información, adecua el sistema de derechos de autor y conexos al entorno digital, asumiendo al mismo tiempo las obligaciones contraídas por la Unión Europea y sus Estados Miembros en el marco de los Tratados Digitales OMPI (WCT y WPPT).

5.-Directiva 2000/31/CE del Parlamento Europeo y del Consejo, de 8 de junio de 2000 (DOCE de 17 de julio) relativa a determinados aspectos jurídicos de los servicios de la sociedad de la información, en particular el comercio electrónico (Directiva sobre el comercio electrónico). En esa Directiva se regula, entre otros aspectos y por lo que interesa a la materia de propiedad intelectual, la responsabilidad de los prestadores de servicios intermediarios (artículos 12 a 15 de la Directiva).

### **15.1.3 Normativa internacional:**

1.-El Convenio de Berna, que protege las obras literarias y artísticas, datando su acta originaria de 1886, siendo España socio fundador del mismo. Entre los principios informadores del Convenio se encuentran el de trato nacional (o asimilación del extranjero al nacional), el de protección automática, el de independencia de la protección, y el de protección mínima (para lograr un conjunto dispositivo uniformemente aplicable).

2.-Tratado OMPI sobre Derecho de Autor (Tratado WCT, 1996): como resultado de la Conferencia Diplomática de la OMPI sobre ciertas cuestiones de derechos de autor y de derechos conexos –celebrada en Ginebra en diciembre de 1996-, se adoptó éste tratado, orientado a ofrecer la necesaria respuesta legislativa a los problemas planteados por la tecnología digital, y particularmente por Internet.

3.-ISO 12083. Marcaje de documentos electrónicos

## **15.2.- PROTECCIÓN JURÍDICA DE PROGRAMAS DE ORDENADOR.**

### **15.2.1 Introducción**

Debemos partir, dada su importancia de la Directiva del Consejo de 14 de mayo de 1991 sobre la protección jurídica de programas de ordenador, allí

nos indica que a efectos de la presente Directiva, el término «programa de ordenador» incluye programas en cualquier forma, incluso los que están incorporados en el «hardware»; que este término designa también el trabajo preparatorio de concepción que conduce al desarrollo de un programa de ordenador, siempre que la naturaleza del trabajo preparatorio sea tal que más tarde pueda originar un programa de ordenador

Dicha directiva fue objeto de transposición al ordenamiento español por la ley 16/1993 de 23 de diciembre, siendo esta a su vez derogada por el Real Decreto Legislativo 1/1996, de 12 de abril, por el que se aprueba el Texto Refundido de la Ley de Propiedad Intelectual, regularizando, aclarando y armonizando las disposiciones legales vigentes sobre la materia.

La Directiva establece que los Estados miembros protegerán mediante derechos de autor los programas de ordenador como obras literarias tal como se definen en el Convenio de Berna para la protección de las obras literarias y artísticas.

A los fines de la presente Directiva, la expresión «programas de ordenador» comprenderá su documentación preparatoria

Respecto a la titularidad de los derechos; se considerará autor del programa de ordenador a la persona física o grupo de personas físicas que lo hayan creado o, cuando la legislación de los Estados miembros lo permita, a la persona jurídica que sea considerada titular del derecho por dicha legislación. Cuando la legislación de un Estado miembro reconozca las obras colectivas, la persona física o jurídica que según dicha legislación haya creado el programa, será considerada su autor.

Cuando un programa de ordenador se cree conjuntamente por varias personas físicas, los derechos exclusivos serán propiedad común.

Cuando un trabajador asalariado cree un programa de ordenador en el ejercicio de las funciones que le han sido confiadas, o siguiendo las instrucciones de su empresario, la titularidad de los derechos económicos correspondientes al programa de ordenador así creado corresponderán, exclusivamente, al empresario, salvo pacto en contrario.

La protección se concederá a todas las personas físicas y jurídicas que cumplan los requisitos establecidos en la legislación nacional sobre derechos de autor aplicable a las obras literarias

De conformidad con la Directiva, los derechos exclusivos del titular incluirán el derecho de realizar o de autorizar:

a) la reproducción total o parcial de un programa de ordenador por cualquier medio y bajo cualquier forma, ya fuere permanente o transitoria.

Cuando la carga, presentación, ejecución, transmisión o almacenamiento de un programa necesitan tal reproducción del mismo, estos actos estarán sujetos a la autorización del titular del derecho;

b) la traducción, adaptación, arreglo y cualquier otra transformación de un programa de ordenador y la reproducción de los resultados de tales actos, sin perjuicio de los derechos de la persona que transforme el programa de ordenador;

c) cualquier forma de distribución pública, incluido el alquiler, del programa de ordenador original o de sus copias. La primera venta en la Comunidad de una copia de un programa por el titular de los derechos o con su consentimiento, agotará el derecho de distribución en la Comunidad de dicha copia, salvo el derecho de controlar el subsiguiente alquiler del programa o de una copia del mismo.

No obstante lo anterior, salvo que existan disposiciones contractuales específicas, no necesitarán la autorización del titular los actos indicados en las letras a) y b) anteriormente citadas cuando dichos actos sean necesarios para la utilización del programa de ordenador por parte del adquirente legítimo con arreglo a su finalidad propuesta, incluida la corrección de errores.

La realización de una copia de salvaguardia por parte de una persona con derecho a utilizar el programa no podrá impedirse por contrato en tanto en cuanto resulte necesaria para dicha utilización.

El usuario legítimo de la copia de un programa estará facultado para observar, estudiar o verificar su funcionamiento, sin autorización previa del titular, con el fin de determinar las ideas y principios implícitos en cualquier elemento del programa, siempre que lo haga durante cualquiera de las operaciones de carga, visualización, ejecución, transmisión o almacenamiento del programa, que tiene derecho a hacer.

#### 15.2.2 Normativa Estatal

Desde el punto de vista de la normativa estatal se regula en el título VII del RD 1/1996, de 12 de abril, por el que se aprueba el Texto Refundido de la Ley de Propiedad Intelectual, que deroga a la citada ley 16/1993 de 23 de diciembre de transposición de la Directiva 91/250/CEE

El derecho de autor sobre los programas de ordenador se regirá por los preceptos del presente Título VII y, en lo que no esté específicamente previsto en el mismo, por las disposiciones que resulten aplicables de la Ley.

A los efectos de la Ley se entenderá por programa de ordenador toda secuencia de instrucciones o indicaciones destinadas a ser utilizadas,

directa o indirectamente, en un sistema informático para realizar una función o una tarea o para obtener un resultado determinado, cualquiera que fuere su forma de expresión y fijación.

A los mismos efectos, la expresión programas de ordenador comprenderá también su documentación preparatoria. La documentación técnica y los manuales de uso de un programa gozarán de la misma protección que este Título dispensa a los programas de ordenador.

El programa de ordenador será protegido únicamente si fuese original, en el sentido de ser una creación intelectual propia de su autor.

La protección prevista se aplicará a cualquier forma de expresión de un programa de ordenador. Asimismo, esta protección se extiende a cualesquiera versiones sucesivas del programa así como a los programas derivados, salvo aquellas creadas con el fin de ocasionar efectos nocivos a un sistema informático.

Cuando los programas de ordenador formen parte de una patente o un modelo de utilidad gozarán, de la protección que pudiera corresponderles por aplicación del régimen jurídico de la propiedad industrial.

No estarán protegidos mediante los derechos de autor las ideas y principios en los que se basan cualquiera de los elementos de un programa de ordenador incluidos los que sirven de fundamento a sus interfaces.

### 1.-Titularidad

Será considerado autor del programa de ordenador la persona o grupo de personas naturales que lo hayan creado, o la persona jurídica que sea contemplada como titular de los derechos de autor en los casos expresamente previstos por la ley, cuando se trate de una obra colectiva



tendrá la consideración de autor, salvo pacto en contrario, la persona natural o jurídica que la edite y divulgue bajo su nombre.

Los derechos de autor sobre un programa de ordenador que sea resultado unitario de la colaboración entre varios autores serán propiedad común y corresponderán a todos éstos en la proporción que determinen.

Cuando un trabajador asalariado cree un programa de ordenador, en el ejercicio de las funciones que le han sido confiadas o siguiendo las instrucciones de su empresario, la titularidad de los derechos de explotación correspondientes al programa de ordenador así creado, tanto el programa fuente como el programa objeto, corresponderán, exclusivamente, al empresario, salvo pacto en contrario.

La protección se concederá a todas las personas naturales y jurídicas que cumplan los requisitos establecidos en la ley para la protección de los derechos de autor.

## 2.-Duración de la protección.

Cuando el autor sea una persona natural la duración de los derechos de explotación de un programa de ordenador será, según los distintos supuestos que pueden plantearse, la prevista en el Capítulo I del Título III del RD 1/1996, de 12 de abril, por el que se aprueba el Texto Refundido de la Ley de Propiedad Intelectual,

Cuando el autor sea una persona jurídica la duración de los derechos a que se refiere el párrafo anterior será de setenta años, computados desde el día 1 de enero del año siguiente al de la divulgación lícita del programa o al de su creación si no se hubiera divulgado.

## 3.-Contenido de los derechos de explotación.

Los derechos exclusivos de la explotación de un programa de ordenador por parte de quien sea su titular, incluirán el derecho de realizar o de autorizar:

- a) La reproducción total o parcial, incluso para uso personal, de un programa de ordenador, por cualquier medio y bajo cualquier forma, ya fuere permanente o transitoria. Cuando la carga, presentación, ejecución, transmisión o almacenamiento de un programa necesiten tal reproducción deberá disponerse de autorización para ello, que otorgará el titular del derecho.
- b) La traducción, adaptación, arreglo o cualquier otra transformación de un programa de ordenador y la reproducción de los resultados de tales actos, sin perjuicio de los derechos de la persona que transforme el programa de ordenador.
- c) Cualquier forma de distribución pública incluido el alquiler del programa de ordenador original o de sus copias.

A tales efectos, cuando se produzca cesión del derecho de uso de un programa de ordenador, se entenderá, salvo prueba en contrario, que dicha cesión tiene carácter no exclusivo e intransferible, presumiéndose, asimismo, que lo es para satisfacer únicamente las necesidades del usuario. La primera venta en la Unión Europea de una copia de un programa por el titular de los derechos o con su consentimiento, agotará el derecho de distribución de dicha copia, salvo el derecho de controlar el subsiguiente alquiler del programa o de una copia del mismo.

#### 4.-Límites a los derechos de explotación.

No necesitarán autorización del titular, salvo disposición contractual en contrario,

a) La reproducción o transformación de un programa de ordenador incluida la corrección de errores, cuando dichos actos sean necesarios para la utilización del mismo por parte del usuario legítimo, con arreglo a su finalidad propuesta.

b) La realización de una copia de seguridad por parte de quien tiene derecho a utilizar el programa no podrá impedirse por contrato en cuanto resulte necesaria para dicha utilización.

c) El usuario legítimo de la copia de un programa estará facultado para observar, estudiar o verificar su funcionamiento, sin autorización previa del titular, con el fin de determinar las ideas y principios implícitos en cualquier elemento del programa, siempre que lo haga durante cualquiera de las operaciones de carga, visualización, ejecución, transmisión o almacenamiento del programa que tiene derecho a hacer.

El autor, salvo pacto en contrario, no podrá oponerse a que el cesionario titular de derechos de explotación realice o autorice la realización de versiones sucesivas de su programa ni de programas derivados del mismo.

No será necesaria la autorización del titular del derecho cuando la reproducción del código y la traducción de su forma sea indispensable para obtener la información necesaria para la interoperabilidad de un programa creado de forma independiente con otros programas, siempre que se cumplan los siguientes requisitos:

1.-Que tales actos sean realizados por el usuario legítimo o por cualquier otra persona facultada para utilizar una copia del programa, o, en su nombre, por parte de una persona debidamente autorizada.

2.-Que la información necesaria para conseguir la interoperabilidad no haya sido puesta previamente y de manera fácil y rápida, a disposición de las personas a que se refiere el párrafo anterior.

3.-Que dichos actos se limiten a aquellas partes del programa original que resulten necesarias para conseguir la interoperabilidad.

#### 5.-Protección registral.

Los derechos sobre los programas de ordenador, así como sobre sus sucesivas versiones y los programas derivados, podrán ser objeto de inscripción en el Registro de la Propiedad Intelectual.

#### 6.-Infracción de los derechos.

De acuerdo con la normativa vigente tendrán la consideración de infractores de los derechos de autor quienes, sin autorización del titular de los mismos, realicen los actos siguientes previstos en el artículo 99 al indicar este que

*“La reproducción total o parcial, incluso para uso personal, de un programa de ordenador, por cualquier medio y bajo cualquier forma, ya fuere permanente o transitoria. Cuando la carga, presentación, ejecución, transmisión o almacenamiento de un programa necesiten tal reproducción deberá disponerse de autorización para ello, que otorgará el titular del derecho.*

*b. La traducción, adaptación, arreglo o cualquier otra transformación de un programa de ordenador y la reproducción de los resultados de tales actos, sin perjuicio de los derechos de la persona que transforme el programa de ordenador.*

*c. Cualquier forma de distribución pública incluido el alquiler del programa de ordenador original o de sus copias."*

Además en particular, se considerarn infractores:

- 1.- Quienes pongan en circulación una o más copias de un programa de ordenador conociendo o pudiendo presumir su naturaleza ilegítima.
- 2.-Quienes tengan con fines comerciales una o más copias de un programa de ordenador, conociendo o pudiendo presumir su naturaleza ilegítima.
- 3.-Quienes pongan en circulación o tengan con fines comerciales cualquier instrumento cuyo único uso sea facilitar la supresión o neutralización no autorizadas de cualquier dispositivo técnico utilizado para proteger un programa de ordenador.

#### 7.- Medidas de protección.

El titular de los derechos reconocidos sobre programas de ordenador que, se disponen en la ley, es decir

- 1.- Podrá pedir el cese de la actividad ilícita, que podrá comprender, la suspensión de la explotación o actividad infractora,
- 2.-La prohibición al infractor de reanudar la explotación o actividad infractora,
- 3.-La retirada del comercio de los ejemplares ilícitos y su destrucción,
- 4.-La retirada de los circuitos comerciales, la inutilización, y, en caso necesario, la destrucción de los moldes, planchas, matrices, negativos y demás elementos materiales, equipos o instrumentos destinados principalmente a la reproducción, a la creación o fabricación de ejemplares ilícitos,

5.-La remoción o el precinto de los aparatos utilizados en la comunicación pública no autorizada de obras o prestaciones.

6.- La remoción o el precinto de los instrumentos utilizados para facilitar la supresión o la neutralización no autorizadas de cualquier dispositivo técnico utilizado para proteger obras o prestaciones aunque aquella no fuera su único uso,

7.-Lla suspensión de los servicios prestados por intermediarios a terceros que se valgan de ellos para infringir derechos de propiedad intelectual

Además las medidas cautelares procedentes, conforme a lo dispuesto en la Ley de Enjuiciamiento Civil.

### **15.3.-TIPOS DE LICENCIAS**

Software Libre o Free Software es un software disponible para cualquiera que desee utilizarlo, copiarlo y distribuirlo, ya sea en su forma original o con modificaciones. La posibilidad de modificaciones implica que el código fuente está disponible. Si un programa es libre, puede ser potencialmente incluido en un sistema operativo también libre. Es importante no confundir software libre con software gratis, porque la libertad asociada al software libre de copiar, modificar y redistribuir, no significa gratuidad. Existen programas gratuitos que no pueden ser modificados ni redistribuidos. Y existen programas pagos.

#### Copyleft.

La mayoría de las licencias usadas en la publicación de software libre permite que los programas sean modificados y redistribuidos. Estas prácticas están generalmente prohibidas por la legislación internacional de copyright, que intenta impedir que alteraciones y copias sean efectuadas sin la autorización del o los autores. Las licencias que acompañan al software libre hacen uso de la legislación de copyright para impedir la utilización no autorizada, pero estas licencias definen clara y

explícitamente las condiciones bajo las cuales pueden realizarse copias, modificaciones y redistribuciones, con el fin de garantizar las libertades de modificar y redistribuir el software registrado. A esta versión de copyright, se le da el nombre de copyleft.

### GPL.

La Licencia Pública General GNU (GNU General Public License GPL) es la licencia que acompaña los paquetes distribuidos por el Proyecto GNU, más una gran variedad de software que incluye el núcleo del sistema operativo Linux. La formulación de GPL es tal que en vez de limitar la distribución del software que protege, llega hasta impedir que este software sea integrado en software propietario. La GPL se basa en la legislación internacional de copyright, lo que debe garantizar cobertura legal para el software licenciado con GPL.

### Debian.

La licencia Debian es parte del contrato realizado entre Debian y la comunidad de usuarios de software libre, y se denomina Debian Free Software Guidelines (DFSG). En esencia, esta licencia contiene criterios para la distribución que incluyen, además de la exigencia de publicación del código fuente: (a) la redistribución libre ; (b) el código fuente debe ser incluido y debe poder ser redistribuido; (c) todo trabajo derivado debe poder ser redistribuido bajo la misma licencia del original; (d) puede haber restricciones en cuanto a la redistribución del código fuente, si el original fue modificado; (e) la licencia no puede discriminar a ninguna persona o grupo de personas, así como tampoco ninguna forma de utilización del software; (f) los derechos otorgados no dependen del sitio en el que el software se encuentra; y (g) la licencia no puede 'contaminar' a otro software.

### BSD.

La licencia BSD cubre las distribuciones de software de Berkeley Software Distribution, además de otros programas. Ésta es una licencia considerada 'permisiva', ya que impone pocas restricciones sobre la forma de uso, alteraciones y redistribución del software. El software puede ser vendido y no hay obligaciones de incluir el código fuente. Esta licencia garantiza el crédito a los autores del software pero no intenta garantizar que las modificaciones futuras permanezcan siendo software libre.

#### X.org.

El Consorcio X distribuye X Window System bajo una licencia que lo hace software libre, aunque sin adherirse al copyleft. Existen distribuciones bajo la licencia de la X.org que son software libre, y otras distribuciones que no lo son. Existen algunas versiones no-libres del sistema de ventanas X11 para estaciones de trabajo y ciertos dispositivos de IBM-PC que son las únicas funciones disponibles, sin otros similares que sean distribuidos como software libre.

#### Software con Dominio Público.

El Software con dominio público es software sin copyright. Algunos tipos de copia o versiones modificadas pueden no ser libres si el autor impone restricciones adicionales en la redistribución del original o de trabajos derivados.

#### Software Semi-libre.

El Software semi-libre es un software que no es libre pero permite que otros individuos lo usen, lo copien, lo distribuyan y hasta lo modifiquen. Ejemplos de software semi-libre son las primeras versiones de Internet Explorer de Microsoft, o algunas versiones de browsers de Netscape, y StarOffice.

#### Freeware.



El término freeware no posee una definición ampliamente aceptada, pero es utilizada para programas que permiten la redistribución pero no la modificación, y que incluyen su código fuente. Estos programas no son software libre.

#### Shareware.

Shareware es el software disponible con el permiso para que sea redistribuido, pero su utilización implica el pago. Generalmente, el código fuente no se encuentra disponible, y por lo tanto es imposible realizar modificaciones.

#### Software Propietario.

El Software propietario es aquel cuya copia, redistribución o modificación están, en alguna medida, prohibidos por su propietario. Para usar, copiar o redistribuir, se debe solicitar permiso al propietario o pagar.

#### Software Comercial.

El Software comercial es el software desarrollado por una empresa con el objetivo de lucrar con su utilización. Nótese que "comercial" y "propietario" no son lo mismo. La mayor parte del software comercial es propietario, pero existe software libre que es comercial, y existe software no-libre que no es comercial.

### **15.4 .-SOFTWARE DE FUENTES ABIERTAS.**

El software libre y de código abierto (también conocido como FOSS o FLOSS, siglas de free/libre and open source software, en inglés) es el software que está licenciado de tal manera que los usuarios pueden estudiar, modificar y mejorar su diseño mediante la disponibilidad de su código fuente.

El término "software libre y de código abierto" abarca los conceptos de software libre y software de código abierto, que, si bien comparten modelos de desarrollo similares, tienen diferencias en sus aspectos filosóficos. El software libre se enfoca en las libertades filosóficas que les otorga a los usuarios mientras que el software de código abierto se enfoca en las ventajas de su modelo de desarrollo. "FOSS" es un término imparcial respecto a ambas filosofías.

El software gratis no necesariamente tiene que ser libre o de código abierto

### Comparación entre software libre y de código abierto

Para que un software sea definido como libre o de código abierto, o ambos, debe cumplir ciertas reglas o normas para poseer esta denominación:

#### Las 4 libertades del software libre

- 1.- Ejecutar el programa con cualquier propósito (libertad 0)  
(privado, educativo, público, comercial, militar, etc.)
- 2.-Estudiar y modificar el programa (libertad 1)  
(para lo cual es necesario poder acceder al código fuente)
- 3.-Distribuir el programa de manera que se pueda ayudar al prójimo (libertad 2).
- 4.-Distribuir las versiones modificadas propias (libertad 3)

#### Las 10 premisas del software de código abierto

- 1.-Libre redistribución: el software debe poder ser regalado o vendido libremente.
- 2.-Código fuente: el código fuente debe estar incluido u obtenerse libremente.
- 3.-Trabajos derivados: la redistribución de modificaciones debe estar permitida

- 4.-Integridad del código fuente del autor: las licencias pueden requerir que las modificaciones sean redistribuidas sólo como parches.
- 5.- Sin discriminación de personas o grupos: nadie puede dejarse fuera.
- 6.-Sin discriminación de áreas de iniciativa: los usuarios comerciales no pueden ser excluidos.
- 7.-Distribución de la licencia: deben aplicarse los mismos derechos a todo el que reciba el programa.
- 8.-La licencia no debe ser específica de un producto: el programa no puede licenciarse solo como parte de una distribución mayor.
- 9.-La licencia no debe restringir otro software: la licencia no puede obligar a que algún otro software que sea distribuido con el software abierto deba también ser de código abierto.
- 10.-La licencia debe ser tecnológicamente neutral: no debe requerirse la aceptación de la licencia por medio de un acceso por clic de ratón o de otra forma específica del medio de soporte del software.

## Organizaciones y licencias tras el FOSS

Existen organizaciones detrás de cada iniciativa de distinción del software.

Por parte del software libre, existe la Free Software Foundation (FSF); apoyando el concepto de software de código abierto existe la Open Source Initiative (OSI). Ambas se enfocan en diferentes aspectos del uso y distribución del software, y su disponibilidad y responsabilidades que competen al usuario tener. Por este motivo existen diferentes licencias que las diferencian:

Licencias de código abierto (para el software de código abierto), licencias de software libre (para el software libre), entre otras, sin protección heredada y con protección heredada.

Autor:

Alfonso García Magariños

Director Asesoría Jurídica Municipal Concello de A Coruña

**16. LEGISLACIÓN SOBRE  
PROTECCIÓN DE DATOS. LEY  
ORGÁNICA 15/1999, DE  
PROTECCIÓN DE DATOS DE  
CARÁCTER PERSONAL. REAL  
DECRETO 1720/2007, DE 21 DE  
DICIEMBRE, POR EL QUE SE  
APRUEBA EL REGLAMENTO DE  
DESARROLLO DE LA LEY  
ORGÁNICA 15/1999, DE 13 DE  
DICIEMBRE, DE PROTECCIÓN  
DE DATOS DE CARÁCTER  
PERSONAL.**

**Tema 16.- Legislación sobre protección de datos. Ley Orgánica 15/1999, de Protección de Datos de Carácter Personal. Real Decreto 1720/2007, de 21 de diciembre, por el que se aprueba el Reglamento de desarrollo de la Ley Orgánica 15/1999, de 13 de diciembre, de Protección de Datos de Carácter Personal.**

## **INDICE**

### 16.1 Legislación sobre protección de datos

#### 16.1.1 Normativa Española

#### 16.1.2 Normativa Comunitaria

### 16.2.- Ley Orgánica 15/1999, de 13 de diciembre, de Protección de Datos de Carácter Personal

#### 16.2.1 Introducción

#### 16.2.2 Principios.

#### 16.2.3 Derechos

#### 16.2.4 Ficheros Públicos y Privados

#### 16.2.5 Agencia Protección Datos.

### 16.3.- Real Decreto 1720/2007 por el que se aprueba el Reglamento de desarrollo de la Ley Orgánica 15/1999, de 13 de diciembre, de Protección de Datos de Carácter Personal

#### 16.3.1 Introducción

#### 16.3.2 Análisis

## **16.1 LEGISLACION SOBRE PROTECCION DE DATOS.**

Sin perjuicio del desarrollo a lo largo del tema de las normas más importantes, cabe destacar:

### **16.1.1 Normativa Española**

1.-Constitución Española de 1978.

2.-Ley Orgánica 15/1999, de 13 de diciembre, de Protección de Datos de Carácter Personal.

3.-Ley 2/2011, de 4 de marzo, de Economía Sostenible. Modificación de la LOPD. Disposición final quincuagésima sexta.

4.-Real Decreto 1720/2007, de 21 de diciembre, por el que se aprueba el Reglamento de desarrollo de la Ley Orgánica 15/1999, de 13 de diciembre, de protección de datos de carácter personal.

5.-Real Decreto 3/2010, de 8 de enero, por el que se regula el Esquema Nacional de Seguridad en el ámbito de la Administración Electrónica (DA.4ª)

6.-Real Decreto 1665/2008, de 17 de octubre, por el que se modifica el Estatuto de la Agencia Española de Protección de Datos, aprobado por el Real Decreto 428/1993, de 26 de marzo.

7.-Real Decreto 156/1996, de 2 de febrero, por el que se modifica el Estatuto de la Agencia Española de Protección de Datos.

8.-Real Decreto 428/1993, de 26 de marzo, por el que se aprueba el Estatuto de la Agencia Española de Protección de Datos

## 16.1.2 NORMATIVA COMUNITARIA

### General

En la Carta de los derechos fundamentales de la Unión Europea, de 7 de diciembre de 2000, el artículo 8 dispone:

*“Toda persona tiene derecho a la protección de los datos de carácter personal que la conciernan.*

*Estos datos se tratarán de modo leal, para fines concretos y sobre la base del consentimiento de la persona afectada o en virtud de otro fundamento legítimo previsto por la ley. Toda persona tiene derecho a acceder a los datos recogidos que la conciernan y a su rectificación.*

*El respeto de estas normas quedará sujeto al control de una autoridad independiente.”*

En las versiones consolidadas del Tratado de la Unión Europea y del Tratado de Funcionamiento de la Unión Europea. Publicadas en el Diario Oficial de la Unión Europea el 30 de marzo de 2010, se recogen aspectos relacionados con la protección de datos, así en el artículo 16 del Tratado de Funcionamiento de la U. E se dispone *“Toda persona tiene derecho a la protección de los datos de carácter personal que le conciernan.*

*2. El Parlamento Europeo y el Consejo establecerán, con arreglo al procedimiento legislativo ordinario, las normas sobre protección de las personas físicas respecto del tratamiento de datos de carácter personal por las instituciones, órganos y organismos de la Unión, así como por los Estados miembros en el ejercicio de las actividades comprendidas en el ámbito de aplicación del Derecho de la Unión, y sobre la libre circulación de estos datos. El respeto de dichas normas estará sometido al control de autoridades independientes.*



*Las normas que se adopten en virtud del presente artículo se entenderán sin perjuicio de las normas específicas previstas en el artículo 39 del Tratado de la Unión Europea. “*

*El artículo 39 del Tratado de la Unión Europea establece “De conformidad con el artículo 16 del Tratado de Funcionamiento de la Unión Europea, y no obstante lo dispuesto en su apartado 2, el Consejo adoptará una decisión que fije las normas sobre protección de las personas físicas respecto del tratamiento de datos de carácter personal por los Estados miembros en el ejercicio de las actividades comprendidas en el ámbito de aplicación del presente capítulo, y sobre la libre circulación de dichos datos. El respeto de dichas normas estará sometido al control de autoridades independientes.*

#### Reglamentos:

1.- Reglamento del Eurodac, No 2725/2000 Del Consejo de 11 de diciembre de 2000 relativo a la creación del sistema «Eurodac» para la comparación de las impresiones dactilares para la aplicación efectiva del Convenio de Dublín, allí se crea un sistema denominado «Eurodac», cuya finalidad será ayudar a determinar el Estado miembro responsable, con arreglo al Convenio de Dublín, del examen de las solicitudes de asilo presentadas en los Estados miembros y, además, facilitar la aplicación del Convenio de Dublín en las condiciones establecidas en el presente Reglamento, se considera que Las impresiones dactilares constituyen un elemento importante para determinar la identidad exacta de dichas personas. Es necesario crear un sistema para comparar sus datos dactiloscópicos, considerando los mismos como datos de carácter personal.

2.- Reglamento (CE) No 45/2001 del Parlamento Europeo y del Consejo de 18 de diciembre de 2000 relativo a la protección de las personas físicas en lo que respecta al tratamiento de datos personales, allí se establece que

las instituciones y los organismos creados por los Tratados constitutivos de las Comunidades comunitarios», garantizarán, de conformidad con el presente Reglamento, la protección de los derechos y las libertades fundamentales de las personas físicas, y en particular su derecho a la intimidad, en lo que respecta al tratamiento de los datos personales, y no limitarán ni prohibirán la libre circulación de datos personales entre ellos o entre ellos y destinatarios sujetos al Derecho nacional de los Estados miembros adoptado en aplicación de la Directiva 95/46/CE.

La autoridad de control independiente establecida por el presente Reglamento, denominada «Supervisor Europeo de Protección de Datos», supervisará la aplicación de las disposiciones del presente Reglamento a todas las operaciones de tratamiento realizadas por las instituciones y organismos comunitarios.

Se entiende por «datos personales»: toda información sobre una persona física identificada o identificable ; se considerará identificable toda persona cuya identidad pueda determinarse, directa o indirectamente, en particular mediante un número de identificación o uno o varios elementos específicos, característicos de su identidad física, fisiológica, psíquica, económica, cultural o social;

«tratamiento de datos personales» cualquier operación o conjunto de operaciones, efectuadas o no mediante procedimientos automatizados, aplicadas a datos personales, como la recogida, registro, organización, conservación, adaptación o modificación, extracción, consulta, utilización, comunicación por transmisión, difusión o cualquier otra forma que permita el acceso a los mismos, así como la alineación o interconexión, y el bloqueo, supresión o destrucción;

Las disposiciones del Reglamento se aplicarán al tratamiento de datos personales por parte de todas las instituciones y organismos

comunitarios, en la medida en que dicho tratamiento se lleve a cabo para el ejercicio de actividades que pertenecen al ámbito de aplicación del Derecho comunitario

Los datos personales deberán ser:

- a) tratados de manera leal y lícita;
- b) recogidos con fines determinados, explícitos y legítimos, y no ser tratados posteriormente de manera incompatible con dichos fines; no se considerará incompatible el tratamiento posterior de datos con fines históricos, estadísticos o científicos, siempre y cuando el responsable del tratamiento establezca las garantías oportunas, en particular para asegurar que los datos no serán tratados con otros fines y que no se utilizarán en favor de medidas o decisiones que afecten a personas concretas;
- c) adecuados, pertinentes y no excesivos con relación a los fines para los que se recaben y para los que se traten posteriormente;
- d) exactos y, si fuera necesario, actualizados; se tomarán todas las medidas razonables para la supresión o rectificación de los datos inexactos o incompletos en relación con los fines para los que fueron recogidos o para los que son tratados posteriormente;
- e) conservados en una forma que permita la identificación de los interesados durante un período no superior al necesario para la consecución de los fines para los que fueron recogidos o para los que se traten posteriormente. La institución o el organismo comunitario establecerá para los datos personales que deban ser archivados por un período más largo del mencionado para fines históricos, estadísticos o científicos, que dichos datos se archiven bien únicamente en forma anónima, o, cuando ello no sea posible, sólo con la identidad codificada del interesado. En cualquier caso, deberá imposibilitarse el uso de los datos salvo para fines históricos, estadísticos o científicos.

3.- Decisión del Consejo de 13 de septiembre de 2004 por la que se adoptan las normas de desarrollo del Reglamento (CE) no 45/2001 del Parlamento Europeo y del Consejo relativo a la protección de las personas físicas en lo que respecta al tratamiento de datos personales por las instituciones y los organismos comunitarios y a la libre circulación de estos datos.

Directivas:

1.-Directiva 2009/136/CE del Parlamento Europeo y del Consejo, de 25 de noviembre de 2009, por la que se modifican la Directiva 2002/22/CE relativa al servicio universal y los derechos de los usuarios en relación con las redes y los servicios de comunicaciones electrónicas, la Directiva 2002/58/CE relativa al tratamiento de los datos personales y a la protección de la intimidad en el sector de las comunicaciones electrónicas y el Reglamento (CE) no 2006/2004 sobre la cooperación en materia de protección de los consumidores

2.-Directiva 2006/24/CE, del Parlamento Europeo y del Consejo de 15 de marzo de 2006, sobre la conservación de datos generados o tratados en relación con la prestación de servicios de comunicaciones electrónicas de acceso público o de redes públicas de comunicaciones y por la que se modifica la Directiva 2002/58/CE.

3.-Directiva 2004/82/CE, del Consejo de 29 de abril de 2004, sobre la obligación de los transportistas de comunicar los datos de las personas transportadas.

4.-Directiva 2002/58/CE del Parlamento Europeo y del Consejo de 12 de julio de 2002, relativa al tratamiento de los datos personales y a la protección de la intimidad en el sector de las comunicaciones electrónicas (Directiva sobre privacidad y las comunicaciones electrónicas).

5.-Directiva 2002/22/CE del Parlamento Europeo y del Consejo, de 7 de marzo de 2002, relativa al servicio universal y los derechos de los usuarios en relación con las redes y los servicios de comunicaciones electrónicas (Directiva servicio universal).

6.-Directiva 2002/21/CE, del parlamento Europeo y del Consejo de 29 de abril de 2004, sobre la obligación de los transportistas de comunicar los datos de las personas transportadas.

7.-Directiva 2002/20/CE, del Parlamento Europeo y del Consejo, de 7 de marzo de 2002, relativa a la autorización de redes y servicios de comunicaciones electrónicas (Directiva de autorización).

8.-Directiva 2002/19/CE, del Parlamento Europeo y del Consejo, de 7 de marzo de 2002, relativa al acceso a las redes de comunicaciones electrónicas y recursos asociados, y a su interconexión (Directiva de acceso).

9.-Directiva 2000/31/CE, del Parlamento Europeo y del Consejo, de 8 de junio de 2000, relativa a determinados aspectos jurídicos de los servicios de la sociedad de la información, en particular el comercio electrónico en el mercado interior (Directiva sobre el comercio electrónico).

10.-Directiva 1999/93/CE, del Parlamento Europeo y del Consejo, de 13 de diciembre de 1999, por la que se establece un marco comunitario para la firma electrónica.

11.-Directiva 97/66/CE del Parlamento Europeo y del Consejo de 15 de diciembre de 1997 relativa al tratamiento de los datos personales y a la protección de la intimidad en el sector de las telecomunicaciones. (Derogada por la Directiva 2002/58/CE).

12.-Directiva 95/46/CE del Parlamento Europeo y del Consejo de 24 de octubre de 1995 relativa a la protección de las personas físicas en lo que respecta al tratamiento de datos personales y a la libre circulación de estos datos.

#### Decisiones:

1.-Decisión de la Comisión de 31 de enero de 2011 de conformidad con la Directiva 95/46/CE del Parlamento Europeo y del Consejo, relativa a la protección adecuada de los datos personales por el Estado de Israel en lo que respecta al tratamiento automatizado de los datos personales

2.-Decisión de la Comisión de 19 de octubre de 2010 de conformidad con la Directiva 95/46/CE del Parlamento Europeo y del Consejo, relativa a la adecuada protección de los datos personales en Andorra

3.-Decisión de la Comisión de 5 de marzo de 2010 con arreglo a la Directiva 95/46/CE del Parlamento Europeo y del Consejo, relativa a la protección adecuada dada en la Ley de las Islas Feroe sobre el tratamiento de datos personales

4.-Decisión de la Comisión (2010/87/UE), de 5 de febrero de 2010, relativa a las cláusulas contractuales tipo para la transferencia de datos personales a los encargados del tratamiento establecidos en terceros países, de conformidad con la Directiva 95/46/CE del Parlamento Europeo y del Consejo.

5.-Decisión de la Comisión de 8 de mayo de 2008 de conformidad con la Directiva 95/46/CE del Parlamento Europeo y del Consejo, relativa a la protección adecuada de los datos personales en Jersey

6.-Decisión de la Comisión de 6 de septiembre de 2005, relativa al carácter adecuado de la protección de los datos personales incluidos en los registros de nombres de los pasajeros (PNR) que se transfieren a la Agencia de Servicios de Fronteras de Canadá.

7.-Decisión de la Comisión de 27 de diciembre de 2004, por la que se modifica la Decisión 2001/497/CE en lo relativo a la introducción de un conjunto alternativo de cláusulas contractuales tipo para la transferencia de datos personales a terceros países.

8.-Decisión de la Comisión de 14 de mayo de 2004 relativa al carácter adecuado de la protección de los datos personales incluidos en los registros de nombres de los pasajeros que se transfieren al Servicio de aduanas y protección de fronteras de los Estados Unidos (Bureau of Customs and Border Protection).

9.-Decisión de la Comisión de 29 de abril de 2004 por la que se establece una lista de organismos cuyos investigadores pueden acceder, con fines científicos, a datos confidenciales.

10.-Decisión de la Comisión de 28 de abril de 2004 relativa al carácter adecuado de la protección de datos personales en la Isla de Man.

11.-Decisión de la Comisión de noviembre de 2003, relativa al carácter adecuado de la protección de datos personales en Guernsey.

12.-Decisión de la Comisión de 30 de junio de 2003, con arreglo a la Directiva 95/46/CE del Parlamento Europeo y del Consejo sobre la adecuación de la protección de los datos personales en Argentina.

13.-Decisión reguladora de la Unidad de Cooperación Judicial Eurojust.

14.-Decisión 2002/16/CE de la Comisión, de 27 de diciembre de 2001, relativa a 'Cláusulas contractuales tipo para la transferencia de datos personales a los encargados del tratamiento establecidos en terceros países, de conformidad con la Directiva 95/46/CE. (queda derogada a partir de 15 de mayo de 2010)..

15.-Decisión de la comisión de 20 de diciembre de 2001 con arreglo a la Directiva 95/46/CE del Parlamento Europeo y del Consejo, sobre la adecuación de la protección de los datos personales conferida por la ley canadiense Personal Information and Electronic Documents Act.

16.-Decisión 2001/497/CE de la Comisión, de 15 de junio de 2001, relativa a 'Cláusulas contractuales tipo para la transferencia de datos personales a un tercer país previstas en la Directiva 95/46/CE.

17.-Decisión de la Comisión de 26 de julio de 2000 con arreglo a la Directiva 95/46/CE del Parlamento Europeo y del Consejo relativa al nivel de protección adecuado de los datos personales en Suiza.

18.-Decisión de la Comisión de 26 de Julio de 2000 con arreglo a la Directiva 95/46/CE del Parlamento Europeo y del Consejo relativa a la protección adecuada de los datos personales en Hungría.

19.-Decisión de la Comisión de 26 de Julio de 2000 con arreglo a la Directiva 95/46/CE del Parlamento Europeo y del Consejo, sobre la adecuación conferida por los principios de puerto seguro para la protección de la vida privada y las correspondientes preguntas más frecuentes, publicadas por el Departamento de Comercio de Estados Unidos de América.

Convenios:



1.-Convenio de Europol.

2.-Convenio de Schengen.

3.-Convenio del Sistema de Información de Aduanas.

## **16.2 LEY ORGÁNICA 15/1999, DE 13 DE DICIEMBRE, DE PROTECCIÓN DE DATOS DE CARÁCTER PERSONAL.**

### **16.2.1 Introducción**

La Ley tiene por objeto garantizar y proteger, en lo que concierne al tratamiento de los datos personales, las libertades públicas y los derechos fundamentales de las personas físicas, y especialmente de su honor e intimidad personal y familiar

Será de aplicación a los datos de carácter personal registrados en soporte físico que los haga susceptibles de tratamiento, y a toda modalidad de uso posterior de estos datos por los sectores público y privado

A los efectos de la Ley Orgánica se entenderá por

- a) Datos de carácter personal: Cualquier información concerniente a personas físicas identificadas o identificables.
- b) Fichero: Todo conjunto organizado de datos de carácter personal, cualquiera que fuere la forma o modalidad de su creación, almacenamiento, organización y acceso.
- c) Tratamiento de datos: Operaciones y procedimientos técnicos de carácter automatizado o no, que permitan la recogida, grabación, conservación, elaboración, modificación, bloqueo y cancelación, así como las cesiones de datos que resulten de comunicaciones, consultas, interconexiones y transferencias.

- d) Responsable del fichero o tratamiento: Persona física o jurídica, de naturaleza pública o privada, u órgano administrativo, que decida sobre la finalidad, contenido y uso del tratamiento.
- e) Afectado o interesado: Persona física titular de los datos que sean objeto del tratamiento a que se refiere el apartado c) del presente artículo.
- f) Procedimiento de disociación: Todo tratamiento de datos personales de modo que la información que se obtenga no pueda asociarse a persona identificada o identificable.
- g) Encargado del tratamiento: La persona física o jurídica, autoridad pública, servicio o cualquier otro organismo que, solo o conjuntamente con otros, trate datos personales por cuenta del responsable del tratamiento.
- h) Consentimiento del interesado: Toda manifestación de voluntad, libre, inequívoca, específica e informada, mediante la que el interesado consienta el tratamiento de datos personales que le conciernen.
- i) Cesión o comunicación de datos: Toda revelación de datos realizada a una persona distinta del interesado.
- j) Fuentes accesibles al público: Aquellos ficheros cuya consulta puede ser realizada por cualquier persona, no impedida por una norma limitativa, o sin más exigencia que, en su caso, el abono de una contraprestación. Tienen la consideración de fuentes de acceso público, exclusivamente, el censo promocional, los repertorios telefónicos en los términos previstos por su normativa específica y las listas de personas pertenecientes a grupos de profesionales que contengan únicamente los datos de nombre, título, profesión, actividad, grado académico, dirección e indicación de su pertenencia al grupo. Asimismo, tienen el carácter de fuentes de acceso público, los Diarios y Boletines oficiales y los medios de comunicación.

#### 16.2.2 Principios.

1.- Calidad de los datos, los datos de carácter personal sólo se podrán recoger para su tratamiento, así como someterlos a dicho tratamiento, cuando sean adecuados, pertinentes y no excesivos en relación con el

ámbito y las finalidades determinadas, explícitas y legítimas para las que se hayan obtenido.

2.- Derecho de información en la recogida de datos, los interesados a los que se soliciten datos personales deberán ser previamente informados de modo expreso, preciso e inequívoco:

a) De la existencia de un fichero o tratamiento de datos de carácter personal, de la finalidad de la recogida de éstos y de los destinatarios de la información.

b) Del carácter obligatorio o facultativo de su respuesta a las preguntas que les sean planteadas.

c) De las consecuencias de la obtención de los datos o de la negativa a suministrarlos.

d) De la posibilidad de ejercitar los derechos de acceso, rectificación, cancelación y oposición.

e) De la identidad y dirección del responsable del tratamiento o, en su caso, de su representante.

3.- Principio del consentimiento del afectado, el tratamiento de los datos de carácter personal requerirá el consentimiento inequívoco del afectado, salvo que la Ley disponga otra cosa.

4.- Comunicación de datos, los datos de carácter personal objeto del tratamiento sólo podrán ser comunicados a un tercero para el cumplimiento de fines directamente relacionados con las funciones legítimas del cedente y del cesionario con el previo consentimiento del interesado.

5.- Deber de secreto, el responsable del fichero y quienes intervengan en cualquier fase del tratamiento de los datos de carácter personal están obligados al secreto profesional respecto de los mismos y al deber de guardarlos, obligaciones que subsistirán aun después de finalizar sus relaciones con el titular del fichero o, en su caso, con el responsable del mismo.

6.- Especial protección a determinados datos, nadie podrá ser obligado a declarar sobre su ideología, religión o creencias.

7.- Seguridad de los datos. El responsable del fichero, y, en su caso, el encargado del tratamiento, deberán adoptar las medidas de índole técnica y organizativas necesarias que garanticen la seguridad de los datos de carácter personal y eviten su alteración, pérdida, tratamiento o acceso no autorizado, habida cuenta del estado de la tecnología, la naturaleza de los datos almacenados y los riesgos a que están expuestos, ya provengan de la acción humana o del medio físico o natural.

#### 16.2.3 Derechos de las personas

Los ciudadanos tienen derecho a no verse sometidos a una decisión con efectos jurídicos, sobre ellos o que les afecte de manera significativa, que se base únicamente en un tratamiento de datos destinados a evaluar determinados aspectos de su personalidad, cabe destacar :

1.-Derecho de consulta al Registro General de Protección de Datos, cualquier persona podrá conocer, recabando a tal fin la información oportuna del Registro General de Protección de Datos, la existencia de tratamientos de datos de carácter personal, sus finalidades y la identidad del responsable del tratamiento. El Registro General será de consulta pública y gratuita

2.-Derecho de acceso, el interesado tendrá derecho a solicitar y obtener gratuitamente información de sus datos de carácter personal sometidos a tratamiento, el origen de dichos datos, así como las comunicaciones realizadas o que se prevén hacer de los mismos

3.-Derecho de rectificación y cancelación. El responsable del tratamiento tendrá la obligación de hacer efectivo el derecho de rectificación o cancelación del interesado en el plazo de diez días

4.-Derecho de tutela, las actuaciones contrarias a lo dispuesto en la Ley pueden ser objeto de reclamación por los interesados ante la Agencia Española de Protección de Datos.

5.- Derecho indemnización. Los interesados que, como consecuencia del incumplimiento de lo dispuesto en la presente Ley por el responsable o el encargado del tratamiento, sufran daño o lesión en sus bienes o derechos tendrán derecho a ser indemnizados.

#### 16.2.4 Ficheros Públicos y Privados.

La creación, modificación o supresión de los ficheros de las Administraciones públicas sólo podrán hacerse por medio de disposición general publicada en el Boletín Oficial del Estado o Diario oficial correspondiente.

Las disposiciones de creación o de modificación de ficheros deberán indicar:

a.-La finalidad del fichero y los usos previstos para el mismo.

b.-Las personas o colectivos sobre los que se pretenda obtener datos de carácter personal o que resulten obligados a suministrarlos.

c.-El procedimiento de recogida de los datos de carácter personal.

d.-La estructura básica del fichero y la descripción de los tipos de datos de carácter personal incluidos en el mismo.

e.-Las cesiones de datos de carácter personal y, en su caso, las transferencias de datos que se prevean a países terceros.

f.-Los órganos de las Administraciones responsables del fichero.

g.-Los servicios o unidades ante los que pudiesen ejercitarse los derechos de acceso, rectificación, cancelación y oposición.

h.-Las medidas de seguridad con indicación del nivel básico, medio o alto exigible.

Podrán crearse ficheros de titularidad privada que contengan datos de carácter personal cuando resulte necesario para el logro de la actividad u objeto legítimos de la persona, empresa o entidad titular y se respeten las garantías que esta Ley establece para la protección de las personas.

Toda persona o entidad que proceda a la creación de ficheros de datos de carácter personal lo notificará previamente a la Agencia Española de Protección de Datos.

Entre los extremos que debe contener la notificación, figurarán necesariamente el responsable del fichero, la finalidad del mismo, su ubicación, el tipo de datos de carácter personal que contiene, las medidas de seguridad, con indicación del nivel básico, medio o alto exigible y las cesiones de datos de carácter personal que se prevean realizar y, en su caso, las transferencias de datos que se prevean a países terceros.

Deberán comunicarse a la Agencia Española de Protección de Datos los cambios que se produzcan en la finalidad del fichero automatizado, en su responsable y en la dirección de su ubicación.

El Registro General de Protección de Datos inscribirá el fichero si la notificación se ajusta a los requisitos exigibles.

En caso contrario podrá pedir que se completen los datos que falten o se proceda a su subsanación.

Transcurrido un mes desde la presentación de la solicitud de inscripción sin que la Agencia Española de Protección de Datos hubiera resuelto sobre la misma, se entenderá inscrito el fichero automatizado a todos los efectos.

#### 16.2.5 Agencia Protección Datos

La Agencia Española de Protección de Datos es un ente de derecho público, con personalidad jurídica propia y plena capacidad pública y privada, que actúa con plena independencia de las Administraciones públicas en el ejercicio de sus funciones

Son funciones de la Agencia Española de Protección de Datos:

- 1.-Velar por el cumplimiento de la legislación sobre protección de datos y controlar su aplicación, en especial en lo relativo a los derechos de información, acceso, rectificación, oposición y cancelación de datos.
- 2.-Emitir las autorizaciones previstas en la Ley o en sus disposiciones reglamentarias.
- 3.-Dictar, en su caso, y sin perjuicio de las competencias de otros órganos, las instrucciones precisas para adecuar los tratamientos a los principios de la Ley.

4.-Atender las peticiones y reclamaciones formuladas por las personas afectadas.

5.-Proporcionar información a las personas acerca de sus derechos en materia de tratamiento de los datos de carácter personal.

6.-Requerir a los responsables y los encargados de los tratamientos, previa audiencia de éstos, la adopción de las medidas necesarias para la adecuación del tratamiento de datos a las disposiciones de esta Ley y, en su caso, ordenar la cesación de los tratamientos y la cancelación de los ficheros, cuando no se ajuste a sus disposiciones.

7.-Ejercer la potestad sancionadora en los términos previstos por la ley orgánica de protección da datos.

8.-Informar, con carácter preceptivo, los proyectos de disposiciones generales que desarrollen esta Ley.

9.-Recabar de los responsables de los ficheros cuanta ayuda e información estime necesaria para el desempeño de sus funciones.

10.-Velar por la publicidad de la existencia de los ficheros de datos con carácter personal, a cuyo efecto publicará periódicamente una relación de dichos ficheros con la información adicional que el Director de la Agencia determine.

11.-Redactar una memoria anual y remitirla al Ministerio de Justicia.

12.-Ejercer el control y adoptar las autorizaciones que procedan en relación con los movimientos internacionales de datos, así como desempeñar las funciones de cooperación internacional en materia de protección de datos personales.



13.-Velar por el cumplimiento de las disposiciones que la Ley de la Función Estadística Pública establece respecto a la recogida de datos estadísticos y al secreto estadístico, así como dictar las instrucciones precisas, dictaminar sobre las condiciones de seguridad de los ficheros constituidos con fines exclusivamente estadísticos

Las resoluciones de la Agencia Española de Protección de Datos se harán públicas, una vez hayan sido notificadas a los interesados. La publicación se realizará preferentemente a través de medios informáticos o telemáticos.

#### 16.2.6 Infracciones y Sanciones.

Los responsables de los ficheros y los encargados de los tratamientos estarán sujetos al régimen sancionador establecido en la presente Ley.

Las infracciones se calificarán como leves, graves o muy graves.

Son infracciones leves:

- 1.-No remitir a la Agencia Española de Protección de Datos las notificaciones previstas en esta Ley o en sus disposiciones de desarrollo.
- 2.-No solicitar la inscripción del fichero de datos de carácter personal en el Registro General de Protección de Datos.
- 3.-El incumplimiento del deber de información al afectado acerca del tratamiento de sus datos de carácter personal cuando los datos sean recabados del propio interesado.

4.-La transmisión de los datos a un encargado del tratamiento sin dar cumplimiento a los deberes formales establecidos en la Ley.

Son infracciones graves:

1.-Proceder a la creación de ficheros de titularidad pública o iniciar la recogida de datos de carácter personal para los mismos, sin autorización de disposición general, publicada en el Boletín Oficial del Estado o diario oficial correspondiente.

2.-Tratar datos de carácter personal sin recabar el consentimiento de las personas afectadas, cuando el mismo sea necesario conforme a lo dispuesto en la Ley y sus disposiciones de desarrollo.

3.-Tratar datos de carácter personal o usarlos posteriormente con conculcación de los principios y garantías establecidos en la Ley y las disposiciones que lo desarrollan, salvo cuando sea constitutivo de infracción muy grave.

4.-La vulneración del deber de guardar secreto acerca del tratamiento de los datos de carácter personal

5.-El impedimento o la obstaculización del ejercicio de los derechos de acceso, rectificación, cancelación y oposición.

6.-El incumplimiento del deber de información al afectado acerca del tratamiento de sus datos de carácter personal cuando los datos no hayan sido recabados del propio interesado.

7.-El incumplimiento de los restantes deberes de notificación o requerimiento al afectado

8.-Mantener los ficheros, locales, programas o equipos que contengan datos de carácter personal sin las debidas condiciones de seguridad que por vía reglamentaria se determinen.

9.-No atender los requerimientos o apercibimientos de la Agencia Española de Protección de Datos o no proporcionar a aquélla cuantos documentos e informaciones sean solicitados por la misma.

10.-La obstrucción al ejercicio de la función inspectora.

11.-La comunicación o cesión de los datos de carácter personal sin contar con legitimación para ello salvo que la misma sea constitutiva de infracción muy grave.

Son infracciones muy graves:

1.-La recogida de datos en forma engañosa o fraudulenta.

2.-Tratar o ceder los datos de carácter personal especialmente protegidos (ideología, afiliación sindical, religión y creencias, origen racial, a la salud y a la vida sexual, comisión de infracciones penales o administrativas)

3.-No cesar en el tratamiento ilícito de datos de carácter personal cuando existiese un previo requerimiento del Director de la Agencia Española de Protección de Datos para ello.

4.-La transferencia internacional de datos de carácter personal con destino a países que no proporcionen un nivel de protección equiparable sin autorización del Director de la Agencia Española de Protección de Datos salvo en los supuestos en los que dicha autorización no resulta necesaria.

Tipo de sanciones.

Las infracciones leves serán sancionadas con multa de 900 a 40.000 euros.

Las infracciones graves serán sancionadas con multa de 40.001 a 300.000 euros.

Las infracciones muy graves serán sancionadas con multa de 300.001 a 600.000 euros.

La cuantía de las sanciones se graduará atendiendo a los siguientes criterios:

a.-El carácter continuado de la infracción.

b.-El volumen de los tratamientos efectuados.

c.-La vinculación de la actividad del infractor con la realización de tratamientos de datos de carácter personal.

d.-El volumen de negocio o actividad del infractor.

e.-Los beneficios obtenidos como consecuencia de la comisión de la infracción.

f.-El grado de intencionalidad.

g.-La reincidencia por comisión de infracciones de la misma naturaleza.

h.-La naturaleza de los perjuicios causados a las personas interesadas o a terceras personas.

i.-La acreditación de que con anterioridad a los hechos constitutivos de infracción la entidad imputada tenía implantados procedimientos adecuados de actuación en la recogida y tratamiento de los datos de carácter personal, siendo la infracción consecuencia de una anomalía en el funcionamiento de dichos procedimientos no debida a una falta de diligencia exigible al infractor.

j.-Cualquier otra circunstancia que sea relevante para determinar el grado de antijuridicidad y de culpabilidad presentes en la concreta actuación infractora.

### **16.3 Real Decreto 1720/2007 por el que se aprueba el Reglamento de desarrollo de la Ley Orgánica 15/1999, de 13 de diciembre, de Protección de Datos de Carácter Personal**

#### **16.3.1 Introducción**

La Ley Orgánica 15/1999, de 13 de diciembre de Protección de datos de carácter personal adaptó nuestro ordenamiento a lo dispuesto por la Directiva 95/46/CE del Parlamento Europeo y del Consejo de 24 de octubre de 1995, relativa a la protección de las personas físicas en lo que respecta al tratamiento de datos personales y a la libre circulación de estos datos, derogando a su vez la hasta entonces vigente Ley Orgánica 5/1992, de 29 de octubre, de Regulación del tratamiento automatizado de datos de carácter personal.

El Reglamento comparte con la Ley Orgánica la finalidad de hacer frente a los riesgos que para los derechos de la personalidad pueden suponer el acopio y tratamiento de datos personales. Por ello, ha de destacarse que esta norma reglamentaria nace con la vocación de no reiterar los contenidos de la norma superior y de desarrollar, no sólo los mandatos contenidos en la Ley Orgánica de acuerdo con los principios que emanan

de la Directiva, sino también aquellos que en estos años de vigencia de la Ley se ha demostrado que precisan de un mayor desarrollo normativo.

El reglamento parte de la necesidad de dotar de coherencia a la regulación reglamentaria en todo lo relacionado con la transposición de la Directiva y de desarrollar los aspectos novedosos de la Ley Orgánica 15/1999, junto con aquellos en los que la experiencia ha aconsejado un cierto grado de precisión que dote de seguridad jurídica al sistema.

### 16.3.2 Análisis

El título I contempla el objeto y ámbito de aplicación del reglamento. A lo largo de la vigencia de la Ley Orgánica 15/1999, se ha advertido la conveniencia de aclarar qué se entiende por ficheros y tratamientos relacionados con actividades personales o domésticas, aspecto muy relevante dado que está excluido de la normativa sobre protección de datos de carácter personal.

Se aporta un conjunto de definiciones que ayudan al correcto entendimiento de la norma, lo que resulta particularmente necesario en un ámbito tan tecnificado como el de la protección de datos personales. Por otra parte, fija el criterio a seguir en materia de cómputo de plazos con el fin de homogeneizar esta cuestión evitando distinciones que suponen diferencias de trato de los ficheros públicos respecto de los privados.

El título II, se refiere a los principios de la protección de datos. Reviste particular importancia la regulación del modo de captación del consentimiento atendiendo a aspectos muy específicos como el caso de los servicios de comunicaciones electrónicas y, muy particularmente, la captación de datos de los menores. Asimismo, se ofrece lo que no puede definirse sino como un estatuto del encargado del tratamiento, que sin duda contribuirá a clarificar todo lo relacionado con esta figura. Las

previsiones en este ámbito se completan con lo dispuesto en el título VIII en materia de seguridad dotando de un marco coherente a la actuación del encargado.

El título III se ocupa de una cuestión tan esencial como los derechos de las personas en este ámbito. Estos derechos de acceso, rectificación, cancelación y oposición al tratamiento, según ha afirmado el Tribunal Constitucional en su sentencia número 292/2000, constituyen el haz de facultades que emanan del derecho fundamental a la protección de datos y sirven a la capital función que desempeña este derecho fundamental: garantizar a la persona un poder de control sobre sus datos personales, lo que sólo es posible y efectivo imponiendo a terceros los mencionados deberes de hacer.

A continuación, los títulos IV a VII permiten clarificar aspectos importantes para el tráfico ordinario, como la aplicación de criterios específicos a determinado tipo de ficheros de titularidad privada que por su trascendencia lo requerían -los relativos a la solvencia patrimonial y crédito y los utilizados en actividades de publicidad y prospección comercial-, el conjunto de obligaciones materiales y formales que deben conducir a los responsables a la creación e inscripción de los ficheros, los criterios y procedimientos para la realización de las transferencias internacionales de datos, y, finalmente, la regulación de un instrumento, el código tipo, llamado a jugar cada vez un papel más relevante como elemento dinamizador del derecho fundamental a la protección de datos.

El título VIII regula un aspecto esencial para la tutela del derecho fundamental a la protección de datos, la seguridad, que repercute sobre múltiples aspectos organizativos, de gestión y aún de inversión, en todas las organizaciones que traten datos personales. La repercusión del deber de seguridad obligaba a un particular rigor ya que en esta materia han confluído distintos elementos muy relevantes. Por una parte, la

experiencia dimanante de la aplicación del Real Decreto 994/1999 permitía conocer las dificultades que habían enfrentado los responsables e identificar los puntos débiles y fuertes de la regulación. Por otra, se reclamaba la adaptación de la regulación en distintos aspectos. En este sentido, el reglamento trata de ser particularmente riguroso en la atribución de los niveles de seguridad, en la fijación de las medidas que corresponda adoptar en cada caso y en la revisión de las mismas cuando ello resulte necesario. Por otra parte, ordena con mayor precisión el contenido y las obligaciones vinculadas al mantenimiento del documento de seguridad. Además, se ha pretendido regular la materia de modo que contemple las múltiples formas de organización material y personal de la seguridad que se dan en la práctica. Por último, se regula un conjunto de medidas destinadas a los ficheros y tratamientos estructurados y no automatizados que ofrezca a los responsables un marco claro de actuación.

Finalmente en el título IX, dedicado a los procedimientos tramitados por la Agencia Española de Protección de Datos, se ha optado por normar exclusivamente aquellas especialidades que diferencian a los distintos procedimientos tramitados por la Agencia de las normas generales previstas para los procedimientos en la Ley 30/1992, de 26 de noviembre, de Régimen Jurídico de las Administraciones Públicas y del Procedimiento Administrativo Común, cuya aplicación se declara supletoria al presente Reglamento.

Autor:

Alfonso García Magariños

Director Asesoría Jurídica Municipal Concello de A Coruña



**17. PLAN DIRECTOR DE  
SEGURIDAD DE INFORMACIÓN  
DE LA XUNTA DE GALICIA.  
DECRETO 230/2008 POR EL  
QUE SE ESTABLECEN LAS  
NORMAS DE BUENAS  
PRÁCTICAS EN LA UTILIZACIÓN  
DE LOS SISTEMAS DE  
INFORMACIÓN DE LA  
ADMINISTRACIÓN DE LA  
COMUNIDAD AUTÓNOMA DE  
GALICIA.**

**Tema 17.- Plan Director de Seguridad de Información de la Xunta de Galicia. Decreto 230/2008 por el que se establecen las normas de buenas practicas en la utilización de sistemas de informacion en la Administración de la CCAA de Galicia.**

**INDICE**

17.1 Plan Director de Seguridad e Información de la Xunta de Galicia

- 17.1.1. Introducción
- 17.1.2 Responsabilidades
- 17.1.3 Objetivos
- 17.1.4 Agentes involucrados

17.2 Decreto 230/2008 por el que se establecen las normas de buenas practicas en la utilización de sistemas de informacion en la Administración de la CCAA de Galicia.

- 17.2.1 Objeto y ámbito de aplicación
- 17.2.2 Órganos responsables y de coordinación
- 17.2.3 Acceso a información, redes de comunicaciones e Internet
- 17.2.4 Servicio de mensajería corporativo
- 17.2.5 Deber de personas usuarias
- 17.2.6 Inspección
- 17.2.7 Responsabilidad de personas usuarias que tengan la condición de empleados públicos

## **17.1 PLAN DIRECTOR DE SEGURIDAD E INFORMACION DE LA XUNTA DE GALICIA**

### **17.1.1 Introducción.**

El Plan Director de Seguridad de la Información establece las actuaciones a llevar a cabo por la Administración de la Comunidad Autónoma de Galicia en materia de seguridad de la información así como los agentes involucrados y sus respectivas responsabilidades. Además, define las directrices necesarias para gestionar de forma segura los sistemas de información de la Administración y manifiesta el reconocimiento de la importancia que tiene la seguridad de la información sobre la confianza que los ciudadanos depositan en la administración.

Las tecnologías de la información y las comunicaciones (TIC) constituyen un instrumento de alto nivel estratégico por su potencial para impulsar la modernización de la Administración de la Comunidad Autónoma Gallega, así como por su capacidad para estimular y sustentar el desarrollo social y económico de Galicia.

Este plan contempla todos los sistemas de información de la Xunta de Galicia. Nace con la voluntad de ser aplicado progresivamente al conjunto de Consellerías de la Administración autonómica, a sus organismos autónomos, sociedades públicas, fundaciones del sector público autonómico y demás entidades de derecho público vinculadas o dependientes de la Comunidad Autónoma de Galicia. Sin perjuicio del anterior, está directamente destinado a garantizar la seguridad de los sistemas corporativos.

Dentro de este plan, corresponde a la Secretaría General de Modernización e Innovación Tecnológica el análisis de necesidades, planificación, diseño, gestión e implantación de los sistemas de información y elementos tecnológicos en los órganos de la Administración de Justicia en Galicia, en coordinación con las administraciones y órganos competentes en la materia de sistemas de información de justicia.

En este Plan también se perfilan una serie de iniciativas orientadas a asesorar y sensibilizar al personal de la Administración Local en materia de protección de datos, de seguridad informática y de acceso electrónico de los ciudadanos a los servicios públicos.

El alcance temporal del presente plan abarca el período 2010 - 2014. En el plan se efectúa una priorización de las distintas actuaciones en orden a su urgencia e impacto. La seguridad de la información requiere una visión global que recoja una mejora pragmática y asequible a corto plazo y, a la vez, un modelo objetivo ambicioso y de largo plazo.

#### 17.1.2 Responsabilidades.

A la Secretaría General de Modernización e Innovación Tecnológica según el Decreto 325/2009, del 18 de junio, de estructura orgánica de los órganos superiores dependientes de la Presidencia de la Xunta de Galicia, le corresponde establecer la política de seguridad informática corporativa de la Xunta de Galicia y la promoción de buenas prácticas en lo relativo al tratamiento de datos de carácter personal. A través del Centro de Seguridad de la Información (CSI), elaborará los planes, medidas y directrices de seguridad informática, supervisará el cumplimiento de todas las medidas de seguridad informática en los diferentes ámbitos y departamentos y diseñará y realizará las acciones encaminadas a garantizar el cumplimiento de la normativa vigente en materia de

## Protección de Datos de Carácter Personal en la Administración de la Comunidad Autónoma.

Las Consellerías de la Administración de la Comunidad Autónoma de Galicia, según el Decreto 230/2008, de 18 de septiembre, por lo que se establecen las normas de buenas prácticas en la utilización de los sistemas de información de la Administración de la Comunidad Autónoma de Galicia designarán el órgano que será responsable de los sistemas de información de su propiedad y de establecer los medios tecnológicos que necesitan las personas a su servicio, así como de velar por el correcto funcionamiento de las infraestructuras y del equipamiento informático y de comunicaciones de que dispongan. Cuando estas atribuciones no estén asignadas reglamentariamente a un órgano, la designación se hará por las secretarías generales de cada departamento. Asimismo, cada departamento de la Xunta de Galicia deberá designar a una persona como responsable de seguridad

El Comité de Seguridad de los Sistemas de Información (CSSI), creado en el Decreto de buenas prácticas, es un órgano colegiado formado por las personas responsables de seguridad de los distintos departamentos de la Xunta de Galicia, que tiene como objetivo definir la política de seguridad corporativa. Este comité estará coordinado y asesorado por el órgano competente en materia de seguridad corporativa a través del Centro de seguridad de la Información

Las personas que prestan servicios a la Administración de la Comunidad Autónoma de Galicia, además de cumplir a con las medidas indicadas en el Decreto de buenas prácticas tienen deber de sigilo y confidencialidad respecto de la información a la que puedan tener acceso por razón de sus funciones, limitándose a emplearla para el estricto cumplimiento de las tareas encomendadas.

La Xunta de Galicia expresa su compromiso con la seguridad, de forma que dará a conocer este Plan Director de Seguridade entre todo el persoal que preste sus servicios en la Administración de la Comunidade Autónoma, velará por su cumplimiento e impulsará la implantación y difusión de la gestión de la seguridad de la información en las personas y empresas de Galicia

### 17.1.3 Objetivos

El presente plan trata de mejorar el nivel de seguridad existente en la Administración de la Comunidad Autónoma de Galicia. Esto significa gestionar la seguridad de la información de tal forma que las medidas de seguridad implantadas alcancen un alto grado de efectividad que reduzca al máximo el impacto de las incidencias de seguridad

Con este fin, la Xunta de Galicia se fija, en materia de seguridad de la información, los siguientes objetivos:

- . Concienciar e implicar en la gestión de la seguridad a toda la dirección y personal de la Administración autonómica, a sus organismos autónomos, sociedades públicas, fundaciones del sector público autonómico y demás entidades de derecho público vinculadas o dependientes de la Comunidad Autónoma de Galicia, contando con la colaboración de proveedores y especialistas
- . Promover el máximo aprovechamiento de las tecnologías de la información y las comunicaciones en la actividad administrativa y asegurar a la vez el respecto de las garantías y derechos de los ciudadanos en sus relaciones con la Administración, estableciendo que la seguridad en los sistemas sea atendida, revisada y auditada por personal calificado y que su configuración y diseño garanticen la seguridad por defecto.

- . Gestionar los riesgos que puedan afectar a los componentes de los sistemas de información para poder identificarlos y evaluarlos y tomar las medidas de seguridad informática adecuadas (físicas, lógicas, técnicas, normativas y organizativas).
- . Garantizar la disponibilidad de los sistemas de información de acuerdo con los requisitos establecidos para los servicios prestados, gestionando y controlando el acceso físico y lógico, certificando que los productos de seguridad usados para dar servicio al ciudadano cumplen con los estándares establecidos, revisando el nivel de actualización, asegurando la confidencialidad de la información gestionada por la Administración y evitando accesos o alteraciones indebidas y pérdidas de información, implantando la seguridad perimetral necesaria ante los posibles riesgos de interconexión con otros sistemas de redes externas y registrando toda actividad de los usuarios que accedan al sistema.
- . Gestionar la continuidad de los servicios TIC proporcionados por la Administración, estableciendo los sistemas de protección a nivel organizativo, lógico y físico que permitan reducir la probabilidad de que se produzca un incidente y, en caso de que se produzca, acortar el tiempo de vuelta a la normalidad y minimizar su impacto.
- . Evaluar periódicamente el sistema de gestión de seguridad, basándose en el registro y tratamiento de incidencias, garantizando que las medidas implantadas alcancen un nivel de madurez optimizado con políticas, normas, procedimientos y estructuras organizacionales que promuevan la concienciación y formación continua de los usuarios en temas de seguridad de la información.
- . Proporcionar un contorno de seguridad que garantice el cumplimiento de los requisitos legales para la validez y eficacia de los procedimientos administrativos que utilicen los medios electrónicos, informáticos y

telemáticos, estableciendo para cada sistema de información responsables diferenciados de la información, del servicio y de la seguridad.

. La Xunta de Galicia actuará como elemento generador de e-confianza que promueva un tejido empresarial sólido en seguridad de la información e incremente la confianza y la protección de los derechos de la ciudadanía gallega en la Sociedad de la información.

#### 17.1.4 Agentes involucrados

1.- Subdirección general de calidad, interoperabilidad y seguridad (sxcis). La Subdirección General de Calidad, Interoperabilidad y Seguridad (SXCIS) de la SXMIT tiene, entre sus funciones, la de que establezca la política de seguridad informática corporativa de la Xunta de Galicia y la de promover las buenas prácticas en lo relativo al tratamiento de datos de carácter personal.

Para eso, se creó, dentro de la SXCIS, el Servicio de Calidad y Seguridad Informática con las siguientes competencias en materia de seguridad de la información:

. Dirección de las políticas corporativas de seguridad informática de la Xunta de Galicia a través del Centro de Seguridad de la Información.

. Elaboración de los planes, medidas y directrices de seguridad informática para el conjunto de órganos y unidades de la Xunta.

. Supervisión del cumplimiento de todas las medidas de seguridad informática en los distintos ámbitos y departamentos de la Xunta.

. Diseño y realización de acciones encaminadas a garantizar el cumplimiento de la normativa vigente en materia de protección de datos



de carácter personal en la Administración de la Comunidad Autónoma de Galicia

## 2.- Centro de Seguridad de la información (CSI)

El Centro de Seguridad de la Información tiene una función transversal dentro de la Xunta. Entre sus atribuciones, se encuentra la resolución de incidencias de seguridad y el asesoramiento a las distintas consejerías en materia de seguridad de la información, tanto en materia de protección de datos como en la evaluación de soluciones de seguridad

El CSI supervisará la seguridad de los sistemas corporativos de la Administración de la Comunidad Autónoma de Galicia y será el centro de resolución de incidentes de seguridad (ataques activos y pasivos, pérdida de información, falta de disponibilidad, etc.).

Dentro de los servicios de apoyo y asesoramiento a las distintas consejerías en su cometido de adecuación de sus sistemas de información a las exigencias de la legislación vigente en materia de protección de datos, el Centro de Seguridad de la Información será el interlocutor con la Agencia Española de Protección de Datos.

El Centro de Seguridad de la Información se encargará de evaluar, analizar y probar en una contornea controlada las distintas soluciones de seguridad existentes en el mercado y de aplicación e interés para la Administración de la Comunidad Autónoma de Galicia. Los resultados de estas evaluaciones se reflejarán en una serie de informes de análisis y de estudios comparativos. Asimismo, podrán realizarse demostraciones a las consejerías, ya sea a través de la contornea de pruebas o mediante charlas divulgativas. Estas evaluaciones podrán realizarse en función de las necesidades detectadas por el propio Centro de Seguridad de la

Información o en base a la valoración de peticiones de los distintos departamentos y podrán referirse la seguridad perimetral, de centros de proceso de datos, de aplicaciones, de puesto informático, etc.

Con el fin de promover el desarrollo del presente Plan en las diferentes consejerías, se incluirán dentro de las funciones del CSI, las siguientes tareas

- . Realizar el seguimiento y revisar el presente Plan Director de forma periódica.
- . Apoyar y asesorar a las consejerías para alcanzar los objetivos fijados en el presente Plan
- . Impulsar y apoyar el Comité de Seguridad de los Sistemas de Información de la Xunta.
- . Velar por la mejora del nivel de seguridad de la información de la Administración de la Comunidad Autónoma de Galicia.

Para el desarrollo de estas funciones el Centro de Seguridad de la Información dispone de un amplio conocimiento en el campo de la seguridad de la información, y podrá contar con el asesoramiento de expertos externos.

### 3.- Consellerías da Xunta .

Las Consellerías son las máximas responsables del estado de seguridad de sus sistemas de información. Deberán velar por su seguridad garantizando la confidencialidad, integridad y disponibilidad de la información. Son también las responsables de acometer las actuaciones recomendadas en el presente Plan Director y alcanzar los objetivos marcados.

Dentro de cada consellería, el personal de soporte técnico desempeñará funciones de asesoría técnica, ejecución, propuesta, coordinación y supervisión de los planes de informatización.

Asimismo, los órganos responsables de los sistemas de información de cada departamento de la Xunta de Galicia, con el apoyo del personal de soporte técnico, serán los competentes para velar por el cumplimiento de los objetivos de este Plan. Les corresponderá a ellos asegurarse de que los equipos se utilizan adecuadamente y atendiendo a la finalidad a la que están destinados.

#### 4.- Comité de seguridad de sistemas de información.(CSSI).

El Comité de Seguridad de los Sistemas de Información, tiene como objetivo establecer el marco de trabajo que impulse la implantación y difusión de la gestión de la seguridad de la información en el ámbito de la Administración de la Comunidad Autónoma de Galicia.

Los miembros del Comité se reúnen para revisar el estado de la seguridad de la información en la Administración de la Comunidad Autónoma de Galicia, aprobar las políticas de seguridad, revisar y aprobar los proyectos de seguridad, revisar los procesos de monitorización de las incidencias de seguridad y realizar otras tareas de gestión de la seguridad de alto nivel que sean necesarias.

Las principales funciones asumidas por este Comité son las que se presentan a continuación:

- .Identificar, revisar y proponer objetivos estratégicos en materia de seguridad de la información.
- . Establecer roles y responsabilidades en materia de seguridad de la información.

- . Proponer y aprobar políticas, normas y directrices de seguridad de la información para la Xunta de Galicia y velar por su cumplimiento.
- . Proveyera un soporte al esfuerzo de seguridad de la información, dando una visión más transversal para el análisis y decisiones, a fin de lograr la mejor relación coste-efectividad en su gestión.
- . Constituir el canal primario de discusión de aspectos de seguridad de la información que se deban abordar en la Administración de la Comunidad Autónoma de Galicia.
- . Apoyar a los coordinadores de seguridad en el desarrollo de estrategias de mitigación de riesgos, basándose en el conocimiento que sus integrantes tienen de sus respectivas áreas.
- . Impulsar la implantación y difusión de la gestión de la seguridad de la información
- . Revisar y aprobar anualmente la Política de Seguridad.
- . Realizar otras tareas de gestión de la seguridad de alto nivel que sean necesarias.
- . Revisar periódicamente el presente Plan Director.

Para el desarrollo de estas funciones, el CSSI podrá contar con el apoyo, en materia de actualización normativa, de la Asesoría Jurídica General. Puntualmente, podrá solicitar la colaboración de la Junta Consultiva de Contratación Administrativa de la Xunta de Galicia y del equipo de auditores y analistas de gestión del rendimiento y calidad forma de la Dirección General de Evaluación y Reforma Administrativa de la Consellería de Presidencia, Administraciones Públicas y Justicia.

## 5.- Asesoría Xurídica Xeral.

El papel del derecho en las tecnologías de la información es amplio y abarca varios campos como pueden ser los que enunciamos a continuación:

- . Protección de datos de carácter personal.
- . Propiedad Intelectual.

- . Servicios de la sociedad de la información.
- . Firma electrónica.
- . Constitución de prueba por medios informáticos.
- . Uso de herramientas informática por el personal.

La Asesoría Jurídica General, acercará al Comité de Seguridad de los Sistemas de Información su conocimiento en el ámbito legal como apoyo a la actualización normativa, mediante la emisión de dictámenes o informes en derecho, la formulación de criterios generales de asesoramiento jurídico y el estudio de los proyectos de reglamentos con examen de su adecuación al ordenamiento constitucional, estatutario y legal.

Por su conocimiento en materias jurídicas vinculadas al derecho de las tecnologías de la información, asegurará que las iniciativas del Comité de Seguridad de los Sistemas de Información y de la Secretaría General competente son acordes a la legislación vigente aplicable. Por otra banda, la Asesoría Jurídica de cada Consellería le prestará un servicio de apoyo en las materias anteriormente mencionadas.

## 6.- Junta Consultiva Contratación Administrativa

La Junta Consultiva de Contratación Administrativa de la Comunidad Autónoma de Galicia, esta adscrita a la Consejería responsable de los asuntos tributarios como un órgano consultivo específica en materia de contratación administrativa.

Dentro de las funciones y competencias de la Junta , se encuentran:

∴. Elaborar y proponer las normas, instrucciones y medidas que considere precisas para la mejora y eficacia de la contratación de la Administración autonómica, sus organismos y sociedades, fundaciones del sector público y demás entidades de derecho público de ella dependientes.

. Realizar estudios e investigaciónes sobre contratación administrativa, trasladando a los órganos de contratación las recomendaciones que se deriven a la sazón

La Xunta de Galicia recurre a proveedores para la prestación de servicios como pueden ser desarrollo de nuevas aplicaciones, infraestructuras, externalización de una actividad.

Es importante para garantizar la seguridad de la información gestionar la actuación de los proveedores. Por eso es necesario identificar los requisitos de seguridad vinculados a la prestación del servicio e incluirlos en los contratos.

Por tanto, la Junta Consultiva de Contratación Administrativa trasladará a los órganos de contratación todas las recomendaciones que considere oportunas en orden a que se incluyan cláusulas sobre disponibilidad, confidencialidad, integridad y autenticidad de la información manejada por los sistemas de información.

Con la finalidad de asegurar este aspecto, ejercerá una función de análisis y propuesta de inclusión en las frunces de contratación de cláusulas medio ambientales, sociales, de comercio justo y de protección de datos que los adjudicatarios deberán cumplir en función del servicio contratado.

#### 7. Equipo de auditores e analistas de gestión do rendimientto y calidad

El equipo de auditores y analistas de gestión del rendimiento y calidad forma de la Dirección General de Evaluación y Reforma Administrativa de la Consellería de Presidencia, Administraciones Públicas y Justicia.

Esta Dirección tiene entre otras las siguientes competencias:

. En materia de evaluación del rendimiento y gestión de la calidad:

1.-El desarrollo y la gestión de las medidas para la implantación de sistemas de mejora de la calidad tendiendo a promover la mejora continua de los servicios de la Administración autonómica, tanto de los que se prestan directamente al ciudadano como los servicios internos.

. En materia de racionalización y simplificación de procedimientos administrativos:

1.-Coordinar la aplicación de la normativa europea y estatal sobre simplificación y mejora de la gestión administrativa.

. En materia de información administrativa y atención al ciudadano:

1.-Evaluar periódicamente la calidad del sistema de información administrativa, proponiendo las medidas de mejoras convenientes con el fin de facilitar a los ciudadanos y usuarios los servicios que solicitan.

2.- Tramitar, sin perjuicio de las competencias que corresponden a las secretarías generales y en colaboración con estas, las quejas y las propuestas que formulen los ciudadanos y usuarios sobre el funcionamiento de los servicios prestados por la Administración autonómica de acuerdo con lo establecido en los artículos 25 y siguientes del Decreto 164/2005, del 16 de junio.

Entre los principios de protección de datos se encuentran el principio de información y el principio de consentimiento. El principio de información, que se regula en el artículo 5 de la LOPD, establece que el interesado debe estar informado el momento de la recogida de datos. En cuanto al principio de consentimiento exige que se obtenga el consentimiento del afectado para el tratamiento de los datos, salvo excepciones recogidas en la LOPD. Estos derechos fundamentales deben garantizarse no sólo para cumplir con la legislación vigente sino también para garantizar la calidad del servicio prestado al ciudadano.

Los procedimientos administrativos son unos de los puntos en los que es necesario prestar atención teniendo en cuenta que recogen, a menudo, datos de carácter personal.

Cuando la recogida de datos se realiza mediante formulario bien sea en formato papel o en formato electrónico se introducirá una cláusula informativa en el formulario cubriendo las exigencias del artículo 5 de la Ley Orgánica protección datos. Con respecto al consentimiento se recogerá mediante la firma del formulario papel y la aprobación del formulario electrónico.

El equipo revisa los procedimientos administrativos antes de su publicación en el DOG.

## 17.2 DECRETO 230/2008 DE 18 DE SEPTIEMBRE POR EL QUE SE ESTABLECEN BUENAS PRACTICAS

### 17.2.1 Objeto y ámbito de aplicación.

Este Decreto tiene por objeto regular las normas de utilización de los sistemas de información y de comunicaciones, fijos y móviles, de los que dispone la Administración de la Comunidad Autónoma de Galicia, estableciendo los derechos y los deberes de las personas usuarias de estos sistemas en lo relativo a su seguridad y buen uso.

La finalidad de la presente norma es conseguir el mejor aprovechamiento de las tecnologías de la información y las comunicaciones en la actividad administrativa, así como garantizar la protección de la información de las personas y de las empresas en sus relaciones con la Administración de la Comunidad Autónoma de Galicia



Será de aplicación a todas las personas que presten servicios para la Administración de la Comunidad Autónoma de Galicia y utilicen para el desempeño de sus funciones los sistemas de información o las redes de comunicaciones propiedad de la Administración autonómica.

El contenido de este Decreto será de aplicación en la utilización del equipamiento informático y de comunicaciones, fijo y móvil, incluyendo cualquier dispositivo puesto a disposición de las personas que prestan servicios para la Administración autonómica

#### 17.2.2 Órganos responsables y de coordinación.

##### 1.- Órganos responsables

Las secretarías generales designarán, dentro de cada departamento de la Xunta de Galicia, al órgano que será responsable de los sistemas de su propiedad y de establecer los medios tecnológicos que necesitan las personas a su servicio, así como de velar por el correcto funcionamiento de las infraestructuras y del equipamiento informático y de comunicaciones de que dispongan. En aquellos casos en que dichas atribuciones ya estén asignadas reglamentariamente a un órgano, no será precisa esta designación.

Estos órganos, con el apoyo del personal de soporte técnico, son los competentes para velar por el cumplimiento de las normas contenidas en este Decreto. Le corresponderá a ellos asegurarse de que los equipos se utilizan adecuadamente y atendiendo a la finalidad a la que están destinados.

Para el mejor cumplimiento de estas atribuciones sobre los sistemas, cada departamento de la Xunta de Galicia deberá designar a una persona como responsable de seguridad. Las personas designadas deberán comunicar,

dentro de su ámbito, las normas, procedimientos y políticas de seguridad para su conocimiento por el personal, así como impulsar su implantación.

## 2.- Órganos de coordinación.

Son órganos de coordinación:

a) La Comisión de Informática de la Xunta de Galicia, regulada por el Decreto 290/1992, de 8 de octubre .

b) La Dirección General de Calidad y Evaluación de las Políticas Públicas de la Consellería de Presidencia, Administraciones Públicas y Justicia.

c) El Comité de Seguridad de los Sistemas de Información de la Xunta de Galicia. Es un órgano colegiado, adscrito a la Consellería de Presidencia, Administraciones Públicas y Justicia, formado por las personas responsables de seguridad de los distintos departamentos de la Xunta de Galicia, que tiene como objetivo definir la política de seguridad corporativa. Este comité estará coordinado y asesorado por la dirección general competente a través del Centro de Seguridad Informática. Su régimen básico de funcionamiento se regulará por orden de la Consellería de Presidencia, Administraciones Públicas y Justicia.

### 17.2.3 Acceso a información, redes de comunicaciones e Internet.

#### 1.- Acceso a informacion

Las personas usuarias tendrán autorizado el acceso únicamente a aquella información y recursos que precisen para el desarrollo de sus funciones. El acceso a la información contenida en los sistemas de la Administración de la Comunidad Autónoma de Galicia estará restringido a aquellas personas poseedoras de la correspondiente autorización, que será personal e

intransferible y compuesta al menos de un identificador y de una contraseña.

Los órganos responsables de los sistemas establecerán los mecanismos adecuados para evitar que las personas puedan acceder o modificar datos sin autorización. Exclusivamente el personal de soporte técnico, conforme a los criterios establecidos por el responsable de cada uno de los sistemas de información, podrá conceder, alterar o anular la autorización de acceso a los datos y recursos.

No se podrán obtener derechos de acceso a la información distintos a los autorizados, ni se utilizará el identificador de otra persona, aunque se disponga de permiso de ésta, salvo indicación expresa y puntual del órgano responsable de dicha información o recurso. Con este fin, las unidades de personal de los distintos departamentos de la Xunta de Galicia comunicarán al servicio de informática todos los cambios que se produzcan en los puestos de trabajo.

Las personas al servicio de la Administración de la Comunidad Autónoma de Galicia deberán velar por la seguridad de los datos a los que tengan acceso por las tareas de su puesto de trabajo, especialmente los confidenciales o de carácter personal.

Por motivos de seguridad, la Administración de la Comunidad Autónoma de Galicia podrá monitorizar los accesos a la información contenida en sus sistemas, cumpliendo los requisitos que al efecto establezca la normativa vigente.

## 2 Redes de comunicaciones.

. La conexión a la red corporativa de la Xunta de Galicia será facilitada por la Dirección General de Calidad y Evaluación de las Políticas Públicas en

uso de las competencias atribuidas en el decreto de estructura orgánica de la Consellería de Presidencia, Administraciones Públicas y Justicia.

No se podrá conectar a esta red de comunicaciones ningún dispositivo por medios distintos a los definidos y autorizados por el Centro de Gestión de Red de dicha dirección general.

En el caso de aquellas redes de comunicaciones de la Administración de la Comunidad Autónoma de Galicia ya gestionadas por otras consellerías, la conexión a las mismas será facilitada por el órgano responsable de cada una de ellas.

### 3 Internet

La Administración de la Comunidad Autónoma de Galicia proveerá de conexión a Internet a las personas a su servicio con una finalidad exclusivamente profesional.

El equipo que tenga acceso a Internet, a través de las redes de comunicación gestionadas por la Administración de la Comunidad Autónoma de Galicia, deberá disponer de «software» de protección frente a virus y demás códigos maliciosos.

Los datos de conexión y tráfico serán monitorizados y se guardará un registro durante el tiempo que establece la normativa vigente en cada supuesto. En ningún caso esta retención de datos afectará al secreto de las comunicaciones.

Las conexiones a sitios Web que contengan material ofensivo o software malicioso serán bloqueadas, salvo excepciones debidamente autorizadas

#### 17.2.4 Servicio de mensajería corporativo

La Administración de la Comunidad Autónoma de Galicia proveerá de servicio de mensajería a las personas a su servicio con una finalidad exclusivamente profesional.

Por razones de seguridad y rendimiento, los órganos responsables del servicio podrán monitorizar el servicio de mensajería corporativa. Esta monitorización no será nunca selectiva o discriminatoria sino que será realizada de forma sistemática o aleatoria y sin vulneración de la intimidad personal ni del secreto de las comunicaciones.

Aquellas cuentas en las que se detecte un uso inadecuado, que se definirá en el documento de política de seguridad corporativa, podrán ser bloqueadas o suspendidas temporalmente. En ningún caso, se podrá utilizar el servicio de mensajería para:

- a) La difusión de mensajes ofensivos o discriminatorios.
- b) El uso de la cuenta de correo corporativo para expresar opiniones personales en foros temáticos fuera del ámbito de las administraciones.
- c) La difusión masiva no autorizada; suscripción indiscriminada a listas de correo o cualquier ataque con el objeto de impedir o dificultar el servicio de correo.

#### 17.2.5 Deber de personas usuarias

Las personas que prestan servicios a la Administración de la Comunidad Autónoma de Galicia, además de cumplir con las medidas indicadas en este Decreto relativas al equipamiento informático y de comunicaciones, a las aplicaciones informáticas, a la información y al uso de los servicios corporativos, son responsables del buen uso de los medios electrónicos,

informáticos, telemáticos y de comunicaciones, fijos y móviles, puestos a su disposición para las actividades propias de las funciones que desarrollan.

No se podrá acceder a los recursos informáticos y telemáticos para desarrollar actividades que persigan o tengan como consecuencia:

- a) La degradación de los servicios.
- b) La destrucción o modificación no autorizada de la información de manera premeditada.
- c) La violación de la intimidad, del secreto de las comunicaciones y del derecho a la protección de datos personales.
- d) El deterioro intencionado del trabajo de otras personas.
- e) El uso de los sistemas de información para fines ajenos a los de la Administración.
- f) Incurrir en actividades ilícitas de cualquier tipo.
- g) Dañar intencionadamente los recursos informáticos de la Administración de la Comunidad Autónoma de Galicia o de otras instituciones.
- h) Instalar o utilizar «software» que no disponga de la licencia correspondiente.

3. Para garantizar unos mínimos de seguridad en el equipamiento asignado, se deberá:

- a) Utilizar y guardar en secreto la contraseña que protege la cuenta de acceso, responsabilidad directa de la persona usuaria. Ésta debe cerrar su cuenta al final de cada sesión o cuando deja desatendido el equipo, con el fin de que no pueda ser usado por terceras personas.
- b) Revisar de forma periódica sus ordenadores, eliminando cualquier virus, programa o fichero que pueda causar daños a otros equipos de la red u otras actuaciones que contravengan la legislación vigente.
- c) En el caso de que su equipo contenga información importante que no esté guardada en un servidor, realizar copias de seguridad periódicas para garantizar su disponibilidad.

Las personas usuarias, en el ejercicio de sus funciones, deberán colaborar con el órgano competente en materia de seguridad de los sistemas de información y seguir sus recomendaciones y, en particular, las del Centro de Seguridad Informática, en aplicación de la política de seguridad corporativa definida por el Comité de Seguridad de los Sistemas de Información de la Xunta de Galicia.

También estarán obligadas al cumplimiento de aquellas otras medidas adicionales que especifiquen los órganos responsables de los sistemas.

#### 17.2.6 Inspección

La Administración de la Comunidad Autónoma de Galicia, mediante los medios tecnológicos y personales que estime oportunos, revisará periódica y puntualmente, por razones de seguridad y de calidad del servicio, el estado de los equipos, dispositivos y redes de comunicaciones de su responsabilidad, así como su correcta utilización, con el objeto de verificar su correcto funcionamiento, eficiencia y el cumplimiento de las medidas y protocolos de seguridad establecidos en la legislación vigente.

La dirección general competente en materia de seguridad corporativa velará por el cumplimiento de la presente normativa e informará al Comité de Seguridad de los Sistemas de Información de la Xunta de Galicia sobre los incumplimientos o deficiencias de seguridad observados, con el objeto de que tomen las medidas oportunas.

Los servicios para los que se detecte un uso inadecuado o que no cumplan los requisitos de seguridad, que se definirán en el documento de política de seguridad corporativa, podrán ser bloqueados o suspendidos temporalmente para aquellas cuentas en las que se detecte un daño para los de los sistemas de información y de comunicaciones. El servicio se restablecerá cuando la causa de su degradación desaparezca

#### 17.2.7 Responsabilidad de personas usuarias que tengan la condición de empleados públicos

La Administración de la Comunidad Autónoma de Galicia exigirá de los empleados públicos la responsabilidad en la que incurriesen por dolo, culpa o descuido graves de las que se deriven daños y perjuicios en sus bienes o derechos o indemnizaciones para particulares, previa instrucción del procedimiento correspondiente en los términos previstos en la normativa de aplicación.

El incumplimiento de los deberes y obligaciones impuestos por el presente Decreto, que sean constitutivos de infracción disciplinaria, según la tipificación efectuada en la normativa aplicable, dará lugar a la incoación del correspondiente procedimiento disciplinario que se tramitará conforme a lo establecido en la normativa aplicable a los empleados públicos en función de la naturaleza jurídica de su vínculo con la Administración. No obstante lo anterior, la incoación de los expedientes y la imposición de las



sanciones requerirá informe previo de la Dirección General de Calidad y Evaluación de las Políticas Públicas.

Autor:

Alfonso García Magariños

Director Asesoría Jurídica Municipal Concello de A Coruña

**18. IMPLANTACIÓN DE LA  
ADMINISTRACIÓN ELECTRÓNICA.  
SEDE ELECTRÓNICA Y SERVICIOS  
DE SEDE. REGISTRO  
ELECTRÓNICO. EXPEDIENTE  
ELECTRÓNICO. ARCHIVO  
ELECTRÓNICO DE DOCUMENTOS.  
DIGITALIZACIÓN, COMPULSA  
ELECTRÓNICA. FACTURA Y  
LICITACIÓN ELECTRÓNICAS.  
INICIATIVAS DEL GOBIERNO  
GALLEGO: OSIMGA, REDE CEMIT,  
PLAN DE BANDA ANCHA 2013,  
AGENDA DIGITAL 2014.GAL,  
ESTRATEGIA DE IMPULSO AL  
SECTOR TIC.**

**TEMA 18. IMPLANTACIÓN DE LA ADMINISTRACIÓN ELECTRÓNICA. SEDE ELECTRÓNICA Y SERVICIOS DE SEDE. REGISTRO ELECTRÓNICO. EXPEDIENTE ELECTRÓNICO. ARCHIVO ELECTRÓNICO DE DOCUMENTOS. DIGITALIZACIÓN, COMPULSA ELECTRÓNICA. FACTURA Y LICITACIÓN ELECTRÓNICAS. INICIATIVAS DO GOBERNO GALEGO. OSIMGA. REDE CeMIT, PLAN DE BANDA LARGA 2013. AXENDA DIXITAL 2014.gal. ESTRATEXIA DE IMPULSO DO SECTOR TIC**

**18.1. IMPLANTACIÓN DE LA ADMINISTRACIÓN ELECTRÓNICA**

**18.2. SEDE ELECTRÓNICA Y SERVICIOS DE LA SEDE**

**18.3. REGISTRO ELECTRÓNICO**

**18.4. EXPEDIENTE ELECTRÓNICO**

**18.5. ARCHIVO ELECTRÓNICO DE DOCUMENTOS**

**18.6. DIGITALIZACIÓN, COMPULSA ELECTRÓNICA.**

**18.7. FACTURA Y LICITACIÓN ELECTRÓNICAS**

**18.8. OSIMGA**

**18.9. REDE CeMIT**

**18.10. PLAN DE BANDA LARGA 2013.**

**18.11. AXENDA DIXITAL 2014.gal**

**18.12. ESTRATEXIA DE IMPULSO DO SECTOR TIC**

**18.13. REFERENCIAS**

**18.1. IMPLANTACIÓN DE LA ADMINISTRACIÓN ELECTRÓNICA**

Según establece la Comisión Europea, la Administración electrónica se define como el uso de las Tecnologías de la Información y las Comunicaciones en las Administraciones Públicas, combinada con cambios organizativos y nuevas aptitudes, con el fin de mejorar los servicios

públicos y los procesos democráticos y reforzar el apoyo a las políticas públicas.

La e-Administración o Administración electrónica hace referencia a la incorporación de las tecnologías de la información y las comunicaciones en dos vertientes:

- Desde un punto de vista organizativo, transformando las oficinas tradicionales, convirtiendo los procesos en papel en procesos electrónicos.
- Desde una perspectiva de las relaciones externas, habilitando la vía electrónica como un nuevo medio para la relación con el ciudadano, empresas y otras instituciones.

La Administración electrónica tiene su mayor impulso en la última década, motivado en parte por un marco legal que ha permitido llevar las garantías jurídicas que existen en el mundo real al mundo virtual y en otra parte por la evolución de las tecnologías relacionadas y el desarrollo de proyectos emblemáticos, como el DNI electrónico.

Pero sobre todo es necesario hacer mención a la Ley 11/2007, de 22 de junio, de acceso electrónico de los ciudadanos a los Servicios Públicos, muchas veces denominada simplemente "Ley de Administración Electrónica" o mencionada por siglas LAECSPC<sup>1</sup>, que consagra el concepto de Administración electrónica en el marco jurídico español y la eleva a la categoría de derecho de los ciudadanos. Es por tanto la norma legal de referencia en esta materia y establece un marco homogéneo para las tres administraciones en la materia.

Su principal objetivo es reconocer y garantizar el derecho del ciudadano a relacionarse por medios electrónicos con las Administraciones Públicas. Por otra parte se pretende impulsar el uso de los servicios electrónicos en la

---

<sup>1</sup> Dadas las múltiples referencias que se hacen en este texto a dicha Ley, optaremos habitualmente por utilizar esta expresión o bien simplemente Ley 11/2007.

Administración creando las condiciones necesarias, y de manera indirecta ejercer con ello un efecto arrastre sobre la sociedad de la información en general.

La LAECSPC además ha supuesto el punto de partida para un desarrollo normativo que permite avanzar en aspectos concretos, y que también se mencionarán en estos capítulos y aparecen en el apartado de referencias.

En particular, y con respecto a los aspectos técnicos, destacan el Esquema Nacional de Seguridad y el Esquema Nacional de Interoperabilidad, y con respecto a este último, las recientes Normas Técnicas de Interoperabilidad.

Además de reconocer el derecho de los ciudadanos a relacionarse con las Administraciones Públicas por medios electrónicos, regula los aspectos básicos de la utilización de las tecnologías de la información en la actividad administrativa, en las relaciones entre las Administraciones Públicas, así como en las relaciones de los ciudadanos con las mismas.

La LAECSPC obliga a las Administraciones a asegurar la disponibilidad, el acceso, la integridad, la autenticidad, la confidencialidad y la conservación de los datos, informaciones y servicios que gestionen en el ejercicio de sus competencias.

También establece una serie de fines, que definen con claridad que debe perseguir todo proyecto de Administración Electrónica:

1. Facilitar el ejercicio de derechos y el cumplimiento de deberes por medios electrónicos.
2. Facilitar el acceso por medios electrónicos de los ciudadanos a la información y al procedimiento administrativo, con especial atención a la eliminación de las barreras que limiten dicho acceso.

3. Crear las condiciones de confianza en el uso de los medios electrónicos.
4. Promover la proximidad con el ciudadano y la transparencia administrativa, así como la mejora continuada en la consecución del interés general.
5. Contribuir a la mejora del funcionamiento interno de las Administraciones Públicas.
6. Simplificar los procedimientos administrativos y proporcionar oportunidades de participación y mayor transparencia, con las debidas garantías legales.
7. Contribuir al desarrollo de la sociedad de la información en el ámbito de las Administraciones Públicas y en la sociedad en general.

También establece unos principios: El respeto al derecho a la protección de datos de carácter personal, de igualdad, de accesibilidad a la información y a los servicios, de legalidad, de cooperación, de seguridad, proporcionalidad, de responsabilidad, de neutralidad tecnológica y de adaptabilidad, de simplificación administrativa y de transparencia y publicidad.

Establece una serie de derechos para los ciudadanos:

- a. A elegir, entre aquellos que en cada momento se encuentren disponibles, el canal a través del cual relacionarse por medios electrónicos con las Administraciones Públicas.
- b. A no aportar los datos y documentos que obren en poder de las Administraciones Públicas.
- c. A la igualdad en el acceso electrónico a los servicios de las Administraciones Públicas (entendida como no discriminación de personas que no tenga un acceso fácil a los medios electrónicos).
- d. A conocer por medios electrónicos el estado de tramitación de los procedimientos en los que sean interesados.

- e. A obtener copias electrónicas de los documentos electrónicos que formen parte de procedimientos en los que tengan la condición de interesado.
- f. A la conservación en formato electrónico por las Administraciones Públicas de los documentos electrónicos que formen parte de un expediente.
- g. A obtener los medios de identificación electrónica necesarios, pudiendo las personas físicas utilizar en todo caso los sistemas de firma electrónica del Documento Nacional de Identidad para cualquier trámite electrónico con cualquier Administración Pública.
- h. A la utilización de otros sistemas de firma electrónica admitidos en el ámbito de las Administraciones Públicas.
- i. A la garantía de la seguridad y confidencialidad de los datos que figuren en los ficheros, sistemas y aplicaciones de las Administraciones Públicas.
- j. A la calidad de los servicios públicos prestados por medios electrónicos.
- k. A elegir las aplicaciones o sistemas para relacionarse con las Administraciones Públicas siempre y cuando utilicen estándares abiertos o, en su caso, aquellos otros que sean de uso generalizado por los ciudadanos.

En particular, en los procedimientos relativos al acceso a una actividad de servicios y su ejercicio, los ciudadanos tienen derecho a la realización de la tramitación a través de una ventanilla única, por vía electrónica y a distancia, y a la obtención de la siguiente información a través de medios electrónicos, que deberá ser clara e inequívoca:

- a. Los requisitos aplicables a los prestadores establecidos en territorio español, en especial los relativos a los procedimientos y trámites necesarios para acceder a las actividades de servicio y para su ejercicio.

- b. Los datos de las autoridades competentes en las materias relacionadas con las actividades de servicios, así como los datos de las asociaciones y organizaciones distintas de las autoridades competentes a las que los prestadores o destinatarios puedan dirigirse para obtener asistencia o ayuda.
- c. Los medios y condiciones de acceso a los registros y bases de datos públicos relativos a prestadores de actividades de servicios.
- d. Las vías de reclamación y recurso en caso de litigio entre las autoridades competentes y el prestador o el destinatario, o entre un prestador y un destinatario, o entre prestadores.

Con el fin de que los ciudadanos puedan ejercer su derecho a no aportar datos que ya obren en poder de la Administración Pública, cada Administración deberá facilitar el acceso de las restantes Administraciones Públicas a los datos relativos a los interesados que obren en su poder y se encuentren en soporte electrónico.

## **18.2. SEDE ELECTRÓNICA Y SERVICIOS DE LA SEDE**

La LAECSPC define la sede electrónica como aquella dirección electrónica disponible para los ciudadanos a través de redes de telecomunicaciones cuya titularidad, gestión y administración corresponde a una Administración Pública, órgano o entidad administrativa en el ejercicio de sus competencias.

Cada Administración Pública determinará las condiciones e instrumentos de creación de las sedes electrónicas, con sujeción a los principios de publicidad oficial, responsabilidad, calidad, seguridad, disponibilidad, accesibilidad, neutralidad e interoperabilidad. En todo caso deberá garantizarse la identificación del titular de la sede, así como los medios disponibles para la formulación de sugerencias y quejas.



Las sedes electrónicas dispondrán de sistemas que permitan el establecimiento de comunicaciones seguras siempre que sean necesarias. Este apartado cobra especial importancia cuando se ofrecen servicios de tramitación.

La publicación en las sedes electrónicas de informaciones, servicios y transacciones respetará los principios de accesibilidad y usabilidad de acuerdo con las normas establecidas al respecto, estándares abiertos y, en su caso, aquellos otros que sean de uso generalizado por los ciudadanos.

La sede electrónica se diferencia de cualquier sede institucional tradicional en entorno telemático por la responsabilidad del titular, el hecho de que la publicación de diarios o boletines oficiales reviste los mismos efectos que la publicación impresa y que puede actuar como sustituta o complemento al tablón de anuncios o edictos. Además, han de tenerse en cuenta las condiciones que han de cumplir las sedes electrónicas de la Administración General del Estado en el marco del Real Decreto 1671/2009. Para el resto de las Administraciones Públicas, si bien están fuera su ámbito, es indudable su validez como elemento de referencia.

En todo caso, a través de su articulado, la LAECSP establece los siguientes requisitos para las sedes electrónicas:

- Debe permitir el acceso de los ciudadanos para la realización de cualquier tipo de trámite o interacción con la Administración.
- Debe permitir a los ciudadanos realizar consultas sobre el estado de tramitación de expedientes en los que tengan la condición de interesado.
- Tanto la sede como los elementos y contenidos de la misma deben basarse en aplicaciones y sistemas que utilicen estándares abiertos o sean de uso generalizado por los ciudadanos.

- Debe contener la información sobre los pasos a seguir para cada uno de los trámites y procedimientos de las Administraciones Públicas.
- Debe contener la información sobre las autoridades competentes para cada actividad de los servicios ofrecidos por las Administraciones Públicas.
- La AGE deberá disponer de una sede electrónica que sirva como Punto de acceso general único a los servicios que presta la AGE y sus Organismos.
- El Punto de acceso general creado por la AGE deberá estar integrado con el resto de sedes de la AGE y Organismos Públicos para la prestación de los distintos servicios.
- Debe garantizar la identificación de su titular.
- Debe permitir establecer las conexiones seguras cuando sean necesarias.
- Debe cumplir los principios de accesibilidad y usabilidad de acuerdo con las normas establecidas al respecto (Según el BOE n. 141 de 13/6/2003 en la disposición 7, artículo 2: se deben cumplir los requisitos AA).
- Permitirá la publicación de actos y comunicaciones que, por disposición legal o reglamentaria deban publicarse en tablón de anuncios o edictos.
- Debe contener la lista de sistemas de firma electrónica avanzada admitidos.
- Debe contener la lista de sellos electrónicos utilizados por cada Administración.
- Debe contener las disposiciones de creación de registros electrónicos.
- La sede permitirá la publicación electrónica del boletín oficial de la Administración, órgano o Entidad competente.
- Debe contener los distintos tipos de escritos, comunicaciones, solicitudes, etc. que pueden presentarse.

- Deberá publicar los medios electrónicos disponibles para que el ciudadano se relacione con las Administraciones Públicas.
- Deberá mostrar de manera visible la fecha y hora garantizando su integridad.
- Deberá publicar una lista con los días considerados inhábiles.
- En aquéllas administraciones que tengan lenguas cooficiales, se debe garantizar el acceso en ambas lenguas.

Desde el punto de vista técnico, la sede electrónica no presenta características tecnológicas distintas a la de cualquier sitio web tradicional, si bien es necesario establecer las medidas de seguridad que permitan garantizar la responsabilidad establecida por la LAECSP.

En particular, es importante la identificación segura de la sede electrónica. En este sentido, la Ley establece la posibilidad de creación de certificados de sede electrónica con este propósito.

La Fábrica Nacional de la Moneda y Timbre – Real Casa de la Moneda (FNMT-RCM), que ya ofrecía certificados de identificación segura, dispone ya de la versión actualizada para el cumplimiento de la LAE SCP (son los llamados genéricamente certificados APE).

### 18.3. **REGISTRO ELECTRÓNICO**

Las tareas fundamentales del registro electrónico son tomar una referencia de tiempo, anotar el asiento de la entrada/salida, guardar los datos de la presentación de información, y devolver un acuse de recibo con el número de registro y momento de la presentación. El registro podrá asimismo incluir funcionalidades adicionales, como por ejemplo, el sellado de tiempo para obtener la referencia temporal, el cotejo/compulsa electrónica de documentos presentados físicamente o el funcionamiento como registro único para toda la Administración.

Con respecto a los registros electrónicos, la LAECSP establece que las Administraciones Públicas crearán registros electrónicos para la recepción y remisión de solicitudes, escritos y comunicaciones. Los registros electrónicos podrán admitir:

- a. Documentos electrónicos normalizados correspondientes a los servicios, procedimientos y trámites que se especifiquen conforme a lo dispuesto en la norma de creación del registro, cumplimentados de acuerdo con formatos preestablecidos.
- b. Cualquier solicitud, escrito o comunicación distinta de los mencionados en el apartado anterior dirigido a cualquier órgano o entidad del ámbito de la administración titular del registro.

En cada Administración Pública existirá, al menos, un sistema de registros electrónicos suficiente para recibir todo tipo de solicitudes, escritos y comunicaciones dirigidos a dicha Administración Pública.

Los registros electrónicos emitirán automáticamente un recibo consistente en una copia autenticada del escrito, solicitud o comunicación de que se trate, incluyendo la fecha y hora de presentación y el número de entrada de registro.

Podrán aportarse documentos que acompañen a la correspondiente solicitud, escrito o comunicación, siempre que cumplan los estándares de formato y requisitos de seguridad que se determinen en los Esquemas Nacionales de Interoperabilidad y de Seguridad.

Los registros electrónicos se regirán a efectos de cómputo de los plazos imputables tanto a los interesados como a las Administraciones Públicas por la fecha y hora oficial de la sede electrónica de acceso, que deberá contar con las medidas de seguridad necesarias para garantizar su

integridad y figurar visible. Permitirán la presentación de solicitudes, escritos y comunicaciones todos los días del año durante las veinticuatro horas.

A los efectos del cómputo de plazo fijado en días hábiles o naturales, y en lo que se refiere a cumplimiento de plazos por los interesados, la presentación en un día inhábil se entenderá realizada en la primera hora del primer día hábil siguiente, salvo que una norma permita expresamente la recepción en día inhábil.

Desde el punto de vista técnico, el registro electrónico debe permitir la presentación de documentación en formato electrónico y la generación del correspondiente asiento en el registro de entrada y salida de la Administración. Ello implica probablemente la necesidad de integración con un sistema de gestión de registro de entrada y salida general, que permita la introducción de asientos manual desde los distintos departamentos.

Además, para poder realizar esta función con las suficientes garantías, debe permitir lo siguiente:

- a) Presentar la documentación en formato electrónico y firmada por el ciudadano mediante el correspondiente certificado electrónico que garantice su autenticidad, integridad y no repudio.
- b) Registrar de modo fehaciente el proceso de registro de dicha documentación. A tal efecto se puede utilizar algún tipo de resguardo firmado mediante el uso de un certificado de la Administración Pública (previsiblemente un certificado de sello electrónico) que incluya sellado de tiempo para garantizar la fecha y hora de entrega.
- c) Entregar al ciudadano el resguardo de dicho registro en formato electrónico, convenientemente firmado por la Administración, en el que constará la fecha y hora de entrada y

los documentos entregados, y pudiendo incorporar un checksum o sistema de equivalente que permita verificar que los contenidos presentados son los que constan en dicho resguardo.

#### **18.4. EXPEDIENTE ELECTRÓNICO**

La LAECSP define expediente electrónico como el conjunto de documentos electrónicos correspondientes a un procedimiento administrativo, cualquiera que sea el tipo de información que contengan.

El foliado de los expedientes electrónicos se llevará a cabo mediante un índice electrónico, firmado por la Administración, órgano o entidad actuante, según proceda. Este índice garantizará la integridad del expediente electrónico y permitirá su recuperación siempre que sea preciso, siendo admisible que un mismo documento forme parte de distintos expedientes electrónicos.

La remisión de expedientes podrá ser sustituida a todos los efectos legales por la puesta a disposición del expediente electrónico, teniendo el interesado derecho a obtener copia del mismo.

#### **18.5. ARCHIVO ELECTRÓNICO DE DOCUMENTOS**

Las Administraciones Públicas podrán emitir válidamente por medios electrónicos los documentos administrativos a los que se refiere el artículo 46 de la Ley 30/1992, de Régimen Jurídico de las Administraciones Públicas y del Procedimiento Administrativo Común, siempre que incorporen una o varias firmas electrónicas conforme a lo establecido en la Sección III del Capítulo II de la LAECSP.

Los documentos administrativos incluirán referencia temporal, que se garantizará a través de medios electrónicos cuando la naturaleza del documento así lo requiera.

Podrán almacenarse por medios electrónicos todos los documentos utilizados en las actuaciones administrativas. Los documentos electrónicos que contengan actos administrativos que afecten a derechos o intereses de los particulares deberán conservarse en soportes de esta naturaleza, ya sea en el mismo formato a partir del que se originó el documento o en otro cualquiera que asegure la identidad e integridad de la información necesaria para reproducirlo. Se asegurará en todo caso la posibilidad de trasladar los datos a otros formatos y soportes que garanticen el acceso desde diferentes aplicaciones.

Los medios o soportes en que se almacenen documentos, deberán contar con medidas de seguridad que garanticen la integridad, autenticidad, confidencialidad, calidad, protección y conservación de los documentos almacenados. En particular, asegurarán la identificación de los usuarios y el control de accesos, así como el cumplimiento de las garantías previstas en la legislación de protección de datos.

Desde el punto de vista técnico, el almacenamiento y tratamiento de documentos en formato electrónico implica la utilización de una firma avanzada que permita garantizar su integridad y el no repudio. En caso de tratarse de documentos en otro formato, como el papel, puede ser necesaria su conversión previa a algún formato electrónico mediante técnicas como la del escaneado.

Además, el sistema informático de soporte debe garantizar la seguridad de los documentos tanto desde el punto de vista de la disponibilidad como del control de acceso. Por último, implica también la implantación sistema de custodia de documentos electrónicos con mecanismos que permitan

garantizar su validez a lo largo del tiempo, mediante el uso, por ejemplo, de firmas longevas o de la actualización periódica de formatos para evitar su obsolescencia.

#### **18.6. DIGITALIZACIÓN, COMPULSA ELECTRÓNICA.**

En primer lugar, debemos distinguir entre cotejo o copia compulsada y copia auténtica:

- El cotejo y la compulsa de documentos es la técnica consistente en la comprobación de que una copia coincide con su original, que lleva a poder afirmar que la misma es exacta. La copia cotejada o compulsada en ningún caso acredita la autenticidad del documento original.
- La copia auténtica de un documento acredita la autenticidad de los datos contenidos en la misma, no sólo desde la perspectiva de su identidad con el documento original, si no por sus efectos certificativos, en cuanto que garantiza, igualmente, la autenticidad de los datos contenidos en este último.

Por consiguiente, la copia auténtica goza de la misma validez y eficacia que el documento original, no limitando sus efectos a un procedimiento administrativo concreto.

Las copias realizadas por medios electrónicos de documentos electrónicos emitidos por el propio interesado o por las Administraciones Públicas, manteniéndose o no el formato original, tendrán inmediatamente la consideración de copias auténticas con la eficacia prevista en el artículo 46 de la Ley 30/1992, de Régimen Jurídico de las Administraciones Públicas y del Procedimiento Administrativo Común, siempre que el documento electrónico original se encuentre en poder de la Administración, y que la



información de firma electrónica y, en su caso, de sellado de tiempo permitan comprobar la coincidencia con dicho documento.

Las copias realizadas por las Administraciones Públicas, utilizando medios electrónicos, de documentos emitidos originalmente por las Administraciones Públicas en soporte papel tendrán la consideración de copias auténticas siempre que se cumplan los requerimientos y actuaciones previstas en el artículo 46 de la Ley 30/1992, de Régimen Jurídico de las Administraciones Públicas y del Procedimiento Administrativo Común.

Las Administraciones Públicas podrán obtener imágenes electrónicas de los documentos privados aportados por los ciudadanos, con su misma validez y eficacia, a través de procesos de digitalización que garanticen su autenticidad, integridad y la conservación del documento imagen, de lo que se dejará constancia. Esta obtención podrá hacerse de forma automatizada, mediante el correspondiente sello electrónico.

Las copias realizadas en soporte papel de documentos públicos administrativos emitidos por medios electrónicos y firmados electrónicamente tendrán la consideración de copias auténticas siempre que incluyan la impresión de un código generado electrónicamente u otros sistemas de verificación que permitan contrastar su autenticidad mediante el acceso a los archivos electrónicos de la Administración Pública, órgano o entidad emisora.

Desde el punto de vista técnico, la compulsa y creación de copias auténticas implica la firma de un documento electrónico mediante el uso de un certificado electrónico. En caso de que dicho documento se encuentre en formato papel, debe ser previamente digitalizado mediante un procedimiento establecido y seguro.

El proceso tecnológico de conversión a formato electrónico se denomina “digitalización certificada”, entendiendo como tal la definición dada compatible con el término “digitalización” que se puede encontrar en el Anexo del Esquema Nacional de Interoperabilidad (RD 4/2010).

Al ser necesario que la copia sea fiel e íntegra, el proceso de digitalización debe cumplir una serie de características:

- a) Debe ser completamente automático y realizarse de forma atómica, sin intervención humana, obteniendo como entrada el documento original e devolviendo como resultado la copia electrónica.
- b) El proceso tecnológico debe ser diseñado de tal forma que non produzca alteración con respecto al documento original. En este aspecto hay que tener en cuenta que la obtención de la copia implica la realización de distintas operaciones. Así por ejemplo, será necesario en un primer momento obtener a partir del contenido en papel una representación en formato electrónico, posiblemente mediante una operación de escaneado o similar. En esta fase será importante definir las características técnicas de los dispositivos a utilizar, y parámetros del proceso como puede ser el nivel de resolución (término definido en el Anexo del RD 4/2010) mínima. A partir de esta información, ya en formato electrónico, muy probablemente será necesario además realizar conversiones entre formatos o aplicar algoritmos de compresión con o sin pérdida de información, para os que será necesario establecer unos límites de tolerancia.

Para la firma, la LAECSP establece la posibilidad de utilizar certificados de personal adscrito a la Administración o funcionario. La Fábrica Nacional de la Moneda y Timbre – Real Casa de la Moneda (FNMT-RCM), que ya ofrecía

certificados de identificación segura, dispone ya de la versión actualizada para el cumplimiento de la LAESCP.

El certificado para el personal de la Administración Pública, es la certificación electrónica emitida por la FNMT-RCM que vincula a su titular con unos datos de verificación de firma y confirma, de forma conjunta:

- la identidad de su titular, número de identificación personal, cargo, puesto de trabajo y/o condición de autorizado.
- al órgano, organismo o entidad de la Administración Pública, bien sea ésta General, autonómica, Local o institucional, donde ejerce sus competencias, presta sus servicios, o desarrolla su actividad.

## **18.7. FACTURA Y LICITACIÓN ELECTRÓNICAS**

### **18.7.1. FACTURA ELECTRÓNICA**

La facturación electrónica es un equivalente funcional de la factura en papel y consiste en la transmisión de las facturas o documentos análogos entre emisor y receptor por medios electrónicos (ficheros informáticos) y telemáticos (de un ordenador a otro), firmados digitalmente con certificados reconocidos.

La Ley 56/2007, de Medidas de Impulso de la Sociedad de la Información, define la factura electrónica como “un documento electrónico que cumple con los requisitos legal y reglamentariamente exigibles a las facturas y que, además, garantiza la autenticidad de su origen y la integridad de su contenido, lo que permite atribuir la factura a su obligado tributario emisor”.

De esta definición extendida en todo el mercado, se transmite tres condicionantes para la realización de e-Factura:

- Se necesita un formato electrónico de factura de mayor o menor complejidad (EDIFACT, XML, PDF, html, doc, xls, gif, jpeg o txt, entre otros).
- Es necesario una transmisión telemática (tiene que partir de un ordenador, y ser recogida por otro ordenador).
- Este formato electrónico y transmisión telemática, deben garantizar su integridad y autenticidad a través de una firma electrónica reconocida. El artículo 3.3 de la Ley 59/2003 de 19 de diciembre define la firma electrónica reconocida como: “la firma electrónica avanzada basada en un certificado reconocido y generada mediante un dispositivo seguro de creación de firma”.

El certificado que se usa es el del expedidor real de la factura. Ya sea éste el obligado tributario, un tercero que actúe en su nombre o el destinatario de la factura, si se ha acordado auto-facturación.

Para cumplir con la norma y que una factura electrónica tenga la misma validez legal que una emitida en papel, el documento electrónico que la representa debe contener los campos obligatorios exigibles a toda factura, estar firmado mediante una firma electrónica avanzada basado en certificado reconocido y ser transmitido de un ordenador a otro recogiendo el consentimiento de ambas partes.

Para homogenizar estos aspectos técnicos se ha desarrollado la Orden PRE/2971/2007, sobre la expedición de facturas por medios electrónicos cuando el destinatario de las mismas sea la Administración General del Estado u organismos públicos vinculados o dependientes de aquélla y sobre la presentación ante la Administración General del Estado o sus organismos públicos vinculados o dependientes de facturas expedidas entre particulares.

En esta orden se crea el formato de factura electrónica Facturae, junto con la previsión de compatibilidad en futuras con normas como UBL (Universal Business Language). Facturae define fundamentalmente las tecnologías de firma a utilizar en las facturas y una estructura en XML que éstas deben cumplir. Se puede encontrar amplia información sobre este formato en el sitio web <http://www.facturae.es/>.

Obligaciones legales del expedidor:

1. Creación de la factura. Mediante una aplicación informática, con los contenidos obligatorios mínimos requeridos.
2. Uso de firma electrónica reconocida
3. Remisión telemática
4. Conservación de copia o matriz de la factura. Esta obligación se regula en el artículo 1 del RD 1496/2003, donde se especifica la obligación de expedir, entregar y conservar facturas.
5. Contabilización y anotación en registros de IVA
6. Conservación durante el período de prescripción
7. Garantía de accesibilidad completa. Deber de gestionar las facturas de modo que se garantice una accesibilidad completa: visualización, búsqueda selectiva, copia o descarga en línea e impresión. Esta es una obligación inherente a la conservación de las facturas por medios electrónicos que el legislador denomina acceso completo a datos, tratando de facilitar la auditoria e inspección de las facturas electrónicas. (Artículo 9 del RD 1496/2003)
8. Subcontratación a un tercero. Todas las fases anteriores pueden ser subcontratadas a un tercero, sin perder su responsabilidad. Regulado en el artículo 5.1 del RD 1496/2003 el legislador deja claro en ese mismo párrafo que, aunque se permite la subfacturación a terceros, es el obligado tributario el responsable de cumplir todas estas obligaciones.

Obligaciones legales del destinatario:

1. Recepción de la factura por medio electrónico

- Verificación de los contenidos mínimos exigibles
- Verificación segura de la firma electrónica. Regulado en el artículo 21 e inherente a las obligaciones de la conservación de las facturas electrónicas se indica que: “el destinatario se debe asegurar de la legibilidad en el formato original en el que se haya recibido, así como, en su caso, de los datos asociados y mecanismo de verificación de firma”.

A diferencia del emisor, al que se permite construir la factura desde la matriz, el destinatario debe conservar los originales firmados.

2. Contabilización y anotación en registros de IVA

3. Conservación durante el período de prescripción. Deber de gestionar las facturas de modo que se garantice una accesibilidad completa.

4. Todas las fases anteriores puede subcontratarlas a un tercero, sin perder su responsabilidad.

La obligación del uso de facturas electrónicas nace de las previsiones de la Ley 30/2007, de Contratos del Sector Público, la cual regula, entre otras muchas materias, el establecimiento de una plataforma de contratación electrónica del Estado y la utilización de medios electrónicos, informáticos o telemáticos por parte de las empresas del sector privado para la contratación con Administraciones Públicas.

#### 18.7.2. LICITACIÓN ELECTRÓNICA

La Ley 30/2007, de Contratos del Sector Público, en su disposición final novena, dedicada a la habilitación normativa en materia de uso de medios electrónicos, informáticos o telemáticos, y uso de factura electrónica, autoriza al Ministro de Economía y Hacienda para aprobar, previo dictamen

### 18.7.2. LICITACIÓN ELECTRÓNICA

La Ley 30/2007, de Contratos del Sector Público, en su disposición final novena, dedicada a la habilitación normativa en materia de uso de medios electrónicos, informáticos o telemáticos, y uso de factura electrónica, autoriza al Ministro de Economía y Hacienda para aprobar, previo dictamen del Consejo de Estado, las normas de desarrollo que puedan ser necesarias para hacer plenamente efectivo el uso de medios electrónicos, informáticos o telemáticos en los procedimientos regulados en esta Ley, las especificaciones técnicas de las comunicaciones de datos y los modelos que deban utilizarse.

En la práctica, un sistema de licitación electrónica debe permitir:

- Consultar en Internet las convocatorias de los contratos y obtener los pliegos. Para ello se crea la figura del Perfil del contratante, donde es posible consultar toda la información relativa a expedientes de contratación. El perfil del contratante debe garantizar técnicamente la fecha y hora de la publicación, así como la integridad del contenido. Esto se lleva a la práctica mediante un sistema por el cual el contenido que va a ser publicado es primero firmado electrónicamente, incluyendo la firma un sellado de tiempo.
- Presentar por medios electrónicos solicitudes de participación, ofertas y documentos. Los licitantes deben poder presentar durante el plazo previsto ofertas de modo telemático. Para ello podrán hacer uso del correspondiente registro electrónico. Tal y como se explicó en dicho apartado, los licitantes podrán obtener el correspondiente resguardo.
- Obtener información sobre el desarrollo del procedimiento mediante la consulta de un tablón de anuncios electrónico. Nuevamente a través del perfil del contratante.
- Recibir notificaciones telemáticamente. Para la emisión de notificaciones telemáticas de modo fehaciente (de no ser así, en

ocasiones se denominan simplemente comunicaciones), se dispone de servicios como el Sistema de Notificaciones Telemáticas Seguras creado por la Sociedad Estatal de Correos y Telégrafos. En este sistema, el ciudadano dispone de un buzón al que son enviadas las notificaciones. El ciudadano tiene la posibilidad de ignorar, aceptar o rechazar las notificaciones, con las mismas garantías que por el canal tradicional, y el servicio informa a la Administración de la situación en cada caso.

Un sistema de licitación electrónica debe además garantizar que, en función del procedimiento de contratación establecido, las ofertas sólo podrán ser accedidas en la fase de tramitación prevista. Permitirá por lo tanto definir las mesas de contratación, si es el caso, y establecerá los mecanismos necesarios para que las ofertas no puedan ser abiertas hasta que éstas se hayan constituido formalmente.

Técnicamente, esto se resuelve mediante la creación de sobres electrónicos, siguiendo los pasos que se detallan a continuación:

#### PREPARACIÓN DE LA LICITACIÓN

- Se identifica a los miembros de la mesa. El sistema debe tener acceso a la clave pública de cada uno de los miembros.

#### PRESENTACIÓN DE OFERTAS:

- En el momento de la presentación de las ofertas, se genera un par de claves pública y privada para cada una, y la oferta es cifrada con la clave pública, creando el sobre electrónico.
- Dicho sobre sólo puede ser abierto mediante la correspondiente clave privada. Sin embargo, la clave privada no se almacena en el sistema. En su lugar, se le aplica un algoritmo que la divide en varias partes.
- Cada parte es asignada a un miembro de la mesa y cifrada con su clave pública, de tal modo que es la única persona que puede acceder a ella con su clave privada.

#### APERTURA DE OFERTAS:



- En el día y hora de constitución de la mesa de contratación, el sistema considera abierta la mesa de contratación y los miembros pueden acceder al sistema.
- Cada uno de los miembros pueden acceder a la parte de la clave privada del sobre que le corresponde. Para ello, puesto que está cifrada con su clave pública, deben utilizar su clave privada, identificándolos fehacientemente.
- Una vez el sistema dispone de suficientes partes de la clave privada (no es necesario que participen todos los miembros de la mesa, sino que es posible establecer previamente un quórum), recompone la clave privada del sobre.
- Con la clave privada del sobre, el sistema ya puede extraer y mostrar la oferta.

Por último, el sistema de licitación electrónica debe contar con un componente de gestión de expedientes que permita llevar a cabo todos los trámites en la secuencia correcta, así como garantizar el acceso a la documentación generada (actas, informes, etc.) y almacenada.

### **18.8. OSIMGA**

O Observatorio da Sociedade da Información e a Modernización de Galicia (OSIMGA) é un órgano asesor para a valoración da evolución da Sociedade da Información, a Modernización Administrativa e a Administración electrónica nas Administracións Públicas de Galicia, e para a participación e colaboración coas distintas Administracións Públicas nestas materias, regulado polo Decreto 21/2010 do 4 de febreiro (DOG, 26/02/2010), e adscrito á Secretaría Xeral de Modernización e Innovación Tecnolóxica da Xunta de Galicia.

Entre as súas funcións están as de desenvolver ou promover estudos e análises de datos que permitan coñecer o nivel de desenvolvemento, a

tendencia e posibles problemáticas que poden afectar á extensión da Sociedade da Información en Galicia e a aplicación do modelo de eGoberno nas Administracións Públicas galegas. O OSIMGA elabora informes de situación e de evolución e facilita datos a outros organismos competentes na materia, contribuíndo á definición estratéxica de políticas públicas.

Así mesmo, o Observatorio analiza o estado de desenvolvemento da Sociedade da Información para toda a cidadanía e, en especial, no que respecta aos colectivos en risco de exclusión, promovendo liñas de actuación que potencien a súa incorporación e permanencia en condicións de igualdade efectiva.

A web do OSIMGA (<http://www.osimga.org/>) é unha canle aberta de comunicación e difusión dos datos, informacións, estudos e outros materiais elaborados polo Observatorio. Ademais subministra unha selección de novas relacionadas coa modernización da Administración autonómica galega e a sociedade da información en Galicia.

Os principais obxectivos do Observatorio son:

1. Desenvolver ou promover compilacións, estudos e análises de datos que permitan coñecer cunha visión global o nivel de desenvolvemento, a tendencia e os posibles problemas que afecten a extensión da sociedade da información en Galicia e á aplicación do modelo de eGoberno nas Administracións Públicas galegas.
2. Facilitar análises comparativas e aliñacións de datos con outros marcos xeográficos.
3. Promover o intercambio de experiencias e información entre Administracións, con outros observatorios, organismos ou entidades.
4. Impulsar a organización de eventos formativos, reunións de expertos ou grupos de traballo.

5. Promover e xestionar a elaboración e difusión de publicacións técnicas, impresas ou electrónicas, específicas e monográficas ou de publicación periódica.
6. Xestionar e ofrecer periodicamente e, como mínimo, a través dunha web específica, información sobre o nivel de desenvolvemento da Sociedade da Información en Galicia, eventos formativos, noticias de actualidade, ou enlaces con outras fontes de información, observatorios ou entidades.
7. Elaborar e presentar publicamente informes que reflectan o estado de situación ou a evolución prevista.
8. Facilitar datos a outros organismos e entidades con competencias específicas na materia.
9. Analizar o estado de desenvolvemento da Sociedade da Información para toda a cidadanía e, en especial, no que respecta ás mulleres, ás persoas con discapacidade, ás persoas maiores e aos colectivos con risco de exclusión, promovendo liñas de actuación que potencien a súa incorporación e permanencia activa en condicións de igualdade efectiva.
10. Avaliar e servir de elemento para a definición de políticas públicas en materia de Sociedade da Información e modernización da Administración.

### **18.9. REDE CeMIT**

A Rede de Centros para a Modernización e a Inclusión Tecnolóxica (CeMIT) é unha iniciativa posta en marcha pola Xunta da Galicia que busca impulsar as TIC e a Sociedade da Información na comunidade galega. Esta rede forma parte da Axenda Dixital 2014.gal, enmarcada no Plan Estratéxico Galicia 2010-2014 que ten como fin acadar a converxencia tecnolóxica con Europa no horizonte do ano 2020.

A Rede CeMIT nace cos obxectivos estratéxicos de vertebrar territorial e socialmente Galicia, en especial onde a fenda dixital se fai máis evidente, e

impulsar, potenciar e difundir os coñecementos nas tecnoloxías da información e a comunicación en tres colectivos principalmente: cidadanía, profesionais TIC e empregados públicos

A Rede CeMIT ofrece un amplo abano de servizos que se divide nos seguintes tres grandes bloques:

- a) **FORMACIÓN:** A cidadáns, profesionais TIC, empresas, empregados públicos e axentes territoriais.
- b) **ACTIVIDADES DE DIFUSIÓN:** Charlas e xornadas de sensibilización e divulgación das actividades da Rede, a captación de usuarios e introdución a temas relacionados coas novas tecnoloxías.
- c) **AULA ABERTA:** Espazos nos que os seus usuarios/as, dentro dun horario establecido, poderán acceder libre e gratuitamente á utilización das aulas, para o que contarán co apoio e asesoramento dos axentes TIC.

**CENTROS DE ALTA ESPECIALIZACIÓN:** A ESCOLA GALEGA DE ADMINISTRACIÓN PÚBLICA E O CENTRO DE NOVAS TECNOLOXÍAS DE GALICIA.

A Escola Galega de Administración Pública (EGAP), é un organismo público pertencente a Xunta de Galicia, e a súa finalidade é deseñar e impartir o programa de formación destinado aos empregados públicos de Galicia.

O Centro de Novas Tecnoloxías de Galicia (CNTG) contribúe activamente na formación e capacitación dos profesionais TIC galegos a través de cursos especializados e certificacións tecnolóxicas de primeiro nivel.

O Plan director de telecomunicación de banda larga busca garantir que as infraestruturas de telecomunicacións contén coa capacidade necesaria para facer posible o acceso de todos os galegos á sociedade da información.

A Xunta de Galicia asume a coordinación dos axentes implicados na elaboración do plan, asegurando orde, eficiencia e un uso óptimo dos recursos. As autoridades rexionais e locais serán as mellor situadas para planificar os proxectos de banda larga.

A Xunta de Galicia aprobou o 18 de febreiro de 2010 o Plan Banda Larga de Galicia 2010-2013, que define a estratexia a seguir para alcanzar as seguintes metas: Reducir o desequilibrio territorial, impulsar a competitividade e innovación nas empresas, modernizar os servizos públicos, e maximizar a cooperación asegurando un despregamento baseado na eficiencia e na cooperación de todos os axentes implicados.

Por tanto, o Plan Director de Banda Larga de Galicia é un instrumento esencial e necesario para a definición das políticas de infraestruturas de Galicia que permitiría, entre outros:

- Impulsar unha estratexia global, encamiñada a situar a Galicia no núcleo avanzado da Sociedade da Información.
- Garantir a capacidade de acceso dos galegos á Sociedade da Información baixo condicións de homoxeneización de calidade de servizo e custo.
- Asegurar a vertebración dixital do noso territorio, como elemento de compensación de desequilibrios culturais, tecnolóxicos e socioeconómico, de inclusión social e de eliminación da fenda dixital.
- Extraer o máximo aproveitamento das posibilidades das novas tecnoloxías como dinamizadoras económicas e xeradoras de

competitividade e innovación nos diferentes sectores produtivos e como medio para promover a equidade, a sustentabilidade e a calidade dos servizos públicos.

- Impulsar a modernización da Administración Pública autonómica e local, empregando toda a potencialidade que ofrece hoxe a tecnoloxía.

A coordinación e tutela do proxecto centralízase na Oficina Técnica, a cal ofrece servizos de xestión, control e seguimento das actuacións do plan, así como, servizos orientados á atención do cidadán.

### **18.11. AXENDA DIXITAL 2014.gal**

Galicia marcouse como obxectivo, dentro do actual marco establecido polo Plan Estratéxico Galicia 2010-2014, o reto de converxer co horizonte europeo para 2020.

Deste Plan xorde a Axenda Dixital de Galicia, como aposta pola definición dunha estratexia clara en materia de Sociedade da Información, que nos permita competir como rexión no novo mercado único dixital europeo definido pola Axenda Dixital para Europa e na nova economía do coñecemento, como camiño para unha recuperación económica sustentable.

Dentro do contexto da Axenda Dixital de Galicia, o último ano supuxo o arranque para iniciativas básicas na creación de infraestruturas, como o Plan de Banda Larga ou o Centro de Proceso de Datos Integral, e a aposta decidida pola Administración electrónica, impulsando o Decreto 198/2010, de 2 de decembro, polo que se regula o Desenvolvemento da Administración Electrónica na Xunta de Galicia y nas entidades dela dependentes, desenvolvendo proxectos para a mellora de servizos públicos

dixitais e fomentando e divulgando o uso do TIC, como a rede CeMIT ou o proxecto Abalar.

A Axenda Dixital de Galicia enfróntase a un importante cambio de enfoque estratéxico: o seu obxectivo é pasar dunha sociedade que utiliza o TIC, a unha sociedade galega que se serve das novas tecnoloxías para xerar un crecemento sustentable, para mellorar as súas cotas de participación na toma de decisións e para contribuír á súa calidade de vida sobre a base do coñecemento. Para logralo, contará co impulso e a participación activa de todos os axentes implicados na Sociedade da Información.

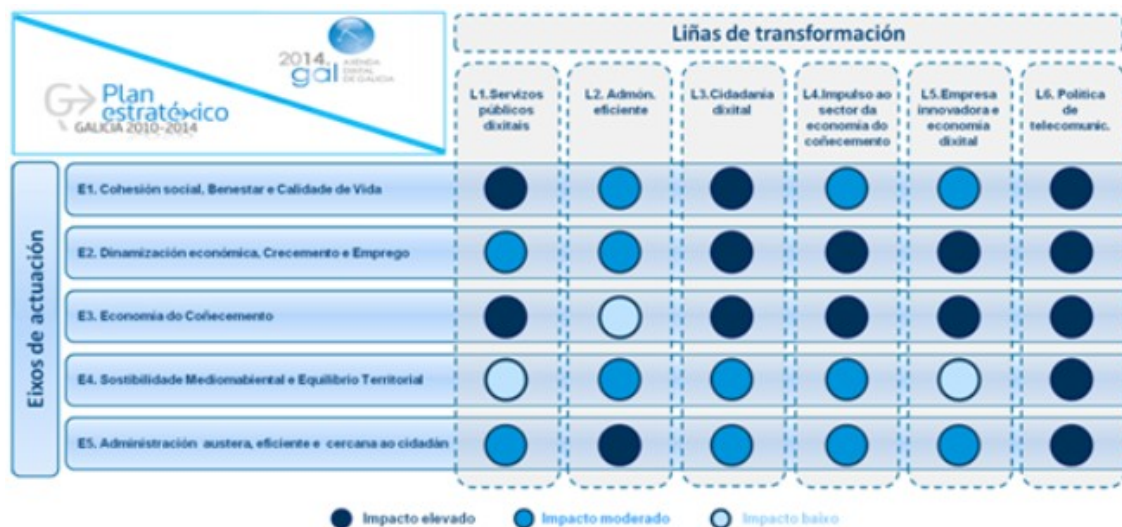
Este enfoque marcouno a Comisión Europea, que deseñou unha estratexia para axudarnos a saír fortalecidos da crise e a converter á UE nunha economía que goce de altos niveis de emprego, de produtividade e de cohesión social. Este é o labor de "Europa 2020: Unha estratexia para un crecemento intelixente, sustentable e integrador" (Bruxelas, 3.3.2010 - COM(2010) 2020. Neste contorno, Europa 2020 establece sete iniciativas para catalizar os avances en cada un dos temas fixados como prioritarios. Unha delas, consiste na definición da Axenda Dixital para Europa (Bruxelas, 19.05.2010 COM (2010) 245) que fai das tecnoloxías da información e da comunicación (TIC) a peza crave para que Europa consiga as súas ambicións para 2020.

A Axenda Dixital impulsará a inclusión de Galicia no novo contexto dixital europeo de forma definitiva no horizonte 2014. O Plan Estratéxico Galicia 2010-2014 define un novo modelo socioeconómico que pretende unha modernización para toda Galicia, que nos implique e inclúanos a todos. As novas tecnoloxías deben servir como catalizadoras dos eixos marcados no Plan Estratéxico, como mecanismo facilitador da cohesión social, da calidade de vida dos galegos, da xeración de emprego de calidade e como impulsor dunha Administración austera, eficiente e próxima ao cidadán.



A continuación pode verse como se alíña a Axenda Dixital de Galicia co Plan

Estratéxico Galicia 2010 -2014



Ademais do **Plan Estratégico Galicia 2010-2014**, a nova Axenda tamén estará integrada con outros plans estratéxicos, como o Plan Galego de I+D+i, co Plan de Banda Larga de Galicia, con Plans de competitividade de diferentes sectores estratéxicos para Galicia (automoción, madeira, téxtil, construción naval, enerxía, pedra e turismo) e coas Axendas Dixitais Locais que promoven o desenvolvemento da Sociedade da Información nunha contorna máis próxima á cidadanía.



ESTRATEGIA



Para alcanzar o salto cualitativo implícito no reto que se plantexa Galicia establécense sete liñas estratéxicas dentro da Axenda Dixital de Galicia:

## L1. SERVIZOS PÚBLICOS DIXITAIS:

Obxectivos:

- Incluír as TIC en todos os ámbitos dos servizos públicos. Pasar do concepto de “Administración electrónica” ao concepto de “eGoberno”.
- Adaptar o modelo de Administración para as novas posibilidades de xestión a través das TIC.
- Prestar os servizos públicos transmitindo seguridade e confianza, asumindo o reto da interoperabilidade.
- Facilitar a vida á cidadanía mediante a mellora dos servizos prestados en ámbitos como: sanidade, educación, xustiza, benestar, etc.
- Mellorar a competitividade do tecido produtivo galego.
- Garantir uns servizos públicos homoxéneos en todo o territorio mediante a colaboración entre as Administracións Públicas.

## L2. ADMINISTRACIÓN EFICIENTE.

Obxectivos:

- Proporcionar servizos de calidade e eficaces aos usuarios aliñados coas necesidades e directrices estratéxicas.
- Xestión eficiente do gasto: Facer máis e mellor con menos.
- Homoxeneizar e consolidar sistemas de xestión TIC, baixo unhas directrices operativas e estratéxicas unificadas de interoperabilidade.
- Trasferencia e compartición do Coñecemento TIC en Galicia.

## L3. CIDADANÍA DIXITAL.

### Obxectivos

- Cohesionar territorial e socialmente Galicia mediante a eliminación da fenda dixital e o impulso dos nativos dixitais.
- Incrementar a empregabilidade grazas ao desenvolvemento das capacidades tecnolóxicas que derivan do uso das TIC.
- Mellorar a calidade de vida dos cidadáns e garantir a súa autonomía persoal universalizando a cultura dixital na nosa comunidade.

## L4. IMPULSO AO SECTOR DA ECONOMÍA DO COÑECEMENTO

### Obxectivos:

- Consolidar un sector TIC competitivo, innovador e xerador de emprego cualificado, capaz de facer fronte aos retos da nova economía do coñecemento.
- Construír un sector TIC cohesionado e integrador, que actúe como motor do resto de sectores estratéxicos de Galicia.
- Converter o sector TIC nun sector forte en busca da excelencia tecnolóxica, que estimule a creación dunha clase emprendedora e creativa.

Deste xeito, o sector TIC non só se converterá nun sector estratéxico en si mesmo, senón que tamén actuará como motor dos sectores estratéxicos de Galicia impulsando un novo modelo produtivo baseado na economía do coñecemento.

## L5. EMPRESA INNOVADORA E ECONOMÍA DIXITAL

### Obxectivos:

- Contribuír á creación de emprego e ao crecemento económico sostible de Galicia creando un novo modelo produtivo.

- Facer máis competitivos os sectores estratéxicos de Galicia mediante a incorporación das TIC aos seus modelos de negocio (automoción, construción naval, enerxía, pedra e rocas ornamentais, téxtil, madeira, turismo).
- Facilitar a incorporación de microPEMEs e autónomos á sociedade do coñecemento, reducindo a fenda dixital para converxer con España e Europa.

## L6. POLÍTICAS DE TELECOMUNICACIÓNS

### Obxectivos:

- Garantir o acceso á Sociedade da Información a toda a sociedade galega.
- Mellorar os servizos ofrecidos á cidadanía a través da modernización dos servizos e infraestruturas de telecomunicacións corporativas soporte dos sistemas e procesos da Xunta.

## L7. MEDIDAS INSTRUMENTAIS DE SEGUIMENTO E COOPERACIÓN.

Por ultimo, una sétima liña, que da soporte ao desenvolvemento das anteriores. Nesta liña, a Xunta de Galicia como elemento dinamizador do sector TIC rexional pretende establecer unha política de austeridade e simplificación de estruturas organizativas, de tal forma que se transforme nunha Administración áxil, de confianza e segura.

Nesta liña, establécese a necesidade de constituír unha nova entidade de xestión das TIC de carácter público e adscrita a todos os niveis á Xunta de Galicia que aglutine as competencias que son responsabilidade da actual Secretaría Xeral de Modernización e Innovación Tecnolóxica e que concentre os recursos humanos, materiais e orzamentarios asociados aos actuais departamentos TIC dispersos nas Consellerías de la Xunta de Galicia.

O Consello da Xunta de Galicia que tivo lugar o 21 de xullo de 2011 aprobou o Plan de posta en marcha da Axencia de Modernización Tecnolóxica de Galicia (AMTEGA) co obxectivo de consolidar un modelo de xestión integrado das TIC na Administración autonómica.

## **18.12. ESTRATEXIA DE IMPULSO DO SECTOR TIC**

O sector das Tecnoloxías da Información e da Comunicación está a converterse nunha das áreas produtivas claves no desenvolvemento económico e social de Galicia, pola súa condición de acelerador do cambio tecnolóxico e polo seu carácter transversal ao resto de sectores. O número de empresas galegas do sector TIC no ano 2009 situábase en 1.542 empresas, o que supón un 9,8 % máis que en 2006.

### **LIÑA TRANSFORMA TIC**

2014.gal persegue a consecución duns obxectivos, a través desta liña:

- Consolidar un hipersector TIC competitivo, innovador e xerador de emprego cualificado, que se converta en sector estratéxico para a economía galega.
- Que ese sector sexa cohesionado e integrador. Que actúe tamén como motor do resto de sectores estratéxicos de Galicia.
- A creación dunha clase emprendedora e creativa estimulada por ese hipersector TIC.
- Busca da excelencia tecnolóxica.

As liñas da Axenda Dixital de Galicia, e principalmente las L4 e L5, explican máis en detalle os obxectivos, enfoque e aspectos clave deste plan de impulso do sector TIC.

## MAPA DE CAPACIDADES TECNOLÓXICAS DE GALICIA.

A Xunta de Galicia está a poñer en marcha, como unha das actuacións dentro da estratexia de impulso do sector TIC, un programa de demanda temperá de tecnoloxía innovadora.

Como primeira actividade dentro deste programa, e dentro do ámbito socio-sanitario, elaborouse o mapa de capacidades tecnolóxicas de Galicia.

O mapa inclúe a experiencia e coñecementos tecnolóxicos, ademais de información detallada de numerosos proxectos, de todos os grupos de investigación TIC de Galicia que desenvolven o seu traballo no ámbito mencionado e de diferentes centros tecnolóxicos galegos ademais do sector empresarial.

### LIÑA EconomiC-IT

Neste mundo globalizado, Galicia debe construír unha economía intelixente, sustentable e integradora, impulsando a competitividade e mellorando a produtividade do tecido empresarial, eliminando as debilidades estruturais existentes, incentivando a innovación e a calidade, apoiando a creación de novas empresas e xerando un alto nivel de emprego.

Nesta contorna, 2014.gal Axenda Dixital de Galicia, definiu unha liña estratéxica “Empresa innovadora e Economía Dixital” co obxectivo de promover a Sociedade da Información no tecido empresarial galego.

Os obxectivos estratéxicos que persegue a Axenda Dixital a través da definición desta liña son os seguintes:

- Contribuír a creación de emprego e ao crecemento económico sostible de Galicia creando un novo modelo produtivo.
- Facer máis competitivos os sectores estratéxicos de Galicia mediante as incorporacións das TIC aos seus modelos de negocio (automoción, construción naval, enerxía, téxtil)
- Facilitar a incorporación das micropemes e autónomos á Sociedade da Información, reducindo a fenda dixital para converxer con España e Europa.

A figura do titor tecnolóxico asociado a rede CeMIT xogará un papel moi importante, axudando a coñecer a situación das empresas identificando cales son as súas necesidades concretas.

## CENTRO DEMOSTRADOR TIC DE GALICIA

O Centro Demostrador TIC de Galicia é o instrumento operativo enmarcado no eixo de actuación económica-T: “Empresas DIXITAL E INNOVADORA” da Axenda Dixital 2014.gal, e ponse en marcha en Galicia a través dun Convenio entre a Secretaría Xeral de Modernización e Innovación Tecnolóxica, a Consellería de Traballo e Benestar e a entidade pública empresarial Red.es.

A misión do Centro é facilitar ás empresas TIC os medios para acercar a súa oferta de produtos ás empresas doutros sectores produtivos, de forma que poidan desenvolver produtos adaptados ás necesidades de mercado, con dous obxectivos: incrementar a demanda de produtos TIC e adaptar esta demanda ás necesidades de innovación dos sectores estratéxicos.

Pódese atopar abundante información sobre os contidos deste tema no enderezo <http://imit.xunta.es/>.

### **18.13. REFERENCIAS**

- Ley 30/1992, de 26 de noviembre, de Régimen Jurídico de las Administraciones Públicas y del Procedimiento Administrativo Común.
- Ley 15/1999, de 13 de diciembre, de Protección de datos de carácter personal
- Ley 53/1999, de 19 de diciembre, de Firma electrónica.
- Ley 34/2002, de 11 de julio, de Servicios de la sociedad de la información y de comercio electrónico.
- Ley 11/2007, de 22 de junio, de Acceso electrónico dos cidadanos a los servicios públicos.
- Ley 30/2007, de 30 de octubre, de Contratos do Sector Público.
- Ley 37/2007, de 16 de noviembre, sobre Reutilización de la información del sector público.
- Real Decreto 1720/2007, de 21 de diciembre, por el que se aprueba el Reglamento de desarrollo de la Ley 15/1999, de 13 de diciembre, de protección de datos de carácter personal.
- Ley 56/2007, de 28 de diciembre, de Medidas de impulso de la sociedad de la Información.
- Real Decreto 1494/2007, de 12 de noviembre, por el que se aprueba el Reglamento sobre las condiciones básicas para el acceso de las personas con discapacidad a las tecnologías, productos y servicios relacionados con la sociedad de la información y medios de comunicación social.
- Ley 17/2009, de 23 de noviembre, sobre el Libre acceso a las actividades de servicios y su ejercicio.
- Ley 25/2009, de 22 de diciembre, de Modificación de diversas leyes para su adaptación a la Ley 17/2009, de 23 de noviembre, sobre el Libre acceso a las actividades de servicios y su ejercicio.
- Real Decreto 3/2010, de 8 de enero, por el que se regula el Esquema Nacional de Seguridad en el ámbito da Administración Electrónica.

- Real Decreto 4/2010, de 8 de enero, por el que se regula el Esquema Nacional de Interoperabilidad en el ámbito de la Administración Electrónica.
- Decreto 198/2010, de 2 de diciembre, por el que se regula el Desarrollo de la Administración Electrónica en la Xunta de Galicia y en las entidades dependientes.
- Resoluciones de la Secretaría de Estado para la Función Pública por la que se aprueban distintas normas técnicas de interoperabilidad.
- iMIT - Iniciativas de Modernización e Innovación Tecnolóxica, da Xunta de Galicia (<http://www.imit.xunta.es/>).
- Área sobre a Axenda Dixital para Europa no sitio web da Comisión Europea ([http://ec.europa.eu/information\\_society/digital-agenda/index\\_en.htm](http://ec.europa.eu/information_society/digital-agenda/index_en.htm)).
- “Manual práctico de supervivencia de la Administración Electrónica”, de Alberto López Tallón, publicado bajo licencia Creative Commons.
- “Anotacións e comentarios ao Decreto de Administración Electrónica da Xunta de Galicia”, editado polo Colexio Profesional de Enxeñaría en Informática de Galicia coa colaboración da Xunta de Galicia. ISBN 978-84-614-7362-5.
- “Las relaciones de la empresa con la Administración Electrónica”, editado polo Colexio Profesional de Enxeñaría en Informática de Galicia. ISBN 978-84-614-9865-9.
- “Empresa, protección de datos y Administración Electrónica”, editado polo Colexio Profesional de Enxeñaría en Informática de Galicia. ISBN 978-84-614-4014-6.



Autor: Jesús Rodríguez Castro

Jefe del Servicio de Informática del Concello de Santiago de  
Compostela

Colegiado del CPEIG

# **SISTEMAS DE INFORMACIÓN.**

**19. DEFINICIÓN Y  
ESTRUCTURAS DE LOS  
SISTEMAS DE INFORMACIÓN.  
SUBSISTEMA FÍSICO Y LÓGICO.  
PRINCIPIOS DE  
FUNCIONAMIENTO DE LOS  
ORDENADORES.  
ARQUITECTURA Y  
COMPONENTES DE LOS  
ORDENADORES. UNIDAD  
CENTRAL Y PERIFÉRICOS.  
SERVIDORES. PUESTO DE  
TRABAJO. DISPOSITIVOS  
PERSONALES.**

**Tema 19: Definición e estruturas dos sistemas de información.  
Subsistema físico e lóxico. Principios de funcionamento dos  
ordenadores. Arquitectura e compoñentes dos ordenadores.  
Unidade central e periféricos. Servidores. Posto de traballo.  
Dispositivos persoais**

---

## **ÍNDICE**

### **19.1 Definición y estructuras de los sistemas de información. 2**

*19.1.1 Definición de un Sistema de Información 2*

*19.1.2 Componentes y Características 2*

### **19.2 Subsistema físico y lógico. 4**

### **19.3 Principios y fundamentos de los ordenadores. Arquitectura y componentes de los ordenadores. Unidad central y periféricos. 5**

### **19.4 Servidores 9**

*19.4.1 Características de un servidor 9*

*19.4.2 Cluster 10*

*19.4.2.1 Clases de clústeres 10*

*19.4.2.2 Componentes de un clúster 11*

*19.4.3 Servidores Blade 12*

### **19.5 Puesto de trabajo. 14**

### **19.6 Dispositivos personales 15**

*19.6.1 PDA 15*

*19.6.2 TABLET 16*

*19.6.3 Smartphones 16*

### **19.7 BIBLIOGRAFIA 19**

### ***Definición y estructuras de los sistemas de información.***

Los Sistemas de Información fueron considerados inicialmente como un elemento que podía proporcionar ahorros de coste en las organizaciones, en la medida que estos podrían dar soporte a actividades de índole operativo en las que la información constituía el principal elemento a procesar.

Hoy el Sistema de Información constituye la base para el desarrollo de nuevos productos o servicios en la empresa actual. Es el soporte principal del trabajo de la empresa y de las actividades de la dirección ya que permite coordinar el trabajo dentro y entre las organizaciones y sobre todo, permite mejorar el funcionamiento de las mismas, desarrollando nuevos modelos organizativos con una clara orientación a la información.

### ***Definición de un Sistema de Información***

No existe una definición aceptada universalmente para el Sistema de Información de la Empresa. Podemos decir que “sistema” es, en un sentido amplio, un conjunto de componentes que interaccionan para cumplir algún objetivo.

Las definiciones más aceptadas en la literatura son:

“El sistema de información puede definirse técnicamente como un conjunto de componentes interrelacionados que recogen, procesan, almacenan y distribuyen la información para dar soporte a la toma de decisiones y al control en una organización. Además del apoyo a la toma de decisiones, coordinación y control, los sistemas de información pueden ayudar a los gestores y trabajadores en general a analizar problemas o situaciones y a desarrollar nuevos productos”.

“Un sistema de información es un conjunto de componentes interrelacionados que recogen, procesan y distribuyen datos e información, incluyendo asimismo los mecanismos de retroalimentación implicados”.

Las definiciones de sistema de información de una organización, implican a todas las personas de la organización, ya sea como usuario, técnico, directivo, o especialista todas las personas de la organización están

involucradas en alguna de las actividades de entrada, proceso o salida del sistema de información.

Dos ideas fundamentales relacionadas con la definición de sistema de información en el ámbito de la empresa:

La primera es que el sistema de información de la organización no es un departamento nuevo, ni mucho menos una parte dependiente de alguno de los departamentos funcionales clásicos; aunque por motivos históricos los primeros tratamientos automáticos de datos cayeron dentro de las competencias del área financiera.

Un proyecto de sistemas de información no debe considerarse como una tarea funcional o departamental, sino como un proyecto que compromete a representantes de todos los departamentos sea cuál sea su nivel en la estructura jerárquica.

La segunda, que el sistema de información no es sólo el centro de proceso de datos (CPD). El centro de proceso de datos sólo sería una parte de los recursos de información, esto es, una parte de las actividades y componentes del sistema de información.

Como la información es la base de la mayoría de las actividades realizadas en una compañía, es necesario desarrollar sistemas que produzcan y administren esta información. Un sistema de información debe proveer información que apoye las operaciones, la administración y las funciones de toma de decisiones en una empresa. En términos muy generales, la tarea del diseño y desarrollo de un sistema de información consiste en analizar un problema de administración, determinar los requerimientos de información por parte del usuario, diseñar un nuevo sistema y conseguir su implantación a la vista de unas restricciones de tiempo, recursos y presupuesto determinadas por la dirección.

Un sistema de información, entendido como un conjunto de personas, datos y procedimientos que funcionan de manera unitaria ejecuta tres actividades generales. En primer lugar, recibe datos de fuentes internas o externas de la empresa, en segundo lugar actúa sobre los datos para producir información y por último el sistema produce la información para los usuarios. O sea, el sistema es un generador de información que utiliza procedimientos que determinan cómo se elabora la información. El objetivo principal de estos sistemas es asegurar que la información sea exacta, esté disponible en el momento preciso y que se presente de forma adecuada.

Los sistemas de información no necesitan estar basados en la informática, aunque la mayoría de las veces lo estén. El factor determinante consiste en

saber si un sistema puede ser mejorado incluyendo en él la capacidad del procesamiento con ordenadores. Si se tiene un sistema manual que ejecuta el trabajo eficientemente y sin error, existen pocos motivos para utilizar ordenadores. De todas maneras, cuando crece el volumen de trabajo, los procedimientos aumentan su complejidad y las actividades llegan a estar más interrelacionadas, obteniéndose grandes mejoras al introducir la ayuda de sistemas informáticos.

### ***Componentes y Características***

Como ya se ha comentado anteriormente, es importante matizar la función de “soporte” que las tecnologías de la información juegan para la implantación de sistemas de información.

Los sistemas de información precisan los siguientes elementos:

**Hardware:** consiste en el equipamiento informático para llevar a cabo la entrada, proceso y salida de la información. Los dispositivos de entrada incluyen teclados, lectores de cinta magnética, etc. Los dispositivos de proceso incluyen la unidad central de proceso, memoria y almacenamiento. Existen múltiples dispositivos de salida como impresoras y pantallas de ordenador. Sin ánimo de profundizar en este tipo de recursos, podemos clasificar a los recursos de hardware en tres grandes categorías:

**Ordenadores,** que además de los miniordenadores y mainframes utilizados desde hace varias décadas, merece destacar la presencia habitual en los puestos de trabajo del ordenador personal.

**Periféricos,** los cuales incluyen toda una variada gama que abarca desde los elementos más comunes de entrada, visualización, almacenamiento y distribución de datos e información como pueden ser los teclados, módems, pantallas de rayos catódicos, cintas magnéticas e impresoras de impacto hasta los elementos más sofisticados, tales como un escáner, pantallas de vídeo gigantes, discos compactos e impresoras láser y de inyección de tinta, etc.

**Otro hardware,** en especial el que permite la interconexión entre equipos informáticos.

**Software:** consiste en los programas e instrucciones que se dan al ordenador. Estos programas e instrucciones permiten al ordenador procesar la información. Podemos clasificar el software en los tres siguientes grupos:

**Software de sistemas;** se trata de un componente estrechamente asociado al propio hardware, que permite a éste operar y soportar software específico para las aplicaciones.

Software de aplicación; que es el conjunto de programas que permitirán realizar las funciones previstas para el sistema de información.

Otro software; como pueden ser las herramientas de desarrollo, compiladores, software de comunicaciones, etc.

Bases de datos: donde se encuentra organizada la información de la empresa. La base de datos de la organización puede recoger hechos e información sobre clientes, empleados, ventas de competencia, etc. Supone uno de los elementos claves para el sistema de información basado en T.I.

Telecomunicaciones: permiten a las organizaciones enlazar sistemas informáticos constituyendo redes. Las redes de área local interconectan equipos en un entorno reducido, típicamente en una oficina o edificio. Las redes de área amplia (WAN) pueden interconectar sistemas alejados mediante redes de comunicación exteriores, habitualmente públicas.

Personas: es el elemento más importante en la mayor parte de los sistemas de información basados en ordenador. Podemos distinguir dos tipos de personas en relación al sistema de información:

Personal específico del área de sistemas: incluye aquellas personas que gestionan, desarrollan programas, y mantienen el sistema informático.

Usuarios: directivos y usuarios en general que interaccionan de algún modo con el sistema.

Procedimientos: incluye las estrategias, políticas, métodos y reglas que en general se aplican en el uso y gestión del sistema de información.



### ***Subsistema físico y lógico.***

Un computador es un conjunto de dispositivos físicos integrados y de elementos lógicos (programas) destinado al tratamiento automatizado de la información. El Sistema Informático, a partir de unos datos de entrada, realiza funciones de almacenamiento, proceso y control proporcionando datos de salida, por eso consideramos que el usuario utiliza los datos que le proporciona el Sistema Informático, pero no forma parte de él. El usuario es un componente más del Sistema de Información, que interacciona, en el caso de un Sistema de Información automatizado con otro componente del mismo denominado Sistema Informático.

La Arquitectura de un Sistema Informático se define como el subconjunto de reglas, normas y procedimientos que especifican las interrelaciones que deben existir entre los componentes y elementos, físicos y lógicos, de un Sistema Informático y las características que deben cumplir cada uno de estos componentes.

El Sistema Informático, como soporte del Sistema de Información, ha evolucionado desde una primera situación en que todos los componentes del Sistema - físicos, lógicos y personales - se encontraban centralizados en una sala de ordenadores a la situación actual en que los componentes del Sistema se encuentran ampliamente distribuidos en diferentes lugares físicos.

Este camino hacia la implantación progresiva de sistemas distribuidos ha pasado por diferentes fases y no se puede dar por finalizado pensando en la evolución futura hacia sistemas repartidos. Las fases son las siguientes:

- 1) En una primera fase, coincidente con los inicios de los procesos de informatización en las organizaciones, los recursos se encuentran totalmente centralizados.
- 2) Una segunda fase, se inicia con la distribución de los componentes físicos y en algunos casos humanos del sistema. Esta fase se caracteriza por la introducción de los terminales no inteligentes asociados a las primeras redes de teleproceso. En esta segunda fase toda la capacidad de proceso y almacenamiento se encuentra centralizada, sin embargo, la entrada y salida de datos se ha distribuido físicamente. Los inconvenientes de este tipo de Sistemas eran los costes de las comunicaciones y la gran complejidad de los Sistemas centralizados que soportaban las redes de teleproceso. Estos Sistemas consumían gran cantidad de recursos en su gestión interna.

3) En la tercera fase se distribuyen además los elementos lógicos, introduciendo cierta inteligencia en los terminales. En realidad ésta es una fase de transición porque el abaratamiento de costes de los equipos y el desarrollo tecnológico permitieron pasar rápidamente a la informática distribuida de hoy día. No obstante, este tipo de organización del sistema informática en el que existen terminales con cierta capacidad de proceso conectados a un equipo central, se sigue manteniendo en muchas Organizaciones, ya que es netamente preferible al modelo anterior, sobre todo por el menor coste de las comunicaciones.

## ***Principios y fundamentos de los ordenadores. Arquitectura y componentes de los ordenadores. Unidad central y periféricos.***

El ordenador se puede ver como un dispositivo electrónico destinado al tratamiento automatizado de la información. Para que un ordenador trate la información es necesario un sistema de información, que ante una entrada, ejecute una serie de instrucciones y devuelva un resultado.

Una arquitectura de ordenador consiste en el diseño, estudio de la estructura, y funcionamiento de un ordenador. Especifica las interrelaciones que deben existir entre los componentes y elementos físicos y lógicos.

### **Modelos de arquitecturas de ordenadores:**

#### **Arquitectura Von Newmann:**

Consiste en una unidad central de proceso que se comunica a través de un solo bus con un banco de memoria en donde se almacenan tanto las instrucciones del programa, como los datos que serán procesados por éste. Esta arquitectura es la más empleada en la actualidad.

En la memoria se almacenan tanto los datos como las instrucciones que forman el programa, con lo cual el cambio de un programa a otro sólo implica un cambio en el valor de posiciones de memoria.

En la arquitectura de von Newman se produce en la CPU una cierta ralentización debido a que tanto las instrucciones como los datos deben pasar de la memoria a la CPU por un único canal (bus). A este efecto se le conoce como "el cuello de botella de Von Newmann". Esto limita el grado de paralelismo (acciones que se pueden realizar al mismo tiempo), y por lo tanto, el desempeño de la computadora.

En esta arquitectura se asigna un código numérico a cada instrucción. Dichos códigos se almacenan en la misma unidad de memoria que los datos que van a procesarse para ser ejecutados en el orden en que se encuentran almacenados en memoria. Esto permite cambiar rápidamente la aplicación de la computadora y dio origen a las computadoras de propósito general.

#### **Arquitectura Harvard**

Esta arquitectura surgió en la universidad del mismo nombre, poco después de que la arquitectura Von Newman. Al igual que en la arquitectura Von Newman, el programa se almacena como un código numérico en la memoria, pero no en el mismo espacio de memoria ni en el mismo formato que los datos. Por ejemplo, se pueden almacenar las instrucciones en doce bits en la memoria de programa, mientras los datos se almacenan en 8 bits en una memoria aparte.

El hecho de tener un bus separado para el programa y otro para los datos permite que se lea el código de operación de una instrucción, al mismo tiempo que se lee de la memoria de datos los operandos de la instrucción previa. Así se evita el problema del cuello de botella de Von Newman y se obtiene más rendimiento.

La complejidad de esta arquitectura sólo compensa cuando el flujo de instrucciones y de datos es más o menos el mismo. Por eso **no** es ampliamente utilizada en ordenadores de propósito general. Sin embargo, sí se utiliza en algunos casos para construir procesadores de señal (DSP).

## **Arquitecturas segmentadas**

Buscan mejorar el rendimiento realizando paralelamente varias etapas del ciclo de instrucción al mismo tiempo. El procesador se divide en varias unidades funcionales independientes y se dividen entre ellas el procesamiento de las instrucciones.

Si un procesador tiene un ciclo de instrucción sencillo consistente solamente en una etapa de búsqueda del código de instrucción y en otra etapa de ejecución de la instrucción, en un procesador sin segmentación, las dos etapas se realizarían de manera secuencial para cada una de las instrucciones; por el contrario en un procesador con segmentación, cada una de estas etapas se asigna a una unidad funcional diferente, la búsqueda a la unidad de búsqueda y la ejecución a la unidad de ejecución. Estas unidades pueden trabajar en forma paralela en instrucciones diferentes. Estas unidades se comunican por medio de una cola de instrucciones en la que la unidad de búsqueda coloca los códigos de instrucción que leyó para que la unidad de ejecución los tome de la cola y los ejecute.

La mejora en el rendimiento no es proporcional al número de segmentos debido a que cada etapa no toma el mismo tiempo en realizarse, además de que se puede presentar competencia por el uso de algunos recursos como la memoria principal. Otra razón por la que las ventajas de este esquema se pierden es cuando se encuentra un salto en el programa y todas las instrucciones que ya se buscaron y se encuentran en la cola, deben descartarse y comenzar a buscar las instrucciones desde cero a partir de la dirección a la que se saltó. Esto reduce el desempeño del

procesador y aún se investigan maneras de predecir los saltos para evitar este problema.

## **Arquitectura multiprocesamiento.**

Cuando se desea incrementar el rendimiento es necesario utilizar más de un procesador para la ejecución del programa.

Para hacer una clasificación de este tipo de arquitecturas se utiliza la taxonomía de Flynn que se basa en el número de instrucciones concurrentes y en los flujos de datos sobre los que operar:

- SISD (Simple Instruction Simple Data). Computador secuencial que no explota el paralelismo ni en las instrucciones ni en los flujos de datos, por ejemplo, las máquinas con monoprocesador.
- MISD (Multiple Instruction Simple Data). Poco común debido al hecho de que la efectividad de los múltiples flujos de instrucciones suele precisar de múltiples flujos de datos. Se utilizan en situaciones de paralelismo redundante, como por ejemplo en navegación aérea.
- SIMD (Simple Instruction Multiple Data). Un computador que explota varios flujos de datos dentro de un único flujo de instrucciones para realizar operaciones que pueden ser paralelizadas de forma natural. En esta clasificación entrarían los Procesadores matriciales y los Procesadores vectoriales (aplican un mismo algoritmo numérico a una serie de datos matriciales).
- MIMD (Multiple Instruction Multiple Data). Se tienen múltiples procesadores que de forma sincronizada ejecutan instrucciones sobre diferentes datos. El tipo de memoria que estos sistemas utilizan es distribuida. En esta arquitectura se engloban los sistemas distribuidos, distinguiendo aquellos que explotan un único espacio compartido de memoria (Procesadores superescalares, Multiprocesador simétrico (SMP) y Acceso no uniforme a memoria (NUMA)) de aquellos que trabajan con espacios de memoria distribuida, como los Clusters.
  - o En los sistemas SMP (Simetric Multiprocesors), varios procesadores comparten la misma memoria principal y periféricos de E/S, normalmente conectados por un bus común. Se conocen como simétricos, ya que ningún procesador toma el papel de maestro y los demás de esclavos, sino que todos tienen derechos similares en cuanto al acceso a la memoria y periféricos y ambos son administrados por el sistema operativo.
  - o Los Clusters son conjuntos de computadoras independientes conectadas en una red de área local o por un bus de interconexión y que trabajan cooperativamente para resolver un problema. Es clave en su funcionamiento contar con un

sistema operativo y programas de aplicación capaces de distribuir el trabajo entre las computadoras de la red.

Las partes físicas (hardware) que componen un ordenador se pueden esquematizar en las siguientes:

1. **PROCESADOR** también conocido como **CPU** (Central Process Unit). Se encarga de interpretar y ejecutar las instrucciones de los programas, realizando cálculos aritméticos y lógicos con los datos. También es el encargado de comunicarse con las demás partes del sistema.

Internamente está constituida por una colección compleja de circuitos electrónicos. Cuando se incorporan todos estos circuitos en un chip de silicio, a este chip se le denomina microprocesador.

La CPU está compuesta por la unidad aritmética lógica, la unidad de control y los registros del sistema:

- a. Unidad de Control (UC): La función de la unidad de control consiste en leer las instrucciones que residen en la memoria principal, interpretarlas y ejecutarlas dando las oportunas órdenes a la unidad aritmético-lógica y a los restantes elementos del sistema.
- b. Unidad Aritmética Lógica (ALU): Ejecuta las operaciones aritméticas lógicas que le señala la instrucción residente en la unidad de control.
- c. Registros del sistema: Son circuitos que sirven como área interna de trabajo. Almacenan una palabra de bits. Estos circuitos son muy rápidos y forman parte del propio procesador.

Hay que hacer mención especial a los **microprocesadores multinúcleo** que combinan dos o más procesadores independientes en un solo circuito integrado. Un dispositivo de doble núcleo contiene solamente dos microprocesadores independientes. En general, los microprocesadores multinúcleo permiten que un dispositivo computacional exhiba una cierta forma del paralelismo a nivel de subproceso, también llamado hilo o thread (thread level parallelism, TLP) sin incluir múltiples microprocesadores en paquetes físicos separados. Esta forma de TLP se conoce a menudo como multiprocesamiento a nivel de chip (chip-level multiprocessing) o CMP.

2. **MEMORIA PRINCIPAL.** Lugar donde se almacenan los datos y las instrucciones de los programas en ejecución, donde se pueden recuperar y grabar en ella datos a través de las dos operaciones básicas definidas sobre ella: lectura o escritura.

Está constituida por celdas o elementos capaces de almacenar 1 bit de información. La memoria se organiza en conjuntos de elementos de un tamaño determinado llamados *palabras de memoria*. A cada palabra le corresponde una dirección única.

Cada palabra es una unidad direccionable en la memoria. El mapa de memoria se corresponde con el espacio de memoria direccionable. Este espacio viene determinado por el tamaño de las direcciones.

3. **BUSES.** Para funcionar el hardware necesita unas conexiones que permitan a los componentes comunicarse entre sí e interaccionar. Estas conexiones se denominan buses o canales. Un bus constituye un sistema común interconectado que coordina y transporta información entre las partes del ordenador.

Un bus se caracteriza por dos propiedades:

- La cantidad de información que puede manipular simultáneamente, llamada “ancho de bus”.
- La rapidez con que puede transferir dichos datos.

Existen tres tipos de buses en un ordenador, en función del tipo de datos que transporten:

- *Bus de Control:* Se encarga de transmitir datos que serán utilizados como órdenes de control.
- *Bus de Direcciones:* Se encarga de transmitir datos que serán utilizados como direcciones de memoria.
- *Bus de Datos:* Se encarga de transportar datos como tales.

El conjunto de estos tres buses forma el **Bus del Sistema**.

**PERIFÉRICOS.** Una de las funciones básicas del computador es enviar y recibir datos desde dispositivos externos a la CPU. A estos dispositivos se les conoce con el nombre genérico de periféricos, pudiendo ser de lectura, de escritura y de lectura y escritura. A pesar de que el término periférico implica a menudo el concepto de “adicional pero no esencial”, muchos de ellos son elementos fundamentales para un sistema informático.

Los periféricos se clasifican según el uso que hagan de la información en:

- *Dispositivos periféricos de entrada.* Introducen datos e instrucciones en la CPU. por ejemplo: un ratón, un teclado.
- *Dispositivos periféricos de salida.* Permiten ver los resultados, por ejemplo: un monitor, una impresora.

- *Dispositivos periféricos de ENTRADA/SALIDA (E/S).* Tienen comunicación bidireccional con la CPU, por ejemplo, un dispositivo de almacenamiento.

Según su función tenemos las siguientes clases principales de periféricos:

#### PERIFÉRICOS DE ALMACENAMIENTO

Estos periféricos se encargan de guardar o salvar los datos de los que hace uso la CPU para que ésta pueda hacer uso de ellos una vez que han sido eliminados de la memoria principal la cuál se borra cada vez que se apaga el ordenador. Pueden ser internos, como un disco duro, o extraíbles, como un CD. Los más comunes son: Disco duro, Grabadora/lector de CD o de DVD, Memorias Flash (usb, sd, etc.), Cintas magnéticas.

#### PERIFÉRICOS DE SALIDA

Son los que reciben información es procesada por la CPU y la reproducen para que sea perceptible para el usuario. Los principales son: Monitor, Impresoras, Altavoces, Auriculares.

#### PERIFÉRICOS DE COMUNICACIONES

Su función es permitir o facilitar la interacción entre dos o más ordenadores, o entre un ordenador y otro periférico externo la mismo. Entre ellos se encuentran los siguientes: Modem, Tarjeta de red (Ethernet o Wireless), Tarjeta Bluetooth, Controladores de puertos (serie, paralelo, infrarrojo, etc.).



## **Servidores**

Como se observa en apartados anteriores, en la clasificación de los ordenadores no aparece el término servidor. Este término surge originalmente del mundo software debido a la arquitectura cliente/servidor, en la que unos programas denominados *clientes* realizan peticiones a otros programas denominados *servidores* los cuales atienden dichas peticiones realizando las acciones necesarias.

- Un **servidor** se define entonces como un programa que acepta conexiones con objeto de atender peticiones mediante el envío de respuestas.
- Un **cliente** se define como un programa que establece conexiones con el propósito de realizar peticiones.

Este uso dual puede llevar a confusión. Por ejemplo, en el caso de un servidor web, este término podría referirse a la máquina que almacena y maneja los sitios web, y en este sentido es utilizada por las compañías que ofrecen hosting u hospedaje. Alternativamente, el servidor web podría referirse al software, como el servidor de http de Apache, que funciona en la máquina y maneja la entrega de las páginas web como respuesta a peticiones de los navegadores de los clientes.

Debido a la especialización y criticidad de muchos tipos de servidores, el término “Servidor” se utiliza para referirse al ordenador (hardware) donde está instalado el programa que atiende a las peticiones.

Así, un servidor en el ámbito profesional es un ordenador específicamente diseñado para optimizar la ejecución de un determinado programa servidor o un conjunto de ellos.

Lógicamente este hardware específico necesita un sistema operativo (SO) personalizado para ejecutar programas servidores, siendo habitual que las compañías proporcionen SO para usuario final (Windows 7, Ubuntu Desktop) y SO para servidores (Windows 2008 Server R2, Ubuntu Server...).

### **Características de un servidor**

Existen factores como la fiabilidad, el rendimiento o el coste que determinan el tipo de servidor (hardware) que se requiere para albergar un software servidor. Así se puede tener en un mismo servidor hardware varios programas servidores, o bien se puede tener un servidor hardware por cada programa servidor.

Desde este punto de vista, cualquier computador que albergue un determinado software servidor podría ser considerado un servidor. Sin

embargo, las máquinas que se diseñan con el propósito de albergar programas servidores tienen una serie de características particulares que hace necesario emplear hardware especializado, orientado a una alta fiabilidad y rendimiento.

- Tienen que procesar numerosas peticiones de clientes en un tiempo corto, por eso necesitan CPUs con velocidades altas de procesamiento. Si por características de la aplicación se requiere una gran cantidad de procesamiento, es más recomendable añadir más CPUs para trabajar en paralelo, en lugar de aumentar la velocidad de una única CPU, por cuestiones de redundancia y fiabilidad.
- Si el servidor recibe peticiones concurrentemente es necesario que cuente con una cantidad de memoria principal o RAM elevada que le permita abrir threads y atender de forma adecuada a los clientes.
- Los buses por los que circula la información dentro del servidor tienen que ser de alto rendimiento para no provocar cuellos de botella.
- Algunos tipos de servidores (ficheros y bases de datos sobre todo) necesitan una tecnología de almacenamiento altamente eficiente siendo normal encontrar dos tipos de tecnologías distintas:
  - SAN (Storage Area Network). Es una red especializada que permite un acceso rápido y confiable entre servidores y recursos de almacenamiento independientes o externo. De esta forma un dispositivo de almacenamiento no es propiedad exclusiva de un servidor, sino que los dispositivos de almacenamiento son compartidos entre todos los servidores de la red como recursos individuales. Esta arquitectura implica disponer de una infraestructura de red de alta velocidad dedicada sólo para Almacenamiento y Backup, optimizada para mover grandes cantidades de datos, y consistente en múltiples recursos de almacenamiento geográficamente distribuidos.
  - NAS (Network Attached Storage). Los dispositivos NAS son dispositivos de almacenamiento a los que se accede a través de protocolos de red.

Los dispositivos NAS utilizan usualmente más de un dispositivo de almacenamiento, en la mayoría de los casos están compuestos por RAIDs (Redundant Arrays of Independent Disks) de discos lo que aumenta la capacidad de almacenamiento, la seguridad, y la velocidad de acceso a la información.

- Los servidores están preparados para ofrecer servicios con un grado de disponibilidad de más del 99%. Esto implica:
  - Que está encendido las 24 horas del día, con lo que es necesario un sistema de refrigeración adecuado. Para ello se

ubican en Centros de Procesos de Datos donde existe la temperatura y humedad óptimas de funcionamiento.

- o Que tienen que contar con Sistemas de Alimentación Ininterrumpida para evitar que un corte eléctrico los deje indisponibles.
- o Que es necesario utilizar componentes **hot swap**, son componentes que se pueden sustituir “en caliente sin parar el servidor”. Esto tiene especial importancia con servidores críticos que no pueden estar parados por una acción planificada. Los componentes **hot swap** más comunes son:
  - Los discos duros configurados en RAID
  - Las fuentes de alimentación
- Los servidores pueden estar ubicados en armarios RACKs o no. La configuración de servidores en RACKs es modular permitiendo agregar o quitar componentes con más facilidad (añadir una cabina de cintas de backup, una nueva fuente de alimentación o un nuevo servidor).

### **Cluster**

Un clúster es un tipo de computador distribuido o paralelo que consiste en un grupo de computadoras interconectadas que trabajan conjuntamente en la solución de un problema. Estos sistemas constituyen una solución flexible, de bajo coste y de gran escalabilidad para aplicaciones que requieren una elevada capacidad de cómputo y memoria.

La tecnología de clústeres ha evolucionado en apoyo de actividades que van desde aplicaciones de supercómputo y software de misiones críticas, servidores Web y comercio electrónico, hasta bases de datos de alto rendimiento, entre otros usos.

Los clústeres ofrecen las siguientes características:

- *Alto rendimiento*: diseñados para dar altas prestaciones en cuanto a capacidad de cálculo y velocidad de proceso.
- *Alta disponibilidad*: diseñados para garantizar la total y absoluta disponibilidad del servicio en el tiempo ofreciendo un funcionamiento ininterrumpido. Todas las máquinas de este clúster están sincronizadas y monitorizadas entre sí. Si se produce un fallo en alguna de las máquinas del clúster, se detecta dicho fallo automáticamente y las otras máquinas asumen las funciones y siguen funcionando manteniendo así la disponibilidad del sistema el software. Son tolerantes a fallos.
- *Balanceo de carga*: Un clúster estará compuesto por uno o más nodos que actúan como *frontend* del clúster, y que se ocupan de repartir las

peticiones de servicio que reciba el clúster a otros ordenadores del clúster que forman el *backend* de éste, evitando así los cuellos de botella.

- *Escalabilidad*: Es relativamente asequible aumentar un nodo en un sistema clúster.

Un clúster de servidores tiene principalmente dos ventajas considerables sobre las soluciones de servidores estándar:

- Garantizan la alta disponibilidad de servicios y datos.
- Permite aprovechar al 100% la capacidad de los nodos introducidos (no hay nodos en stand-by).

### **Clases de clústeres**

La forma en que operará el clúster está determinada por la función que éste deberá desempeñar:

- Clúster de Alto Rendimiento: diseñado para dar altas prestaciones en cuanto a capacidad de cálculo. Existen distintas aplicaciones que se les puede dar a este tipo de clúster, entre las cuales encontramos: cálculos matemáticos, renderizaciones de gráficos, compilación de programas, descifrado de códigos.
- Clúster de Alta Disponibilidad: están diseñados para garantizar el funcionamiento ininterrumpido de ciertas aplicaciones. La idea principal de este tipo de clúster es proporcionar un servicio ininterrumpido las 24 horas del día, los 7 días de la semana.

Están formados por un conjunto de dos o más máquinas que comparten los discos de almacenamiento de datos, y se monitorizan mutuamente. Si se produce un fallo del hardware o de las aplicaciones de alguna de las máquinas del clúster, el software de Alta Disponibilidad es capaz de rearrancar automáticamente los servicios que han fallado en cualquiera de las otras máquinas del clúster. Y cuando la máquina que ha fallado se recupera, los servicios son nuevamente migrados a la máquina original.

- Clúster de Alta Eficiencia: Son clústeres cuyo objetivo de diseño es el ejecutar la mayor cantidad de tareas en el menor tiempo posible. Existe independencia de datos entre las tareas individuales.

Los clúster de alta eficiencia y alta disponibilidad suelen utilizarse para entornos empresariales y esta funcionalidad solamente puede ser efectuada por hardware especializado, mientras que los clúster de alto rendimiento son propios de universidades y centros de cálculo.

### **Componentes de un clúster**

Para que un clúster funcione como tal, no basta sólo con conectar entre sí los ordenadores, sino que es necesario proveerlos de un sistema de manejo del clúster, el cual se encargue de interactuar con el usuario y los procesos que corren en él para optimizar el funcionamiento. Es decir que, para poder funcionar, requiere tantos componentes hardware como software.

- **Nodos.** Son los ordenadores en sí mismos, existiendo ordenadores personales, sistemas multi-procesador o estaciones de trabajo (workstations). Pueden ser:
  - *Dedicados:* su uso está exclusivamente dedicado a realizar tareas relacionadas con el clúster.
  - *No dedicados:* su uso no está exclusivamente dedicado a realizar tareas relacionadas con el clúster, utilizándose los ciclos de reloj del computador cuando éste no se utiliza.
- **Almacenamiento.** Puede consistir en una NAS, una SAN, o almacenamiento interno en el servidor. El protocolo más comúnmente utilizado es NFS (Network File System), sistema de ficheros compartido entre servidor y los nodos. Sin embargo existen sistemas de ficheros específicos para clústeres como Lustre (CFS) y PVFS2.
- **Red de interconexión.** Se utilizan Redes de Alta Velocidad como solución de alto rendimiento para que las comunicaciones no sean el cuello de botella del rendimiento del sistema.

Las redes de interconexión son un componente fundamental de los clústeres que proporcionan: alto ancho de banda, baja latencia, fiabilidad y escalabilidad.

Las redes de interconexión comunes en clúster son:

- Ethernet: Estándar de redes de computadoras de área local con acceso al medio por contienda CSMA/CD.
- Fast Ethernet: Serie de estándares de IEEE de redes Ethernet de 100 Mbps (megabits por segundo).
- Gigabit Ethernet: Ampliación del estándar Ethernet que consigue una capacidad de transmisión de 1 gigabit por segundo.
- SCI (Scalable Coherent Interface): Estándar de interconexión de redes de alta velocidad utilizado para multi-procesamiento con memoria compartida y paso de mensajes.
- ATM (Asynchronous Transfer Mode): Tecnología de telecomunicación desarrollada para hacer frente a la gran demanda de capacidad de transmisión para servicios y aplicaciones.

- o Myrinet: Red de interconexión de clústeres de altas prestaciones. El procesamiento de las comunicaciones de red se hace a través de chips integrados en las tarjetas de red de Myrinet (Lanai chips), descargando a la CPU de gran parte del procesamiento de las comunicaciones.
- o HIPPI (High Performance Parallel Interface): Bus para conexiones de alta velocidad para dispositivos de almacenamiento en superordenadores. Ha sido sustituido progresivamente por otras tecnologías más rápidas.
- o FiberChannel: Tecnología de red utilizada principalmente para redes de almacenamiento, disponible primero a la velocidad de 1 Gbps y posteriormente a 2, 4 y 8 Gbps.
- o Infiniband: Es una red surgida de un estándar desarrollado específicamente para realizar la comunicación en clústeres. La conexión básica es de 2Gbps efectivos y se podrían alcanzar los 96Gbps.
- **Sistema Operativo.** Tiene que ser multiproceso y multiusuario.
- **Middleware.** Actúa entre el sistema operativo y las aplicaciones, recibiendo los trabajos entrantes al clúster y redistribuyéndolos de manera que el proceso se ejecute más rápido y el sistema no sufra sobrecargas en un servidor determinado. Está compuesto de dos subniveles de software:
  - o *SSI (Single System Image):* ofrece a los usuarios un acceso unificado a todos los recursos del sistema.
  - o *Disponibilidad del sistema:* que permite servicios como puntos de chequeo, recuperación de fallos, soporte para tolerancia a fallos.

### **Servidores Blade**

Blade Server es una arquitectura que ha conseguido integrar en tarjetas todos los elementos típicos de un servidor. Cada servidor blade es una tarjeta (llamada *blades*) que contiene la memoria RAM, el disco duro y la CPU. Los servidores blade en tarjetas se insertan en un chasis que se coloca en un rack estándar ocupando entre 4U y 6U dentro del rack, permitiendo albergar un máximo de 16 servidores blade en un chasis. Este chasis, a su vez integra y permite compartir los elementos comunes como son:

- La ventilación y la refrigeración.
- Los switches de red redundante con el cableado.
- Las fuente de alimentación y el SAI tipo hot swap.

- Interfaces de almacenamiento.

Al estar todo integrado en el chasis se consigue reducir el consumo eléctrico, cableado, sistemas de enfriamiento y el espacio dentro del rack.

Las empresas que requieren de la actualización de sus sistemas se enfrentan al problema de consumo eléctrico, espacio, control de temperatura y ubicación de los nuevos equipos. Tradicionalmente, hasta la llegada de los servidores Blade, el método para incrementar nuevos requerimientos era agregar más servidores en rack, lo que ocupa más espacio, complica el cableado, hace más compleja la gestión de administración de los sistemas, consume más recursos, etc.

La tecnología blade supone un diseño más eficiente en cuanto a coste y espacio. Para ello se ha reducido el chasis, se ha bajado el consumo, simplificado el cableado y el mantenimiento, mientras se incrementan las funcionalidades.

Estos son los principios básicos en los que se fundamenta la arquitectura blade y que al final proporciona una reducción del coste total.

### **Diferencias entre un sistema de servidores montados en rack y blade server**

La principal diferencia es que en un sistema montado en rack, el servidor es una unidad completa en sí mismo. Esto significa que contiene la CPU, memoria, fuente de alimentación, ventiladores y disipadores. Estos servidores son atornillados en el rack, y cada uno es conectado a la red corporativa usando un cable separado.

Los blade servers son una versión compacta de sistemas montados en rack. El blade incluye una CPU, memoria y dispositivos para almacenar datos. Pero no tiene fuente de alimentación eléctrica ni ventiladores. Los blades son insertados en slots y enlazados entre sí gracias a un bus de alta velocidad dentro del chasis.

### **Ventajas**

- Reduce la gestión gracias a su infraestructura simplificada.
- Comparte fuentes de alimentación y ventiladores y una gestión del sistema centralizada disminuyendo costes porque requieren menos electrónica y consumen menos energía.
- El chasis elimina la mayoría del cableado que se encuentra en los sistemas montados en rack.

- Intercambio en caliente (Hot-Swap): si un blade falla puede ser reemplazado sin ningún impacto en los otros blades.
- Facilitan la gestión y reducen tiempo y coste administrativo al estar todos los servidores en un sólo equipo.
- Se reduce el espacio al integrar en un sólo chasis muchos servidores, sin reducir poder de cómputo.
- Escalabilidad horizontal: porque nos ofrece ampliar el número de servidores fácilmente a medida que va creciendo la demanda.
- Alta disponibilidad, pues la mayoría de los equipos poseen elementos redundantes que garantizan el funcionamiento continuado de los servidores sin interrupciones.



### ***Puesto de trabajo.***

Lo más habitual en cualquier empresa es que en el puesto de trabajo exista un microcomputador, es decir, o bien un PC o una estación de trabajo (en inglés workstation). El concepto de PC u ordenador personal es ampliamente aceptado.

Una Workstation es un microordenador de altas prestaciones especialmente diseñado para niveles de alto rendimiento en ciertas tareas, como pueden ser, diseño gráfico, edición de video, gestión de redes de Internet, aplicaciones de alto consumo, etc. Estos potentes ordenadores han encontrado su sitio en la ingeniería y desarrollo de software entre otras cosas, debido a su habilidad multitarea.

En la actualidad los PCs son bastantes potentes en cuanto a memoria y capacidad de procesamiento. Sin embargo, el hardware de las estaciones de trabajo está optimizado para situaciones que requieren un alto rendimiento y fiabilidad, mucha cantidad de memoria, computación de multitarea, etc. donde generalmente se mantienen operativas en situaciones en las cuales cualquier computadora personal tradicional dejaría rápidamente de responder.

Los profesionales cuando escuchan la palabra estación de trabajo, piensan que es una máquina que no necesitan y que tiene un coste muy superior a las expectativas. La realidad es que eso ha cambiado, especialmente en todo lo relativo al factor precio, y ahora, con una inversión mínima, un profesional puede, gracias a una estación de trabajo, obtener hasta un 50 por ciento más de rendimiento en sus tareas diarias.

Existen profesionales que se compran un PC potente, con más de 2 GB de memoria, con una tarjeta gráfica de alto nivel, con alta capacidad de memoria interna, etc. porque necesitan trabajar con aplicaciones de software. Lo hacen porque no conocen la existencia de las estaciones de trabajo pero, sobre todo, porque no saben las diferencias que tienen con un PC y las funcionalidades y ventajas que pueden ofrecerle. Un PC, por ejemplo, en el apartado de memoria, llega hasta donde llega y ahí surgen los problemas. Existen estaciones de trabajo que puede alcanzar los 128 GB de memoria, cinco discos duros, biprocesadores, etc.

Las principales aplicaciones de una Workstation son:

- CAD (Computer Aided Design, Diseño Asistido por Ordenador): destinadas al diseño y análisis de sistemas de ingeniería y arquitectura.
- AEC (Architecture Engineering Construction): aplicables a la edición de planos de construcción y arquitectura, elaboración de presupuestos y seguimientos de obras.
- CAM (Computer Aided Manufacturing): aplicables en el diseño, análisis y prueba de circuitos integrados, tarjetas y otros sistemas electrónicos.
- CASE (Computer Aided Software Engineering): ayuda a la gestión completa de los ciclos de vida de los desarrollos de aplicaciones lógicas.
- GIS (Geographic Information System): para edición, captura y análisis de información sobre determinadas zonas geográficas, con base en referencias de mapas digitalizados.
- Sistemas expertos: basados en técnicas de programación de inteligencia artificial, para aplicaciones tales como detección electrónica de errores, funciones de diagnóstico o configuración de ordenadores.
- Aplicaciones empresariales: investigación cuantitativa, seguridad, simulación de análisis reales...
- Edición electrónica: creación para su posterior publicación de periódicos, revistas, presentaciones y documentación en general.
- Telecomunicaciones: gestión de redes, desarrollo de aplicaciones de telecomunicaciones basadas en inteligencia artificial, aplicaciones de apoyo a la investigación y desarrollo (I+D), edición electrónica y procesado de imágenes.
- Las estaciones de trabajo también pueden ser utilizadas como pasarelas (gateways), para acceder a grandes ordenadores, y para ejecutar remotamente utilizando protocolos de comunicaciones.

## ***Dispositivos personales***

Por dispositivos personales, entendemos aquellos dispositivos con una relativa capacidad de cálculo y que por su tamaño pueden ser transportados de forma sencilla por el usuario y lo que es más importante, ser utilizados fuera de un entorno regulado de trabajo como puede ser la oficina. Los tres dispositivos más habituales hoy en día son: PDA's, Tablet's y Smartphones.

### ***PDA***

Una PDA (Personal Digital Assistant) es un ordenador de bolsillo diseñado como una agenda electrónica, pero que actualmente poseen una potencia razonable y son capaces de realizar numerosas funciones más allá de las de mera agenda electrónica constituyéndose como una extensión misma del ordenador personal, que podremos sincronizar con éste.

Otros términos asociados son palmtop y handhelds. Un palmtop es un ordenador pequeño que literalmente coge en la palma de la mano. Un handheld es un ordenador sumamente pequeño que se puede sostener con la mano.

Los términos PDA, palmtop y handhelds surgieron para cubrir necesidades diferentes. Actualmente la división entre ambas es muy difusa; ambos términos se utilizan indistintamente.

Las tecnologías de comunicaciones inalámbricas (Bluetooth, Wi-Fi, IrDA (infrarrojos), GPS...) permiten que con una PDA se pueda consultar el correo electrónico, usarlos como navegador GPS o para temas relativos a la domótica.

Pero más allá de las funciones y software con las que viene equipado la PDA, lo que lo hace verdaderamente potente es la posibilidad de personalización casi ilimitada al permitir cargar las aplicaciones "bajo demanda".

### **Características:**

- Tienen un tamaño físico muy reducido para que quepa en la mano.
- Son bastantes ligeros para que sea fácil su transporte en un bolsillo.
- La pantalla es táctil ocupando gran parte del dispositivo y dejando poco espacio para ubicar botones hardware. No suelen disponer de un teclado con botones (salvo algunos dispositivos) por lo que para

agregar texto se utiliza un teclado virtual o se le añade un teclado externo por USB.

- Tienen capacidad multimedia, ya que integran altavoz, micrófono y grabadora de voz.
- Disponen de conexión de periféricos: para dispositivos de almacenamiento externo y para módulos de expansión.
- Amplio soporte de conexiones inalámbricas: Bluetooth, infrarrojos, Wi-fi.
- Funcionamiento con baterías de Litio-ion.
- Capacidad de almacenamiento por encima de los 64 MB que se puede ampliar mediante el uso de tarjetas de memoria Flash.
- La sincronización con los ordenadores personales permite la actualización del directorio, haciendo que la información del computador y de la PDA sea la misma. La sincronización también evita la pérdida de la información almacenada en caso de que el accesorio se pierda, sea robado o destruido.
- Utilizan sistemas operativos específicos como son Windows Mobile, HP webOS, Linux.

### **Limitaciones:**

- Potencia de computación reducida, debido a que los microprocesadores tienen que tener en cuenta la duración de las baterías, el sobrecalentamiento, etc.
- Capacidad de almacenamiento, aunque hoy en día con tarjetas de memoria de varios GB es una limitación menor.
- Baja duración de las baterías.
- Comunicaciones.
- Software específico.

## **TABLET**

El Tablet es un ordenador portátil de tamaño reducido, con pantalla sobre la cual el usuario puede escribir usando un lápiz especial (el stylus). El texto manuscrito es digitalizado mediante reconocimiento de escritura. El lápiz también se utiliza para moverse dentro del sistema y utilizar las herramientas y funciones de las Tablet

El Tablet combina la potencia de un ordenador portátil con la comodidad de un PDA.

En función de si disponen o no de teclado se distinguen:

- Tablet “Slate”: no dispone de teclado y es necesario utilizar un lápiz o los dedos para manipularlo.
- Tablet “Convertible”: posee un teclado. Puede ser deslizable para poder deslizarse debajo de la pantalla o de modo que la pantalla pueda girar.

### **Características:**

- Los microprocesadores empleados en estos dispositivos están basados en soluciones para móvil.
- Para el almacenamiento se suelen utilizar discos EIDE convencionales pero de 2,5” (más finos).
- La memoria que suelen utilizar es SODIMM (small online DIMM), especiales para laptop e impresoras.
- Pantalla táctil.
- Nuevas formas de control mediante voz y escritura manual.

### **Smartphones**

Estos dispositivos hacen las funciones de un teléfono móvil convencional, pero están dotados de una mayor versatilidad, ya que también incluyen algunas de las funciones de un ordenador personal. En la actualidad todos ellos tienen en común un conjunto amplio de características, como una pantalla táctil de gran formato, conectividad WiFi, Bluetooth, 3G... No obstante, existen otros importantes parámetros que conviene tener en consideración:

### **Pantalla**

Es un componente de extrema importancia debido a que las pantallas táctiles de los smartphones son la interfaz directa de comunicación entre el usuario y el propio dispositivo.

Existen dos tecnologías aplicables a estas superficies táctiles:

- Las capacitivas: es la más adecuada para facilitar la interacción directa con el dedo en lugar de los habituales stylus, ya que para que respondan al instante basta con deslizarlo, por lo que el usuario no necesita ejercer ningún tipo de presión sobre la superficie. Además, pueden detectar varias pulsaciones de manera simultánea, por lo que la experiencia para el usuario es más atractiva que en el caso de las resistivas.
- Las resistivas: están formadas por varias capas, por lo que cuando las presionamos entran en contacto. Esto produce un cambio de

corriente, facilitando, de este modo, la detección de la pulsación. Por esta razón, la experiencia de usuario en este caso parece ser menos atractiva que en el anterior, ya que la respuesta del dispositivo es algo más lenta, o al menos es la sensación que puede brindarnos.

## **Sistema operativo**

El funcionamiento de un S.O. afecta directamente al rendimiento del dispositivo, la usabilidad de su interfaz y las funcionalidades que ponen a disposición de los usuarios.

Actualmente el S.O. que se implanta en un Smartphone ha adoptado tanta trascendencia como el equipo mismo. A tal punto que se habla, de “Smartphones Android”, para referirse a los teléfonos que funcionan a través de este desarrollo de Google. Por lo tanto, la elección del sistema es casi tan importante como la de un smartphone en sí. Hay que tener en cuenta que además estos S.O. también se utilizan en los tablets, por ejemplo el iOS de Apple se encuentra en su smartphone iPhone y en su tablet iPad, el HP webOS se implanta en los smartphones PalmPre y en su tablet TouchPad, etc. A continuación se exponen los S.O. más relevantes en el mercado:

- **HP webOS** es un sistema operativo multitarea basado en Linux, desarrollado por Palm, Inc., ahora propiedad de Hewlett-Packard Company. Cabe destacar que usa tecnologías web como HTML5, JavaScript y CSS y soporta Flash.

webOS incluye una característica llamada "Synergy" que permite conectar el sistema con numerosos servicios de redes sociales e integrar información de varias fuentes.

webOS hace uso del cloud computing para la sincronización de datos

- **Android**. es un sistema operativo multitarea basado en Linux no sólo en su núcleo, sino también en su concepto: de código abierto y gratuito. Esto significa que cualquier fabricante que desee podrá instalar Android en sus equipos posibilitando que el sistema esté disponible en una amplia gama de smartphones. Fue diseñado originalmente para dispositivos móviles, tales como teléfonos inteligentes, tablets, pero que actualmente se encuentra en desarrollo para usarse en netbooks y PC.

El *Android Market* es un catálogo de aplicaciones que pueden ser descargadas e instaladas en dispositivos Android sin la necesidad de un PC

- **BlackBerry OS** es un sistema operativo móvil desarrollado por Research In Motion para sus dispositivos BlackBerry. Al comienzo de su andadura los BlackBerry estuvieron orientados al público corporativo, pero tras la aparición del iPhone se abrió al uso personal

(al igual que muchos smartphones). La interfaz más cómoda para usar un BlackBerry es el teclado físico, que no es sólo un accesorio como en otros smartphones sino que es la llave para acceder a todas las funcionalidades.

- **Windows Phone** es un sistema operativo móvil desarrollado por Microsoft, como sucesor de la plataforma Windows Mobile.[2] Está pensado para el mercado de consumo generalista en lugar del mercado empresarial[3] por lo que carece de muchas funcionalidades que proporciona la versión anterior

El Hub Marketplace es el lugar en el que se pueden comprar y descargar todo tipo de contenido como aplicaciones, música, películas, programas de TV, podcast.

- **iOS** es el sistema operativo móvil de Apple desarrollado originalmente para el iPhone, siendo después usado en el iPod Touch e iPad. Se dice que es un SO que marca tendencias. En la última versión del SO (iOS 4) se soporta la multitarea. Uno de los aspectos más criticados es su falta de soporte para Flash.

La interfaz de usuario de iOS se basa en el concepto de manipulación mediante gestos multitáctil. Los elementos de la interfaz se componen por deslizadores, interruptores y botones. La respuesta es inmediata y se provee de una interfaz fluida. La interacción con el sistema operativo se realiza mediante gestos como deslizar, tocar y pellizca

El App Store de Apple es donde se pueden comprar y descargar contenidos. Fue pionera en ese aspecto.

La carga de las aplicaciones se realiza casi instantáneamente, brindando fluidez al desempeño general del teléfono.

- Otros sistemas operativos: Bada, Meego, Symbian, etc.

## **BIBLIOGRAFIA**

- John L. Hennessy, David A. Patterson Computer architecture: a quantitative approach, Elsevier, Morgan Kaufmann, 2007.
- Carl Hamacher, Zvonko Vranesic and Safwat Zaky. Organización de Computadores, 5ª edición. Ed. McGraw Hill, 2002.
- Fundamentos de sistemas de información. Madrid: Prentice Hall. Edwards, C.; Ward, J.; Bytheway, A. (1998).
- Essentials of Management Information Systems. Organisation and Technology. Englewoods Cliffs: Prentice Hall. Laudon, K.C.; Laudon, J.P. (2002).
- Administración de los Sistemas de Información. Prentice-Hall. Laudon, K.C. y Laudon, J.P. (2002)
- PCWORLD Marzo 2010.
- <http://es.wikipedia.com>

**Autor:** Francisco Javier Rodríguez Martínez

Subdirector de Sistemas Escola Superior Enxeñaría Informática Ourense

Colegiado del CPEIG



## **20. ESTRUCTURAS DE LA INFORMACIÓN. MODELO ENTIDADE-RELACIÓN. ENTIDADES Y ATRIBUTOS. RELACIONES. DIAGRAMAS DE ENTIDAD-RELACIÓN.**

## **Tema 20. Estruturas da información. Modelo entidade-relación. Entidades e atributos. Relacións. Diagramas de entidade- relación.**

---

### **ÍNDICE**

<b>20.1 Estructuras de la información.....</b>	<b>2</b>
<b>20.2 Modelo Entidad - Relación (E-R).....</b>	<b>4</b>
<i>20.2.1 Modelo Entidad Relación Extendido.....</i>	<i>4</i>
20.1.1.1 Cardinalidad.....	4
20.2.1.1 Jerarquía Subconjunto.....	4
20.2.1.2 Generalización.....	4
20.2.1.3 Tipos de relaciones.....	5
20.2.1.4 Control de redundancias.....	6
20.2.1.5 Dimensión temporal.....	6
<b>20.3 Entidades y Atributos.....</b>	<b>7</b>
20.3.1 Entidad.....	7
20.3.2 Relación o Interrelación.....	7
20.3.3 Dominio.....	8
20.3.4 Atributos.....	8
<b>20.4 Relaciones.....</b>	<b>9</b>
20.4.1 Elementos.....	9
20.4.2 Restricciones.....	10
20.4.2.1 Restricciones inherentes.....	10
20.4.2.2 Restricciones semánticas.....	10
<b>20.5 Diagramas entidad-Relación.....</b>	<b>12</b>

<i>20.5.1Tipo de correspondencia.....</i>	<i>12</i>
<i>20.5.2Entidades Débiles.....</i>	<i>12</i>
<i>20.5.3Papel o rol.....</i>	<i>13</i>
<i>20.5.4Atributos multivaluados y compuestos.....</i>	<i>13</i>
<i>20.5.5Atributos Derivados.....</i>	<i>13</i>
<b>20.6Bibliografía.....</b>	<b>14</b>

## **20.1 ESTRUCTURAS DE LA INFORMACIÓN**

Se llama datos al conjunto de propiedades que caracterizan un fenómeno, e información al conjunto de valores que pueden tomar estas propiedades junto con las relaciones o dependencias entre las mismas.

Los modelos de datos son herramientas de abstracción que permiten representar la realidad captando las restricciones semánticas que en ella se puedan dar.

Cuando en el mundo real se da información de cualquier suceso u objeto siempre los datos suministrados van acompañados de una semántica o de un significado. De la misma manera estos datos están sujetos a unas restricciones y nosotros entendemos los datos suministrados sólo si entendemos el dominio y las restricciones de significados que acompañan a la información. Sin embargo, cuando aparecen los ordenadores y las bases de datos se empiezan a informatizar ocurre que se tiende a almacenar datos separando a estos de su interpretación, esto es, de su semántica. Como consecuencia fue necesaria la aparición de los modelos de datos como una herramienta que ayudara a incorporar significado a los datos almacenados.

De esta manera, los modelos de datos proporcionan mecanismos de abstracción que permiten la representación de la parte del mundo real que nos interesa registrar. Esta representación se concibe en dos niveles: el de las estructuras que hacen posible la representación de la información, y el de la información en sí misma.

Esto nos lleva a diferenciar entre lo que se denomina el esquema de la base de datos (descripción específica en términos de un modelo de datos) y la colección de datos en sí misma que es lo que denominamos base de datos.

El primer paso en la representación de un problema del mundo real es la caracterización del mismo, o lo que es lo mismo, la determinación mediante un proceso de simplificación de los datos de interés (de entre

todos los que intervienen en el problema) y sus límites (universo del discurso).

Los modelos de datos ofrecen distintos niveles de abstracción que facilitan la representación de los datos:

1. **Clasificación.** Es la acción de crear una categoría a partir de las características comunes a un conjunto de ejemplares. Por ejemplo, a partir de los elementos Pedro, Juan y Cristina podemos crear la categoría profesor de instituto.

Su proceso inverso es la particularización.

2. **Agregación.** Es la capacidad de considerar un objeto en base a los elementos que lo constituyen. Por ejemplo, podemos crear la clase coche a partir de las clases volante, ruedas, motor y carrocería.

Su proceso inverso es la desagregación.

3. **Generalización.** Es similar a la clasificación, pero creando una categoría a partir de las características comunes a un conjunto de otras categorías. Por ejemplo, a partir de las categorías profesor de matemáticas, profesor de física y profesor de informática, podemos crear la categoría profesor de instituto.

Su proceso inverso es la especialización.

Según el nivel de abstracción que apliquemos, podemos hablar de tres tipos de modelos de datos:

- **Modelo conceptual.** Describe los tipos o clases de objetos desde un punto de vista estructural. Para cada uno de estos tipos de objetos describe sus propiedades y el dominio y restricciones de cada una, así como las relaciones entre ellos. (Modelo Entidad/Relación).
- **Modelo lógico.** Representa el problema bajo las restricciones específicas del tipo de Sistema Gestor de Base de Datos (SGBD) que se aplique en cada caso específico. (Modelo Relacional para el caso de los SGBD Relacionales).

- **Modelo físico.** Representa el problema desde el punto de vista de su implementación en el sistema de tratamiento utilizado y los métodos y mecanismos que se van a usar en su almacenamiento.

Un modelo de datos define las reglas mediante las cuales se han de estructurar los datos del mundo real.

La representación de un mundo real mediante un determinado modelo da lugar a un esquema, el cual describe las categorías existentes. Sin embargo, la realidad contempla además de los aspectos estáticos, los aspectos dinámicos. Por tanto las propiedades del mundo real son de dos tipos:

- **Estáticas:** relativamente invariantes en el tiempo, que es lo que se suele conocer como estructuras. Este tipo de propiedades está compuesto por:

- o Elementos permitidos como: los objetos (entidades, relaciones, registros, ...), asociaciones entre objetos, propiedades de los objetos y asociaciones (atributos, elementos de datos, ...), dominios (conjuntos de valores que pueden tomar las propiedades).

- o Elementos no permitidos o restricciones, puesto que no todos los valores, cambios de valor o estructuras están permitidos en el mundo real. Cada modelo tiene por sí mismo limitaciones en cuanto a las estructuras que permite:

- Las restricciones impuestas por el modelo se conocen como restricciones inherentes

- Las restricciones que permiten capturar la semántica del universo de discurso que se quiere modelar y verificar la corrección de los datos almacenados en la base de datos. Estas últimas restricciones se conocen como restricciones de integridad o semánticas.

- Las restricciones de integridad son impuestas por el usuario, mientras que las restricciones inherentes al modelo son impuestas directamente por el modelo.

- **Dinámicas:** son las operaciones que se aplican a los datos o valores almacenados en las estructuras, los cuales varían a lo largo del tiempo al aplicarles dichas operaciones. La aplicación de cualquier operación sobre los valores de los elementos, debe dejar a estos con un estado válido, es decir los valores de los elementos deben pertenecer a alguna de las categorías definidas en el esquema y deben cumplir las restricciones de integridad.

La componente estática de un modelo de datos se define a través del lenguaje de definición de datos (DDL) y la componente dinámica se define a través del lenguaje de manipulación de datos (DML), constituyendo ambas componentes el lenguaje de datos. También se puede mencionar el lenguaje de control de datos (DCL) que añade una capa de seguridad.

## **20.2            *MODELO ENTIDAD - RELACIÓN (E-R)***

Propuesto por Peter Chen en dos artículos (1976 y 1977). Es un modelo muy extendido que ha experimentado una serie de ampliaciones a lo largo de los años.

El modelo se apoya en dos conceptos, el concepto de entidad y el concepto de relación. Este modelo de datos permite representar casi cualquier restricción del diseño de datos.

El modelo E/R percibe el mundo real como una serie de objetos relacionados entre sí y pretende representarlos gráficamente mediante un mecanismo de abstracción. Este mecanismo de abstracción está basado en una serie de símbolos, reglas y métodos que permitirán representar los datos de interés del mundo real, ofreciendo al diseñador una herramienta para aislar al modelo de consideraciones relativas a la máquina y a los usuarios.

### **20.2.1            *Modelo Entidad Relación Extendido***

El Modelo E/R con el paso del tiempo ha sufrido una serie de modificaciones tanto en su simbolismo gráfico, como en la ampliación de sus elementos.

#### **20.1.1.1                                    *Cardinalidad***

Este primer concepto en cierto modo estaba tratado de forma implícita en el Modelo E/R original. Sin embargo, ha sido posteriormente cuando se le ha dado cierta relevancia e incluso una forma de representación.

El concepto cardinalidad, también denominado “clase de pertenencia”, permite especificar si todas las ocurrencias de una entidad participan o no en la interrelación establecida con otra(s) entidad(es):

- Si toda ocurrencia de la entidad A debe estar asociada con al menos una ocurrencia de la entidad B a la que está asociada por una determinada interrelación, se dice que la clase de pertenencia es obligatoria, es decir, la cardinalidad mínima es 1.



- Por el contrario, si no toda ocurrencia de la entidad A necesita estar asociada con alguna ocurrencia de la entidad B asociada, se dice que la clase de pertenencia es opcional, es decir, la cardinalidad mínima es 0.

Podemos definir la **Cardinalidad de un tipo de Entidad** como el número mínimo y máximo de ocurrencias de un tipo de entidad que pueden estar relacionadas con una ocurrencia del otro tipo de entidad que participan en el tipo de interrelación.

Surepresentación gráfica es una etiqueta del tipo (0,1), (1,1), (0,n) ó (1,n) según corresponda, al lado de las entidades asociadas por la relación tal como se puede observar en el siguiente ejemplo, donde el primer elemento de la tupla es la cardinalidad mínima, y el segundo elemento de la tupla es la cardinalidad máxima, que coincide con el tipo de correspondencia

Ejemplo: 'Un libro puede estar escrito por ninguno, uno o varios autores. Un autor escribe al menos un libro y puede escribir varios.'

### **20.2.1.1**

### **Jerarquía Subconjunto**

La descomposición de tipos de entidad en varios subtipos es una necesidad muy habitual en el modelado conceptual. En el mundo real se pueden identificar varias jerarquías de entidades. La interrelación que se establece entre un supertipo y sus subtipos corresponde a la noción de "ES-UN" (IS-A) o más exactamente "es un tipo de".

Para su representación se utiliza un triángulo invertido, con la base paralela al rectángulo que representa el supertipo.

El concepto jerarquía Subconjunto establece que una entidad A es un subconjunto de otra entidad B cuando toda ocurrencia de la primera también es una ocurrencia de la segunda, y lo contrario no tiene por qué ser cierto.

Por tanto, tendremos una jerarquía subconjunto cuando cada ocurrencia de una entidad genérica pueda ser también una ocurrencia de otras entidades que, potencialmente, son subconjuntos no disjuntos (solapados). Es decir, en las entidades subconjunto pueden aparecer ocurrencias repetidas.

#### Características:

- Toda ocurrencia de un subtipo es una ocurrencia del supertipo, las cardinalidades serán siempre (1,1) en el supertipo y (0,1) o (1,1) en los subtipos.
- Todo atributo del supertipo pasa a ser un atributo de los subtipos.

La entidad subconjunto puede tener atributos, además de tener los atributos de la entidad genérica, pero siempre las entidades subconjunto se encuentran identificadas por la clave de la entidad genérica. Además todos los atributos comunes de las entidades subconjunto deberían aparecer en la entidad genérica para evitar repetir los atributos en cada una de las entidades subconjunto.

#### **20.2.1.2**

#### **Generalización**

El concepto de jerarquía de generalización o generalización establece que una entidad genérica X es una generalización de otras entidades especializadas si cada ocurrencia de la primera es una ocurrencia y solamente una de las otras entidades. A veces este concepto se conoce también como jerarquía de especialización.

Se tendrá una jerarquía de generalización cuando la entidad genérica se divida en una serie de entidades en función del valor que tome un determinado atributo de la entidad genérica.

La generalización tiene dos restricciones semánticas asociadas:

- **Totalidad** si todo ejemplar del supertipo tiene que pertenecer a algún subtipo. El caso contrario se llama **Parcialidad**.
- **Solapamiento** si un mismo ejemplar del supertipo puede pertenecer a más de un subtipo. El caso contrario se llama **Exclusividad**.

Pueden existir interrelaciones de cada una de las cuatro combinaciones posibles, y se representarían de la siguiente forma:

#### TOTAL SIN SOLAPAMIENTO

Tanto un hombre como una mujer son persona.

Una persona no puede ser a la vez hombre y mujer.

Toda persona tiene que ser un hombre o una mujer.

#### PARCIAL SIN SOLAPAMIENTO

Tanto un artículo como un libro son documentos.

Un mismo documento no puede ser a la vez un artículo y un libro.

Puede haber documentos que no sean ni artículos ni libros.

#### TOTAL CON SOLAPAMIENTO

Tanto un empleado como un estudiante son personas.

Una misma persona puede ser estudiante a la vez que empleado.

Toda persona en nuestra BD tiene que ser un estudiante y/o empleado.

#### PARCIAL CON SOLAPAMIENTO

Tanto un docente como un investigador son empleados.

Un mismo empleado puede ser docente a la vez que investigador.

### **20.2.1.3**

### **Tipos de relaciones**

A. RELACIONES REFLEXIVAS Son interrelaciones en las que interviene un único tipo de entidad (unarias).

Ejemplo: Un trabajador puede ser jefe de ningún trabajador o puede serlo de varios

trabajadores, mientras que un trabajador sólo es dirigido por ninguno o un trabajador

**B. INTERRELACIONES CON RESTRICCIONES DE EXCLUSIVIDAD.** Dos interrelaciones que implican a un mismo tipo de entidad participan de una restricción de exclusividad si los ejemplares de esa entidad pueden participar de una u otra interrelación, pero no de ambas.

Se ha recogido en el esquema que en una determinada biblioteca los artículos están publicados en revistas o aparecen en recopilaciones, pero no en ambos

**C. INTERRELACIONES CON RESTRICCIONES DE EXCLUSIÓN.** Dos interrelaciones entre los mismos dos tipos de entidad son exclusivas si un ejemplar del primer tipo de entidad y otro ejemplar del segundo tipo de entidad sólo pueden estar relacionados por una de las dos interrelaciones, nunca por ambas simultáneamente.

Un profesor no puede recibir e impartir el mismo curso, aunque al contrario que en la restricción anterior puede impartirlo o recibirlo.

**D. INTERRELACIONES CON RESTRICCIONES DE INCLUSIVIDAD.** Son dos interrelaciones que implican a un mismo tipo de entidad, en las que los ejemplares de la entidad tienen que haber participado de una interrelación con una cardinalidad determinada para poder participar de la otra.

Para que un profesor pueda impartir un curso, tiene que haber recibido el curso un mínimo de 3 veces.

E. INTERRELACIONES CON RESTRICCIONES DE INCLUSIÓN. Son aquellas que se establecen entre los mismos dos tipos de entidad y que restringen una interrelación entre dos ejemplares de cada una de las entidades a la vinculación de esos dos mismos ejemplares a través de la otra interrelación.

Todo ejemplar de profesor que esté unido a un ejemplar de curso mediante la interrelación imparte tiene necesariamente que estar unido al mismo ejemplar de curso mediante la interrelación recibe.

F. AGREGACIÓN. Es un tipo especial de interrelación que permite representar tipos de entidad compuestos que se forman a partir de otros más simples. Existen dos clases de agregaciones

Compuesto/Componente El supertipo de entidad se obtiene por la unión de los subtipos. Se representa de la siguiente forma

Miembro/Colección: El supertipo de entidad es una colección de elementos de un mismo subtipo. Se representa como:

#### **20.2.1.4**

#### **Control de redundancias**

En el modelo E/Res necesario evitar las redundancias para no tener problemas de inconsistencias de la representación. Un elemento de un esquema es redundante si puede ser eliminado sin pérdida de semántica.

Existen dos formas principales de redundancia:

- En los atributos (atributos derivados o calculados): Aunque son redundantes, no dan lugar a inconsistencias siempre que en el esquema se indique su condición de derivados y la fórmula mediante la que han de ser calculados.
- En las interrelaciones (también llamadas interrelaciones derivadas): Una interrelación es redundante si su eliminación no implica pérdida de semántica porque existe la posibilidad de realizar la misma asociación de ejemplares por medio de otras interrelaciones. Para ello es condición necesaria pero no suficiente que forme parte de un ciclo.

La existencia de un ciclo no implica la existencia de interrelaciones redundantes.

Para que una interrelación pueda ser eliminada por redundante se tiene que cumplir:

- Que exista un ciclo.
- Que las interrelaciones que componen el ciclo sean equivalentes semánticamente,
- Que después de eliminar la interrelación se puedan seguir asociando los ejemplares de las dos entidades que estaban interrelacionadas
- Que la interrelación no tenga atributos o que éstos puedan ser transferidos a otro elemento del esquema a fin de no perder su semántica.

### **20.2.1.5**

### **Dimensión temporal**

Es necesario establecer un método semántico y gráfico que recoja de algún modo, en el esquema conceptual, el transcurso del tiempo y su influencia en la forma en que cambian los datos. Existen varias aproximaciones:

- La más simple la constituyen los atributos de tipo fecha asociados a algunas entidades o interrelaciones:
  - o Para sucesos instantáneos, es decir, sin duración, bastará con un sólo atributo de este tipo.

- o Para poder almacenar hechos que transcurren en un intervalo de tiempo determinado necesitaremos una `fecha_inicio` y una `fecha_fin`.
  - o En las bases de datos históricas, en las que una interrelación entre dos ejemplares concretos se pueda repetir en el tiempo, el atributo `fecha` será multivaluado.
- Cuando es necesario representar la evolución de un tipo de entidad a lo largo del tiempo se utiliza un atributo de estado, que indicará en qué estado concreto se encuentra la entidad.

En muchos casos lleva asociado otro atributo, que es la fecha en la que se ha producido el cambio de estado o el intervalo de tiempo en que ha permanecido en dicho estado.

## 20.3 **ENTIDADES Y ATRIBUTOS**

Chen distingue en el modelo E/R los siguientes elementos: Entidad, Relación, Atributo y Dominio

### 20.3.1 **Entidad**

Una entidad es un objeto real o abstracto de interés en una organización y acerca del cual se puede y se quiere obtener una determinada información; personas, cosas, lugares, etc., son ejemplos de entidades

La estructura genérica que describe un conjunto de entidades aplicando la abstracción se denomina tipo de entidad, mientras que entidad se refiere a cada una de las ocurrencias o ejemplares de ese tipo de entidad. Así pues, asociado al concepto de entidad surge el concepto de ocurrencia de entidad. Una ocurrencia de entidad es una realización concreta de una entidad. De esta manera, Hospital es un tipo de entidad mientras que “CHOU” es una ocurrencia o ejemplar.

Una entidad debe cumplir las siguientes reglas:

- Debe tener existencia propia (veremos que hay un tipo de entidades que en puro rigor no cumple esta restricción como son las entidades débiles)
- Cada ocurrencia de un tipo de entidad tiene que poder distinguirse de los demás
- Todas las ocurrencias de un mismo tipo de entidad deben tener las mismas propiedades o atributos

Una entidad se representa gráficamente en el modelo E/R mediante un rectángulo y en el interior del mismo se escribe en mayúsculas el nombre del tipo de entidad.

ENTIDAD

Existen dos tipos de entidades:



- **Regulares:** sus ejemplares tienen existencia por sí mismos, p.ej. LIBRO.
- **Débiles** en las que la existencia de un ejemplar depende de que exista un cierto ejemplar de otro tipo de entidad. Se verán en el apartado 2.2.2

### 20.3.2 *Relación o Interrelación*

Una interrelación es una asociación entre entidades y se caracterizará por unas determinadas restricciones que determinarán las entidades que pueden o no participar de dicha relación.

La interrelación se representa gráficamente por un rombo etiquetado con el nombre de la interrelación en mayúsculas unido mediante arcos a las relaciones que vincula.

Asociado al concepto de interrelación surge el concepto de ocurrencia de interrelación.

Una ocurrencia de interrelación es la asociación concreta de ocurrencias de entidad de diferentes entidades. Por ejemplo, si tenemos las entidades MEDICO y HOSPITAL, y la interrelación “trabaja en”, una ocurrencia de interrelación será: MARTA GARCÍA trabaja en el CHOU.

Una interrelación queda caracterizada por tres propiedades:

- **Nombre:** las interrelaciones deben tener un nombre que las identifique unívocamente.
- **Grado:** número de tipos de entidad sobre las que se realiza la asociación. La interrelación del ejemplo anterior será binaria, es decir, su grado sería dos.
- **Tipo de Correspondencia:** Número máximo de ocurrencias de cada tipo de entidad que pueden intervenir en una ocurrencia del tipo de interrelación.

Las relaciones pueden tener atributos propios.

### 20.3.3

### *Dominio*

Representa el conjunto de valores posibles de una determinada propiedad o atributo de un tipo de entidad o de un tipo de interrelación. En términos de abstracción, es una especialización de un conjunto. Con lo que se puede decir que el dominio es un conjunto de valores homogéneos con un nombre.

Se representa por un círculo pequeño acompañado de su nombre en minúsculas.

Es importante resaltar en este punto que los dominios tienen existencia propia y es el que realmente captura una semántica del mundo real. Lo que ha ocurrido muy a menudo es que se tiende a confundir dominio con atributo.

### 20.3.4

### *Atributos*

Es cada una de las posibles propiedades o características de un tipo de entidad o tipo de interrelación. Los atributos toman valor en un dominio por lo que un atributo es una determinada interpretación de un dominio y varios atributos pueden tomar valores en el mismo dominio. Por ejemplo, si tenemos el atributo COLOR el dominio sobre el que se define podría ser: (NARANJA, BLANCO, AZUL y NEGRO).

Se puede representar gráficamente de 2 formas:

- a) Por un círculo pequeño unido por un arco al tipo de entidad y acompañado del nombre del atributo.
- b) Encerrando en un ovalo el nombre del atributo unido por un arco al tipo de entidad.

En función de las características del atributo respecto de la entidad se distinguen dos tipos de atributos:

- **Atributo identificador clave:** distingue de manera única una ocurrencia de entidad del resto de ocurrencias de entidad. Normalmente, el atributo identificador es único, pero puede haber

casos en los que haya varios atributos identificadores, por lo que denominaremos a cada uno de ellos **atributo identificador candidato**. Elegiremos uno como identificador clave y el resto serán atributos identificadores. Se representa gráficamente:

- **Atributo descriptor:** caracteriza una ocurrencia de entidad pero no la distingue del resto de ocurrencias de entidad.

Una relación puede tener atributos al igual que las entidades.

## 20.4

## RELACIONES

El modelo relacional es un modelo lógico de datos, desarrollado por Codd, que introdujo la teoría matemática de las relaciones en el campo de las BD y supuso un importante paso en la investigación de los SGBD. El documento de Codd propone un modelo de datos basado en la “Teoría de las Relaciones”, donde los datos se estructuran lógicamente en forma de relaciones -tablas-, siendo un objetivo fundamental del modelo mantener la independencia de esta estructura lógica respecto al modo de almacenamiento y a otras características de tipo físico (independencia de ordenación, indexación y de los caminos de acceso).

Este nuevo modelo de datos perseguía los siguientes objetivos:

- Independencia lógica: añadir, eliminar o modificar cualquier elemento de la BD no debe repercutir en los programas y/o usuarios que accedan a vistas de los mismos.
- Independencia física: el modo en que se almacenan los datos no debe influir en su manipulación lógica y, por tanto, los usuarios que acceden a esos datos no han de modificar sus programas por cambios en el almacenamiento físico.
- Flexibilidad: poder ofrecer a cada usuario los datos de la forma más adecuada a su aplicación.
- Uniformidad: Las estructuras lógicas de los datos presentan un aspecto uniforme (tablas), lo que facilita la concepción y manipulación de la BD por parte de los usuarios.
- Sencillez: Las características anteriores, así como unos lenguajes de usuario muy sencillos, producen como resultado que el modelo de datos relacional sea fácil de comprender y utilizar por parte del usuario final.

### 20.4.1

### Elementos

El modelo relacional introduce su propia terminología para denominar los objetos y elementos utilizados:

1. **Relación.** Es el elemento central del modelo relacional. Son matrices bidimensionales (tablas) caracterizadas por un nombre, un conjunto de atributos (dimensión vertical de la tabla = columnas) y un conjunto de tuplas (dimensión horizontal = filas).

Cada tupla está formada por el conjunto de valores que toma cada uno de los atributos para un elemento de la relación.

En las relaciones podemos hablar de dos componentes:

- **Intensión** es la parte definitoria y estática de la relación. Define la estructura abstracta de datos y las restricciones de integridad de la misma. Es lo que llamaremos *esquema de relación*.
- **Extensión** es el conjunto de tuplas que satisfacen el esquema de relación en un instante dado y se encuentran almacenadas en la base de datos. Varía con el transcurso del tiempo.

Al número de tuplas de una relación en un instante dado se le denomina **cardinalidad** de la relación, y normalmente varía con el transcurso del tiempo. Al número de columnas o atributos se le denomina **grado** de la relación.

2. **Dominio.** Es el conjunto definido, finito y homogéneo de los valores atómicos posibles de un determinado atributo.

Cada atributo está ligado a un determinado dominio y representa el uso de un dominio para una determinada relación.

Los dominios pueden estar definidos por intención (conjunto definido mediante una serie de reglas abstractas) o por extensión (conjunto finito de valores posibles).

3. **Claves de una relación.** Una clave es una(s) columna(s) cuyos valores identifican una única fila de una tabla. Hay varias clases de claves

- *Clave candidata* Cada uno de los conjuntos mínimos de atributos que identifiquen sin ambigüedad y de forma única cada una de las tuplas de una relación.
- *Clave primaria o principal* De entre todas las claves candidatas de una relación, en la definición del esquema se deberá especificar cuál de ellas se considera como identificador primario. Al resto de las claves candidatas se les denominará *claves alternativas*.
- *Claves foráneas o claves ajenas* son el conjunto de atributos de una relación que se corresponden con la clave primaria de otra relación del modelo. Proporcionan al modelo relacional los mecanismos adecuados para representar las (inter)relaciones existentes entre los objetos del problema.
  - o Puede referenciar la clave primaria de la misma tabla (relaciones reflexivas)
  - o Debe tener siempre un valor correspondiente en la tabla donde es clave primaria
  - o Debe estar formada por toda la clave primaria y no solo una parte de ella
  - o Puede tener nulos
  - o Puede tener valores duplicados
  - o Una tabla puede contener múltiples claves foráneas, donde cada una representa la relación con otra tabla

## 20.4.2

## ***Restricciones***

En el modelo relacional existen restricciones, es decir, estructuras u ocurrencias no permitidas, siendo preciso distinguir entre restricciones inherentes (propias del modelo) y restricciones semánticas (de usuario).

### 20.4.2.1

### ***Restricciones inherentes***

- i.1. No se define ningún orden en los elementos que forman una relación, ni en el sentido horizontal (tuplas) ni en el vertical (atributos). El orden es siempre irrelevante.

i.2. En toda relación es obligatoria la existencia de la clave primaria, y por tanto no puede haber dos tuplas iguales.

i.3. Cada atributo de una tupla sólo puede tomar un único valor del dominio sobre el cual está definido.

i.4. Regla de integridad de clave o entidad: ninguno de los atributos que forman parte de una clave primaria de una relación puede tomar un valor nulo para ninguna tupla de esa relación.

#### **20.4.2.2** **Restricciones semánticas**

1. Declaración de Clave primaria (PRIMARY KEY): Permite declarar un atributo o un conjunto de atributos como clave primaria de una relación, por lo que sus valores no se podrán repetir ni admitirán nulos.
2. Unicidad (UNIQUE): indica que los valores de un atributo (o conjunto) no pueden repetirse en una relación. Esta restricción permite definir claves candidatas.
3. Obligatoriedad (NOT NULL) , indica que un atributo (o conjunto) no admite nulos
4. Integridad referencial (restricción de clave ajena): permiten que las claves foráneas de una relación referencien una tupla válida de la relación padre. El usuario puede especificar, en la definición del esquema relacional, las operaciones que deben llevarse a cabo cuando se produce el borrado o modificación de una tupla en la relación padre. Las posibilidades son:
  - Borrado/modificación en cascada (CASCADE). El borrado o modificación de una tupla en la relación padre, provoca el borrado o modificación de todas las tuplas relacionadas en la relación hija.
  - Borrado / modificación restringido (NO ACTION). Si existen tuplas relacionadas en la relación hija, no se permite el borrado o modificación de las tuplas de la relación padre.
  - Borrado / modificación con puesta a nulos (SET NULL). Pone a nulo los valores de todos los atributos que conforman la clave ajena en la relación hija. Sólo está permitido cuando esos valores se puedan poner a nulo.

- Borrado / modificación con puesta a un valor por defecto (SET DEFAULT). Similar al anterior, pero los atributos que conforman la clave ajena en la relación hija se ponen a un valor especificado previamente en la definición del esquema.
5. Restricciones de rechazo. En la definición del esquema relacional pueden imponerse otra serie de restricciones que garanticen la integridad del modelo, y por tanto, de la información almacenada en la base de datos. Estas restricciones deben ser verificadas en toda operación de actualización para que el nuevo estado constituya una ocurrencia válida del esquema; en caso de que la operación intente violar la condición se impide que la operación se lleve a cabo, como son:
- Restricciones de verificación (CHECK). Especifican condiciones que debencumplir los valores de determinados atributos de una relación, como pueden ser los atributos de existencia obligatoria (NOT NULL).
  - Aserciones (ASSERTION). Permiten especificar condiciones entre los elementos de distintas relaciones del esquema.
  - Disparadores (TRIGGER). Permiten especificar condiciones y acciones que se lleven a cabo cuando se efectúe una acción determinada sobre alguna relación del esquema



## 20.5 **DIAGRAMAS ENTIDAD-RELACIÓN**

Decíamos que el interés de los modelos de datos es captar tanta semántica como sea posible del mundo real. Con lo que hemos visto hasta el momento comprobamos que se permite establecer cualquier número de relaciones diferentes entre tipos de entidad pero no podemos establecer restricciones del tipo:

- Un médico sólo trabaja en un hospital
- En un hospital trabajan n médicos
- Todos los socios del videoclub han alquilado al menos una película

A continuación se analiza en detalle todos los aspectos de las relaciones que nos permitirán captar toda la semántica deseada.

### 20.5.1 **Tipo de correspondencia**

Se denomina tipo de correspondencia al tipo de asociación que se establece entre las entidades relacionadas. Concretamente, se puede definir el tipo de correspondencia como el número máximo de ocurrencias de una entidad asociada a una ocurrencia de otra o de la misma entidad a través de una relación.

Para una relación binaria, es decir, de grado dos, entre las entidades A y B, existen tres tipos posibles de correspondencias:

- **Correspondencia 1:1** Una ocurrencia de la entidad A se asocia como máximo con una ocurrencia de la entidad B y viceversa, como se puede observar en la figura

Un ejemplo de este tipo de correspondencia puede ser que un cliente tiene una única cuenta bancaria en una sucursal determinada y una cuenta determinada de una sucursal pertenece a un único cliente.

- **Correspondencia 1:N** Una ocurrencia de la entidad A se asocia con un número indeterminado de ocurrencias de la entidad B, pero una ocurrencia de la entidad B se asocia como máximo con una ocurrencia de la entidad A. Si fuera al revés la correspondencia sería N:1.

Un ejemplo de este tipo de correspondencia puede ser que una persona vive en una ciudad y en una ciudad viven muchas personas.

- **Correspondencia N:M** Una ocurrencia de la entidad A se asocia con un número indeterminado de ocurrencias de la entidad B y viceversa.

Un ejemplo de este tipo de cardinalidad puede ser que un proveedor suministra varios productos y cada producto puede ser suministrado por varios proveedores.

## 20.5.2

### *Entidades Débiles*

El concepto de entidad débil está directamente relacionado con las restricciones de tipo semántico del modelo E/R y, más concretamente, con la denominada restricción de existencia.

Esta restricción establece el hecho de que la existencia de una entidad no tiene sentido sin la existencia de otra, es decir, una entidad tiene dependencia de existencia de otra cuando sin la primera la segunda carecería de sentido.

Esto conlleva que la desaparición de las ocurrencias de la entidad de la cual depende su existencia lleve a la desaparición de las ocurrencias de la entidad débil que dependan de ellas. Por ejemplo, un ejemplar de la entidad EDICIÓN no existiría si no hubiera un ejemplar correspondiente en la entidad LIBRO. Las entidades débiles se representan gráficamente por dos rectángulos concéntricos y en el interior el nombre de la entidad.

**ENTIDAD DÉBIL**

Hay dos tipos de dependencias de las entidades débiles respecto a las entidades regulares:

*Dependencia en existencia* los ejemplares de la entidad débil no pueden existir si desaparece el ejemplar de la entidad regular con el que están relacionados, pero la entidad débil puede ser identificada sin necesidad de identificar la entidad fuerte relacionada, es decir, la entidad débil tiene un atributo identificador clave.

En el ejemplo es evidente que si desaparece un empleado de la base de datos la existencia de sus familiares carece de sentido, es decir, la entidad FAMILIAR tiene dependencia de existencia respecto de la entidad EMPLEADO. Sin embargo, cada una de las ocurrencias de la entidad familiar puede identificarse por sí misma.

*Por identificación* la entidad débil no tiene sentido en sí misma y no puede ser identificada sin la entidad fuerte relacionada, es decir no tiene un atributo identificador clave sino tan sólo un descriptor discriminador y necesita el atributo clave de la entidad

fuerte para poder identificar de manera única sus ocurrencias de entidad.

En el ejemplo, el atributo identificador clave será Cod\_Libro (como clave de la entidad fuerte LIBRO) más ID como discriminador de la entidad EJEMPLAR.

Como conclusión al concepto de entidad débil conviene resaltar las circunstancias siguientes:

1. La dependencia en existencia no implica una dependencia en identificación, hecho que si sucede en el caso inverso pues una entidad que depende de otra por su Atributo Claveno tendrá sentido sin la existencia de esta última.
2. En una interrelación con cardinalidad N:M nunca habrá entidades débiles. La razones que la supuesta ocurrencia de la entidad débil que se tuviera que borrar podría estar asociada a más de una ocurrencia de la supuesta entidad fuerte, lo que implicaría la imposibilidad de su borrado, hecho éste en clara contraposición con la definición de entidad débil.

### 20.5.3

### *Papel o rol*

Es la función que cada una de las entidades realiza en una interrelación concreta. Gráficamente se representa indicando el nombre del rol en la línea que une las entidades con las relaciones.

Los roles juegan un papel especialmente importante en relaciones reflexivas donde es necesario conocer los dos roles que el mismo tipo de entidad juega en una determinada relación.

La razón está en que estamos asociando entre sí ocurrencias de una misma entidad de forma que cada una de ellas tiene un significado diferente. En el ejemplo, una ocurrencia de PERSONAS hará papel de 'padre' y la otra papel de 'hijo'.

#### **20.5.4      *Atributos multivaluados y compuestos***

Un último tipo de restricciones que se deben tener en cuenta a la hora de realizar el diseño conceptual de una base de datos con el Modelo E/R son las que afectan a la tipología de los diferentes atributos. Desde este punto de vista podemos definir dos tipos diferentes de atributos respecto a los manejados hasta el momento, que son los siguientes:

**Atributos multivaluados.** Son aquellos atributos que para una misma ocurrencia de la entidad toman más de un valor. Por, ejemplo si cada cliente puede tener más de un teléfono y es de interés guardar todos sus posibles valores, el atributo teléfono sería multivaluado. Se representa etiquetando su arco con un valor de cardinalidad N.

**Atributos Compuestos.** Son aquellos que agrupan en sí mismos, por afinidad o por forma de uso, más de un atributo. Por ejemplo el atributo "dirección" engloba los atributos calle, número, ciudad, provincia y código postal.

Se representa especificando sus atributos componentes rodeando al mismo y enlazándolos al símbolo del atributo compuesto mediante arcos.

#### **20.5.5**

#### ***Atributos Derivados***

Son aquellos que pueden calcularse a partir de otros. Por ejemplo, si tenemos la entidad PERSONA con los atributos DNI, Nombre, Fecha\_Nacimiento y Edad, el último atributo (Edad) puede obtenerse a partir de otro atributo (la fecha de nacimiento) y es, por lo tanto, redundante. Este tipo de atributos deben eliminarse del esquema.

## 20.6

## **BIBLIOGRAFÍA**

- The Entity/Relationship Model: Toward a unified view of data. CACM, 1,1. 1976
- The Entity/Relationship Model: A basis for the enterprise view of data. AFIPS Conference Proceedings, Vol 46. 1977
- Introducción a los sistemas de bases de datos. C.J. Date. Pearson Educación, 2001.
- Fundamentos de Sistemas de Bases de Datos. Ramez A. Elmasri & Shamkant B. Navathe. Addison-Wesley, 2002 [3ª edición].

**Autor:** Francisco Javier Rodríguez Martínez

Subdirector de Sistemas Escola Superior Enxeñaría Informática Ourense

Colegiado del CPEIG

# **21. SISTEMAS DE GESTIÓN DE BASES DE DATOS. MODELO RELACIONAL. NORMALIZACIÓN. SQL. LENGUAJE DE DEFINICIÓN DE DATOS (DDL), LENGUAJE DE MANIPULACIÓN DE DATOS (DML) Y LENGUAJE DE CONTROL DE DATOS (DCL).**



## **Tema 21: Sistemas de gestión de bases de datos. Modelo relacional. Normalización. SQL. Lenguaje de definición de datos (DDL), lenguaje de manipulación de datos (DML) y lenguaje de control de datos (DCL).**

---

### **ÍNDICE**

<b>21.1 Sistemas de Gestión de Bases de Datos (SGBD).....</b>	<b>2</b>
21.1.1 <i>Introducción.....</i>	2
21.1.2 <i>SGBD objetivo y características.....</i>	2
21.1.3 <i>Evolución.....</i>	2
21.1.3.1 <i>Arquitectura en 2 capas.....</i>	2
21.1.3.2 <i>Arquitectura en 3 capas.....</i>	2
21.1.4 <i>Estructura General de un SGBD.....</i>	3
21.1.5 <i>SGBD Relacionales (SGBD-R).....</i>	3
21.1.5.1 <i>Características de los SGBD-R.....</i>	3
<b>21.2 MODELO RELACIONAL .....</b>	<b>4</b>
21.2.1 <i>Elementos.....</i>	4
21.2.2 <i>Restricciones.....</i>	4
<b>21.3 NORMALIZACIÓN.....</b>	<b>5</b>
<b>21.4 SQL.....</b>	<b>6</b>
21.4.1 <i>Partes del lenguaje SQL.....</i>	6
21.4.2 <i>Modos de trabajo con SQL.....</i>	6
<b>21.5 Lenguaje y definición de datos (DDL).....</b>	<b>6</b>
21.5.1 <i>Objetos de la Base de Datos.....</i>	7
21.5.2 <i>Gestión de tablas.....</i>	7
21.5.3 <i>Gestión de vistas.....</i>	9

21.5.4Gestión de índices.....	9
<b>21.6 Lenguaje de manipulación de datos (DmL) .....</b>	<b>9</b>
21.6.1Inserción de valores en una tabla.....	9
21.6.2Borrado de valores de una tabla.....	10
21.6.3Modificación de valores de una tabla.....	10
21.6.4Consulta de datos.....	10
21.6.5Consultas sobre múltiples tablas.....	12
21.6.6Subconsultas.....	12
21.6.7Operaciones con consultas.....	13
<b>21.7. Lenguaje de control de datos DCL.....</b>	<b>13</b>
21.7.1Seguridad.....	13
21.7.2Transacciones.....	13
<b>21.8Bibliografía.....</b>	<b>15</b>

## **21.1 SISTEMAS DE GESTIÓN DE BASES DE DATOS (SGBD)**

### **21.1.1**

### **Introducción**

Para que una base de datos funcione correctamente precisa de un software que gestione todas sus operaciones y que además proporcione una interfaz de comunicación para los usuarios facilitando el acceso a los datos contenidos en ella. Este tipo de software son los denominados *Sistemas de Gestión de Bases de Datos* o *S.G.B.D.*

Según una definición formal, un Sistema de Gestión de Bases de Datos, *“es un conjunto coordinado de programas, procedimientos, lenguajes, etc... que suministra, tanto a los usuarios no informáticos, como a los analistas programadores, o al administrador, los medios necesarios para describir y manipular los datos integrados en la base, manteniendo su integridad, confidencialidad y seguridad”*.

Este grupo definido de software encargado de ofrecer soporte a las bases de datos aporta una serie de facilidades para el manejo de las bases de datos. Estas facilidades se traducen en una serie de Lenguajes que permiten operar contra las distintas bases de datos que soporta el SGBD, siendo los principales:

- **Lenguaje de Definición de Datos o DDL** (Data Definition Language): permite definir los esquemas conceptuales de una base de datos.
- **Lenguaje de Manipulación de Datos o DML** (Data Manipulation Language): suministra operaciones para la realización de consultas y actualizaciones de la información contenida en las bases de datos.
- **Lenguaje de Control de Datos o DCL** (Data Control Language): permite administrar y controlar el acceso a los datos existentes en una base de datos.

Desde los años 70, el grupo ANSI/X3/SPARC es el encargado de ocuparse de la normalización de los SGBD, publicando en 1975 un informe provisional en donde propone una arquitectura de 3 capas para los SGBD que posteriormente se revisaría y detallaría en 1977. Sin embargo, hasta 1985 no se presentó el *Modelo de Referencia* para la estandarización de los SGBD (Modelo ANSI).

### **21.1.2 SGBD objetivo y características**

Para un sistema gestor de bases de datos su objetivo principal es el de ofrecer un entorno idóneo para la extracción, manipulación almacenamiento de la información de las bases de datos. Los SGBD realizan una gestión centralizada de todas las peticiones de acceso funcionando de interfaz entre los usuarios y la base de datos. También es el gestor de la estructura física de los datos y de su almacenamiento, por lo que en definitiva, libera a los usuarios de conocer la organización física exacta de los datos, así como de crear los algoritmos de almacenamiento, actualización o consulta de la información contenida en la base de datos.

No todos los SGBD son iguales ni tienen las mismas funcionalidades, puesto que dependen de cada producto y del modelo de base de datos que gestionen. Independientemente de esto, existen una serie de características que se podrían identificar como comunes a todos ellos, las cuales fueron definidas por Codd y revisadas con posterioridad, a medida que las nuevas necesidades se fueron integrando. Las características necesarias para que un SGBD pueda cubrir las necesidades de un usuario son:

- Con el fin de simplificar el mantenimiento de las aplicaciones que hacen uso de las bases de datos, un SGBD ha de mantener la independencia entre las soluciones software y la estructura de la base de datos. Esta independencia no es completa pero cada vez se aproximan más a esta exigencia.
- En la medida de lo posible, no debe existir redundancia de datos, es decir, estos no deben de estar almacenados varias veces. Con esto se consigue asegurar la coherencia de los datos.
- Un SGBD ofrece las herramientas necesarias a un usuario para:

- o Almacenar datos.
- o Acceder a la información.
- o Actualizar los datos.

Estas herramientas han de proporcionar estos servicios de tal manera que resulte transparente al usuario.

- Permite el acceso de múltiples usuarios a la misma base de datos y en el mismo momento. Cuando esto se produce, si alguno de los usuarios está realizando operaciones de actualización de los datos, el SGBD ha de asegurarse de realizar una correcta gestión de la concurrencia evitando la corrupción de los datos o que estos se vuelvan inconsistentes. Para realizar esta gestión el SGBD hace un correcto uso de los bloqueos de las bases de datos.
- Ofrece a los usuarios un catálogo al cual pueden acceder donde se almacenan de una manera centralizada, las descripciones de los datos. Este servicio es el que permite la eliminación y detección de las redundancias de datos y se denomina *diccionario de datos*.
- Realiza las transacciones garantizando que las actualizaciones correspondientes a una transacción se realizaran, o en caso de que no sean posible realizar alguna, ninguna tendrá efecto. Esto se debe a que durante una transacción, se producen acciones que cambian el contenido de la base de dato. Si por algún motivo la transacción falla, entonces al base de datos pasa a un estado de inconsistencia, puesto que no todos los cambios de la transacción se produjeron, obligando al SGBD a deshacer los cambios para que la base de datos vuelva a un estado consistente.
- Garantiza la recuperación de las bases de datos. Si ocurre algún problema que provoque que la información se vea afectada, un fallo de hardware o software que hagan abortar al SGBD, o que un usuario interrumpa una operación antes de que se finalice la transacción, el SGBD debe proporcionar los mecanismos necesarios para solventar este tipo de situaciones y recuperar la base de datos a un estado consistente.

- Proporciona seguridad a las bases de datos, es decir, restringe mediante diferentes niveles, que el acceso a las bases de datos sólo lo realizaran usuarios autorizados, protegiendo las bases de accesos no autorizados, ya sean accidentales o no.
- Un SGBD garantiza la integridad de las bases de datos. Aporta un conjunto de reglas que la base de datos no puede violar consiguiendo así la validez y consistencia de los datos.
- Proporciona herramientas para la administración de las bases de datos. Este conjunto de herramientas permiten una serie de funcionalidades:
  - o Importación y extracción de datos
  - o Monitorización del funcionamiento y obtención de datos estadísticos sobre el uso de las bases de datos.
  - o Reorganización de índices
  - o Optimización del espacio liberado para su reutilización.
- Mantiene una disponibilidad continua, garantizando que en todo momento las bases de datos están accesibles. Proporciona que las tareas de administración, gestión y mantenimiento se puedan llevar a cabo sin interrumpir el correcto funcionamiento de las bases de datos.
- Todo SGBD se integra con un gestor de comunicaciones, software encargado de gestionar el acceso de los usuarios que realicen la conexión con la máquina que sirve de soporte al SGBD de una manera remota a través de una red de datos. El gestor de comunicaciones no forma parte de un SGBD pero si es preciso que el SGBD se integre con él.
- Posee un DDL, Lenguaje de Definición de Datos para la creación y modificación de las bases de datos.

- Posee un DML, Lenguaje de Manipulación de Datos para la inserción, manipulación y consulta de la información contenida en las bases de datos.
- Posee un DCL, Lenguaje de Control de Datos para controlar el acceso a la información de las bases de datos.
- Garantiza la escalabilidad y elevada capacidad de proceso, es decir, es capaz de aprovechar los recursos de la máquina disponibles, aumentando su capacidad de procesado a medida que disponga de recurso.
- Es capaz de almacenar enormes cantidades de datos sin que el usuario perciba una degradación en el rendimiento del sistema.

### **21.1.3**

### ***Evolución***

En los primeros inicios de la informática los datos formaban parte de los programas, integrados como constantes. Posteriormente, con la aparición de los ficheros como colección de datos homogénea, se empieza a diferenciar la estructura lógica que representa el punto de vista del usuario y la estructura física de los datos.

Esta diferenciación se hace más evidente con la aparición en los sistemas operativos de subsistemas de gestión de datos, pero no resulta suficiente para romper la dependencia entre los datos y los programas y viceversa, y de ambos con respecto a la máquina.

Para limar estas dependencias existentes entre los datos y las aplicaciones, se comienzan a utilizar arquitecturas que diferencian la estructura lógica de los datos, representación de los datos orientados hacia el problema, de la estructura física, representación orientada hacia la máquina. Esta diferenciación funciona a través de una transformación o mapping que hace las tareas de conversión entre una estructura y otra.

#### **21.1.3.1**

#### **Arquitectura en 2 capas**

Con la aparición de los primeros SGBD en los años 60, los sistemas pasan de una orientación centrada en el proceso a una orientación enfocada hacia las bases de datos.

Esto produce que los datos y las relaciones entre los mismos se ubiquen en la bases de datos, consiguiendo aislarlos de las aplicaciones. Esta evolución provoca un cambio en las tendencias de las estructuras de datos haciendo que la estructura lógica sea más flexible y sencilla, mientras que la estructura física se vuelva más compleja con el fin de mejorar el rendimiento.

Los primeros SGBD facilitan en gran medida la descripción y el almacenamiento de las relaciones permitidas caracterizando al mismo tiempo los distintos modelos de datos: jerárquico, red, relacional.

La arquitectura seguida por estos SGBD estaba definida en dos niveles:

- *Estructura Global*, con la características lógicas y físicas: Esquema
- *Vistas Lógicas Externas* de los usuarios: Subesquemas

### **21.1.3.2**

### **Arquitectura en 3 capas**

La organización ANSI (American National Standards Institute) publica en el año 1975 un informe que resultaría clave para el desarrollo de los SGBD. En este informe se indica la necesidad de evolucionar los SGBD con el fin de conseguir una total independencia entre los datos y las aplicaciones. Para tal propósito propone un modelo arquitectónico en 3 capas y define a su vez el modelo conceptual para conseguirlo.

En este informe la estructura global del modelo de 2 niveles se divide dando lugar a dos estructuras, en una se quedan los aspectos lógicos y el esquema conceptual, mientras que en la otra se queda con los aspectos físico o esquema interno.

Mediante esta definición, los SGBD que siguen esta normativa muestran internamente los tres niveles perfectamente diferenciados:

- **Nivel Interno o Físico:** representa el nivel más bajo de abstracción y en este nivel es donde se describe en detalle la estructura física de la base de datos, dispositivos de almacenamiento físico, estrategias de acceso, índices, etc. Para



ello interactúa con el sistema operativo y con el gestor de ficheros. En definitiva el esquema interno especifica que datos son almacenados y cómo, además de describir la estructura de la base de datos en forma de modelo conceptual de almacenamiento.

- **Nivel Conceptual:** se corresponde con el nivel intermedio de abstracción y describe los datos que son almacenados en la base de datos y las relaciones existentes entre ellos. También describe la base de datos según su estructura de diseño. En este nivel la base de datos resulta una colección de registros lógicos sin descriptores de almacenamiento. Mediante este nivel se consigue el aislamiento de la representación de la información de los requerimientos de la máquina y de las exigencias de los usuarios.
- **Nivel Externo o Lógico:** supone el de mayor grado de abstracción y contiene las vistas externas de la base de datos asociadas a un esquema externo. Proporciona a cada tipo de usuario únicamente la parte del esquema que resulta relevante para él y para la cual tiene acceso. Cada base de datos puede tener tantas vistas como necesite.

Con esta arquitectura se pretende conseguir que el esquema conceptual sea una descripción estable e independiente del nivel superior y del inferior, es decir, independiente tanto de las vistas como del almacenamiento de los datos. Con la consecución de esta independencia las bases de datos se convierten en sistema más flexibles y adaptables.

#### **21.1.4 Estructura General de un SGBD**

Los principales módulos del SGBD son:

- **El compilador del DDL.** Chequea la sintaxis de las sentencias del DDL y actualiza las tablas del diccionario de datos o catálogo que contienen los metadatos.
- **El precompilador del DML.** Convierte las sentencias del DML embebidas en el lenguaje anfitrión, en sentencias listas para su procesamiento por parte del compilador de lenguaje anfitrión y

además extrae dichas sentencia DML para que puedan ser procesadas de forma independiente por el compilador del DML.

- **El compilador del DML.** Chequea la sintaxis de las sentencias del DML y se las pasa al procesador de consultas.
- **El procesador de consultas.** Realiza la transformación de las consultas en un conjunto de instrucciones de bajo nivel que se dirigen al gestor de la base de datos.
- **El gestor de la base de datos.** Sirve de interfaz para los programas de aplicación y las consultas de los usuarios. El gestor de la base de datos acepta consultas y examina los esquemas externo y conceptual para determinar qué registros se requieren para satisfacer la petición. Entonces el gestor de la base de datos realiza una llamada al gestor de ficheros para ejecutar la petición.

Los principales componentes del gestor de la base de datos son los siguientes:

- o *El gestor de transacciones.* Realiza el procesamiento de las transacciones.
- o *El gestor de buffers.* Transfiere los datos entre memoria principal y los dispositivos de almacenamiento secundario.
- o *El gestor de ficheros.* Gestiona los ficheros en disco en donde se almacena la base de datos. Este gestor establece y mantiene la lista de estructuras e índices definidos en el esquema interno. Para acceder a los datos pasa la petición a los métodos de acceso del sistema operativo que se encargan de leer o escribir en los ficheros físicos que almacenan la información de la base de datos.

En el esquema propuesto se reflejan distintos bloques en los que se indican:

- Tipos de usuarios que pueden acceder al SGBD
- Métodos utilizados por los usuarios para acceder a la información.

- SGBD que se divide en:
  - o El primer subsistema es el encargado de recibir las peticiones y dirigirlas al gestor de la base de datos o al diccionario de datos.
  - o El segundo es el gestor de la base de datos, que posee un gestor de transacciones, un gestor de buffer y el gestor de ficheros.
  - o La base de datos con sus índices y el diccionario de datos.

### **21.1.5 SGBD Relacionales (SGBD-R)**

Los Sistemas Gestores de Base de Datos Relacionales están basados en el **Modelo Relacional** el cual intenta representar el Universo del Discurso mediante el álgebra relacional y sus principales características son:

Basado en un modelo matemático con un conjunto de reglas y algoritmos establecidos, permitiendo que se desarrollen lenguajes de acceso y manipulación muy potentes y fiables.

La estructuración de los datos se realiza mediante relaciones que son modeladas utilizando tablas bidimensionales que representan las entidades como sus relaciones.

Estable reglas de integridad que posibilitan la incorporación de aspectos semánticos y el traslado de restricciones o comportamientos de los datos al esquema conceptual, que de otra forma, no se podrían modelar sólo con las tablas.

#### **21.1.5.1 Características de los SGBD-R**

Sus tres principales características son las estructuras de datos, los operadores asociados y los aspectos semánticos.

##### **21.1.5.1.1 Estructuras de datos: Relaciones y Claves**

Elementos:

- *Relación*: subconjunto de un producto cartesiano entre conjuntos de atributos que en el modelo relacional se muestra como una tabla con  $m$  fila y  $n$  columnas.
- *Atributo*: representan las columnas de una tabla y se corresponden con las propiedades de las entidades. Estos atributos se encuentran limitados por un dominio que especifica el rango de valores que pueden tomar pudiendo ser compartido por varios atributos.
- *Dominio*: rango de valores que un atributo puede adoptar. Este rango es dependiente del tipo de atributo y los valores del dominio han de ser homogéneos.
- *Tuplas*: nombre que se le asocia a cada una de las filas de una tabla que se corresponden con cada una de las ocurrencias de la relación que se representa en la tabla. Su orden no es relevante.
- *Cardinalidad de la relación*: número de tuplas de una relación.
- *Grado de la relación*: número de atributos de una relación.

Dentro de los elementos que conforman la estructura de datos los más importantes son las relaciones, cuyas características más importantes son:

- Todas las tuplas de una relación están formadas por el mismo número, tipo de atributos y en el mismo orden.
- El orden de las tuplas carece de relevancia.
- En cada atributo de una tupla sólo puede aparecer un valor que además ha de pertenecer al dominio correspondiente.
- No pueden existir dos tuplas iguales en la misma relación. Esto provoca que exista uno o varios atributos que sirvan para distinguir unas tuplas de otras denominados *claves candidatas*.

Alguna de estas claves candidatas son seleccionadas por el administrador o diseñador de la base de datos para la identificación de tuplas, entonces la clave se denomina *clave primaria* y no puede adoptar

nunca el valor nulo. El resto de claves candidatas que no son seleccionadas como primarias se denominan *claves alternativas o secundarias*.

Además una relación puede incluir dentro de sus atributos la clave primaria de otra relación, pasando ésta a ser *clave foránea* de la primera relación.

#### **21.1.5.1.2**

#### **Operadores asociados**

El álgebra con la que se mueve el modelo relacional está formada por un conjunto de operadores asociados y es completa, es decir, garantiza matemáticamente que con ella se puede realizar cualquier acceso a la base de datos.

Los operadores utilizan las relaciones del modelo como operandos. Los operadores más importantes se muestran a continuación:

- **Unión.** La unión de dos relaciones “A” y “B” produce el conjunto de tuplas formado por las tuplas de “A” y las tuplas de “B”. Solo es aplicable a relaciones con el mismo grado y con los mismos atributos.
- **Diferencia.** La diferencia entre dos relaciones “A” y “B” es el conjunto de tuplas de la relación A que no están en “B”. Solo es aplicable a relaciones con el mismo grado y con los mismos atributos.
- **Producto Cartesiano.** El producto cartesiano de dos relaciones “A” de grado m y “B” de grado n, está formado por el conjunto de todas las posibles tuplas de m+n atributos con los m primeros valores de “A” y los n restantes de “B”.
- **Proyección.** Considerando “x” un subconjunto de atributos de la relación “A”, la *proyección* del atributo “x” sobre la relación “A” es la relación formada por los atributos de “A” correspondientes con los del subconjunto “x”.
- **Selección.** Si “F” resulta una fórmula que está compuesta por operadores lógicos, aritméticos y de comparación y los operandos se corresponden con valores de los atributos de una relación “A”,

entonces la selección de “F” sobre “A” es el conjunto resultante formado por las tuplas de “A” que cumplen la condición establecida por “F”.

Partiendo de este conjunto de operadores se pueden generar otros derivados como la intersección, el cociente o la unión natural.

#### **21.1.5.1.3**

#### **Aspectos Semánticos**

Cuando una característica del entorno o del universo del discurso no se puede modelar mediante la definición de una relación, ésta ha de definirse mediante un nivel de descripción superior pasando a formar parte de los aspectos semánticos. Estos aspectos, desde un punto de vista práctico, son restricciones que se añaden a las propias del modelo relacional y que su propósito es el de garantizar la integridad y validez de los datos. A su vez también aportan un mayor grado de información al esquema lógico de datos.

Dentro de este conjunto de restricciones se pueden identificar dos grupos:

- *Restricciones de Usuario.* Son restricciones que se aplican a los valores pertenecientes al dominio de los atributos, como por ejemplo en un atributo fecha limitar los meses a 12 y los días a 31.
- *Integridad Referencial.* Las restricciones pertenecientes a la integridad referencial se ocupan del mantenimiento de las referencias existentes entre las propias relaciones.

Para mantener la integridad referencial, cuando se realiza alguna tarea de borrado o modificación de las tuplas se ha de realizar alguna de las siguientes acciones.

- o Impedir la operación, para asegurarse que una vez establecida la relación entre dos tuplas de distintas tablas no se puede deshacer.
- o Transmitir en cascada, es decir si se borra o modifica una tupla, todas aquellas que hace referencia a ella se han también de borrar o modificar.

- o Poner a nulo, mantener la integridad asignando el valor nulo al atributo que realiza las tareas de clave foránea.
- o Establecer valor por omisión o lanzar un procedimiento de usuario que lo establezca.

## 21.2

## **MODELO RELACIONAL**

Es un modelo lógico de datos, desarrollado por Codd, que introdujo la teoría matemática de las relaciones en el campo de las BD y supuso un importante paso en la investigación de los SGBD. El documento de Codd propone un modelo de datos basado en la “Teoría de las Relaciones”, donde los datos se estructuran lógicamente en forma de relaciones -tablas-, siendo un objetivo fundamental del modelo mantener la independencia de esta estructura lógica respecto al modo de almacenamiento y a otras características de tipo físico (independencia de ordenación, indexación y de los caminos de acceso).

Este nuevo modelo de datos perseguía los siguientes objetivos:

- Independencia lógica: añadir, eliminar o modificar cualquier elemento de la BD no debe repercutir en los programas y/o usuarios que accedan a vistas de los mismos.
- Independencia física: el modo en que se almacenan los datos no debe influir en su manipulación lógica y, por tanto, los usuarios que acceden a esos datos no han de modificar sus programas por cambios en el almacenamiento físico.
- Flexibilidad: poder ofrecer a cada usuario los datos de la forma más adecuada a su aplicación.
- Uniformidad: Las estructuras lógicas de los datos presentan un aspecto uniforme (tablas), lo que facilita la concepción y manipulación de la BD por parte de los usuarios.

- Sencillez: Las características anteriores, así como unos lenguajes de usuario muy sencillos, producen como resultado que el modelo de datos relacional sea fácil de comprender y utilizar por parte del usuario final.

### 21.2.1

### Elementos

El modelo relacional introduce su propia terminología para denominar los objetos y elementos utilizados:

1. **Relación.** Es el elemento central del modelo relacional. Son matrices bidimensionales (tablas) caracterizadas por un nombre, un conjunto de atributos (dimensión vertical de la tabla = columnas) y un conjunto de tuplas (dimensión horizontal = filas).

Cada tupla está formada por el conjunto de valores que toma cada uno de los atributos para un elemento de la relación.

En las relaciones podemos hablar de dos componentes:

- Intensión es la parte definitoria y estática de la relación. Define la estructura abstracta de datos y las restricciones de integridad de la misma. Es lo que llamaremos *esquema de relación*.
- Extensión es el conjunto de tuplas que satisfacen el esquema de relación en un instante dado y se encuentran almacenadas en la base de datos. Varía con el transcurso del tiempo.

Al número de tuplas de una relación en un instante dado se le denomina **cardinalidad** de la relación, y normalmente varía con el transcurso del tiempo. Al número de columnas o atributos se le denomina **grado** de la relación.

2. **Dominio.** Es el conjunto definido, finito y homogéneo de los valores atómicos posibles de un determinado atributo.

Cada atributo está ligado a un determinado dominio y representa el uso de un dominio para una determinada relación.



Los dominios pueden estar definidos por intensión (conjunto definido mediante una serie de reglas abstractas) o por extensión (conjunto finito de valores posibles).

3. **Claves de una relación.** Una clave es una(s) columna(s) cuyos valores identifican una única fila de una tabla. Hay varias clases de claves

- *Clave candidata* Cada uno de los conjuntos mínimos de atributos que identifiquen sin ambigüedad y de forma única cada una de las tuplas de una relación.

- *Clave primaria o principal* De entre todas las claves candidatas de una relación, en la definición del esquema se deberá especificar cuál de ellas se considera como identificador primario. Al resto de las claves candidatas se les denominará *claves alternativas*.

- *Claves foráneas o claves ajenas* son el conjunto de atributos de una relación que se corresponden con la clave primaria de otra relación del modelo. Proporcionan al modelo relacional los mecanismos adecuados para representar las (inter)relaciones existentes entre los objetos del problema.

- o Puede referenciar la clave primaria de la misma tabla (relaciones reflexivas)

- o Debe tener siempre un valor correspondiente en la tabla donde es clave primaria

- o Debe estar formada por toda la clave primaria y no solo una parte de ella

- o Puede tener nulos

- o Puede tener valores duplicados

- o Una tabla puede contener múltiples claves foráneas, donde cada una representa la relación con otra tabla

## 21.2.2

## Restricciones

En el modelo relacional existen restricciones, es decir, estructuras u ocurrencias no permitidas, siendo preciso distinguir entre restricciones inherentes (propias del modelo) y restricciones semánticas (de usuario).

### Restricciones inherentes

i.1. No se define ningún orden en los elementos que forman una relación, ni en el sentido horizontal (tuplas) ni en el vertical (atributos). El orden es siempre irrelevante.

i.2. En toda relación es obligatoria la existencia de la clave primaria, y por tanto no puede haber dos tuplas iguales.

i.3. Cada atributo de una tupla sólo puede tomar un único valor del dominio sobre el cual está definido.

i.4. Regla de integridad de clave o entidad: ninguno de los atributos que forman parte de una clave primaria de una relación puede tomar un valor nulo para ninguna tupla de esa relación.

### Restricciones semánticas

1. Declaración de Clave primaria (PRIMARY KEY): Permite declarar un atributo o un conjunto de atributos como clave primaria de una relación, por lo que sus valores no se podrán repetir ni admitirán nulos.
2. Unicidad (UNIQUE): indica que los valores de un atributo (o conjunto) no pueden repetirse en una relación. Esta restricción permite definir claves candidatas.
3. Obligatoriedad (NOT NULL) , indica que un atributo (o conjunto) no admite nulos
4. Integridad referencial (restricción de clave ajena): permiten que las claves foráneas de una relación referencien una tupla válida de la relación padre. El usuario puede especificar, en la definición del esquema relacional, las operaciones que deben llevarse a cabo cuando se produce el borrado o modificación de una tupla en la relación padre. Las posibilidades son:

- Borrado/modificación en cascada (CASCADE). El borrado o modificación de una tupla en la relación padre, provoca el borrado o modificación de todas las tuplas relacionadas en la relación hija.
  - Borrado / modificación restringido (NO ACTION). Si existen tuplas relacionadas en la relación hija, no se permite el borrado o modificación de las tuplas de la relación padre.
  - Borrado / modificación con puesta a nulos (SET NULL). Pone a nulo los valores de todos los atributos que conforman la clave ajena en la relación hija. Sólo está permitido cuando esos valores se puedan poner a nulo.
  - Borrado / modificación con puesta a un valor por defecto (SET DEFAULT). Similar al anterior, pero los atributos que conforman la clave ajena en la relación hija se ponen a un valor especificado previamente en la definición del esquema.
5. Restricciones de rechazo. En la definición del esquema relacional pueden imponerse otra serie de restricciones que garanticen la integridad del modelo, y por tanto, de la información almacenada en la base de datos. Estas restricciones deben ser verificadas en toda operación de actualización para que el nuevo estado constituya una ocurrencia válida del esquema; en caso de que la operación intente violar la condición se impide que la operación se lleve a cabo, como son:
- Restricciones de verificación (CHECK). Especifican condiciones que deben cumplir los valores de determinados atributos de una relación, como pueden ser los atributos de existencia obligatoria (NOT NULL).
  - Aserciones (ASSERTION). Permiten especificar condiciones entre los elementos de distintas relaciones del esquema.
  - Disparadores (TRIGGER). Permiten especificar condiciones y acciones que se lleven a cabo cuando se efectúe una acción determinada sobre alguna relación del esquema

## 21.3

## NORMALIZACIÓN

Al estudiar la estructura del modelo relacional, se deduce que la información de nuestra base de datos puede representarse por medio de un conjunto de objetos (relaciones y dominios) y de un conjunto de reglas de integridad.

En el modelo relacional, como en los demás modelos de datos, el diseño de una base de datos se puede abordar de dos formas distintas:

- Obteniendo el esquema relacional directamente a partir de la observación de nuestro universo del discurso, de forma que plasmemos nuestra percepción del mismo en un conjunto de esquemas de relación, los cuales contendrán los atributos y las restricciones de integridad que representan los objetos y reglas que hemos podido captar en nuestro análisis del mundo real.
- Realizando el proceso de diseño en dos fases, en la primera se lleva a cabo el diseño conceptual, por ejemplo en el modelo E/R, obteniéndose el correspondiente esquema conceptual; en la segunda, éste se transforma en un esquema relacional siguiendo unas determinadas reglas de transformación.

Estas relaciones que resultan de la observación del mundo real o de la transformación al modelo relacional del esquema E/R elaborado en la etapa de modelado conceptual, pueden presentar algunos problemas, derivados de fallos en la percepción del universo del discurso, en el diseño del esquema E/R, o en el paso al modelo relacional;

Entre estos problemas cabe destacar los siguientes:

- Incapacidad para almacenar ciertos hechos.
- Redundancias y, por tanto, posibilidad de inconsistencias,
- Ambigüedades.
- Pérdida de información (aparición de tuplas repetidas).
- Pérdida de dependencias funcionales, es decir, de ciertas restricciones de integridad que dan lugar a interdependencias entre los datos.

- Existencia de valores nulos.
- Aparición, en la base de datos, de estados que no son válidos en el mundo real (anomalías de inserción, borrado y modificación).

En definitiva, el esquema relacional debe ser siempre analizado para comprobar que no presenta los problemas anteriormente citados, evitando la pérdida de información y la aparición de estados que no son válidos en el mundo real.

Para evitar que se puedan dar estos problemas, existen una serie de reglas o formas normales. Estas formas normales serán aplicadas normalmente a bases de datos ya implantadas en forma de relaciones (tablas), lo que nos permitirá pasar a otras relaciones (tablas) que no den los problemas anteriormente descritos.

Existen seis formas normales. Las tres primeras en la mayor parte de los casos son suficientes para normalizar los esquemas de relación.

### **1. Primera Forma Normal (1FN).**

Se dice que una relación está en 1FN cuando cada atributo sólo toma un valor del dominio simple subyacente. Es decir, cada atributo tiene asociado un dominio del cual sólo toma un valor en cada tupla.

Es una restricción inherente al modelo relacional, por lo que su cumplimiento es obligatorio y afecta al número de valores que pueden tomar los atributos de una relación.

### **2. Segunda Forma Normal (2FN).**

Se dice que una relación está en 2FN si:

- Está en 1FN.
- Cada atributo no principal de la relación tiene dependencia funcional completa respecto de la clave primaria de esa relación, esto es, el valor de los atributos no principales de la relación viene determinado por el valor de todos los atributos de la clave.

Por ejemplo, la relación:

**Matrícula (dni, asignatura, nombre, apellidos, curso, nota, aula, lugar)**

No está en 2FN, puesto que nombre y apellidos dependen únicamente de dni y no del valor de asignatura. Igualmente, curso depende únicamente de asignatura, y no del valor de dni.

Para pasar a 2FN se descompone la relación en otras tres, de la forma:

**Matrícula2 (dni, asignatura, nota, aula, lugar)**

**Alumno (dni, nombre, apellidos)**

**Materia (asignatura, curso)**

### **3. Tercera Forma Normal (3FN).**

Una relación R satisface la tercera forma normal (3FN), si y sólo si está en 2FN, y, cada atributo no principal (atributo que no forma parte de la clave) de la relación no depende funcionalmente de otros atributos no principales de esa relación. Es decir, no pueden existir dependencias entre los atributos que no forman parte de la clave primaria de la relación R.

Si se considera, como es lógico, que cada aula se encuentra ubicada físicamente en un único lugar, se puede observar que la relación Matrícula2, la cual se encuentra en 2FN, sigue presentando problemas debidos a que existe una dependencia entre los atributos aula y lugar (no se encuentra, por tanto, en 3FN),

Para eliminar los problemas que ocasiona en la relación Matrícula2 la existencia de esta dependencia funcional, esta relación debe descomponerse en dos relaciones, quedando el esquema de la forma:

**Matrícula3 (dni, asignatura, nota, aula)**

**Ubicación (aula, lugar)**

**Alumno** (dni, nombre, apellidos)

**Materia** (asignatura, curso)

#### **4. Forma Normal de Boyce-Codd (FNBC).**

Es una redefinición más estricta de la 3FN, ya que ésta presentaba ciertos problemas en relaciones con varias claves candidatas compuestas que se solapaban. Por ello en 1974, Boyce y Codd definieron la llamada forma normal que lleva su nombre (FNBC). Se basa en el concepto de determinante funcional.

Se llama determinante funcional a un atributo o a un conjunto de atributos de una relación R del cual depende funcionalmente de forma completa algún otro atributo de la misma relación.

Una relación R satisface la forma normal de Boyce-Codd (FNBC) si, y sólo si se encuentra en 1 FN, y cada determinante funcional es una clave candidata de la relación R, esto es, ningún atributo facilita información de otro atributo que no es clave candidata.

Por ejemplo, en el esquema:

**Matrícula4** (dni, asignatura, apellidos, nombre, nota, aula)

**Ubicación** (aula, lugar)

**Materia** (asignatura, curso)

(donde *asignatura*, *apellidos*, *nombre* es una clave candidata y *dni*, *asignatura* es otra clave candidata de la relación Matrícula4).

En este caso, el esquema está en 3FN pero no está en FNBC, puesto que dni es un determinante funcional (apellidos y nombre dependen de dni) y no es clave candidata de la relación. Para poner el esquema en FNBC hay que descomponer Matrícula4 en 2 relaciones, de tal forma que queda un esquema similar al obtenido en el apartado anterior:

**Matrícula3** (dni, asignatura, nota, aula)

**Alumno** (dni, nombre, apellidos)

**Ubicación** (aula, lugar)

**Materia** (asignatura, curso)

## **5. Cuarta Forma Normal (4FN)**

Está basada en la eliminación de las dependencias multivaluadas. Se dice que en una relación existe una dependencia multivaluada ( $\alpha \twoheadrightarrow \beta$ ), si los valores de un conjunto de atributos  $\beta$  depende únicamente del valor que tome otro conjunto de atributos  $\alpha$ , de forma independiente al resto de atributos de la relación.

Por ejemplo, si tenemos una relación para un concesionario con todos los modelos de coches que se vende, con su color y equipamiento respectivo:

**Concesionario** (modelo, color, equipamiento)

y sabemos que puede vender dos modelos, *utilitario* y *berlina*. Si el modelo utilitario se puede vender en color azul o verde con dos tipos de equipamiento (base o normal) y el modelo berlina se puede vender en color plata o azul con equipamientos normal o lujo.

En este caso, si sabemos que el modelo es utilitario, sabemos los posibles valores para el color y el equipamiento, y por tanto existen dos dependencias multivaluadas:  $\text{modelo} \twoheadrightarrow \text{color}$  y  $\text{modelo} \twoheadrightarrow \text{equipamiento}$ , y la relación se encuentra en 4 FN pero no en FNBC

Para poner el esquema en 4FN se debe descomponer la relación en otras dos de la forma:

**Concesionario1** (modelo, color)

**Concesionario 2** (modelo, equipamiento)



## **6. Quinta Forma Normal (5FN).**

Está basada en la eliminación de las dependencias de reunión. Se dice que existe una dependencia de reunión si la relación puede ser construida en base a la reunión natural de las proyecciones de esa relación sobre los atributos que la forman.

Una relación está en 5FN si y sólo si toda dependencia de reunión en esa relación está implicada por las claves candidatas entre sí, y no por cualquier otro atributo de esa relación.

Por ejemplo, en la relación: **Docencia (dni, asignatura, aula)**

que define las aulas que se asignan a cada asignatura y los alumnos matriculados en la misma (cada alumno matriculado en una asignatura recibe clase en todas las aulas asignadas a la misma), existe dependencia de reunión entre los atributos dni, asignatura y aula.

Para eliminar las dependencias de reunión y ponerlo en 5FN, se descompone la relación en otras tres:

**Docencia1 (dni, asignatura)**

**Docencia2 (asignatura, aula)**

**Docencia3 (dni, aula)**

## **21.4**

## **SQL**

EL SQL (Structured Query Language) es un lenguaje estandarizado de peticiones (query) a bases de datos relacionales. Es el lenguaje de manipulación de bases de datos relacionales más extendido, habiéndose convertido en un estándar de facto.

Permite realizar consultas utilizando los recursos del álgebra relacional combinados con el cálculo relacional de tuplas.

**SQL sea un lenguaje declarativo** en lo que lo importante es definir qué se desea hacer, por encima de cómo hacerlo. Con este lenguaje se pretendía que las instrucciones se pudieran escribir como si fueran órdenes humanas; es decir, utilizar un lenguaje lo más natural posible. De ahí que se le considere un lenguaje de cuarta generación.

La base teórica de SQL es bastante fuerte. Las operaciones funcionan en términos de conjuntos y no de registros individuales. Además no incluye ninguna especificación de localización de los datos o ruta de acceso dejando esta tarea al intérprete del lenguaje.

La primera definición del modelo relacional de bases de datos fue publicada por Codd en 1970. El trabajo de Codd fue inmediatamente desarrollado por empresas y universidades. El SQL fue desarrollado en el centro de investigación de IBM bajo el nombre SEQUEL (Structured English Query Language) en 1974 y 1975. La versión SEQUEL/2 cambió de nombre a SQL por motivos legales. IBM comenzó a trabajar en una implementación de SEQUEL/2 (SQL) llamada System R que estuvo operativa en 1977.

En 1986 se publicó el estándar ANSI del lenguaje que, posteriormente fue adoptado por ISO en 1987, lo que convierte a SQL en estándar mundial como lenguaje de bases de datos relacionales.

En 1989 aparece el estándar ISO (y ANSI) llamado SQL89 o SQL1. En 1992 aparece la nueva versión estándar de SQL (a día de hoy sigue siendo la más conocida) llamada SQL92. En 1999 se aprueba un nuevo SQL estándar que incorpora mejoras que incluyen triggers, procedimientos, funciones,... y otras características de las bases de datos objeto-relacionales; dicho estándar se conoce como SQL99.

El último estándar es el del año 2008 (SQL2008). ISO/IEC 9075-1:2008

### **21.4.1 Partes del lenguaje SQL**

Podemos considerar dos fases en la vida de la base de datos: la etapa de preparación y puesta en marcha y la etapa de explotación. Esta última es el

objetivo final de todo el sistema y las tareas de preparación se realizarán antes de entrar en esa fase de utilidad para la organización.

El lenguaje SQL se compone de un conjunto de instrucciones al igual que cualquier lenguaje tradicional. Debido a la existencia de las dos fases antes descritas se suelen agrupar estas instrucciones en tres “sublenguajes” que en particular en bases de datos relacionales son los siguientes:

- **Lenguaje de Definición de Datos, DDL** (Data Description Language): Es el lenguaje utilizado para la creación y mantenimiento de la estructura de la base de datos. Se utiliza para definir y modificar los esquemas de las relaciones, crear o destruir índices y eliminar relaciones. Permite también la definición de vistas y permisos de acceso para los usuarios. Es el lenguaje que utiliza el administrador de las bases de datos para realizar sus tareas.
- **Lenguaje de Manipulación de datos, DML** (Data Manipulation Language). Incluye todas las instrucciones para realizar consultas a las bases de datos, insertar, modificar o eliminar datos. Éste es el que utilizan los usuarios finales en la fase de explotación de la base de datos. A la parte del DML que permite realizar consultas sobre los datos de la BD se le llama DQL (Data Query Language), pero es una parte del DML.
- **Lenguaje de Control de Datos, DCL** (Data Control Language). Existen una serie de tareas relacionadas con las bases de datos que no están incluidas en ninguno de los dos grupos antes descritos ya que no son propiamente de descripción ni de manipulación de datos. Mediante este lenguaje se establecen las restricciones adicionales necesarias para controlar la privacidad y la integridad de la información almacenada en la base de datos.

Además, los lenguajes comerciales, y SQL en particular, incluyen otras facilidades como son control de principio y final de las operaciones o el bloqueo de datos mientras dura una consulta.

### 21.4.2

### *Modos de trabajo con SQL*

Los modos en que el lenguaje SQL actúa sobre una base de datos son los siguientes.

- Modo interactivo: El usuario de la base de datos establece un diálogo con el Sistema Gestor de Base de Datos (SGBD) a través del intérprete de SQL. De esta forma puede realizar operaciones interactivamente sobre la base de datos introduciendo cualquier sentencia SQL y sin restricción en el orden de éstas.

Las sentencias SQL así introducidas son traducidas por el intérprete de SQL que producirá la solicitud correspondiente para el gestor de la base de datos que la gestionará y generará una respuesta para el usuario.

- Desde un programa: El usuario ejecuta una aplicación sobre el sistema operativo. La aplicación puede ser de dos tipos:
  - o Programa escrito íntegramente en SQL o mejor en extensiones de éste que incluyen las estructuras de programación habituales (bucles y selección). Un ejemplo de esto es PL/SQL de Oracle que es un lenguaje procedural que permite desarrollar programas que acceden a la base de datos vía SQL. Estos programas o módulos de sentencias SQL son en realidad guiones que el intérprete SQL va siguiendo.
  - o Programa escrito en un lenguaje convencional de programación con partes escritas en SQL. Esto es lo que se llama SQL embebido y al lenguaje que contiene el texto SQL se le llama lenguaje anfitrión (host). En este caso las instrucciones del lenguaje de programación se ejecutan por el procedimiento habitual mientras que las sentencias SQL se le pasan a un módulo especial de ejecución del SGBD. Una implementación de SQL embebido establece las relaciones que deben mantener los objetos de la base de datos con los objetos del programa anfitrión y restricciones de funcionamiento. Este modo de trabajo admite dos variantes:

- SQL estático: el programa no admite cambios durante la ejecución. Este es el método utilizado en la mayoría de las aplicaciones.
- SQL dinámico: si durante la ejecución se debe modificar algún parámetro se tiene que utilizar SQL dinámico. Esto resulta menos eficiente que un programa SQL estático y utiliza técnicas dinámicas de manejo de variables, lo que dificulta la tarea de programación.

## **21.5 LENGUAJE Y DEFINICIÓN DE DATOS (DDL)**

Los datos en bases de datos relacionales se almacenan en tablas. Una tabla se compone de filas y columnas o registros y campos correspondientes a entidades y atributos de una relación. A todas las estructuras lógicas de datos se les llama de una forma genérica objetos de las bases de datos.

Normalmente en un SGBD existen varios usuarios que deberán disponer en primer lugar de acceso al sistema operativo de la máquina. Cada usuario para el SGBD estará identificado por un nombre, una palabra clave y unas propiedades. Todo ello es fijado y administrado por el administrador de la base de datos (ABD) que es un usuario especial con derechos sobre todos los objetos.

Lo normal es que las estructuras de datos que van a ser compartidas sean creadas por el administrador pero esto no es un requisito imprescindible dado que puede haber usuarios para los que sea de interés mantener sus propias bases de datos que compartirán con parte o toda la organización.

El creador o propietario de una tabla puede permitir el acceso a su tabla a otros usuarios del sistema pero puede que le interese también permitir el acceso a una parte de la tabla. Para estos casos se creará una vista (view).

En una vista se seleccionan algunos atributos o algunas tuplas de la tabla y se permite que usuarios seleccionados las puedan manejar.

La búsqueda de un registro en una tabla es una búsqueda secuencial. Una forma de agilizar la búsqueda consiste en ordenar los registros según algún criterio y preguntar por los registros según ese orden. Si existe una tabla en la que se anota el valor característico de cada registro y su dirección, la búsqueda se llama indexada y esta tabla auxiliar se llama índice.

En un sistema multiusuario cada usuario tiene un conjunto de objetos que le pertenecen de diferentes tipos como tablas, índices o vistas. A este conjunto se le suele llamar esquema.

Todos los objetos citados hasta ahora deben crearse y configurarse antes de pasar al uso de la base de datos. Estas tareas se realizan con DDL.

Cada objeto de una base de datos tiene un nombre que sirve para localizarlo a lo largo de su vida. Estos nombres tienen un ámbito donde son reconocidos. Cada nombre debe ser único en su ámbito. Los objetos que pertenecen a un esquema (tablas, índices, etc.) tienen ese esquema como ámbito máximo. Por ello un nombre de tabla, por ejemplo, puede repetirse en distintos esquemas de usuarios diferentes pero no dentro del mismo esquema.

Además de estas normas básicas las diferentes implementaciones tienen sus propias reglas sobre ámbitos.

Resumiendo, con un DDL se puede hacer:

- Gestión de tablas
- Gestión de vistas
- Gestión de índices

### **21.5.1**                      ***Objetos de la Base de Datos***

Según los estándares, una base de datos es un conjunto de objetos pensados para gestionar datos. Estos objetos están contenidos en esquemas. Un **esquema** representa la estructura de la base de datos. Los

elementos que incluye un esquema son tablas, dominios, vistas, restricciones, disparadores y otros constructores.

En el estándar SQL existe el concepto de **catálogo** que sirve para almacenar esquemas. Así el nombre completo de un objeto vendría dado por:

catálogo.esquema.objeto

Si no se indica el catálogo se toma el catálogo por defecto. Si no se indica el esquema se entiende que el objeto está en el esquema actual.

### **Tipos de datos y dominios**

Un dominio es un conjunto del cual toma sus valores una columna o atributo de una relación. Según este concepto los tipos de datos predefinidos son dominios.

Algunos de los tipos de datos predefinidos en el estándar son:

- Integer (4 Bytes) y SmallInt (2 Bytes)
- Decimal (precisión, (escala)). Representa un decimal de coma fija. Si se omite la escala se supone 0.
- Float. Representa un decimal de coma variable.
- Char (n) Cadena de caracteres de longitud fija (de n caracteres).
- Varchar (n) Cadena de caracteres de longitud variable (de máximo n caracteres).
- Date (fecha), Time (hora), TimeStamp (fecha y hora).
- Boolean, bit
- CLOB. Representa textos de gran longitud.
- BLOB. Representa binarios de gran longitud.

### **Definición de dominios**

Una definición de dominios es un tipo de datos especializado que se puede utilizar en la definición de columnas. Su sintaxis es la siguiente.

```
CREATE DOMAIN nombre dominio tipo de datos
```

```
[ DEFAULT valor defecto ]
```

```
[ definición de restricciones de dominio ] ;
```

Donde:

- tipo de datos: uno de los proporcionados por SQL
- valor defecto: Especifica el valor por omisión para columnas definidas de este dominio. Será asignado a cada columna con dicho dominio, si no tiene ya su propia cláusula DEFAULT
- definición de restricciones de dominio: implica una restricción que se aplica a toda columna definida sobre el dominio. Se define con la cláusula

```
[CONSTRAINT nombre_restricción] CHECK (expresión condicional)
```

Ejemplo: Enumeración de posibles valores de los colores para un dominio particular

```
CREATE DOMAIN Color VARCHAR(8) DEFAULT 'sinColor'
```

```
CONSTRAINT color_valido
```

```
CHECK (VALUE IN ( 'rojo', 'amarillo', 'azul', 'verde', 'sinColor' )
```

```
);
```

## 21.5.2

## *Gestión de tablas*

Una tabla es un objeto de la base de datos que almacena datos.



Para describir una tabla se utiliza la sentencia CREATE, especificando el nombre y las características de la tabla.

Las características básicas que debe de tener una tabla son:

- Definición de los atributos o columnas
- Restricciones de integridad

Para definir una columna habrá que decir el nombre y tipo de datos que se van a almacenar en ese campo de la tabla

### **Restricciones de integridad**

Se denomina restricción de integridad a una propiedad que debe cumplirse para mantener en las tablas los criterios de diseño.

Las restricciones tienen un nombre para poder ser manipuladas posteriormente y pueden afectar a una tabla entera o a una columna o atributo (restricciones de tabla o restricciones de columna).

La restricción de tabla es alguna característica de mantenimiento de la integridad que se asocia a la tabla al crearla. Esta restricción se aplicará a los valores que contenga la tabla, luego supondrá una validación de los datos al introducirlos o modificarlos.

Las restricciones de columnas indican ciertas características que deben asociarse a la columna que se está describiendo.

En SQL las restricciones de integridad se fijan mediante una sentencia CONSTRAINT.

La sintaxis de restricciones para tablas es la siguiente:

```
[CONSTRAINT nombre_restricción] (  
    [UNIQUE | PRIMARY KEY (atrib1[,atrib2]...)]  
    [FOREIGN KEY (atrib1 [,atrib2]...)  
        REFERENCES tabla(atrib1[,atrib2]...)]
```

```
SET          [ON UPDATE [CASCADE | NO ACTION | RESTRICT |  
              NULL | SET DEFAULT]  
  
SET          [ON DELETE [CASCADE | NO ACTION | RESTRICT |  
              NULL | SET DEFAULT] ]  
  
            [CHECK condición]  
  
        )
```

La sintaxis de restricciones para atributos es la siguiente:

```
[CONSTRAINT nombre_restricción] (  
    [NOT NULL]  
  
    [UNIQUE | PRIMARY KEY]  
  
    REFERENCES tabla(atrib1[,atrib2]...)  
  
    [ON UPDATE [CASCADE | NO ACTION | RESTRICT |  
    SET          NULL | SET DEFAULT]  
  
    [ON DELETE [CASCADE | NO ACTION | RESTRICT |  
    SET          NULL | SET DEFAULT] ]  
  
    [CHECK condición]  
  
    )
```

Las restricciones que se pueden establecer son:

- NOT NULL: se aplica a un campo e indica que no puede contener valores nulos.
- UNIQUE: la(s) columna(s) NO puede contener valores duplicados. Debe declararse como NOT NULL y NO pueden formar parte de la clave primaria.

- **PRIMARY KEY:** denota la(s) columna(s) que son clave principal de la tabla. Tienen valor único y no nulo.
- **CHECK:** se puede aplicar a un atributo o a toda la tabla. Indica una condición que debe satisfacer cada atributo o cada fila de la tabla antes de ser insertada borrada o actualizada.
- **FOREING KEY / REFERENCES:** cuando entre dos tablas se establece una relación las tuplas de una se relacionan con las de la otra mediante ciertos campos clave en cada tabla. La tabla de la que parte la relación se llama tabla primaria y la otra se llama tabla secundaria.

Se llama **clave ajena** (foreing key) al conjunto de atributos de la tabla secundaria que es clave principal en la tabla primaria. Esta última debe ser clave principal o única.

Una relación mantiene la integridad referencial si cumple las siguientes dos condiciones:

- o Toda tupla de la tabla hija está asociada con una tupla de la tabla padre.
- o Si una tupla de la tabla hija no cumple lo anterior, el valor que tiene la columna de la clave ajena es nulo.

La restricción de integridad referencial debe establecerse en la tabla hija. Se usan ciertos parámetros para fijarla

- o **FOREING KEY:** indica que columna o columnas constituyen la clave ajena en una restricción de integridad referencial. Se aplica en la restricción de tabla.
- o **REFERENCES:** especifica la tabla padre. Si no indica a qué clave primaria o única se refiere la clave ajena se entiende que la clave referenciada es la clave primaria de la tabla indicada.

Las siguientes opciones hacen que el gestor mantenga la integridad referencial

- a) ON [DELETE | UPDATE] CASCADE: Borra o actualiza el registro en la tabla padre y automáticamente borra o actualiza los registros coincidentes en la tabla hija. Sin esta cláusula no se permite borrar o actualizar un registro principal que tenga registros secundarios asociados.
- b) ON [DELETE | UPDATE] RESTRICT: No se puede borrar o actualizar un registro en la tabla padre mientras no se borre o actualice el registro secundario asociado en la tabla hija.
- c) ON [DELETE | UPDATE] SET NULL: Borra o actualiza el registro en la tabla padre y establece en NULL la o las columnas de clave foránea en la tabla hija.
- d) ON [DELETE | UPDATE] NO ACTION: Significa ninguna acción en el sentido de que un intento de borrar o actualizar un valor de clave primaria no será permitido si en la tabla referenciada hay un valor de clave foránea relacionado.
- e) ON [DELETE | UPDATE] SET DEFAULT: Borra o actualiza el registro en la tabla padre y establece el valor por defecto de la o las columnas de clave foránea en la tabla hija.

## **1) Creación de una tabla**

La sintaxis básica de la instrucción es la siguiente.

```
CREATE TABLE nombretabla (  
    atributo1 tipo1 [restricciones de atributo]  
    [,atributo2 tipo2 [NOT NULL] [UNIQUE] [DEFAULT valor]  
    [Restricciones De Tabla])
```

La creación de una tabla engloba las definiciones de atributos y/o restricciones:

- a) **La DEFINICIÓN DE ATRIBUTOS** se realiza dando el nombre del atributo (se ajusta a las mismas reglas que los nombres de tablas) y su tipo. Opcionalmente se puede indicar unas restricciones de atributos:
- a. Not null: restricción de valor no nulo.
  - b. Definiciones de restricciones de clave primaria, valor UNIQUE, clave ajena.
  - c. Definición de restricciones generales con la cláusula check.
- b) **La DEFINICIÓN DE RESTRICCIONES DE INTEGRIDAD / SEMÁNTICAS:** Permiten al diseñador restringir el rango de valores de una tabla. Las restricciones pueden ser de columna si afectan a una sola columna, o de tabla si afectan a una o más columnas, tal como se vio en el apartado anterior.

Ejemplo:

```
CREATE TABLE DPTO(  
    DEPTNO    INTEGER(4),  
    DNAME     VARCHAR(14) NOT NULL,  
    LOC       VARCHAR(13) DEFAULT "OURENSE",  
    PRIMARY KEY (DEPTNO)  
)
```

```
CREATE TABLE EMP (  
    EMPNO     INTEGER (4) NOT NULL ,  
    ENAME     VARCHAR(10),  
    JOB       VARCHAR(9),
```

```
MGR          INTEGER(4),  
HIREDATE     DATE,  
SAL          DECIMAL(7,2),  
COMM         DECIMAL(7,2),  
DEPTNO       INTEGER(4);  
  
CONSTRAINT pk_empl PRIMARY KEY (EMPNO),  
FOREIGN KEY (DEPTNO) REFERENCES DEPT (DEPTNO)  
                ON UPDATE CASCADE ON DELETE RESTRICT,  
CHECK SAL > 600,04  
)
```

## 2) Modificación de una tabla

Para modificar una tabla ya creada se utiliza el comando ALTER TABLE con el que pueden especificarse nuevas columnas, nuevas restricciones (ADD) o bien modificarse una columna (MODIFY). La sintaxis es:

```
ALTER TABLE table (  
    [ADD (col1|restric1) (,col2|restric2)...)]  
    [ALTER (col1 tipo1(,col2 tipo2)...)]  
    [DROP CONSTRAINT restricción]  
    [DROP COLUMN columna [CASCADE | RESTRICT] ]  
)
```

- Al añadir una columna a una tabla ya existente hay que tener en cuenta que **NO** está permitido NOT NULL en la definición de una nueva columna, y si se desea introducir un valor para la columna, en cada fila existente, hay que especificarlo con la cláusula DEFAULT al añadir la columna.
- Sólo se puede cambiar el tipo o disminuir el tamaño de una columna si tiene valores nulos en todas las columnas.
- Sólo se pueden borrar restricciones que tengan nombre.
- Al eliminar una columna de una tabla podemos indicar la opción:
  - o CASCADE: elimina la columna y toda restricción o vista que le hace referencia.
  - o RESTRICT: sólo elimina la columna si ninguna vista ni restricción le referencia.

Por ejemplo, la sentencia:

```
ALTER TABLE EMP (  
    ADD ADDRESS VARCHAR(12) DEFAULT "Unknow";  
    ALTER ENAME VARCHAR(30)  
)
```

Modifica la longitud del atributo ENAME y añade el campo ADDRESS.

### **3) Destrucción de una tabla**

La sintaxis para eliminar una tabla es:

```
DROP TABLE nombre_de_tabla [CASCADE | RESTRICT]
```

Para ejecutar esta instrucción se deben tener suficientes privilegios en el sistema o ser el propietario de la tabla.

El parámetro opcional :

- **RESTRICT:** Destruye la tabla sólo si no se le hace referencia desde ninguna otra tabla (clave ajena), ni es tabla base de una vista.
- **CASCADE:** Elimina la tabla junto con las restricciones y vistas que la referencian.

Después de la ejecución de la sentencia DROP cualquier referencia a la tabla dará un error. La tabla se borrará independientemente de que contenga datos o no.

### **21.5.3** *Gestión de vistas*

Una vista es una tabla virtual, es decir, una tabla que no existe físicamente en la base de datos pero aparece al usuario como si existiera. Las vistas no tienen datos almacenados propios, distinguibles y físicamente almacenados. En su lugar, el sistema almacena la definición de la vista (es decir, las reglas para acceder a las tablas base físicamente almacenadas para materializar la vista).

Las vistas tienen varias utilidades:

- Mostrar a los usuarios los datos que les interesan.
- Proteger los datos.
- Reestructurar datos que pueden estar distribuidos en diferentes soportes de manera que aparezcan como una tabla.
- Crear interfaces para aplicaciones que esperan una estructura de tablas distinta a la de creación. Mediante las vistas las aplicaciones se independizan de la estructuración real de los datos.

#### **Creación de una vista**

Se usa la sentencia CREATE VIEW cuya sintaxis es:

```
CREATE [OR REPLACE] VIEW nombre_de_vista [lista_de_campos]
```

AS consulta



Cuando se ejecuta una sentencia de creación de vista se realiza una consulta que selecciona tuplas y atributos de una o varias tablas. Ejemplo: si se le quiere dar una lista de empleados a la empresa de seguridad que controla el acceso al edificio, se utilizará una vista sobre la tabla original:

```
CREATE VIEW EMP_SECURITY  
  
AS      SELECT EMPNO, ENAME  
  
FROM EMP
```

Por defecto, la vista toma los nombres de los atributos seleccionados desde las tablas base, siempre que ningún atributo sea el resultado de una operación aritmética o función de agregado. Si se desea cambiar los nombres de los atributos se utiliza la lista\_de\_campos.

### **Modificación de la estructura de una vista**

La estructura de una vista no puede ser modificada como tal. Lo que se puede hacer es utilizar una sentencia de creación con la cláusula OR REPLACE que sustituirá una vista por otras.

El comando de SQL ALTER VIEW se utiliza para recompilar una vista, Esto se debe hacer cuando se han modificado las tablas bases de la vista para actualizar la vista sobre las nuevas estructuras. Su sintaxis es:

```
ALTER VIEW vista COMPILE
```

Después de esta instrucción toda referencia a la vista desde otro objeto se destruye.

### **Destrucción de una vista**

La instrucción DROP VIEW permite eliminar una vista que haya sido creada anteriormente

```
DROP VIEW view
```

#### 21.5.4

### *Gestión de índices*

En una base de datos un índice es un medio de acceder a los registros de una forma más rápida que con el simple recorrido secuencial de una tabla. El índice es un objeto de la base de datos que contendrá una entrada para cada valor de las columnas indexadas, con la dirección del registro donde debe buscarse ese valor.

Uno de los usos más comunes de los índices es el mantenimiento de una tabla ordenada por distintos criterios.

Sólo es recomendable crear índices para aquellos campos que tengan muchas búsquedas pues el índice ocupa espacio y tiene que actualizarse cada vez que se borra, actualiza o inserta un elemento en una tabla.

#### **Creación de un índice**

La instrucción para crear un índice es CREATE INDEX

CREATE INDEX nombre índice

ON nombre\_tabla (campo{, campo)

[NOSORT]

Se especifica el nombre de la tabla sobre la que se crea el índice, así como el campo o campos sobre el que se indexa. Es posible crear índices concatenados, que se forman con más de una columna. Se emplea en caso de columnas que siempre se consultan juntas.

La opción NOSORT que aparece en algunos sistemas SQL hace que se ahorre tiempo y espacio en la creación de un índice haciendo que si la tabla se ha llenado con registros que están físicamente ordenados con el mismo criterio que el índice se pueda evitar la ordenación que se produce al crear el índice.

CREATE INDEX IND\_EMPRE ON EMPRESA (Abre\_emp)

## **Eliminar un índice**

Para destruir un índice se utiliza la cláusula DROP INDEX

DROP INDEX índice

## **21.6 LENGUAJE DE MANIPULACIÓN DE DATOS (DML)**

Se llaman manipulaciones a aquellas operaciones sobre una base de datos que no afectan a la estructura de ésta sino a su contenido. Estas operaciones se realizan con DML (Data Manipulation Language). Las manipulaciones posibles sobre una base de datos son las siguientes:

- Insertar valores en tuplas (INSERT)
- Eliminar una tupla (DELETE)
- Actualizar el valor de un campo en una o varias tuplas (UPDATE)
- Consultar o listar todos o algunos campos de un grupo de tuplas (SELECT)

Las operaciones de manipulación se pueden hacer tanto en tablas como en vistas ya que éstas no son sino tablas lógicas.

### **21.6.1      *Inserción de valores en una tabla***

Para insertar una fila o tupla en una tabla ya creada se utiliza el comando INSERT cuya sintaxis es:

```
INSERT INTO nombre_tabla [columna (,columna)*]  
VALUES (valor (,valor)*)}
```

Si no se especifican nombres de columnas, los valores insertados deben corresponder en cantidad y tipo de datos con los atributos de la tabla y tienen que estar en el mismo orden con el que se creó la tabla.

Se pueden especificar unas columnas y otras no, teniendo en cuenta que las columnas no especificadas tomarán el valor por defecto, si se definió, o NULL.

```
INSERT INTO DEPT (DEPTNO, DNAME, LOC)
VALUES (90, 'CONTABILIDAD', 'OURENSE')
```

Para insertar varias tuplas con una sola instrucción se puede utilizar una subconsulta que devuelva tuplas de estructura compatible con la de la tabla:

```
INSERT INTO nombre_tabla [columna (,columna)*]
```

consulta

### **21.6.2 Borrado de valores de una tabla**

Para borrar una fila se utiliza el comando DELETE cuya sintaxis es la siguiente:

```
DELETE FROM nombre_tabla
```

```
[WHERE condición]
```

Con esta instrucción se borran todas las tuplas que cumplan la condición WHERE. Si no se incluye ésta se borran todos los elementos de la tabla. Por ejemplo, para borrar el departamento "CONTABILIDAD":

```
DELETE FROM DEPT WHERE DNAME = "CONTABILIDAD"
```

### 21.6.3 *Modificación de valores de una tabla*

Para modificar los valores de determinadas tuplas de una tabla se utiliza la sentencia UPDATE con la siguiente sintaxis:

```
UPDATE nombre_tabla  
  
    SET columna1=valor1{,columna2=valor2}*}  
  
    [WHERE condición]
```

Con la cláusula SET se especifica las columnas a modificar con sus nuevos valores y con la cláusula WHERE se seleccionan las filas a actualizar. Si no hay WHERE, se aplica la modificación a todas las filas. Por ejemplo, si se quiere conceder a todo empleado del departamento de Informática un aumento salarial del 18%.

```
UPDATE EMP  
  
    SET SAL = SAL*1.18  
  
    WHERE DEPTNO IN (SELECT DEPTNO  
  
                     FROM DEPT  
  
                     WHERE DNAME='INFORMATICA')
```

### 21.6.4 *Consulta de datos*

Una consulta sirve para extraer los datos almacenados en una base de datos. La consulta en SQL consta de tres partes:

- Cláusula SELECT: para indicar qué atributos se desean consultar
- Cláusula FROM: indica sobre qué relación o relaciones se quiere hacer la consulta

- Cláusula WHERE: indica las condiciones que deben cumplir las tuplas para ser seleccionadas

Su sintaxis abreviada es la siguiente:

```
SELECT * | {[DISTINCT] columna | expresión [[AS] alias], ...}
```

```
FROM tablas
```

```
[WHERE condiciones_where]
```

```
[GROUP BY columnas_group]
```

```
[HAVING condiciones_having]
```

```
[ORDER BY columnas_orden]
```

El efecto de una consulta como ésta es el siguiente:

- Se realiza el producto cartesiano de las relaciones citadas en la cláusula FROM
- Se aplica el operador selección del álgebra relacional para seleccionar aquellas tuplas del producto cartesiano que hagan verdadero el predicado WHERE
- Se proyecta el resultado obtenido sobre los atributos especificados en SELECT

### **Selección de atributos**

La consulta más sencilla es seleccionar todas las tuplas de una tabla. Por ejemplo, seleccionar todos los datos de todos los empleados.

```
SELECT EMPNO, ENAME, JOB, MGR, HIREDATE, SAL, COMM,  
DEPTNO  
  
FROM EMP
```

Se utiliza el asterisco (\*) como comodín para seleccionar todos los campos. La sentencia anterior es equivalente a:

```
SELECT * FROM EMP;
```

Se pueden seleccionar columnas individuales: "Lista todos los salarios"

```
SELECT SAL FROM EMP
```

En una tabla, algunas columnas tendrán valores repetidos. Si solo se quieren mostrar los valores diferentes de una columna en una tabla hay que usar la palabra clave **DISTINCT** antepuesta al nombre de la columna

```
SELECT DISTINCT SAL FROM EMP
```

Se puede cambiar el nombre que se le da a la cabecera de la columna en el resultado de la instrucción SELECT. Para ello se utiliza un alias con el comando AS después del nombre de la columna.

```
SELECT SAL AS Salario FROM EMP
```

## Orden

Por defecto SQL no ordena los resultados, para ordenarlos, se utiliza la cláusula ORDER BY:

```
ORDER BY campo1 [ASC|DESC], campo2 [ASC|DESC], ...
```

El modo de ordenación se indica para cada campo:

- ASC ordena ascendente

- DESC ordena descendente

### **Consulta con expresiones**

También se pueden realizar consultas en las que se evalúe una expresión.

Por ejemplo, si se dispone de una tabla con el salario bruto anual de los empleados de una empresa, es posible consultar su sueldo neto mensual. Suponiendo que se paguen 14 pagas anuales y que se conozca el tipo de IRPF la instrucción para realizar esta consulta sería:

```
SELECT ENAME, (SAL/14)(1-IRPF/100) FROM EMP
```

### **Condiciones para restringir la consulta**

Para restringir las tuplas que se obtienen, se pueden imponer condiciones. Para ello se usa la cláusula WHERE que admite los siguientes operadores

Operadores Relacionales (comparación): >, >=, <, <=, =, <>: Estos operadores se utilizan para comparar datos.

Operadores lógicos: AND, OR, NOT: Se utilizan para unir condiciones.

```
SELECT * FROM EMP  
  
WHERE DEPTNO = 99  
  
AND SAL > 1250;
```

Además, las condiciones unidas por AND, OR y NOT admiten paréntesis. Si no se ponen paréntesis la prioridad, de mayor a menor, es NOT, AND y OR.

### **Consulta de pertenencia a un rango**

Se puede comprobar si una expresión entra o no dentro de un rango marcado. Se utiliza el operador BETWEEN. La sintaxis es:



expresión[NOT] BETWEEN expresión [AND expresión]

```
SELECT * FROM EMP
```

```
WHERE EMPNO BETWEEN 9 AND 54
```

### **Consultas de pertenencias a una lista**

Con el operador IN se comprueba si un elemento pertenece a una lista de valores. La sintaxis de la condición es:

elemento[NOT] IN lista\_expresiones | subconsulta

Ejemplo:

```
SELECT * FROM EMP
```

```
WHERE ENAME IN('Pepe Martínez' y 'Josefa Martín')
```

Selecciona de la tabla de empleados los datos de 'Pepe Martínez' y 'Josefa Martín'.

### **Consulta con patrones**

Con la utilización del operador LIKE se puede buscar una cadena de caracteres dentro de otra. La sintaxis es la siguiente:

<cadena>[NOT] LIKE <cadena>

Es una comparación de igualdad pero admite comodines:

- %: se puede sustituir por cualquier número de caracteres (0 o más).
- Los SGBDR también admiten sustituir un único carácter.

Ejemplo: Para seleccionar aquellos elementos de la tabla ALUMNO cuyo campo Nombre empiece por la cadena “Ma”

```
SELECT * FROM EMP WHERE ENAME LIKE 'Ma%'
```

## Funciones de agregación

Existen funciones que permiten calcular, desde una sentencia SQL, sumas, medias aritméticas, etc. de datos. Muchas de estas funciones aceptan el parámetro DISTINCT|ALL. Si toma el valor ALL (valor por defecto) indica que deben considerarse todas las apariciones aunque sean repetidas y si es DISTINCT deben ignorarse las repeticiones. Algunas de las más importantes funciones son:

- AVG (Atributo): media aritmética de los valores de atributo.
- MIN (Atributo): valor mínimo de los valores de atributo.
- MAX (Atributo): valor máximo de los valores de atributo.
- SUM (Atributo): suma los valores de atributo.
- COUNT (Atributo): cuenta el número de filas donde atributo no es nulo.
- COUNT (\*): cuenta el número de filas incluyendo aquellas con nulos.
- LCASE (Atributo): transforma Atributo a mayúsculas.
- UCASE (Atributo): transforma Atributo a minúsculas.
- MID (Atributo, m [, n]): devuelve una porción de Atributo comenzando en el carácter m y con n caracteres de longitud.
- LEN (Atributo): devuelve la longitud de Atributo.

```
SELECT AVG (SAL) FROM EMP
```

```
SELECT COUNT (*) FROM EMP
```

## **Consulta con agrupamiento de filas**

Las funciones de agregación se suelen utilizar combinadas con la cláusula de agrupamiento GROUP BY, que agrupa el resultado por una serie de atributos.

Una instrucción SELECT con este parámetro devuelve grupos de tuplas en lugar de tuplas individuales. Como resultado de la consulta aparecerá un resumen de la información por cada grupo en lugar de todas las filas.

Por ejemplo para listar los números de departamento y la suma de los salarios de cada uno de ellos se utilizaría:

```
SELECT DEPTNO, SUM(SAL)
FROM EMP
GROUP BY DEPTNO
```

La expresión de un GROUP BY puede contener referencias a cualquier campo de las tablas nombradas en FROM. Sin embargo la lista de expresiones que siguen al SELECT no puede contener más que:

- Constantes
- Funciones de grupo (AVG, MAX, MIN, COUNT, SUM)
- Expresiones idénticas a las de la cláusula GROUP BY
- Expresiones que devuelvan el mismo valor para todas las tuplas que formen parte de un grupo.

No se pueden seleccionar atributos que no se puedan agrupar por los atributos indicados en el GROUP BY.

Si en una consulta aparece una cláusula WHERE y una GROUP BY primero se seleccionarán las tuplas que cumplan la condición del WHERE y después se aplica el agrupamiento.

### Restricciones en los agrupamientos

Cuando se selecciona un conjunto de atributos agrupados por uno o más atributos, se pueden imponer condiciones a los grupos (es decir, condiciones a los atributos que se están seleccionando). Es la cláusula HAVING, que sería el equivalente a la cláusula WHERE pero aplicada a los grupos. Por ejemplo,

“Lista la suma de los sueldos agrupada por departamentos, pero sólo aquellos en los que la suma sea mayor que 7.000”

```
SELECT SUM(SAL), DEPTNO  
  
FROM EMP  
  
GROUP BY DEPTNO  
  
HAVING SUM(SAL)>7.000
```

### **21.6.5 Consultas sobre múltiples tablas.**

Es más que habitual necesitar en una consulta datos que se encuentran distribuidos en varias tablas. Las bases de datos relacionales se basan en que los datos se distribuyen en tablas que se pueden relacionar mediante un campo.

Para realizar consultas a más de una tabla, basta con indicar en la cláusula FROM las tablas separadas por comas y añadir las condiciones necesarias en la cláusula WHERE.

```
SELECT ENAME, DNAME  
  
FROM EMP, DEPT
```

Este ejemplo realiza el producto cartesiano de la tabla EMP y DEPT y devolvería para cada registro de la tabla EMP, todos los registros de la tabla DEPT.

Si se quiere hacer correctamente, asociando nombre de trabajador con el departamento en el que trabaja, se utiliza un criterio de comparación por la clave ajena, por ejemplo:

```
SELECT ENAME, DNAME  
  
FROM EMP, DEPT  
  
WHERE EMP.DEPTNO = DEPT.DEPTNO
```

Los nombres de los atributos, en caso de que las tablas tengan atributos con el mismo nombre, irán precedidos por el nombre de la tabla y un punto, como en EMP.DEPTNO. Si no existe confusión posible pueden indicarse sin el nombre de la tabla, como por ejemplo DNAME, que sólo existe en la tabla DEPT.

Para evitar repetir continuamente el nombre de las tablas, se puede especificar un alias, añadiendo al nombre de la tabla en la clausula from el alias.

A partir de la versión SQL 1999 se ideó una nueva sintaxis para consultar varias tablas. La razón fue separar las condiciones de asociación respecto de las condiciones de selección de registros. La sintaxis completa es:

```
SELECT    tabla1.column1,    tabl1.column2,...    tabla2.column1,  
tabla2.column2,...
```

```
FROM tabla1
```

```
[CROSS JOIN tabla2] |
```

```
[NATURAL JOIN tabla2] |
```

```
[JOIN tabla2 USING (columna)] |
```

```
[JOIN tabla2 ON (tabla1.columa=tabla2.columna)] |
```

```
[LEFT|RIGHT|FULL    OUTER    JOIN    tabla2    ON
```

```
(tbl1.colum=tbl2.column)]
```

- **CROSS JOIN.** Realiza un producto cruzado entre las tablas indicadas. Eso significa que cada tupla de la primera tabla se combina con cada tupla de la segunda tabla. Es decir si la primera tabla tiene 10 filas y la segunda otras 10, como resultado se obtienen 100 filas, resultado de combinar todas entre sí.

```
SELECT ENAME, DNAME  
  
FROM EMP CROSS JOIN DEPT
```

- **NATURAL JOIN.** Establece una relación de igualdad entre las tablas a través de los campos que tengan el mismo nombre en ambas tablas:

```
SELECT ENAME, DNAME  
  
FROM EMP NATURAL JOIN DEPT
```

En ese ejemplo se obtienen la lista de los empleados y los nombres de los departamentos a los que pertenecen a través de los campos que tengan el mismo nombre en ambas tablas. Hay que asegurarse de que sólo son las claves principales y secundarias de las tablas relacionadas, las columnas en las que el nombre coincide, de otro modo fallaría la asociación y la consulta no funcionaría.

- **JOIN USING.** Permite establecer relaciones indicando qué columna (o columnas) común a las dos tablas hay que utilizar. Las columnas deben de tener exactamente el mismo nombre en ambas tablas:

```
SELECT ENAME, DNAME  
  
FROM EMP JOIN DEPT USING (DEPTNO)
```

- **JOIN ON** Permite establecer relaciones cuya condición se establece manualmente, lo cual es útil para asociaciones cuyos campos en las tablas no tienen el mismo nombre:

```
SELECT ENAME, DNAME  
  
FROM EMP e JOIN DEPT d ON (e.DEPTNO=d.DEPTNO)
```

- **OUTER JOIN** Utilizando las formas vistas hasta ahora de relacionar tablas sólo aparecen en el resultado de la consulta filas presentes en las tablas relacionadas. Es decir en la consulta anterior sólo aparecen empleados relacionados con la tabla de departamentos. Si hay empleados que no están en departamentos, éstos no aparecen (y si hay departamentos que no están en la tabla de empleados, tampoco salen).
- Para solventar esto, se utilizan relaciones laterales o externas (outer join):
  - o `tabla1 LEFT OUTER JOIN tabla2 ON`. Obtiene los datos de la tabla1 estén o no relacionados con datos de la tabla 2.
  - o `tabla1 RIGHT OUTER JOIN tabla2 ON`. Obtiene los datos de la tabla2 estén o no relacionados con datos de la tabla 1.
  - o `tabla1 FULL OUTER JOIN tabla2 ON`. Obtiene los registros no relacionados de ambas tablas.

### 21.6.6

### Subconsultas

Son sentencias SELECT que se encuentra anidadas dentro de otras SELECT. Estas sentencias se escriben entre paréntesis para advertir al gestor que se deben de ejecutar primero. Permite solucionar consultas que requieren para funcionar el resultado previo de otra consulta.

```
SELECT ENAME  
  
FROM EMP  
  
WHERE SAL >= (SELECT AVG(SAL) FROM EMP)
```

Da como resultado el listado de los empleados que superan la media de sueldo. Primero se realiza la operación del SELECT de la cláusula Where y seguidamente se ejecuta el SELECT del principio.

Una subconsulta que utilice los valores >,<,>=,... tiene que devolver un único valor, de otro modo ocurre un error. Además tienen que tener el mismo tipo de columna para relacionar la subconsulta con la consulta que la utiliza (no puede ocurrir que la subconsulta tenga dos columnas y ese resultado se compare usando una sola columna en la consulta general).

### **La cláusula (NOT) EXISTS**

La cláusula EXISTS (o NOT EXISTS) comprueba si una subconsulta devuelve algún valor (EXISTS) o no devuelve ninguno (NOT EXISTS). Por ejemplo:

“Lista los departamentos que no hayan contratado a nadie el 28 de diciembre de 2010”

```
SELECT D.DNAME
FROM DEPT D
WHERE NOT EXISTS
  (SELECT * FROM EMP E
   WHERE E.DEPTNO=D.DEPTNO
   AND HIREDATE ='28/12/2010')
```

La consulta de primer nivel busca en la tabla de departamentos los nombres y, para cada fila, comprueba —mediante la subconsulta— que para ese número de departamento no existan empleados que hayan sido contratados el 28 de diciembre de 2010.

### **Consulta con cuantificadores**



Se denominan cualificadores a ciertos predicados que permiten utilizar subconsultas que devuelven varias filas en la columna correspondiente a un atributo.

Por ejemplo si se quiere mostrar el sueldo y nombre de los empleados cuyo sueldo supera al de cualquier empleado del departamento de ventas. La subconsulta necesaria para ese resultado mostraría todos los sueldos del departamento de ventas. Pero no podremos utilizar un operador de comparación directamente ya que esa subconsulta devuelve más de una fila. La solución a esto es utilizar los cuantificadores entre el operador y la consulta, que permiten el uso de subconsultas de varias filas.

La sintaxis de uso dentro de condiciones es la siguiente:

expresión operador\_relacional cuantificador {lista\_exps | subconsulta}

Los cuantificadores son:

- **ANY o SOME:** La comparación con un ANY (o SOME , equivalente) es verdadera si lo es para algún valor de los obtenidos con una subconsulta.
- **ALL.** En este caso la comparación es verdadera si lo es con todos los valores devueltos por la consulta subordinada y falsa en caso contrario. Por ejemplo: para saber el empleado con el sueldo más alto de toda la empresa.

SELECT ENAME

FROM EMP

WHERE SAL > = ALL(SELECT SAL FROM EMP)

### 21.6.7 *Operaciones con consultas*

Existen ciertos operadores que permiten combinar los conjuntos de tuplas que se obtiene de dos consultas SELECT y obtener un nuevo conjunto de tuplas. Estas operaciones corresponden con operadores del álgebra relacional y son los siguientes:

- UNION y UNION ALL realizan la unión de las tuplas obtenidas por dos consultas que se especifican como operandos. UNION no incluye las tuplas repetidas mientras que UNION ALL sí las incluye. UNION corresponde a la unión de relaciones del álgebra relacional.

Para ello ambas instrucciones tienen que utilizar el mismo número y tipo de columnas

```
SELECT nombre FROM empleados
```

### **UNION**

```
SELECT nombre FROM visitantes
```

Esto crea una tabla que incluye los nombres de los empleados y visitantes

- INTERSECT permite unir dos consultas SELECT de modo que el resultado serán las filas que estén presentes en ambas consultas. Equivale al operador intersección del álgebra relacional.

```
SELECT tipo,modelo FROM productos
```

```
WHERE chip="QWER-21"
```

### **INTERSECT**

```
SELECT tipo,modelo FROM productos
```

```
WHERE chip="WDFV-23"
```

- MINUS combina dos consultas SELECT de forma que aparecerán los registros del primer SELECT que no estén presentes en el segundo. Corresponde a la diferencia del álgebra relacional.

```
SELECT tipo,modelo FROM productos
```

```
WHERE chip="QWER-21"
```

### **MINUS**

```
SELECT tipo,modelo FROM productos
```

```
WHERE chip="WDFV-23"
```

Las dos consultas sobre las que se aplique cualquiera de estos operadores deben devolver tuplas con la misma estructura.

## **21.7 . LENGUAJE DE CONTROL DE DATOS DCL**

Con el nombre de lenguaje de control de datos (DCL Data Control Language) se hace referencia a la parte del lenguaje SQL que se ocupa de los apartados de seguridad y de la integridad en el procesamiento concurrente.

### **21.7.1 Seguridad**

El lenguaje SQL supone un nivel general de seguridad del software gestor de la base de datos y sus sentencias se utilizan para especificar restricciones de seguridad. El esquema de seguridad SQL se basa en tres conceptos:

- Los usuarios son los actores de la base de datos. Cada vez que el gestor de la base de datos recupera, inserta, suprime o actualiza datos, lo hace a cuenta de algún usuario.
- Los objetos de la base de datos son los elementos a los cuales se puede aplicar la protección de seguridad SQL. La seguridad se aplica generalmente a tablas, vistas y columnas
- Los privilegios son las acciones que un usuario tiene permitido efectuar para un determinado objeto de la base de datos.

La creación y eliminación de usuarios en SQL no es estándar, dependiendo de cada producto comercial. Sin embargo, la concesión y revocación de privilegios si es estándar y está recogida en las sentencias GRANT y REVOKE.

#### **Concesión de privilegios: GRANT**

La sentencia GRANT se utiliza para conceder privilegios de seguridad sobre objetos de la base de datos a usuarios específicos. Normalmente la sentencia GRANT es utilizada por el propietario de la tabla o vista para proporcionar a otros usuarios acceso a los datos. La sentencia GRANT incluye una lista específica de los privilegios a conceder, el nombre del objeto al cual se aplican los privilegios y la lista de usuarios a los cuales se conceden los privilegios. La sintaxis es la siguiente:

GRANT listaPrivilegios

ON listaObjetos

TO listaUsuarios

[WITH GRANT OPTION]

- ListaPrivilegios: Se conceden todos (ALL) o un subconjunto de privilegios (separados por comas) que permiten borrar, insertar, consultar, actualizar o modificar (DELETE, INSERT, SELECT, UPDATE, ALTER) una tabla, vista o un conjunto de ellas.
- ListaObjetos: Objetos a los que se le aplican los privilegios.
- ListaUsuarios: Se conceden a todos los usuarios (PUBLIC) o a una lista de usuarios.
- WITH GRANT OPTION: Indica que aquellos usuarios a los que se ha concedido estos privilegios pueden a su vez concederlos (nunca más de los que se tienen actualmente) a otros usuarios por medio de sentencias GRANT.

Por norma general los privilegios de acceso se aplican sobre todas las columnas en la tabla o vista, pero también se puede especificar una lista de columnas con el privilegio UPDATE.

GRANT UPDATE (SAL) ON EMP TO grupoNominas

Sólo el propietario de un objeto puede conceder los privilegios del mismo. El propietario es siempre el creador del mismo.

Las operaciones con el esquema de la base de datos (CREATE, DROP, etc.) sólo pueden ser realizadas por el propietario del esquema.

### **Revocación de privilegios: REVOKE**

Los privilegios que se han concedido con la sentencia GRANT pueden ser retirados con la sentencia REVOKE. La sentencia REVOKE tiene una estructura que se asemeja estrechamente a la sentencia GRANT, especificando un conjunto específico de privilegios a ser revocados, para un objeto de la base de datos específico, para uno o más usuarios. Una sentencia REVOKE puede retirar todos o parte de los privilegios que previamente se han concedido a un usuario. Es necesario especificar que un usuario sólo puede retirar los privilegios que él mismo ha concedido a otro usuario.

REVOKE ListaPrivilegios

ON ListaObjetos

FROM ListaUsuarios

La utilización de vistas combinada con una definición de usuarios y concesión juiciosa de privilegios constituye el mecanismo de seguridad que el administrador de la base de datos SQL utiliza para llevar a cabo las políticas de seguridad del sistema.

## **21.7.2**

### ***Transacciones***

Se entiende por transacción el efecto producido por un grupo de instrucciones DML ejecutadas una tras otra, es decir, una transacción es un conjunto de acciones que o bien se realizan todas, o bien no se realiza ninguna.

En SQL una transacción comienza implícitamente en la primera instrucción que altera el estado de la información almacenada en la base de datos. Para preservar las propiedades ACID (Atomic, Consistent, Isolate, Durable) de una transacción, SQL dispone de dos sentencias que permiten que los cambios realizados por una transacción queden reflejados permanentemente en la base de datos (comprometer)

- COMMIT [WORK]. Termina la transacción actual grabando permanentemente las modificaciones.
- ROLLBACK [WORK]. Obliga al sistema a volver al estado anterior al inicio de la transacción.

También es posible que cada entorno de programación y/o SGBD disponga de elementos adicionales para el control de concurrencia que puedan ser utilizados por el usuario, como por ejemplo bloqueos.

Desde el momento en que a una base de datos pueden acceder diferentes usuarios al mismo tiempo, en cada instante podremos tener distintas transacciones que manipulen la base de datos a la vez.

Las transacciones especifican un nivel de aislamiento que define el grado en que se debe aislar una transacción de las modificaciones de recursos o datos realizadas por otras transacciones. En teoría, toda transacción debe estar completamente aislada de otras transacciones, pero en la realidad, por razones prácticas, esto puede no ser cierto siempre. Los niveles de aislamiento se describen en cuanto a los efectos secundarios de la simultaneidad que se permiten, como las lecturas desfasadas o ficticias.

El estándar SQL define cuatro niveles de aislamiento transaccional en función de tres eventos que son permitidos o no dependiendo del nivel de aislamiento. Estos eventos son:

- *Lectura sucia*. Las sentencias SELECT son ejecutadas sin realizar bloqueos, pero podría usarse una versión anterior de un registro. Por lo tanto, las lecturas no son consistentes al usar este nivel de aislamiento.

- *Lectura no repetible.* Una transacción vuelve a leer datos que previamente había leído y encuentra que han sido modificados o eliminados por una transacción cursada.
- *Lectura fantasma.* Una transacción vuelve a ejecutar una consulta, devolviendo un conjunto de registros que satisfacen una condición de búsqueda y encuentra que otros registro que satisfacen la condición han sido insertadas por otra transacción cursada.

Los niveles de aislamiento SQL se definen en base a si permiten cada uno de los eventos definidos anteriormente.

Niveles de aislamiento:

Nivel de aislamiento	Comportamiento Permitido			
		Lect. Sucia	Lect. No Repetible	Lect. Fantasma
Lectura confirmada	no	SI	SI	SI
Lectura confirmada		NO	SI	SI
Lectura repetible		NO	NO	SI
Serializable		NO	NO	NO

La sentencia para controlar el nivel de aislamiento en SQL es:

```
SET TRANSACTION ISOLATION LEVEL {READ UNCOMMITTED |READ
COMMITTED| REPEATABLE READ|SERIALIZABLE }
```

- **READ UNCOMMITTED.** Especifica que las instrucciones pueden leer filas que han sido modificadas por otras transacciones pero todavía no se han confirmado.

- **READ COMMITTED.** Especifica que las instrucciones no pueden leer datos que hayan sido modificados.
- **REPEATABLE READ.** Especifica que las instrucciones no pueden leer datos que han sido modificados pero aún no confirmados por otras transacciones y que ninguna otra transacción puede modificar los datos leídos por la transacción actual hasta que ésta finalice.
- **SERIALIZABLE.** Especifica que las instrucciones no pueden leer datos que hayan sido modificados, pero aún no confirmados, por otras transacciones. Ninguna otra transacción puede modificar los datos leídos por la transacción actual hasta que la transacción actual finalice.



## 21.8

## **BIBLIOGRAFÍA**

- Codd, E.F. "A Relational Model of Data for Large Shared Data Banks". In: Communications of the ACM 13 (6): 377-387, 1970.
- DAFTG of the ANSI/X3/SPARC Database System Study Group, "Reference Model for DBMS Standardization". Sigmod Record, Vol.15, No.1, March 1986)
- de Miguel y M. Piattini. "Fundamentos y Modelos de Bases de Datos". Ed. RA-MA. 1999. ISBN 978-84-78-97361-3
- M. Piattini, E. Marcos, C. Calero y B. Vela. "Tecnología y Diseño de Bases de Datos". Ed. RA-MA 2006. ISBN 978-847-897733-8
- Nguyen Viet Cuong, "XML Native Database Systems Review of Sedna, Ozone, NeoCoreXMS". 2006.
- Jim Gray, "Transaction Processing: Concepts and Techniques". Ed. Morgan Kaufman, 1992. ISBN 15 586 0190 2
- The Entity/Relationship Model: Toward a unified view of data. CACM, 1,1. 1976
- The Entity/Relationship Model: A basis for the enterprise view of data. AFIPS Conference Proceedings, Vol 46. 1977
- Introducción a los sistemas de bases de datos. C.J. Date. Pearson Educación, 2001.
- Fundamentos de Sistemas de Bases de Datos. Ramez A. Elmasri & Shamkant B. Navathe. Addison-Wesley, 2002 [3ª edición].
- Connolly & Begg. (2005). Sistemas de bases de datos. Un enfoque práctico para diseño, implementación y gestión. Pearson Addison Wesley. Madrid.

- Kroenke. (2002). Procesamiento de Bases de Datos. Fundamentos, Diseño e Implementación. Octava Edición. Pearson. Prentice Hall.
- Piattiani, Esparza Marcos, Calero Coral & Vela Belen.(2007). Tecnología y diseño de Bases de Datos. AlfaOmega Ra-Ma. México.

**Autor:** Francisco Javier Rodríguez Martínez

Subdirector de Sistemas Escola Superior Enxeñaría Informática Ourense

Colegiado del CPEIG

## **22. SISTEMAS DE GESTIÓN DE CONTENIDOS. SISTEMAS DE GESTIÓN DOCUMENTAL. GESTIÓN DEL CONOCIMIENTO. LOS SISTEMAS DE INFORMACIÓN GEOGRÁFICA (SIX).**

## **Tema 22: Sistemas de gestión de contenidos. Sistemas de gestión documental. Gestión del conocimiento. Los sistemas de información geográfica (SIG).**

---

### **ÍNDICE**

<b>22.1.Sistemas de Gestión de Contenidos.....</b>	<b>2</b>
22.1.1.Introducción a los Sistemas de Gestión de Contenidos.....	2
22.1.2.Funcionalidad de los SGC.....	2
22.1.3.Arquitectura general de los Sistemas de Gestión de Contenidos.....	3
22.1.4.Categorías.....	3
22.1.5.Criterios de valoración.....	3
22.1.6.JOOMLA!.....	4
22.1.7.WORDPRESS.....	5
22.1.8.DRUPAL.....	6
<b>22.2.Sistemas de Gestión Documental.....</b>	<b>7</b>
22.2.1.Definición de Sistemas de Gestión Documental.....	7
22.2.2.Funciones de la Gestión documental.....	8
22.2.3.Ciclo de Vida de los Documentos.....	9
22.2.4.Beneficios de la Gestión Documental.....	10
<b>22.3.Gestión del Conocimiento.....</b>	<b>11</b>
22.3.1.Definición de Gestión del Conocimiento.....	11
22.3.2.Cuestiones sobre gestión del conocimiento.....	11
<b>22.4.Sistemas de Información Geográfica SIG.....</b>	<b>12</b>
22.4.1.Arquitectura de un SIG.....	13
22.4.2.Clasificación de los SIG.....	14
22.4.3.Áreas de Aplicación.....	15
<b>22.5.Bibliografía.....</b>	<b>17</b>

## **22.1. Sistemas de Gestión de Contenidos**

### **22.1.1. Introducción a los Sistemas de Gestión de Contenidos.**

Los sistemas de gestión de contenidos, en adelante SGC (CMS en inglés), son un tipo especial de software orientado a la creación, administración y distribución de contenidos digitales. Los SGC proporcionan una estructura o framework para dar soporte a tareas básicas y complejas de gestión de contenido. Están principalmente orientados para servir como marco de publicación de contenidos en la red a través de web. El éxito de este tipo de sistemas radica fundamentalmente en la facilidad de uso, estableciendo mecanismos sencillos para la creación de contenidos, su actualización, su administración y categorización, y su publicación. Proporcionar facilidad en el manejo de contenidos implica otorgar un mayor dinamismo en el flujo de la información.

Una de sus principales características es que permiten separar el contenido de la presentación, cuestión que proporciona versatilidad a la hora de realizar cambios en el diseño. Además, aportan herramientas que permiten descentralizar la publicación de contenidos en la Web.

Un aspecto clave en la gestión de contenidos es la categorización de la información. La capacidad de establecer mecanismos que permitan localizar la información útil es otra de las características propias de los sistemas de gestión de contenidos. Esta capacidad se basa en el uso de metadatos que sirven para proporcionar información añadida a los contenidos publicados, y que son utilizados por los buscadores y clasificadores de información.

### **22.1.2. Funcionalidad de los SGC**

La funcionalidad general ofrecida por los SGC puede agruparse en cinco bloques fundamentalmente:

- **Creación de contenido:** Se realiza de forma sencilla. Los CMS aportan herramientas para que los creadores sin conocimientos técnicos en páginas web pueden concentrarse en el contenido. La forma habitual consiste en proporcionar un editor de texto WYSIWYG, en el que el usuario ve el resultado final mientras escribe. El uso de este tipo de editores es muy sencillo. El acceso a los mismos es muy cómodo, ya que sólo se requiere para ello un equipo con acceso a Internet y un navegador web.
- **Gestión de contenido:** Todo el contenido creado se almacena en la base de datos que utiliza el sistema. En la propia base de datos es en donde también se guardan datos relacionados con la estructura de la web o los usuarios autorizados.
- **Gestión de usuarios:** La mayoría de los SGC presentan una gestión de usuarios en la que cada uno cuenta con diferentes permisos para gestionar el contenido. Dependiendo de los permisos de usuarios, se pueden encontrar distintos roles que van desde el administrador general de la plataforma, hasta el usuario final que consulta la información.
- **Publicación de contenido:** Una vez creado el contenido, los SGC proporcionan diferentes mecanismos para proceder a su publicación. Se le puede asignar una fecha de publicación o bien se puede publicar directamente. En la publicación del contenido, el aspecto que tendrá viene marcado por el patrón marcado para la sección donde se encuentre la información, que habitualmente se corresponde con un conjunto de estilos predefinidos. Esta separación

entre contenido y forma representa una característica muy importante de los SGC dado que permite que se pueda modificar el estilo de un portal web sin necesidad de modificar el contenido.

- **Presentación de contenido:** Los SGC gestionan automáticamente la accesibilidad del web, proporcionando mecanismos de adaptación a las necesidades de cada usuario y además son perfectamente compatibles con la mayoría de los navegadores web existentes. El sistema se encarga de gestionar otros aspectos como los menús de navegación, añadiendo enlaces de forma automática. También gestionan todos los módulos, internos o externos, que sean incorporados al sistema.

Dentro del ciclo de actividades que se corresponde con la funcionalidad de los CMS, es necesario definir un conjunto de roles o usuarios a los que se asocian una serie de tareas:

1. **Autor:** Que puede ser cualquier miembro usuario del sistema que desee publicar contenido.
2. **Publicador:** que revisa esa información y autoriza su publicación en tiempo y forma adecuadas.
3. **Administrador del Sistema:** desempeña funciones técnicas que consisten en optimizar el rendimiento y arquitectura del sistema. Pueden además proponer las plantillas de diseño y los sistemas de categorización más adecuados, de acuerdo con los proveedores de información y de mantener el sistema en constante mejora y actualización.

La definición de roles o usuarios es dependiente de la plataforma final que se puede escoger como SGC, y van desde los presentados anteriormente hasta una definición de grano fino, donde se especifican roles intermedios y se diversifican más las tareas. El sistema de gestión de contenidos controla y ayuda a manejar cada paso de este proceso,

incluyendo las labores técnicas de publicar los documentos a uno o más sitios.

### **22.1.3. Arquitectura general de los Sistemas de Gestión de Contenidos.**

La arquitectura de estos sistemas es modular, proporcionando un marco de desarrollo que facilita la implementación de nuevas funcionalidades. En este sentido, los SGC incorporan una gran variedad de módulos que permiten extender el funcionamiento del sistema. Existen módulos para gestión integral de un sitio Web, para la gestión de páginas generadas dinámicamente, y otros módulos que posibilitan la personalización del sistema por parte del usuario.

Los SGC, a nivel de software, pueden definirse como un framework que habitualmente consta de dos partes diferenciadas:

- **Backend o parte administrativa:** A través del backend pueden controlarse todos los aspectos relativos a la configuración del framework, la administración del contenido (creación, categorización, edición, publicación, eliminación), la personalización del entorno de consulta (frontend), actualización y configuración de nuevas funcionalidades.
- **Frontend o parte pública:** A través del frontend se pueden consultar los contenidos publicados, acceder a las funcionalidades proporcionadas para los usuarios configuradas desde el backend, y también sirve para recolectar ciertos datos de entrada.

La separación del sistema en frontends y backends es un tipo de abstracción que ayuda a mantener las dos principales partes del SGC separadas. Dentro de la arquitectura de SGC se contempla la existencia de



una o varias bases de datos, responsables fundamentalmente de la persistencia del contenido publicado a través del CMS y de todos los datos relativos a la configuración del sistema.

El flujo básico es que gestor responde a las solicitudes de páginas que se plantean desde los lectores, recuperándolas la base de datos, componiendo las plantillas definidas y devolviendo al servidor web el contenido final que este ofrece al lector.

Los SGC, al ser aplicaciones web, se ejecutan en el servidor web donde estén alojados. Dependiendo de las tecnologías utilizadas para el desarrollo del SGC, la complejidad del servicio web que soporta la plataforma será mayor. El acceso a los SGC se realiza a través de los navegadores web. Cuando un usuario realiza una petición de una página, el gestor de contenidos es el encargado de interactuar con el servidor para generar una página dinámica, con un formato definido, y cuyo contenido se extrae de la base de datos.

#### **22.1.4. Categorías**

En cuanto a la categorización de los SGC, no existe una clasificación estricta, sino más bien, categorizaciones en función de determinadas características propias de los sistemas. Así pues, podemos clasificarlos en función del lenguaje de programación en el que se desarrollen, según su licencia (código abierto o software privativo), e incluso por su funcionalidad (Blogs, Wikis, Foros, ...)

#### **22.1.5. Criterios de valoración**

A la hora de proceder a la implantación de un SGC, es necesario tener en cuenta una serie de criterios que nos servirán para establecer, en función de la situación, cuál es el SGC más adecuado. Estos criterios son los siguientes:

- **Código abierto o código propietario:** En el caso de los SGC de tipo privativo, es decir, los comercializados por empresas bajo licencias restrictivas, no se permite el acceso al código fuente por parte de terceros. Sin embargo con los SGC de código fuente abierto, esta limitación no existe, dado que los desarrolladores sí que permiten el acceso libre y la modificación del código. Esta característica es muy importante puesto que el poder disponer del código fuente proporciona el poder modificar el producto, aportándole nuevas funcionalidades o incluso corrigiendo posibles errores. Esta es una faceta muy importante relacionada con la evolución del producto. Otra ventaja de los SGC de código libre es el coste, puesto que este tipo de gestores de contenido son gratuitos, sin ningún coste de licencias. En el caso de los SGC comerciales, el coste puede llegar a ser muy elevado, sobre todo para un particular. Además de todo esto, alrededor de los gestores de contenido de código libre suelen existir comunidades de usuarios que comentan sus experiencias con el uso de estos sistemas, aportan novedades y desarrollan nuevas funcionalidades.
- **Arquitectura técnica:** El SGC tiene que ser fiable, robusto y adaptable a futuras necesidades. Para ello, es preciso tener en cuenta cuál es la arquitectura del sistema, que tecnologías se han utilizado, y analizar el diseño de la plataforma, con el objetivo de poder emprender ampliaciones en las funcionalidades ofrecidas en caso de ser necesario. También es conveniente que permita separar contenido, presentación y estructura, de acuerdo con los estándares establecidos para el web. Para ello, es altamente recomendable

decantarse por el uso de sistemas que hagan uso de motores de plantillas, así como uso de definiciones de estilos basadas en hojas de estilo (CSS).

- **Grado de desarrollo:** Es muy importante que la herramienta seleccionada tenga un grado de madurez adecuado para poder desarrollar la funcionalidad requerida, y que se disponga de módulos o componentes para poderle añadir funcionalidad.
- **Soporte, posición en el mercado y opiniones:** La herramienta tiene que tener soporte tanto por los creadores como por los desarrolladores. Es fundamental que una herramienta que sea conocida por muchos usuarios y expertos, este hecho puede ayudar a posibles usuarios a decidirse por el SGC en cuestión. Habitualmente las grandes comunidades de usuarios y desarrolladores se encuentran alrededor de los SGC libres, proporcionando un marco ideal para el rápido desarrollo de estos sistemas así como de su mantenimiento.
- **Usabilidad:** Partiendo de la premisa de que existen diferentes roles con diferenciación clara de tareas, debemos de tener en cuenta que determinados perfiles de usuarios no tienen por qué tener conocimientos técnicos. Ello implica que el SGC tiene que ser fácil de aprender y utilizar.
- **Accesibilidad:** Tenemos que tener en cuenta que en el momento en que trabajamos con SGC, el sistema debe estar preparado para el uso por parte de la mayor cantidad de usuarios posible. Por tanto, es siempre recomendable que el portal web cumpla un estándar de accesibilidad.
- **Velocidad de descarga:** Es importante que las páginas solicitadas por los usuarios se carguen rápido. La naturaleza de las páginas

dinámicas y la separación de estructura, presentación y contenido contribuyen a que las páginas sean más ligeras.

## **22.1.6. JOOMLA!**

### **22.1.6.1 Introducción**

Joomla! Es uno de los SGC con mayor impacto y distribución. Esto ha sido proporcionado por el hecho de ser un sistema de código abierto, desarrollado en uno de los lenguajes mayoritarios para Internet como es PHP. Está recogido bajo licencia GPL y actualmente cuenta con una de las mayores comunidades de usuarios y desarrolladores. Este administrador de contenidos puede trabajar en Internet o intranets y requiere de una base de datos MySQL, así como de un servidor web, preferiblemente HTTP Apache.

### **22.1.6.2 Arquitectura**

En cuanto a su diseño, desde el punto de vista de desarrollo, Joomla! está programado en PHP bajo un patrón Modelo-Vista-Controlador, integrando un motor de plantillas, y permitiendo separar totalmente la capa de presentación de la lógica de los datos. Esta modularidad proporciona una gran facilidad para extender el sistema. Las funcionalidades en Joomla! se añaden a nivel de módulos o componentes. Estos módulos o componentes son partes del sistema que se implementan de forma independiente, bajo el patrón MVC, y se integran perfectamente dentro del SGC principal. Existen repositorios libre de la comunidad de usuarios y desarrolladores donde se pueden encontrar cientos de módulos gratuitos para extender las funcionalidades de Joomla!. Sin embargo, esta cuota también representa un modelo de negocio para muchas empresas que proporcionan sus productos en forma de módulos para Joomla!. Así pues, el diseño patronizado mediante MVC y el uso de tecnologías maduras como PHP y MySQL hace que resulte relativamente sencillo ampliar las

funcionalidades de este SGC a partir de la implementación propia de módulos que satisfagan alguna funcionalidad concreta.

El SGC Joomla presenta una arquitectura en tres niveles: nivel de extensiones, nivel de aplicación y de desarrollo.

El nivel superior, de extensiones, se compone de extensiones del marco de desarrollo de Joomla y de sus aplicaciones. En esta capa se sitúan los módulos, componentes y plantillas (templates). El nivel del medio, de aplicación, consiste en una serie de aplicaciones que extienden del core para dar soporte a los módulos y componentes. Implementa también las aplicaciones necesarias para la administración (backend) así como la arquitectura principal del frontend. El nivel inferior, correspondiente al de desarrollo, consta del conjunto de clases PHP que lo forman, las bibliotecas que son utilizadas por el marco de desarrollo o se instalan para uso por los desarrolladores y finalmente los plugins, que extienden la funcionalidad.

Algunas características básicas que se incluyen en Joomla! son: sistema adaptado para mejorar el rendimiento web, versiones imprimibles de páginas y generación directa en pdf, módulos de flash con noticias, integración con blogs y foros, módulos nativos para la gestión de encuestas, calendarios, búsqueda en el sitio web e internacionalización del lenguaje. El nombre de Joomla! proviene de una pronunciación fonética para anglófonos de la palabra swahili jumla, que significa "todos juntos" o "como un todo". Fue escogido como una reflexión del compromiso del grupo de desarrolladores y la comunidad del proyecto.

### **22.1.6.3 Comunidad de desarrollo**

La comunidad de Joomla, para el desarrollo de sus múltiples frentes, usa diferentes formas de comunicación como son el uso de salas de chat a través de IRC, participación en foros especializados, listas de correo, "wikis" y blogs. La gestión de administración principal del proyecto esta delegada al grupo principal, conocido como "Core Team". Este grupo de

desarrolladores representa la columna vertebral del proyecto, ya que son los encargados de guiar a Joomla! dentro del movimiento de código abierto. Este grupo está compuesto por diferentes perfiles, con variadas experiencias y totalmente multidisciplinar. Lleva activo desde el año 2005, aproximadamente con el nacimiento oficial de Joomla!. Su responsabilidad principal radica en la organización con respecto Joomla en su estructura funcional como organización y no únicamente en la programación del sistema de gestión de contenidos.

Además del grupo principal o Core Team, existen también otros grupos que se han creado para enriquecer el conocimiento que la comunidad Joomla proporciona. Cada uno de los grupos se especializa en un aspecto específico de Joomla! que es importante para la expansión y desarrollo. El Core Team no puede estar en cada discusión de estos temas, por ello existe una estructura jerarquizada en donde un responsable de cada grupo de desarrollo se encarga de comunicarse de forma directa con el Grupo Principal.

Además del trabajo de la comunidad de usuarios y desarrolladores, existe una organización que proporciona soporte para muchos aspectos del proyecto. Se trata de la Open Source Matters Inc (OSM), que es una organización sin ánimo de lucro de origen estadounidense. El objetivo fundamental de esta organización es dar soporte a la parte legal y financiera del proyecto de código abierto Joomla. Recientemente la OSM se ha incorporado como una organización sin ánimo de lucro de Nueva York, proporcionando una garantía de continuidad para el proyecto y actividades futuras, proporcionando el soporte necesario para que las comunidades de usuarios y desarrolladores puedan seguir participando.

#### 22.1.6.4 Principales características

Las principales características que han hecho de Joomla uno de los mejores SGC del momento son las siguientes:

- **Usabilidad de su interfaz:** Esta característica se hace principalmente notoria en la interfaz de administración. El objetivo fundamental es que cualquier persona sin conocimientos técnicos pueda tener control del sistema, para acortar la curva de aprendizaje de las tareas administrativas.
- **Gestión de contenido:** El sistema presenta una estructura jerárquica para gestionar el contenido basada en agrupaciones de artículos (la unidad fundamental de contenido) que se organizan en secciones y categorías. Permite crear menús y submenús, subir imágenes y ficheros, así como syndicar de forma nativa noticias mediante RSS.
- **Gestión de usuarios:** Existen dos tipos de usuarios básicos: los usuarios invitados, que son aquellos que acceden al portal navegando, que no poseen ninguna cuenta en el sistema y que habitualmente están capacitados para consultar los artículos, y los usuarios registrados que son aquellos que disponen de una cuenta (nombre de usuario/contraseña) para autenticarse en el sitio y acceder a funcionalidades específicas. Dentro de los usuarios registrados existen distintos roles cada uno con una serie de privilegios. La gestión de las cuentas y permisos de los usuarios en Joomla! puede hacerse de forma nativa, o bien haciendo uso de un sistema externo como LDAP.
- **Personalizable:** Gracias a la combinación del uso de estándares, y al diseño desacoplado proporcionado por el patrón MVC, la presentación del contenido se puede personalizar de forma muy sencilla. La apariencia del front-end es perfectamente modificable gracias al uso de plantillas. Las plantillas pueden modificarse de

manera sencilla permitiendo que se adapten a las necesidades del sistema.

- **Extensibilidad:** Como ya se ha comentado con anterioridad, una de las principales características que definen a este software es la modularidad de la plataforma, que permite el desarrollo e integración de una gran cantidad de módulos y componentes que permiten extender las funcionalidades del sistema. La facilidad en el desarrollo de estas piezas software ha proporcionado que actualmente exista un gran número de extensiones y módulos existentes, programados por la comunidad de usuarios, que aumentan las posibilidades de la aplicación con nuevas características y que se integran fácilmente en el sistema. Como ejemplo de extensiones disponibles, se citan gestores de documentos, galerías de imágenes multimedia, motores de comercio y venta electrónica, calendarios, etc.
- **Multiplataforma:** Debido a la utilización de tecnologías libres estandarizadas, este SGC puede correr sobre cualquier sistema operativo, ya sea GNU/Linux, en Windows o en Mac OSX. Los únicos requisitos son disponer en la máquina de un servidor web, y de una base de datos MySQL.

## **22.1.7. WORDPRESS**

### **23.1.7.1 Introducción**

La popularidad creciente de los blogs o bitácoras como medio popular para difundir contenido, ha tenido también cabida dentro del desarrollo de los Sistemas de Gestión de Contenidos. WordPress es un SGC enfocado precisamente a la creación de blogs, especialmente orientado a ofrecer comodidad para la ardua tarea de mantener los sitios web periódicamente actualizados.



Wordpress está desarrollado en PHP y MySQL, bajo licencia GPL, lo que también implica que es software libre y por tanto su código es modificable y adaptable. En este sentido, comparte muchas de las ventajas que esta filosofía otorga a otros SGC como Joomla!.

El fundador del proyecto de Wordpress es Matt Mullenweg. WordPress fue creado a partir del desaparecido b2/cafeleg y actualmente es el SGC más popular orientado a la creación de blogs. Las causas de su enorme crecimiento están relacionadas con su licencia libre, la facilidad de uso y las características que proporcionan en general los sistemas de gestión de contenido.

Al igual que la mayoría de los SGC más populares, Wordpress está implementado bajo un patrón MVC. Sumado a esto, al proporcionarse como producto libre, se posibilita la labor de la enorme comunidad de desarrolladores para revisiones e implementación de módulos que añadan nuevas funcionalidades. Éste es otro de los factores que ha proporcionado su creciente expansión.

Como ocurre con Joomla!, sumado al trabajo de la comunidad libre de desarrolladores, el liderazgo del proyecto recae sobre una entidad llamada Automattic.

### **23.1.7.2 Características**

Algunas características básicas que definen Wordpress son las siguientes:

- Proporciona un sistema de publicación web basado en entradas ordenadas por fecha.
- La estructura y diseño visual del sitio depende de un sistema de plantillas, que es independiente del contenido en sí. Separación de la capa de presentación.

- Se apuesta decididamente por las recomendaciones del W3C, pero es dependiente siempre de la plantilla a usar.
- La gestión y ejecución corre a cargo del sistema de administración con los plugins y los widgets que usan las plantillas.
- Como en otros SGC, existe una jerarquía de usuarios/roles, y Wordpress permite múltiples autores o usuarios.
- Aunque el sistema está orientado a configurar un único blog o bitácora por sistema instalado, permite múltiples blogs o bitácoras.
- Dispone de múltiples herramientas para organizar el contenido (artículos) en categorías.
- Dispone de componentes visuales para la edición de los artículos (componentes WYSIWYG "What You See Is What You Get")
- Permite comentarios y herramientas de comunicación entre blogs.
- Dispone de funcionalidades necesarias para la sindicación de contenidos en los principales formatos estándar (RSS 2.0 y ATOM 1.0 ).
- Subida y gestión de adjuntos y archivos multimedia.
- Sistema de búsqueda integrada dentro de la plataforma.

## **22.1.8. DRUPAL**

### **23.1.8.1 Introducción**

Otro de los más conocidos en el mundo de los SGC es Drupal. Drupal es un sistema de gestión de contenido, similar en cuanto su arquitectura y orientación a Joomla! Es un sistema modular multipropósito y muy

configurable. Permite gestionar y publicar artículos, imágenes, u otros archivos. Su diseño modular permite integrar una gran cantidad de servicios diferentes como foros, encuestas, votaciones, blogs y administración de usuarios y permisos.

Drupal es un sistema dinámico. Esto implica que, como el todos los anteriores, el contenido se almacena de forma persistente en una base de datos, y las páginas que se demandan desde el front-end de consulta son generadas dinámicamente. El sistema se encarga de acceder al contenido de la base de datos y montar la página que suministrará al servidor web.

Es un programa libre, con licencia GNU/GPL, escrito en PHP bajo un patrón de diseño MVC, lo que de nuevo facilita su modificación y adaptabilidad, potenciando el trabajo de la extensa comunidad de usuarios.

Algunas características propias de Drupal a nivel de desarrollo son la calidad de su código y de las páginas generadas. Hace especial hincapié en el respeto de los estándares de la web, y un énfasis particular en la usabilidad y consistencia de todo el sistema.

El diseño de Drupal lo hace especialmente idóneo para construir y gestionar comunidades en Internet. Sin embargo, gracias a sus características de flexibilidad y adaptabilidad, así como la gran cantidad de módulos adicionales disponibles, convierte a Drupal en un SGC de propósito general, capaz de adecuarse a muchos tipos diferentes de sitio web.

### **23.1.8.2 Características**

Sus características principales son las siguientes:

- **Extensibilidad:** Gracias a la extensa comunidad de usuarios y desarrolladores, dispone de una gran cantidad de módulos con distintas funcionalidades: foro, galería, encuestas, boletín de noticias, correo electrónico, chat, etc.

- **Código aberto:** Al estar disponible el código fuente bajo los términos de la licencia GNU/GPL, es posible extender o adaptar Drupal según las necesidades.
- **Personalización:** La capa de presentación está perfectamente aislada del resto del sistema, haciendo la apariencia totalmente configurable en función de las preferencias de los usuarios.
- **Gestión de usuarios:** Como todo SGC, dispone de una jerarquía de usuarios/roles y de un sistema interno para gestionarlos y permitir la autenticación. Ésta última puede hacerse bien de forma local o utilizando un sistema de autenticación externo.
- **Gestión de contenidos:** Proporciona un sistema de control de versiones, que permite seguir y auditar todas las actualizaciones del contenido. Dispone de un sistema de temas o plantillas que permite separar el contenido del sitio de la presentación. También cuenta con la posibilidad de exportar el contenido en formato RDF/RSS para ser utilizado por otros sitios web.
- **Multiplataforma:** Puede funcionar con cualquier servidor web (Apache, Microsoft IIS) y en sistemas como Linux, Windows, Solaris, BSD y Mac OS X. Al estar implementado en PHP es portable.

## 22.2. Sistemas de Gestión Documental

### 22.2.1. Definición de Sistemas de Gestión Documental

Un sistema de gestión documental se define como un conjunto de elementos y relaciones entre ellos, que tiene el propósito de normalizar, controlar y coordinar todas las actividades y procesos que afectan en

cualquier medida a los documentos generados en el transcurso de la actividad de una organización. Las operaciones más habituales que se realizan sobre estos documentos, abarcan todo su ciclo de vida, desde su creación hasta su almacenamiento y puesta a disposición de los usuarios.

Además, un sistema de gestión documental tiene que satisfacer lo siguiente:

- Conservar los atributos básicos de los documentos, que les confieren su valor informativo, legal y probatorio.
  - o Originalidad
  - o Autenticidad
  - o Integridad
  - o Veracidad
- Mantener la organización de los documentos integrados en un contexto. Esto implica conservar una interrelación con los otros documentos que surgen de la misma función, actividad, que son producidos por el mismo departamento u organismo, que forman parte de la misma serie, etc.

### *Software de gestión documental*

El software de gestión documental abarca todos aquellos programas software diseñados para gestionar grandes cantidades de documentos. En estos documentos no necesariamente debe existir organización dentro de sus contenidos, de hecho, lo más común es que el contenido de estos documentos no guarde una organización clara.

Existen diversos métodos que usados en combinación con las bibliotecas de documentos y una serie de índices, permiten un acceso rápido a la información almacenada en dichos documentos, los cuales,

habitualmente se encuentran comprimidos y suelen almacenar, además del texto plano, otros contenidos multimedia como imágenes, videos, etc.

Entre los objetivos que se persiguen a la hora de implantar un sistema de gestión documental cabe mencionar:

- Resaltar la importancia que tienen los documentos dentro de cualquier tipo de organización, pública o privada.
- Facilitar la recuperación de información de forma rápida, exacta y efectiva.
- Analizar la producción documental, para evitar documentos innecesarios o que no merece la pena almacenar pasado cierto tiempo.
- Conseguir que los archivos sean útiles y significativos como unidades de información no sólo dentro de la empresa sino también externamente.

Antes de montar un sistema de gestión documental es necesario realizar una serie de consideraciones previas que podemos agrupar en las siguientes categorías:

- Administrativas: se centra todo lo que puede influir en la administración de la empresa.
- Económicas: se refiere a la evaluación del ahorro que genera la gestión de documentos.

Para la implantación de este tipo de sistema, es necesario también realizar un diagnóstico y una evaluación de los requisitos tanto técnicos como administrativos.

### **22.2.2. Funciones de la Gestión documental**

Las principales funciones de la gestión documental son:

- Almacenamiento
- Captura
- Conservación
- Consulta
- Creación
- Difusión
- Eliminación
- Ingreso
- Uso

### **22.2.3. Ciclo de Vida de los Documentos**

El ciclo de vida de un documento abarca todas las fases por las que un documento pasas, desde que se crea hasta que se archiva o elimina.

Los documentos pueden tener distintos valores que son:

- *Valor Primario (Administrativo):* su propósito es dejar constancia de una actividad.
  - o Valor fiscal o contable: acreditar el cumplimiento de las obligaciones contables o tributarias.
  - o Valor legal o jurídico: su finalidad es, entre otras, servir de prueba ante la ley.
- *Valor Secundario:*
  - o Valor Informativo: su propósito es servir de base para la reconstrucción de cualquier actividad realizada.
  - o Valor Histórico: sirve de fuente para la investigación histórica.

Las distintas fases que atraviesa un documento son:

- Archivo de Oficina (Documentación Activa): fase en la cual los documentos son creados o recibidos por algún departamento, sobre los cuales se puede realizar una serie de operaciones de edición.
- Archivo General (Documentación semiactiva): en esta etapa la principal función es la consulta de la documentación y la actividad que recibe este tipo de documentación es menor que en el Archivo de Oficina.
- Archivo Histórico (Documentación Inactiva): en esta etapa la documentación sólo tiene utilidad como fuente de información histórica. Las consultas que recibe son menores.

#### **22.2.4. Beneficios de la Gestión Documental**

Si se realiza una buena gestión de los documentos, esto repercute en la empresa con una serie de beneficios, como son:

- Obtener información precisa de las actividades de la empresa que sirva de apoyo para actividades futuras, toma de decisiones, etc.
- Facilitar la realización de las actividades de la empresa.
- Documentar las políticas y el proceso de toma de decisiones.
- Garantizar la continuidad de la empresa en caso de fallo masivo en los sistemas, catástrofe, etc.
- Cumplir con los requisitos legales que existen con algún tipo de ficheros de datos.
- Almacenamiento de evidencias de las actividades relacionadas con la empresa y entidades externas.
- Mantener un histórico de la evolución de la entidad.
- Centralizar el almacenamiento de documentos.
- Facilitar la prestación de servicios a los usuarios de la empresa.



## **22.3. Gestión del Conocimiento**

### **22.3.1. Definición de Gestión del Conocimiento**

No existe una definición universalmente aceptada de la gestión del conocimiento. Sin embargo, existen numerosas definiciones de diversos expertos. En general, la gestión del conocimiento es la conversión del conocimiento tácito en conocimiento explícito y su intercambio dentro de la organización. La gestión del conocimiento es el proceso mediante el cual las organizaciones generan valor de sus activos intelectuales. Definidos de esta manera, se hace evidente que la gestión del conocimiento tiene que ver con el proceso de identificación, adquisición, distribución y mantenimiento de los conocimientos que son esenciales para la organización.

Si se considera la gestión del conocimiento en un contexto más amplio, entonces existen múltiples definiciones, sin embargo, todas ellas apuntan a la misma idea, aunque cada una se centre en un aspecto particular de la gestión del conocimiento.

- Una definición orientada a los resultados puede afirmar que la gestión del conocimiento es "tener el conocimiento adecuado en el lugar correcto, en el momento adecuado y en el formato correcto".
- Una definición orientada al proceso puede afirmar que la gestión del conocimiento consiste en "la gestión sistemática de los procesos por los cuales el conocimiento se identifica, se crea, se une, se comparte y se aplica".
- Una definición orientada a la tecnología puede presentar una fórmula de gestión del conocimiento como "Business Intelligence + motores de búsqueda + agentes inteligentes".

### **22.3.2. Cuestiones sobre gestión del conocimiento**

Existen dos aspectos principales en la gestión del conocimiento, que son la *gestión de la información* y la *gestión de las personas*. Visto desde esta perspectiva, la gestión del conocimiento es, por un lado, la información y, por otro, la gente.

La mayoría de empresarios y directivos están familiarizados con el manejo de información a largo plazo. Este término se asocia con la gestión del conocimiento en relación con los objetos, que son identificados y controlados por los sistemas de información. La práctica de la gestión de la información fue ampliamente aceptada cuando los ejecutivos se dieron cuenta de que la información era un recurso importante, que debía ser manejado correctamente, para que las empresas puedan mejorar su competitividad.

Como consecuencia del crecimiento de la práctica de la gestión de la información, los conceptos de "*análisis de la información*" y "*planificación de la información*", se han desarrollado, proporcionando herramientas adicionales para los profesionales.

En la vertiente teórica la gestión de la información ha evolucionado convirtiéndose en gestión del conocimiento. En la práctica, la gestión del conocimiento implica, entre otros, la identificación y mapeo de los activos intelectuales de una organización. Esto significa, básicamente, la identificación de quién sabe qué dentro de la empresa. Cuando se mira

desde esta perspectiva, la gestión del conocimiento puede ser considerado como un proceso de realización de una auditoría de los activos intelectuales. Sin embargo, la gestión del conocimiento va más allá de este nivel de la cartografía y también implica la creación de conocimiento para obtener ventajas competitivas y la conversión de grandes cantidades de datos de la organización en información de fácil acceso.

Se ha demostrado una y otra vez que cuando el conocimiento se gestiona bien, hay una reducción significativa en el tiempo necesario para completar las tareas y la duplicación innecesaria se evita.

Como ya se ha comentado anteriormente, un aspecto de la gestión del conocimiento es la gestión de personas. Básicamente, se trata de la gestión del conocimiento tácito que reside dentro de las cabezas de las personas. En la práctica implica la gestión del conocimiento que existe junto a los procesos organizativos que implica una serie compleja de capacidades dinámicas, know-how y otras capacidades relacionadas con el conocimiento.

Con el fin de gestionar de forma eficaz a las personas que poseen el conocimiento tácito que se desea, es esencial tener en cuenta su diversidad cultural y los valores sociales, actitudes, aspiraciones y gustos. Si esto se puede hacer con éxito, puede conducir a la creación de nuevos conocimientos que de otra manera no se puede lograr mediante la gestión de información por sí sola.

A pesar de la importancia de los dos aspectos de la gestión del conocimiento, la cual está bien reconocida por muchas organizaciones, el verdadero potencial de la gestión del conocimiento todavía queda por alcanzarse. De hecho, no todas las organizaciones con algún sistema de gestión del conocimiento son conscientes de que tienen estos sistemas.

La mayoría de las organizaciones tienen algún tipo de sistema para la gestión del conocimiento explícito, ya sea simple o compleja, aunque, no necesariamente se refieran a él como un sistema de gestión del conocimiento. Por otro lado, la gestión del conocimiento tácito no es común y la tecnología actual basada en la gestión del conocimiento no se ha desarrollado de forma plenamente eficaz para la extracción de conocimiento tácito. Aunque el conocimiento tácito es la base de conocimiento organizacional, es algo tan personal que es difícil de formalizar y comunicar.

Ambos aspectos de la gestión del conocimiento presentan dos cuestiones inmediatas:

- Hacer que el conocimiento de la organización sea más productivo.
- Producir beneficios significativamente mayores que los previstos.

La gestión del conocimiento ofrece una excelente oportunidad para adoptar estrategias de negocio que antes eran imposibles. Por ejemplo, se puede abrir la puerta a la creación de una red casi ilimitada que mejore las relaciones con clientes y proveedores. En la mejora de relaciones con los clientes, la gestión del conocimiento hace posible el descubrimiento de nuevos problemas y oportunidades a través del uso óptimo de los activos de conocimiento, tales como el contrato de venta, los registros, los datos demográficos de los clientes, etc. Es precisamente de esta manera la gestión del conocimiento se puede complementar y mejorar el impacto de otras iniciativas de la organización como la gestión de la calidad total, el proceso de reingeniería de negocios, y el aprendizaje organizacional.

Es evidente a partir de esta discusión que las iniciativas de gestión del conocimiento se pueden aplicar en una variedad de ámbitos para lograr resultados superiores en casi cualquier tipo de organización. Y es posible

alcanzar estos resultados, independientemente del nivel de disponibilidad tecnológica o el sector del mercado en cuestión.

## **22.4.     Sistemas de Información Geográfica SIG**

Como información espacial (geográfica, georreferenciada o geodatos) nos referimos a todo tipo de información relativa a sucesos o elementos para la cual se incluye una referencia a su localización, la cual está situada sobre o en las inmediaciones de la superficie de la Tierra. La forma de referenciar la posición de estos elementos o estos sucesos puede realizarse de distintas formas, mediante una simple dirección postal, con coordenadas geográficas (longitud y latitud) o con coordenadas cartesianas en algún sistema de referencia cartográfico.

La mayor parte de la información en formato electrónico almacenada actualmente en sistemas de todo tipo, es información espacial o que podría serlo. El porque de este auge de la información espacial la encontramos en una serie de características que justifican el interés de asociar a una información la referencia de su localización.

Por una parte tenemos la cualidad de la información espacial para su representación en forma gráfica y simbólica mediante mapas. Los mapas son un sistema de comunicación que ha sido utilizado desde las primeras civilizaciones y con el que está familiarizado prácticamente todo el mundo. Además los mapas han tenido gran importancia a lo largo de la historia militar, económica y política de las naciones, por lo que han sido considerados siempre como un recurso clave a cuyo desarrollo se han dedicado importantes esfuerzos.

Por otra parte, la capacidad que posee la información espacial para integrar conjuntos de información que de otra forma serían inconexos,

mediante la aplicación de las relaciones espaciales de coincidencia, proximidad o adyacencia inherentes a dicha localización espacial. Esta característica es probablemente la que mayor potencial otorga a la información espacial, constituyendo la base del análisis espacial.

La primera manifestación de los sistemas de información geográfica los podemos encontrar, como se comentaba anteriormente, en los mapas, sin embargo, ya en épocas más recientes, las aportaciones de las tecnologías de la información en el ámbito de la cartografía han sido muy importantes. Se pueden destacar aquellos avances destinados a la mejora de los procesos de producción cartográfica, las orientadas a la explotación y análisis de la información cartográfica.

En lo tocante a la producción cartográfica, actualmente se cuenta con técnicas muy depuradas para la producción de mapas en todas sus fases, desde la captura de datos (fotogrametría aérea, imágenes de satélite, teledetección, telemetría láser, GPS, etc), hasta los diferentes procesos que componen la fase de elaboración de la cartografía. Estas técnicas han permitido no sólo notables mejoras en la calidad, diversidad y flexibilidad de los productos cartográficos, sino que ha hecho posible, disponer de información cartográfica muy actualizada

En lo tocante al análisis de la información geográfica es necesario destacar en primer lugar las importantes limitaciones prácticas que presentan los mapas tradicionales para su utilización en análisis mediante técnicas manuales. La superación de estas limitaciones es ha sido la motivación inicial para el desarrollo de los Sistemas de Información Geográfica, SIG (o GIS de acuerdo con la terminología anglosajona), que se ha convertido en la otra gran rama de aportaciones de las tecnologías de la información en el ámbito de la cartografía.

El desarrollo de los primeros GIS datan de finales de los años 60 y supuso un gran cambio en la utilización de la información espacial que se hacia hasta ese momento. De hecho las técnicas y metodologías de análisis

espacial de la información, que hasta el momento habían sido poco examinadas por la excesiva complejidad asociada a los tratamientos manuales, se vieron paulatinamente mejoradas y en muchos casos empezó a ser posible su utilización con el procesamiento automatizado de la información espacial en formato digital.

Un Sistema de Información Geográfica está orientado a la captura, manipulación, recuperación, análisis, representación, etc, de información georreferenciada, aquella en la que la posición espacial ocupada por los objetos del mundo real que se modelizan forma parte inherente a dicha información.

Los SIG gozan de gran aceptación desde sus primeras implantaciones lo cual se debe en gran medida a su capacidad para construir modelos orientados a la resolución de problemas cuyo universo de discurso se caracteriza por tener un componente espacial.

Estas primeras realizaciones fueron impulsadas principalmente por organizaciones con responsabilidades en la gestión de recursos con implantación territorial como son ordenación del territorio, recursos naturales, censo, defensa, etc.

Desde estas primeras implantaciones, en los años 60, hasta la década de los 80, el desarrollo de los SIG se produjo de una forma relativamente lenta debido sobretodo a la capacidad y coste de la tecnología digital disponibles en aquel momento. Desde la segunda mitad de los 80 se produce un gran auge, tanto en diversificación de las áreas de aplicación de esta tecnología como en la oferta de productos comerciales, lo que ha otorgado gran popularidad y difusión a los SIG en todo tipo de organizaciones.

Actualmente la evolución se caracteriza por una serie de factores que impiden una plena estabilidad del sector:

- Evolución de las tecnologías en las que se apoyan los SIG, como son, gestores de bases de datos, procesamiento paralelo, visualización, etc.
- Alto grado de relación entre los SIG e Internet, que está en permanente evolución.
- La tecnología disponible se orienta a implementaciones que no aprovechan el potencial de la misma.
- Restricciones institucionales que todavía impiden el acceso y utilización de la información cartográfica de que se dispone a todos los niveles.

#### **22.4.1. Arquitectura de un SIG**

En los primeros desarrollos de los SIG encontramos principalmente sistemas software cerrados de gran tamaño y complejidad (SIG monolítico), que eran utilizados principalmente por grupos de usuarios reducidos con un nivel de especialización bastante elevados, además, se orientaban a tareas muy concretas y generalmente presentaban poca o ninguna integración con otros sistemas. Actualmente los sistemas se diferencian mucho de estos iniciales, son cada vez más habituales y sencillos, no requieren de usuarios expertos para su manejo y permiten la integración de la información con otros sistemas.

En un sistema SIG podemos hablar de arquitecturas de 3 capas, así tenemos:

- Capa de Presentación: incorpora todas las funcionalidades que permiten la interacción entre el usuario y el sistema, para acceso el acceso a la información y presentación de resultados. Habitualmente esto se traduce en una GUI que facilita el acceso a las herramientas de la siguiente capa o también en una aplicación externa que pueda acceder a determinadas funciones de geoproceto.



- Capa de Proceso: abarca una serie de herramientas diferentes que integran el núcleo del SIG.
- Capa de Gestión de Datos: centraliza el acceso a los datos, que pueden localizarse en distintos almacenes. Además, integra también muchas de las funciones que se encargan de proporcionar transparencia sobre los detalles de los datos: sistemas de proyección, formatos, transformación de coordenadas, etc.

Para implementar estas capas funcionales podemos tener los mismos o diferentes sistemas físicos, teniendo una gran cantidad de posibilidades. Si se hace una desagregación completa, cada una de estas capas residirá en uno o más servidores diferentes, adaptado a las necesidades específicas del entorno de implantación del SIG.

En el mercado de los SIG comerciales, se están asumiendo cada vez mas una serie de estándares de facto que han ido surgiendo en los últimos años debido a las tecnologías de componentes (Java Beans, .NET, ...) y de plataformas interoperables de objetos distribuidos (SOAP, CORBA). Estas tecnologías, permiten construir nuevos SIG de forma extensible e integrando funcionalidades proporcionadas por diversos proveedores.

Desde la perspectiva de los almacenes de datos, se tiende cada vez mas al uso de los sistemas post relacionales, que permiten integrar en sistemas relacionales tradicionales algunas características de las BDOO e incluso incluyen extensiones espaciales del modelo multimedia del estándar SQL3.

Por otro lado, el avance en el campo de las telecomunicaciones y más en concreto, de Internet, con un gran potencial tanto para la transmisión de grandes cantidades de información y acceso a datos, ha favorecido la expansión de las arquitecturas de geoproceto basadas en servicios web.

Los servicios web permiten concebir y desarrollar sistemas que integran, con un mínimo nivel de acoplamiento, información y servicios de geoproceso interoperables de múltiples fuentes y en distintos formatos, a los que se accede en un entorno de red distribuido.

Estas nuevas arquitecturas pretenden satisfacer el deseo de la comunidad SIG de disponer de un acceso ilimitado y en cualquier momento a información actualizada e interoperable. La disponibilidad de este tipo de servicios está facilitando una expansión del intercambio y de la difusión electrónica de la información espacial.

Por otra parte, este crecimiento en la implantación de productos y servicios de información cartográfica en la red, establece los principios para el asentamiento real de un entorno en el que sea posible el intercambio de información geográfica y servicios de geoproceso. Cabe destacar también la acción tan importante que en esta línea está desarrollando el consorcio OpenGIS, en el que se aglutinan los principales entes involucrados en el sector de la información espacial y todos los sistemas y tecnologías que la soportan (usuarios, universidades, administraciones, industrias software, ...). El propósito de estos colectivos es elaborar de forma consensuada, especificaciones de interfaces interoperables en el campo de las tecnologías de la información espacial.

Desde la creación de OpenGIS, a mediados de los años 90, han sido ya muchas las realizaciones prácticas en las que ha tomado parte y siendo, sus especificaciones estándares de facto en el área de las tecnologías de la información espacial. Además, en muchos casos estos estándares son la base para la formulación de estándares internacionales.

Por ejemplo, los dos siguientes, son servicios web que ya han sido enteramente especificados:

- Servicio de Entidades Vectoriales: facilita información relativa a la entidad o entidades que se encuentran almacenadas en una capa

vectorial y que reúnen las características especificadas durante la consulta.

- Servicio de Mapas en la Web: genera mapas en el formato deseado para ser visualizados en un navegador u otro tipo de cliente sencillo. Estos mapas serán la respuesta a alguna consulta con ciertos parámetros realizada previamente.

Podemos concluir que las arquitecturas de los SIG tienden a ser distribuidas, interoperables y en red, apoyadas sobre estándares abiertos de Internet.

#### **22.4.2. Clasificación de los SIG**

De acuerdo con la funcionalidad que integran y el tipo de problema que pretenden resolver, podemos distinguir los siguientes grupos de sistemas de información geográfica.

1. *SIG Profesional*: se enfocan hacia usuarios con un alto nivel de especialización y formación en este campo. Integra todas las funciones que se pueden necesitar en un SIG a nivel de recopilación y edición de datos, administración de BD, análisis y geoproceso avanzado y todas las herramientas específicas que puedan ser necesaria para mantenimiento de la información.
2. *SIG de Sobremesa*: se enfocan hacia la explotación y utilización de la información. Incorpora herramientas de análisis de la información, además de mecanismos avanzados para la presentación de resultados como son, informes, gráficos, mapas, etc. Presentan una gran facilidad de manejo, con lo cual los usuarios no necesitan ser expertos en el ámbito, además las herramientas que integran son potentes y facilitan el acceso avanzado a la información.

3. *Visualizadores SIG*: se trata de herramientas sencillas que se centran exclusivamente en la visualización de la información, de distintos tipos y formatos..
4. *WebGIS*: se trata de proporcionar el acceso a datos cartográficos y a las funcionalidades (servicios) de los SIG a través de la red. Cada vez más se tiende hacia la estandarización de este tipo de servicios liderado por OpenGIS.
5. *SIG de Componentes*: con la expansión en el campo de la ingeniería de software de los desarrollos basados en componentes, se ha alcanzado la posibilidad de incorporar funcionalidades espaciales en todo tipo de aplicaciones (captación espacial de aplicaciones), lo que supone un nuevo impulso para la generalización del uso de la información espacial a nuevos campos, en los que se pueden realizar interesantes sinergias.
6. *SIG de Dispositivos Móviles*: se apoya en el uso de PDAs y Smartphones. Estos dispositivos tienen capacidad suficiente como para soportar casi todas las funciones de un sistema tradicional.

### **22.4.3. Áreas de Aplicación**

Los productos SIG comerciales son cada vez más comunes y populares, por lo que recoger todas las áreas posibles de aplicación es una ardua tarea. Sin embargo, en el siguiente listado se presentan las más destacadas o donde el número de desarrollos es mayor.

## **Demografía**

En esta categoría se recogen todas las aplicaciones, que si bien pueden ser de naturaleza muy diversa, comparte el hecho de que utilizan características demográficas y socioeconómicas, y la distribución espacial de las mismas para la toma de decisiones.

Los datos en los que se apoyan este tipo de sistemas suelen proceder de registros estadísticos confeccionados por algún organismo (oficial o no).

Las aplicaciones dentro de esta categoría se suelen centrar en el marketing evaluación del impacto de un servicio, selección de lugares para el establecimiento de negocios o servicios, etc.

## **Gestión y Planificación Urbana**

Esta categoría se orienta a actividades propias de gestión municipal como son, la gestión de servicios de infraestructura (alumbrado, alcantarillado, mobiliario urbano, etc.), la gestión del tráfico, la gestión de tasas y licencia, el emplazamiento para instalaciones y servicios comunitarios, etc.

Este tipo de sistemas suele manejar escalas grandes y se usa como base el callejero del municipio en cuestión. Además este tipo de aplicaciones suele emplear un modelo de datos de tipo vectorial.

## **Gestión de Instalaciones**

En esta categoría se agrupan los desarrollos orientados a compañías de suministros y servicios, como son electricidad, agua, ferrocarril, etc. Las

aplicaciones tipo de este grupo pasan por la gestión del mantenimiento, la relación con el cliente (notificaciones de cortes de suministro), diseño de instalaciones, etc.

Estos sistemas se caracterizan por:

- La precisión necesaria suele ser elevada.
- Existe una fuerte estructura en red, necesaria para la realización de análisis.
- Se establecen conexiones con bases de datos externas.
- Existe una jerarquía de componentes de la red.

## **Aplicaciones de gestión e inventario de recursos**

En esta categoría se incluyen campos como la gestión forestal, la planificación agraria, la evaluación del impacto ambiental, la gestión del territorio, la del patrimonio natural y la del medio ambiente.

Normalmente manejan escalas pequeñas con diversas calidades en los datos e incluso sin contrastar. Estas aplicaciones usan modelos de datos tanto vectoriales como ráster.

## **Gestión Catastral**

Esta categoría se orienta a la gestión de la propiedad inmobiliaria y por su importancia ha adquirido un término específico, Sistemas de Información Territorial (SIT).

En nuestro país contamos con el Sistema de Información Catastral, que cuenta con datos y descripciones de las propiedades tanto a nivel urbano, como a nivel rústico.

## **22.5. Bibliografía**

- Introduction to Knowledge Management. Filemon A. Uriarte Jr.
- “SilverStripe: The Complete Guide to CMS Development”. Ingo Schommer y Steven Broschart. Ed. Wiley, 2009. ISBN: 04 7068183 1.
- “WordPress, The best Content Management System (CMS) Guide by Heinz Duthel”. Heinz Duthel. Ed. IAC Society, 2010.
- “The Official Joomla! Book”. Jennifer Marriott, Elin Waring. Ed. Addison-Wesley Professional, 2010. ISBN: 03 217 0421 5.
- “Using Drupal”. Angela Byron, Addison Berry, Nathan Haug, Jeff Eaton, James Walker, Jeff Robbins. Ed. O'Reilly Media, 2008. ISBN: 05 965 1580 4.
- Información geográfica y sistemas de información geográfica. Juan A. Cebrián de Miguel
- Sistemas de información geográfica. Joaquín Bosque Sendrá
- Sistemas de Información Geográfica Aplicados a la Gestión del Territorio. Juan Peña Llopis

**Autor:** Francisco Javier Rodríguez Martínez

Subdirector de Sistemas Escola Superior Enxeñaría Informática Ourense  
Colegiado del CPEIG



**23. MOTORES DE BUSCA.  
SERVIDORES DE MENSAJERÍA.  
SISTEMAS DE CORREO. LISTAS  
DE DISTRIBUCIÓN. GRUPOS DE  
NOTICIAS DE RED (NNTP).  
SISTEMAS DE  
VIDEOCONFERENCIA.  
MENSAJERÍA INSTANTÁNEA.  
ACCESIBILIDAD Y USABILIDAD.  
W3C. E-LEARNING. WEB 2.0.  
WIKIS. BLOGS. COMUNIDADES  
VIRTUALES. REDES SOCIALES.  
SINDICACIÓN DE CONTENIDOS.  
PODCAST. SUITES DE OFIMÁTICA  
EN WEB. ALMACENAMIENTO EN  
WEB. ESCRITORIOS VIRTUALES.  
P2P. WEB SEMÁNTICA.**

**Tema 23: Motores de busca. Servidores de mensajería. Sistemas de correo. Listas de distribución. Grupos de noticias de red (NNTP). Sistemas de videoconferencia. Mensajería instantánea. Accesibilidad y usabilidad. W3C. E-learning. Web 2.0. Wikis. Blogs. Comunidades virtuales. Redes sociales. Sindicación de contenidos. Podcast. Suites de ofimática en Web. Almacenamiento en web. Escritorios virtuales. P2P. Web semántica.**

---

**ÍNDICE**

<b>23.1 Motores de búsqueda.....</b>	<b>2</b>
23.1.1 Spiders.....	2
23.1.2 Directorios.....	2
23.1.3 Sistemas mixtos (directorio y motor de búsqueda).....	3
23.1.4 Metabuscadores.....	3
23.1.6 Multibuscadores.....	3
<b>23.2 Servidores de Mensajería.....</b>	<b>3</b>
23.2.1 Sistema de mensajería Centralizada.....	4
23.2.2 Sistema de mensajería Distribuido.....	4
<b>23.3 Sistemas de Correo.....</b>	<b>4</b>
23.3.1 Elementos del servicio de correo electrónico.....	5
23.3.2 Dirección de correo electrónico.....	5
23.3.3 Proceso de envío de mensajes.....	5
<b>23.5 Listas de distribución.....</b>	<b>6</b>
<b>23.6 Grupos de noticias de red (NNTP).....</b>	<b>7</b>
<b>23.7 Sistemas de videoconferencia.....</b>	<b>7</b>
<b>23.8 Mensajería instantánea.....</b>	<b>8</b>
<b>23.9 E-LEARNING.....</b>	<b>8</b>
23.9.1 Introducción.....	8
23.9.2 Concepto.....	8
23.9.3 Plataformas de e-Learning.....	8
23.9.4 Ventajas.....	9
23.9.5 Inconvenientes.....	10
23.9.6 Estandarización.....	10
23.9.7 Plataforma Moodle .....	11
<b>23.10 Accesibilidad y Usabilidad.....</b>	<b>11</b>
23.10.1 Accesibilidad como cualidad de los sistemas.....	11
23.10.2 Limitaciones en la accesibilidad.....	12
23.10.3 Promoviendo la accesibilidad.....	12
23.10.4 Usabilidad.....	12
<b>23.11 W3C.....</b>	<b>13</b>
<b>23.12 Web 2.0.....</b>	<b>13</b>
<b>23.13 Wikis.....</b>	<b>15</b>
<b>23.14 Blogs.....</b>	<b>16</b>

<b>23.15 Comunidades Virtuales.....</b>	<b>16</b>
<b>23.16 Redes Sociales.....</b>	<b>18</b>
<b>23.17 Sindicación de contenidos.....</b>	<b>18</b>
23.17.1 Fuente Web.....	19
23.17.2 Agregador de noticias.....	19
23.17.3 Formato RSS.....	19
23.17.4 Estándar Atom.....	20
<b>23.18 Podcast.....</b>	<b>20</b>
23.18.1 Podcasting Vs. Streaming.....	21
<b>23.19 Suites Ofimáticas en Web.....</b>	<b>21</b>
23.19.1 Feng Office.....	21
23.19.2 Google Docs.....	22
23.19.3 Office Web Apps.....	23
<b>23.20 Almacenamiento en web.....</b>	<b>23</b>
23.20.1 Dropbox.....	24
<b>23.21 Escritorios Virtuales.....</b>	<b>24</b>
23.21.1 eyeOS.....	24
<b>23.22 Mashups.....</b>	<b>25</b>
<b>23.23 P2P.....</b>	<b>26</b>
23.23.1 Características.....	26
23.23.2 Tipos de redes P2P.....	26
<b>23.24 Web semántica.....</b>	<b>28</b>
23.24.1 Definición de web semántica.....	28
23.24.2 RDF, SPARQL y OWL.....	28
<b>23.25 Bibliografía.....</b>	<b>30</b>

## **23.1**

### **Motores de búsqueda**

En los inicios, internet comenzaba ofrecer una gran cantidad de información, que difícilmente podía ser catalogada y referenciada. Esto suponía un impedimento a la hora de realizar búsquedas de información relativas a temáticas concretas, provocando una alta ineficiencia a nivel general en el uso de internet y el acceso a la red para la búsqueda y/o divulgación de la información. Como solución para este problema surgieron los motores búsqueda.

Los motores de búsqueda, también conocidos como buscadores, son sistemas software que se encargan de localizar sitios web relacionados con un determinado conjunto de términos clave que se le suministran. En esencia, son un sistema informático que consulta los ficheros de las web que se encuentran almacenadas en los servidores web. Para realizar esta tarea, habitualmente utilizan una pieza de software específicamente diseñado para analizar la red en busca de webs y obtener información que permita clasificarlas mediante términos clave o bien utilizando árboles jerárquicos por temas.

La solución ofrecida por los motores de búsqueda no es total, sino parcial. Esto es debido a que en realidad no buscan en Internet cada vez que realizamos una consulta. La búsqueda la realizan en una base de datos en la cuál almacenan referencias de las páginas accesibles junto con datos concretos, metainformación, que sirve para catalogarlas. Esta información habitualmente se recoge a través de un programa (habitualmente robot) que es el que se encarga de realizar visitas periódicas por todo el contenido disponible del web. No existe unicidad en los criterios de selección para la agregar nuevas páginas a las base de datos de los motores de búsqueda. El

resultado es que cada base de datos contiene información de muy diversa calidad, especificando sus propios criterios de selección, y consecuentemente, estableciendo categorizaciones y resultados diferentes para cada búsqueda en función del motor de búsqueda con el que estemos trabajando.

A nivel general, se pueden distinguir cinco tipos básicos de motores de búsqueda, diferenciados entre sí fundamentalmente por el tipo de información que albergan, o los mecanismos que utilizan para realizar la referenciación de las páginas que ofrecen. Estos cinco tipos son los spiders, directorios, sistemas mixtos, metabuscadores y multibuscadores.

### **23.1.1**

#### ***Spiders***

Los Spiders, también conocidos como arañas web o crawlers, son programas que revisan las páginas web de forma metódica y automática. Habitualmente realizan copias de las páginas web visitadas para un procesamiento posterior que consiste en indexar dichas páginas en función de su contenido, determinado por conjuntos de términos clave, para proporcionar un sistema de búsqueda posterior más optimizado.

El funcionamiento es simple. La araña se inicia con una lista de URLs o páginas a visitar. A medida que va consultando las páginas, va añadiendo todos los hipervínculos que se encuentran en dichas páginas a una lista de URLs que visitará de forma recurrente en función de unas reglas establecidas. Las visitas se realizan de forma periódica, por tanto, es posible que en determinadas ocasiones el contenido no aparezca totalmente actualizado. El orden en que se muestran los resultados de la

consulta está determinado por diversos factores que dependen de cada buscador en particular,.

La gran mayoría de los buscadores que se utilizan habitualmente entran dentro de la categoría de Spiders. Son sistemas costosos, que hacen uso intensivo de una gran cantidad de recursos.

Algunos ejemplos de “Spiders” son Google, Bing o Hotbot

### **23.1.2** ***Directorios***

Los directorios son un tipo de motores de búsqueda con un funcionamiento totalmente distinto de los “Spiders”. Los directorios son simplemente listas categorizadas de recursos, que se estructuran jerárquicamente. Esta estructura se organiza en forma de árbol, permitiendo visualizar los contenidos con diferente grado de granularidad, desde los más generales a los más específicos.

En realidad estos motores no disponen de ningún software específico que analice los contenidos web, sino que realiza las clasificaciones y categorizaciones del material en función de un conjunto de criterios seleccionados de forma manual. Esto implica que la tecnología en la que se basan sea barata y sencilla, sin embargo el coste operacional es alto, ya que siempre se requiere de intervención humana.

Algunos ejemplos de directorios son Yahoo! y Open Directory Project

### **23.1.3** ***Sistemas mixtos (directorio y motor de búsqueda)***

Los sistemas mixtos combinan características de directorios y motores de búsqueda. Disponen habitualmente de alguna pieza software

de tipo “Spider” para realizar el análisis de la web, y además permiten añadir y presentar páginas clasificadas en catálogos según su contenido. La combinación de estas características representa la tendencia actual en los buscadores más importantes.

### **23.1.4**

### ***Metabuscadores***

Los metabuscadores son un tipo de motores de búsqueda que centran sus resultados en búsquedas que realizan sobre otros buscadores. Esto significa que obtienen inicialmente un conjunto de resultados de otro buscador, y a continuación refinan esos resultados presentando una selección propia.

Una de las principales ventajas de los metabuscadores es que amplían el ámbito de las búsquedas que realiza el usuario. Proporciona una gran cantidad de resultados combinados en función de los criterios particulares de cada metabuscador. En muchas ocasiones estos criterios de ordenación no resultan del todo claros.

Por otro lado, el problema principal es que los metabuscadores no distinguen entre las diferentes sintaxis de los buscadores, limitando la especificidad con la que los metabuscadores pueden trabajar para localizar información. Además, al realizar las búsquedas en diferentes fuentes (buscadores), la obtención de resultados suele demorarse mucho más que al utilizar otro tipo de motor de búsqueda.

Algunos ejemplos de metabuscadores son Metacrawler, Ixquick, Dogpile o Metabuscador

### **23.1.5**

### **23.1.6**

### ***Multibuscadores***

Los multibuscadores son un tipo de motores de búsqueda similares a los metabuscadores, pero con una diferencia notable; mientras que los metabuscadores no distinguen las diferentes sintaxis de los buscadores que utilizan para la obtención de resultados, los multibuscadores sí lo hacen. Esto implica que puedan lanzar varias búsquedas en motores seleccionados respetando el formato original de los buscadores.

Los multibuscadores son útiles para realizar búsquedas en diferentes buscadores al mismo tiempo. Su operativa difiere de los buscadores normales, dado que no disponen de componentes software que analicen y almacenen contenido, sino que lo único que contienen es un registro de buscadores y los criterios de adecuación de las expresiones de búsqueda asociadas a cada buscador. Los multibuscadores no almacenan información de páginas relativas a contenido. Simplemente realizan la consulta adecuada a cada buscador dentro de su registro, y realizan un filtrado de los enlaces repetidos, y aplican además criterios de selección como la relevancia de cada enlace en los diferentes buscadores, para generar finalmente una lista de resultados. Un ejemplo de multibuscador es [iniciodirecto.com](http://iniciodirecto.com).

## **23.2**

### **Servidores de Mensajería**

Un servidor de mensajería es una aplicación que posee la capacidad para manejar mensajes entre dos o más entidades, ya sean aplicaciones de usuario u otros sistemas de gestión de mensajes. Los mensajes de un servidor de mensajería son enviados a través de un middleware, el cual facilita la comunicación entre los distintos elementos del sistema, utilizando generalmente un conjunto de reglas y especificaciones que posibilitan la comunicación entre las distintas partes. Otra de las características de los servidores de mensajería es la capacidad de almacenaje de los mensajes, este almacenamiento se produce generalmente en una cola hasta que es



posible el envío del mismo hacia su destinatario, que por lo general resulta ser otra aplicación.

Es muy habitual encontrarse en una empresa u organización un sistema de mensajería funcionando en un servidor y esperando el envío de mensajes a su cola de entrada. Desde allí, el middleware analiza mensaje a mensaje determinando el destino de cada uno. Una vez en el servidor, un mensaje sólo tiene dos posibilidades de entrega, o ser enviado de manera local, o que éste tenga que ser redirigido a otro servidor de mensajería para que sea él el que realice la entrega. Si el mensaje ha de ser entregado a un destino local, entonces es enviado inmediatamente al buzón local. Por el contrario, si el mensaje es determinado como remoto, el servidor de mensajería debe enviar el mensaje a otro servidor de mensajería dentro de su entorno para que sea éste el que realice la entrega del mensaje.

Por lo general, si existen problemas de conexión entre los servidores o no es posible determinar la localización del servidor de mensajería remoto, el usuario el cual realizó el envío del mensaje es informado a través de un mensaje enviado por el servidor de mensajería, informando de la situación. Este tipo de mensaje suelen ser sólo de notificación de que se están teniendo problemas con el envío del mensaje, puesto que el servidor de mensajería continuará intentando enviar el mensaje hasta que se agoten el número máximo de intentos de envío o hasta que el mensaje caduque, es decir, exceda un límite de tiempo de estancia en el servidor.

Los modelos de los servidores de mensajería suelen adaptarse a una arquitectura centralizada o seguir una solución distribuida.

### **23.2.1** ***Sistema de mensajería Centralizada***

Un sistema de mensajería centralizada se fundamenta en un núcleo de datos, el cual aloja todos los recursos y servicios de los servidores que conforman el sistema. Este núcleo de datos permite que cualquier usuario

del sistema de mensajería se conecte a los servicios de mensajería, ya sea de forma local o remota.

Las características de un sistema de mensajería centralizado son:

- **Datos:** Todos los datos y la información se encuentra albergada y se gestiona desde el núcleo, incluso cuando los usuarios establecen una conexión remota para su utilización. Esta centralización facilita en gran medida la administración de los servicios, puesto que provoca que ésta sea más sencilla.
- **Actualizaciones:** Las actualizaciones se han de realizar únicamente en el núcleo central, en donde se encuentra todo el sistema.
- **Ubicación:** El centro de datos añade al sistema dispositivos de aislamiento de la alimentación o sistemas de alimentación ininterrumpida (SAI). Proporciona además la posibilidad de ofrecer servicios incluso cuando se produce alguna incidencia de carácter grave, ya que posibilita la réplica de todo el sistema de una manera eficaz.

### **23.2.2                      *Sistema de mensajería Distribuido***

Un sistema de mensajería distribuido está formado por una serie de sucursales repartidas en distintas ubicaciones conectadas entre sí. Cada sucursal posee un servidor o servidores de mensajería con todos sus servicios de manera independiente del resto de sucursales. Cada uno de los servidores de mensajería realiza el envío de sus mensajes locales y redirige a los otros servidores aquellos mensajes que no son de dominio local y que sí son capaces de resolver alguno de los otros servidores.

- **Datos:** La información se encuentra también distribuida entre cada una de las sucursales y cada una de ellas gestiona y administra esta información y sus servicios, lo que provoca un aumento en la complejidad de estas tareas.

- Actualizaciones: Cada vez que se lleva a cabo una tarea de actualización ésta ha de realizarse en cada una de las sucursales, para que tenga efecto en todo el sistema.
- Ubicación: Cada sucursal posee su propio centro de datos, los cuales pueden ofrecer los mismos servicios que en una arquitectura centralizada.

## 23.3

### Sistemas de Correo

El correo electrónico, o email, está catalogado como un servicio de red, que proporciona a los usuarios la capacidad para enviar y recibir mensajes y archivos de forma rápida y eficiente a través de dispositivos digitales. El sistema trata de representar una analogía con el correo postal habitual, presentándose como una alternativa para el envío de mensajes de texto o cualquier tipo de fichero en formato digital. Dado que su coste operacional es bajo y su eficiencia es elevada, el correo electrónico está actualmente desplazando al correo postal original.

El origen del correo electrónico es anterior incluso a la red Internet. Los primeros pasos para la creación del sistema de correo electrónico se dieron en el MIT entorno a 1961, cuando se desarrolló un sistema que permitía a varios usuarios, desde terminales remotos, ingresar en un mainframe en el cual podían almacenar una copia de sus archivos en el disco. Éste fue uno de los pasos iniciales en la implementación de mecanismos para la compartición de información. En 1965 comenzó a utilizarse un sistema basado en el almacenamiento de mensajes compartidos entre los usuarios de una supercomputadora dando lugar al primer sistema de email utilizado. Posteriormente, en 1971, se incorporó al sistema de mensajería el uso de la arroba (@) como elemento para dividir

el nombre de los usuarios de la máquina en la que se encontraban alojados.

### **23.3.1** *Elementos del servicio de correo electrónico*

Técnicamente hablando, el correo electrónico es un servicio proporcionado en internet, soportado por el protocolo SMTP (Simple Mail Transfer Protocol) y el protocolo POP (Post Office Protocol). Como en todo servicio, la arquitectura del sistema consta de una parte cliente, habitualmente utilizada por los usuarios para enviar y recibir correos, y una parte servidora, que proporciona los mecanismos adecuados para el almacenamiento y transferencia de los correos entre los diferentes clientes. Los protocolos definen los esquemas de comunicación entre los clientes y los servidores para que los mensajes se transmitan de un sitio a otro. El protocolo SMTP es el protocolo encargado del envío de los mensajes, mientras que el protocolo POP es el encargado de la recepción de los mensajes.

El **cliente de correo electrónico**, también llamado **Mail User Agent (MUA)** es un programa que básicamente permite gestionar los mensajes recibidos así como recibir correos nuevos. Habitualmente se hace referencia a cliente de correo electrónico, se hace referencia a aplicaciones stand-alone que proporcionan un amplio abanico de funcionalidades para la gestión de nuestro correo. Sin embargo, actualmente la mayoría de los proveedores de servicio de correo permiten el acceso a través de los navegadores web, proporcionando interfaces web a modo de cliente para la consulta y gestión de nuestro correo electrónico. Existe una gran diferencia respecto del funcionamiento de ambas opciones; cuando se utiliza un cliente de correo electrónico, todos los mensajes disponibles se descargan en el ordenador en el que se esté ejecutando ese cliente de correo electrónico. Sin embargo, cuando se accede a la cuenta de correo a través

de las interfaces web, los mensajes siguen almacenados en el servidor, siendo accesibles a través del cliente web, desde cualquier ordenador que disponga de una conexión a Internet.

Algunos ejemplos de clientes de correo electrónico stand-alone son:

- **Microsoft Outlook:** Cliente privativo de la compañía Microsoft. Es el cliente de correo estándar de Microsoft, incluido en el paquete Microsoft Office.
- **Mozilla Thunderbird:** Alternativa de software libre de Outlook, desarrollado por Mozilla.

Algunos ejemplos de proveedores de servicio de correo web son gmail, hotmail o yahoo!.

En cuanto al **servidor de correo electrónico** consiste en un conjunto de aplicaciones informáticas ubicadas en un equipo servidor, ya sea en red local o en Internet, cuya tarea consiste en realizar una serie de procesos que tienen la finalidad de transportar información entre los distintos usuarios del servicio. El servidor de correo electrónico es el encargado de gestionar a todos los usuarios registrados en el sistema con sus correspondientes identificaciones (direcciones de correo electrónico) que servirán para poder interactuar entre sí mediante el envío de correos.

El **servidor de correo electrónico**, dispone de una pieza software denominada Agente de Transferencia de Correo (MTA) o Agente de Transporte de Mensajes, cuyo objetivo consiste en transmitir los datos de una máquina a otra. En concreto, se centra en la parte de transferencia de datos entre distintos servidores, ejerciendo diferentes roles como servidor de otros servidores de correo, cliente de otros servidores de correo y como intermediario entre el cliente de correo que emite el mensaje y otro servidor de correo externo.

Actualmente, a excepción de las grandes corporaciones que disponen de su propia infraestructura TIC, la mayoría de los usuarios hacen uso de servidores de correo electrónico que pertenecen a alguna entidad proveedora de dicho servicio. Existen diferentes empresas y entidades que ofrecen servicios de correo electrónico, tanto de forma gratuita como de pago. Los servicios de correo gratuito son los más conocidos por los usuarios, y entre ellos podemos destacar los servicios de gmail, hotmail o yahoo!. Las entidades registradoras de dominio son las que habitualmente ofrecen servicios de correo electrónico asociados a la cuenta de dominio contratada. En cuanto a las soluciones software para la implantación de un servidor de correo electrónico, pueden destacarse Microsoft Exchange Server para plataformas Windows o Sendmail, Qmail, Zimbra y Postfix para Unix/GNULinux.

### **23.3.2** ***Dirección de correo electrónico***

La dirección de correo electrónico es una secuencia de palabras que tienen por objeto identificar a un determinado usuario de un servicio de correo de forma inequívoca. Esta dirección de correo representa el identificador mediante el cuál el usuario puede enviar y recibir correos.

La sintaxis de una dirección de correo es la siguiente:

- **Nombre de usuario:** Es conjunto de palabras escogidas por el usuario que habitualmente suele coincidir con el nombre o algún identificativo de la persona o usuario que utilizará la cuenta de correo. Puede contener letras, números y algunos signos.
- **@:** es el signo o símbolo encargado de separar dos partes importantes de la dirección de correo, concretamente el nombre de usuario y el dominio.
- **Nombre de dominio:** El nombre de dominio en internet es una identificación asociada a un dispositivo o grupo de dispositivos.

Habitualmente se corresponde con el nombre del proveedor del servicio de correo.

### **23.3.3** *Proceso de envío de mensajes*

Se detalla a continuación el proceso de envío y recepción de correos electrónicos y los elementos y protocolos que intervienen entre un ordenador A y un ordenador B.

El ordenador con el cliente A redacta un correo electrónico para el cliente B y lo envía. Al realizar la operación de envío, el cliente de correo en A contacta con el Servidor de correo A a través del protocolo SMTP, le transfiere el correo y le da la orden de enviarlo. Al recibir la petición, el servidor de correo A verifica que el correo pertenece a otro dominio. Para resolver la dirección a la cuál tiene que enviar el correo, realiza una consulta a un servidor de DNS para averiguar quién es el encargado de gestionar el dominio asociado al cliente B. Una vez obtenida la respuesta, y resuelto el servidor de correo B, el servidor de correo A se comunica con él a través del protocolo SMTP, enviándole el correo emitido desde el cliente A, y quedando este correo almacenado en el servidor B. Posteriormente, cuando el cliente B decida consultar el correo, accederá mediante el protocolo POP al servidor de correo B y se descargará los mensajes almacenados en dicho servidor que tengan por destinatario al usuario de Cliente B.

## 23.4

## 23.5

### Listas de distribución

Las listas de distribución, o listas de correo electrónico son agrupaciones de usuarios de correo electrónico. Mediante un software apropiado se pueden configurar listados de direcciones de email para el envío masivo de información a múltiples usuarios a la misma vez. Cada lista de distribución de correo electrónico está a su vez referenciada por una dirección email. A grandes rasgos, cada vez que un usuario autorizado emite un email con la dirección de la lista de distribución como destinatario, en realidad la lista reenviará el email a todos los usuarios adscritos a esa lista.

Las listas de correo electrónico son una de las herramientas cada vez más utilizadas en las organizaciones para mantener a los usuarios informados con noticias e información de interés. De forma habitual, es necesario que los propios usuarios se registren en esas listas de las cuáles están interesados en recibir noticias o información.

Las listas de correo electrónico están gestionadas por el propio servidor de correo, o por software adicional específico para su gestión. Para el alta, baja o modificación de los datos de los usuarios, es habitual que los servidores de listas de correo electrónico pongan a disposición de los usuarios una o varias direcciones de correo a las cuáles enviar comandos. Además algunos servidores de listas de correo permiten diferentes modos de subscripción:

- **Modo individual:** el usuario de la lista recibe todos los mensajes que formen parte de la misma. De la misma forma, si el usuario dispone de los privilegios necesarios, puede enviar correos a la lista de distribución.



- Modo no correo: el usuario no recibe los mensajes que se envían a la lista pero puede enviar correos a la lista. Habitualmente esta opción permite la consulta de los correos a través de interfaz web.
- Modo resumen diario: también llamado modo digest, consiste en que el usuario solamente recibe un correo diario que incluye todos los mensajes enviados a la lista de correo.

Tipos de listas de correo electrónico:

- Boletín electrónico: se utiliza como medio unidireccional para la transmisión de información debido a que sólo pueden enviar mensajes a la lista determinadas personas encargadas de la publicación y gestión de dicho boletín.
- Lista de debate: En este tipo de lista, cualquier suscriptor puede enviar correos a la lista de distribución, y el resto de usuarios pueden contestarlos de la misma manera. De esta forma se pueden generar debates e intercambios de información. Las cadenas de correos van generando hilos que pueden ser contestados por cualquiera de los usuarios de la lista.

Existen en internet diferentes servicios que permiten la creación de listas de correo electrónico de forma gratuita, como por ejemplo Google Groups, Yahoo! o eListas.

A nivel de implementación, existen diferentes productos basados en software libre para la configuración y gestión del servicio de listas de correo, como por ejemplo phpList, Sympa, Mailman y Gmane.

## 23.6

### Grupos de noticias de red (NNTP)

Los grupos de noticias son un servicio proporcionado en Internet al cuál los usuarios pueden suscribirse para participar de forma similar a las listas de correo. En esencia, los grupos de noticias serían similares a un tablón electrónico de noticias categorizadas jerárquicamente por temáticas. El contenido dentro del servicio de noticias se organiza como un gran número de grupos, en los que se agrupan los diferentes temas. El nombre de cada uno de los grupos de noticias disponibles en la red consta de un conjunto de identificadores separados por puntos, habitualmente relacionados con el dominio de la entidad que gestiona el servidor de noticias, o bien siguiendo un estándar definido para las principales redes de grupos de noticias. Esto permite al usuario suscribirse a un grupo o grupos determinados que resulten de su interés y recibir todos los mensajes que el resto de usuarios envían a ese grupo.

Este funcionamiento se asemeja bastante al de una lista correo, de hecho comparten un objetivo intrínseco que consiste en la generación de espacios de debate y foros de discusión sobre algún tema concreto. La diferencia es que con las listas de correo se reciben directamente los mensajes en el cliente de correo, mientras que con las news o grupos de noticias, se necesita conectarse a un servidor de noticias y extraer los grupos en los que el usuario esté interesado.

Una vez que el usuario se registra en un grupo o grupos determinados, puede agregar mensajes al sistema. Puede también publicar noticias que contesten o repliquen otras noticias previas, formando hilos de debate. Para la gestión de las noticias se necesita software específico para este servicio. Actualmente la mayoría de los clientes de correo web vienen preparados para desempeñar esta función. Las funcionalidades básicas que deben proporcionar son las de permitir seleccionar los grupos de interés

para el usuario, lectura de noticias publicadas por otros y envío de noticias al servidor.

Dada la cantidad de mensajes que se generan a diario en los grupos de noticias, es habitual que los servidores de noticias públicos dispongan de mecanismo para evitar la saturación de sus sistemas de almacenamiento. Una de ellas consiste en estipular un tiempo de vida determinado para las noticias que se van almacenando en el sistema. Al cabo de ese periodo de vida, el contenido es eliminado.

Los grupos de noticias se clasifican jerárquicamente en función de sus temáticas, proporcionando una ayuda importante a la hora de localizar los temas de interés.

## **23.7** **Sistemas de videoconferencia**

Los sistemas de videoconferencia tienen como principal característica el permitir comunicación simultánea y bidireccional de señales de audio y vídeo, lo que proporciona capacidad para mantener reuniones con personas situadas en lugares alejados.

Éste tipo de sistemas habitualmente integran capacidades que abarcan parte de los sistemas vistos con anterioridad ya que integran la capacidad de establecer comunicación escrita (chat) y gestión de mensajería instantánea. Además pueden incluir capacidad para la transmisión de ficheros y edición en herramientas colaborativas.

La base tecnológica de los sistemas de videoconferencia es la compresión digital de los flujos de audio y vídeo en tiempo real.

En cuanto a la categorización de los sistemas de videoconferencia, podemos clasificarlos fundamentalmente en dos grandes grupos:

- **Sistemas de videoconferencia dedicados** : Disponen de los componentes hardware necesarios para realizar una videoconferencia en remoto. Son sistemas de alta calidad, especiales para las circunstancias en las que se demanda una alta fiabilidad y calidad de los datos transmitidos. Por lo general, éste tipo de dispositivos constan de una cámara de vídeo de alta calidad y una consola. Dentro de este tipo de sistemas podemos distinguir varios tipos de dispositivos hardware en función del objetivo del sistema:
  - o Grupos grandes: son dispositivos grandes , no portátiles, más costosos utilizados para grandes salas y auditorios. Requieren de instalación y mantenimiento adecuados.
  - o Grupos reducidos: no son portátiles, son más pequeños y menos costosos, utilizados para salas de reuniones pequeñas. Requieren de instalación.
  - o videoconferencia individual: Se trata de dispositivos portátiles, destinados a usuarios individuales, tienen cámaras fijas, micrófonos y altavoces integrados en la consola.
- **Sistemas de videoconferencia de escritorio**: Los sistemas de escritorio, o sistemas de usuario, se basan en la combinación de parte software y hardware. En cuanto a parte software, se trata de algún cliente de mensajería con capacidad para realizar videoconferencia mediante la transmisión de una cámara web y un micrófono conectado al ordenador. En cuanto a los dispositivos hardware, simplemente serían necesarios una cámara web y un micrófono. En la actualidad, prácticamente la gran mayoría de los sistemas de mensajería instantánea soportan videoconferencia. Es el caso por ejemplo de los clientes de MSN Messenger o Skype.

## 23.8

### Mensajería instantánea

El término chat es un anglicismo que hace referencia al charla o cibercharla. Se utiliza para designar las comunicaciones escritas realizadas en tiempo real a través de Internet entre dos o más personas. Puede realizarse a través de canales públicos, o mediante canales privados, dependiendo del medio y protocolos que se utilicen. Éste tipo de características determinan las diferentes tipologías de chat:

- **Webchat:** Es un tipo de chat en el que los mensajes se transmiten a través de WWW. Es un tipo de chat de fácil acceso, dado que las interfaces están implementadas como aplicaciones web, accesibles desde cualquier navegador. Resulta sencillo de utilizar dado que existen una gran cantidad de componentes visuales que ayudan a personalizar rápidamente los estilos de escritura y visualización en este tipo de aplicaciones de acceso a los webchats, lo que lo hace resultar atractivo para usuarios noveles. Sin embargo el uso de los webchat está decayendo dado que las tareas de actualización del contenido de la página que carga el webchat, así como la inestabilidad de algunos navegadores hace que mantener las conversaciones en tiempo real sea difícil en algunas ocasiones.
- **IRC (Internet Relay Chat):** Representa la forma más conocida y antigua de chat que existe. IRC es un protocolo de comunicación en tiempo real basado en texto que permite comunicación entre usuarios sin necesidad de acuerdo previo de establecer la comunicación, es decir, que dos usuarios que se encuentren en un canal pueden comunicarse entre sí sin necesidad de haber establecido

una comunicación previa. El IRC presenta un modelo cliente-servidor donde las aplicaciones cliente de los usuarios, que habitualmente son aplicaciones stand-alone, se configuran y conectan contra un determinado servidor de IRC, permitiendo establecer comunicación con el resto de personas que se encuentran conectadas a dicho servidor, bien mediante chat privado, o a través de canales de libre acceso. En este modelo de chat, existen agentes moderadores que intervienen en la administración y control de todo lo que sucede en cada servidor de IRC. El cliente más habitual de este tipo de redes es el *mIRC*.

- **Mensajería instantánea:** Puede considerarse otra modalidad de chat. En esencia es similar al IRC, dado que la arquitectura de este tipo de sistemas que proporcionan mensajería instantánea se basa en un modelo cliente-servidor en el cuál aplicaciones stand-alone se conectan contra el servidor y permiten enviar y recibir mensajes de otros usuarios conectados al servidor. Sin embargo la gran diferencia radica en que en los sistemas de mensajería instantánea, para que dos usuarios se comuniquen, deberá de existir un contacto previo mediante el cuál ambos usuarios accedan a establecer la comunicación. La mayoría de estos sistemas cuentan con su propia red que únicamente es accesible mediante el cliente propio de esa red, desarrollado por una entidad o compañía concreta. En ese sentido, se presenta como un modelo de comunicación limitado y controlado que se está comenzando a adoptar en algunas empresas y corporaciones para establecer un mecanismo de intercambio de información de forma económica, controlada y fiable. Algunos sistemas de

**mensajería instantánea más comunes son el MSN Messenger, Yahoo Messenger o ICQ.**

## **23.9**

## **E-LEARNING**

### **23.9.1**

### ***Introducción***

En los últimos años han aparecido sistemas informáticos orientados a la enseñanza y aunque el objetivo de todos ellos es muy similar, los medios mediante los cuales llegan a dicho objetivo varían en gran medida. El involucrar a las nuevas tecnologías en el ámbito de la enseñanza, introduciendo las mismas como herramienta fundamental del proceso de aprendizaje, ha desembocado en la aparición de un nuevo término conocido como eLearning. Las tecnologías asociadas se engloban en un conjunto de sistemas que tratan de proporcionar los medios y mecanismos adecuados para facilitar los procesos de aprendizaje en prácticamente todas las áreas de conocimiento. Sin embargo, muchos de estos sistemas, mal identificados como “sistemas de eLearning”, únicamente se centran en la gestión y clasificación de documentos para ponerlos a disposición de alumnos y docentes, como es el caso de los Sistemas de Gestión de Contenidos, o de los Sistemas de Gestión Documental. Aunque ciertamente facilitan la tarea de búsqueda y organización de información, este tipo de sistemas no realizan ningún tipo de seguimiento del proceso de aprendizaje de los alumnos.

La idea de eLearning, y en consecuencia, los sistemas de eLearning, pretenden precisamente abarcar esa fase del proceso de aprendizaje, proporcionando los mecanismos necesarios para realizar el seguimiento del proceso de forma íntegra.

### 23.9.2

#### *Concepto*

El concepto de eLearning se define de muchas formas diferentes, fundamentalmente debido a que los actores que de él hacen uso son muy diversos, cada cual con su idiosincrasia y su ámbito de aplicación.

A nivel general, se puede definir el eLearning como la educación a distancia completamente virtualizada a través de las nuevas posibilidades tecnológicas que hay disponibles, como las nuevas redes de comunicación, y fundamentalmente la red de red, Internet. Fundamentalmente se utilizan para ello herramientas o aplicaciones de hipertexto, que proporcionan la ventaja de ser totalmente portables y accesibles desde cualquier plataforma. La idea es que este tipo de sistemas den soporte a los procesos de enseñanza-aprendizaje. Este tipo de sistemas pueden englobarse como un subgrupo de los Sistemas de Gestión de Contenidos, entendiendo estos últimos como una generalización, y asumiendo los sistemas de eLearning como una especialización de los SGC para un propósito específico con funcionalidades propias.

Algunos teóricos dividen el eLearning en tres ramas diferentes:

- computer aid instruction (CAI)
- computer-managed instruction (CMI)
- computer supporter learning resources (CSLR)

El primer término abarca la porción de productos de eLearning que proporcionan enseñanza como tutoriales, simulaciones y ejercicios. El segundo término se refiere a los productos de eLearning que tienen funciones de evaluación, seguimiento y guía de estudio. Finalmente, el tercer término cubre los aspectos del eLearning que dan soporte al desempeño, la comunicación y el almacenamiento. Esta clasificación se refiere sólo a partes del conjunto total representado por el eLearning.



### 23.9.3

### *Plataformas de e-Learning*

En la práctica, para llevar a cabo un programa de formación basado en eLearning, se hace uso de plataformas o sistemas de software que permitan la comunicación e interacción entre profesores, alumnos y contenidos. Se tienen principalmente dos tipos de plataformas:

- LMS (Learning Management Systems), utilizados para impartir y dar seguimiento administrativo a los cursos en línea.
- LCMS (Learning Content Management Systems), empleados para la gestión de los contenidos digitales. Siguen el concepto básico de los CMS, que es la administración de contenidos, pero enfocados al ámbito educativo.

A veces la diferenciación entre ambas es sólo funcional y en lugar de constituir dos herramientas software diferentes se ofrecen en una misma aplicación, que en España se conoce por el nombre de Plataforma Tecnológica o de Teledocencia.

Entre las herramientas más utilizadas para los ambientes o sistemas eLearning están, como ya se ha dicho anteriormente, los Sistemas de Administración de Aprendizaje o LMS, también ampliamente conocidos como plataformas de aprendizaje. Un LMS es un software basado en un servidor web que provee módulos para los procesos administrativos y de seguimiento que se requieren para un sistema de enseñanza, simplificando el control de estas tareas. Los módulos administrativos permiten, por ejemplo, configurar cursos, matricular alumnos, registrar profesores, asignar cursos a un alumno, llevar informes de progreso y calificaciones. También facilitan el aprendizaje distribuido y colaborativo a partir de actividades y contenidos pre-elaborados, de forma síncrona o asíncrona, utilizando los servicios de comunicación de Internet como el correo, los foros, las videoconferencias o el chat.

El alumno interactúa con la plataforma a través de una interfaz web que le permite seguir las lecciones del curso, realizar las actividades programadas, comunicarse con el profesor y con otros alumnos, así como dar seguimiento a su propio progreso con datos estadísticos y calificaciones. La complejidad y las capacidades de las plataformas varían de un sistema a otro, pero en general todas cuentan con funciones básicas como las que se han mencionado. Entre las plataformas comerciales más comunes se encuentran Blackboard y WebCT, mientras que las más reconocidas por parte del software libre son Moodle y Claroline.

#### **23.9.4**

#### ***Ventajas***

El eLearning permite superar algunas de las barreras existentes en los sistemas de enseñanza asistida por ordenador. Algunas de ellas son:

- Elimina las distancias y favorece la movilidad de los alumnos.
- Aumenta el número de destinatarios que pueden seguir un curso simultáneamente.
- Permite flexibilidad horaria.
- Permite alternar diversos métodos de enseñanza.
- Favorece la interacción entre alumnos. Está demostrado que la no presencia física minimiza la timidez y favorece el establecimiento de comunicación entre los alumnos, especialmente en la adolescencia.
- Anonimato.
- Seguimiento y tutoría del progreso del alumno a través de los canales de comunicación establecidos.

- Posibilidad de escoger entre gran variedad de materiales, cursos y especialidades.
- Minimiza los costes de formación continua en la empresa.
- Favorece la convivencia familiar para alumnos con responsabilidades familiares a su cargo.

Además de por las ventajas enumeradas, intervienen otros factores que favorecen la implantación de sistemas eLearning:

- **Factores económicos:** Se alcanza una mejor relación coste-beneficio en la producción y desarrollo aprovechando la reutilización de componentes tecnológicos y materiales de aprendizaje. Es un factor interesante a la hora de aumentar los niveles de formación en países en desarrollo, con un alto ritmo de crecimiento económico y con grandes necesidades de trabajadores cualificados.
- **Alta disponibilidad de recursos digitales:** Las grandes empresas multinacionales necesitan distribuir materiales de aprendizaje a sitios geográficamente dispersos, para que estén disponibles en cualquier momento desde cualquier lugar. La existencia de un gran número de recursos digitales libres y gratuitos en Internet (imágenes, clips de audio y vídeo, animaciones, etc.) favorecen su reutilización y aprovechamiento por parte de las grandes empresas (o terceros, como puede ser una empresa especializada en la creación de cursos o implantación de sistemas de eLearning) para la creación de cursos a través de sistemas eLearning.
- **Penetración social:** La alta penetración en la sociedad de las nuevas tecnologías en general y de Internet en particular, favorece la aceptación de nuevas vías de información y de comunicación.
- **Ayudas estatales:** Los programas de subvenciones por parte del Estado, las Comunidades Autónomas y el Fondo Social Europeo, han

incentivado la creación y desarrollo de un sector empresarial dedicado a la formación on-line. Estas subvenciones han hecho posible la aparición de programas como los de Formación Continua de trabajadores, que contribuyen a la adaptación de los trabajadores a las más nuevas tecnologías.

### **23.9.5**

#### ***Inconvenientes***

Algunos inconvenientes en el empleo de sistemas de eLearning son:

- **Preparación del estudiante:** Es necesario un esfuerzo para asegurar que los estudiantes tienen las habilidades y conocimientos técnicos, así como el acceso al hardware y software necesarios para completar satisfactoriamente el curso basado en las TICs. Tanto la gestión del tiempo y las habilidades metacognitivas están relacionadas con las actitudes y la motivación del estudiante.
- **Personal dedicado:** Al igual que los estudiantes, los profesores deben tener habilidades técnicas, conocimiento y acceso al hardware y software, necesarios en este caso, para facilitar el diseño y desarrollo del curso basado en las TICs. Y deben tener un excelente manejo del tiempo y la motivación para proporcionar asistencia y llevar el seguimiento del estudiante. No obstante algunos autores diferencian rol del profesor, encargado de la selección de contenidos, seguimiento y asistencia al alumno, del rol del técnico encargado del diseño y creación del curso eLearning a partir de los contenidos, objetivos y metodologías, estableciendo de esta forma la necesidad de diferentes perfiles.
- **Gestión de la información:** A pesar de que se posean unas habilidades técnicas y un manejo del tiempo excepcionales, tanto los

profesores como los alumnos requieren de interfaces que reduzcan las cuestiones logísticas y técnicas. El uso de boletines y listas de distribución pueden ayudar a manejar la sobrecarga de información.

- **Equidad:** No todos los usuarios cuentan con las mismas facilidades de acceso a Internet. La tecnología incrementa las diferencias entre los que tienen y los que no tienen tales posibilidades.
- **Ancho de banda:** Este es uno de los mayores inconvenientes desde hace una década y que está desapareciendo rápidamente con la llegada de líneas de banda ancha. Actualmente, en Europa, el ancho de banda es aceptable y permite transmitir con buenos resultados audio y vídeo sincronizados sin los indeseables “saltos” de antaño.

### 23.9.6

### *Estandarización*

Uno de los principales problemas de los sistemas de eLearning siempre ha sido reutilización de los contenidos, de forma que estos puedan ser utilizados en sistemas diferentes, debido a que la mayoría de los sistemas definían sus propios formatos de almacenamiento y procesamiento de los contenidos educativos, así como la forma de acceder y manejarlos. Esta falta de acuerdo se debe en gran medida a la descoordinación en el desarrollo de estándares para eLearning en la década pasada.

Hoy en día existen multitud de sistemas destinados a la enseñanza, ya sean meros gestores de contenidos, gestores del proceso de aprendizaje o sistemas más completos capaces de dar soporte a procesos administrativos, ofrecer herramientas de autoría y edición de cursos, etc. Sin embargo, a pesar de la variedad existente, su heterogeneidad dificulta la compatibilidad entre ellos. No todos son de código abierto, algunos usan

formatos propietarios y generalmente no es posible reutilizar contenidos y estructuras de aprendizaje entre ellos.

Estas incompatibilidades, ya sean totales o parciales, repercuten negativamente en el coste asociado a la implantación de un sistema de eLearning, puesto que en el mejor de los casos, una vez superado el tiempo de aprendizaje de las distintas aplicaciones del sistema, sería necesaria la readaptación de material ya existente para otros sistemas, o en el peor de los casos, crear dicho material desde cero. Una especificación sobre aprendizaje virtual asegura que el nuevo material siga funcionando exactamente igual independientemente de la plataforma que se utilice, siempre que dichas plataformas cumplan la misma especificación.

## **23.9.7**

### ***Plataforma Moodle***

#### **21.9.7.1**

##### **Introducción**

Moodle es un acrónimo de Module Object-Oriented Dynamic Learning Environment. Consiste en una plataforma que proporciona de forma integral mecanismos para la gestión de cursos. Moodle integra además las herramientas necesarias para crear y gestionar comunidades virtuales orientadas al aprendizaje online. Por tanto, podemos categorizar a Moodle como una plataforma tecnológica de tipo LMS (Learning Management System).

Originalmente Moodle fue creado por Martin Dougiamas. Basó el diseño de la plataforma partiendo de que el conocimiento se construye en la mente del estudiante en lugar de ser transmitido sin cambios a partir de libros. Existe también una importante apuesta por el modelo de aprendizaje colaborativo. El propósito es construir un ambiente centrado en el estudiante que le proporcione capacidad para generar ese conocimiento, basado en las habilidades y conocimientos propios de los tutores o

profesores, en lugar de simplemente publicar y transmitir la información que se considera que los estudiantes deben conocer.

En conclusión, Moodle es un paquete de software para la creación de cursos y sitios web basados en internet, orientado a dar soporte a un marco de educación constructivista. El sistema es multiplataforma y está registrado bajo licencia GNU/GPL.

En cuanto a la arquitectura de la plataforma, Moodle es una aplicación web que se ejecuta en servidores que soportan PHP y haciendo uso de base de datos para la persistencia de la información. Esa base de datos es única, y desde la versión 1.7 Moodle cuenta con una capa de abstracción que le permite seleccionar entre diversos motores de bases de datos, siendo MySQL y PostgreSQL las más utilizadas.

### **Principales características**

Moodle, como sistema englobado dentro de los gestores de contenido, y a su vez como sistema específico de eLearning, tiene las siguientes características:

- Promueve una pedagogía constructivista social fundamentada en el trabajo colaborativo, la realización de actividades y debates.
- Su arquitectura y herramientas son apropiadas para clases en línea, además de servir como complemento del aprendizaje presencial.
- Tiene una interfaz de navegador de tecnología sencilla, ligera, y compatible.
- Para su puesta en producción, únicamente es necesaria una plataforma que soporte PHP y la disponibilidad de una base de datos. Gracias a su capa de abstracción, Moodle soporta los principales sistemas gestores de bases de datos.

- Es una plataforma segura. Todos los formularios son revisados y las cookies cifradas.
- Es adaptable y extensible. La mayoría de las áreas de introducción de texto pueden ser editadas usando el editor HTML, tan sencillo como cualquier editor de texto.

## **23.10** **Accesibilidad y Usabilidad**

### **23.10.1** *Accesibilidad como cualidad de los sistemas*

La accesibilidad es una cualidad de los sistemas informáticos vinculada al campo de la interacción entre humanos y ordenadores. Fundamentalmente se centra en la capacidad de acceso al uso de la aplicación o sistema informático que es objetivo por parte del usuario. En el campo concreto de las tecnologías web, accesibilidad hace referencia a la capacidad de acceso a la Web y a sus contenidos por todas las personas. La accesibilidad pretende facilitar el acceso a cualquier tipo de usuario independientemente de la discapacidad (física, intelectual o técnica) que presenten. También está relacionado con aquellas dificultades que se derivan del contexto de uso ya sean tecnológicas o ambientales. Esta cualidad está íntimamente relacionada con la usabilidad de los sistemas.

A la hora de diseñar contenidos, hay que tener en cuenta los factores de accesibilidad que permitirán que cualquier tipo de usuario pueda acceder en condiciones de igualdad a la información almacenada. Existen mecanismos y estándares actualmente que trabajan sobre ello, y las tecnologías proporcionadas por la mayoría de los SGC permiten estructurar nuestros contenidos teniendo en cuenta este tipo de facetas. Un caso concreto se da con los sitios que tienen un código XHTML semánticamente



correcto, permitiendo proporcionar un texto equivalente alternativo a las imágenes y a los enlaces. Esto supone que los usuarios ciegos puedan utilizar lectores de pantalla o líneas Braille para acceder a los contenidos. Lo mismo ocurre cuando los vídeos disponen de subtítulos; usuarios con dificultades auditivas podrán entenderlos perfectamente.

Los sistemas de gestión de contenido actuales permiten además cierta personalización de las características del sitio. Factores como el tamaño de letra o las proporciones de la interfaz comienzan a ser personalizables por cada tipo de usuario, proporcionando ayuda para que los usuarios con problemas visuales puedan leerlos sin dificultad.

### **23.10.2** *Limitaciones en la accesibilidad*

Existen fundamentalmente cuatro tipos de limitaciones en la accesibilidad de los sitios Web:

- Visuales: Abarcando un amplio abanico de patologías y de distintos grados de deficiencia visual, que pueden ir desde la baja visión a la ceguera total, además de problemas para distinguir colores.
- Motrices: Dificultad o la imposibilidad de usar las manos, incluidos temblores, lentitud muscular, debido a enfermedades como el Parkinson, distrofia muscular, parálisis cerebral o amputaciones.
- Auditivas: Sordera o deficiencias auditivas.
- Cognitivas: Dificultades de aprendizaje o discapacidades cognitivas que afecten a la memoria, la atención, las habilidades lógicas, etc.

### **23.10.3** *Promoviendo la accesibilidad*

La tarea de promover la accesibilidad en el entorno web corre a cargo del grupo de trabajo Web Accessibility Initiative (WAI), que depende directamente del World Wide Web Consortium. En 1999 el WAI publicó la versión 1.0 de sus pautas de accesibilidad Web (WCAG). Con el paso del tiempo se han convertido en un referente internacionalmente aceptado hasta que en diciembre del 2008 las WCAG 2.0 fueron aprobadas como recomendación oficial.

Estas pautas se dividen en tres bloques orientadas específicamente para cada uno de los principales perfiles que forman parte de un proyecto de desarrollo web:

- **Pautas de Accesibilidad al Contenido en la Web (WCAG):** Están dirigidas a los profesionales del diseño y desarrollo web y proporcionan información y recomendaciones acerca de cómo hacer que los contenidos del sitio Web sean accesibles.
- **Pautas de Accesibilidad para Herramientas de Autor (ATAG):** Están dirigidas a los desarrolladores del software que usan los webmásters, con el objetivo de proporcionar un mejor soporte para la construcción de sitios accesibles.
- **Pautas de Accesibilidad para Agentes de Usuario (UAAG):** Están dirigidas a los desarrolladores de Agentes de usuario (navegadores y similares), para que estos programas faciliten a todos los usuarios el acceso a los sitios Web.

#### **23.10.4**

#### **Usabilidad**

La usabilidad es otro atributo vinculado a los sistemas software, particularmente importante en el campo de la interacción hombre-computador. Actualmente la usabilidad está reconocida como un

importante atributo de calidad del software. Actualmente no llega con fabricar sistemas con alto rendimiento y fiabilidad, sino que el objetivo es crear sistemas que sean cómodos y manejables, adaptados para los usuarios finales. En el marco de la usabilidad se ha generado un importante centro de servicios en el que empresas especializadas desarrollan sus actividades fundamentalmente orientadas a la asesoría en estos campos.

En los proyectos de desarrollo de software en general, y en los orientados a la distribución y gestión de contenidos en particular, el concepto de usabilidad es de importancia capital. A la hora de distribuir contenido a través de la red, el portal está abierto a todo tipo de usuarios. Esta faceta comparte importancia con el concepto anterior de accesibilidad. Pero además, la usabilidad permite incrementar el atractivo, desarrollando sistemas sencillos e intuitivos que permiten un fácil manejo y rápido aprendizaje.

Desde un enfoque del diseño y evaluación de aplicaciones software, hablamos de usabilidad software como un conjunto de fundamentos teóricos y metodológicos que aseguran el cumplimiento de los niveles de usabilidad requeridos.

## **23.11**

### **W3C**

El World Wide Web Consortium, abreviado W3C, es el máximo organismo a nivel mundial que se encarga de gestionar y publicar las recomendaciones y estándares asociados al World Wide Web. Es decir, el objetivo de este consorcio es estandarizar los protocolos y las tecnologías utilizadas para construir la web, de manera que el contenido este disponible para la mayor parte posible de la población del mundo. Las

principales actividades a las que se dedica son, a la coordinación de los diferentes grupos de trabajo en el ámbito de la generación de:

- Especificaciones y estándares: sobre tecnologías asociadas al WWW.
- Directrices: y recomendaciones de desarrollo para buenas prácticas.
- Herramientas: que permitan validar la aceptación y cumplimiento de los estándares y recomendaciones propuestas.

Está dirigida por Tim Berners-Lee, responsable del grupo de investigación que desarrolló la URL, el protocolo HTTP (HyperText Transfer Protocol, Protocolo de Transferencia de HiperTexto), así como también del lenguaje de etiquetado HTML (Lenguaje de Marcado de HiperTexto) que son las principales tecnologías sobre las que se basa la Web.

Se creó en 1994 en el MIT, actual sede central del consorcio. El consorcio está formado por una gran diversidad de miembros y entidades cada una de las cuales colabora en los ámbitos en los que el W3C ejerce su función. Actualmente está integrado por tres tipos de figuras principales:

- Miembros adscritos del W3C: garantizan la fortaleza y el sentido del Consorcio a través de la inversión y la participación activa en las Actividades del W3C. El W3C cuenta con más de 400 organizaciones Miembro provenientes de más de 40 países, con intereses muy variados. Entre los Miembros del W3C se incluyen proveedores de productos de tecnología y servicios, proveedores de contenido, usuarios corporativos, laboratorios de investigación, organismos de estandarización y administraciones, que trabajan conjuntamente para alcanzar un acuerdo sobre la dirección que debe tomar la Web.
- Equipo W3C (W3C Team): El Equipo del W3C incluye a más de sesenta investigadores e ingenieros de todo el mundo que dirigen las actividades técnicas del W3C y gestionan las operaciones del Consorcio. La mayoría de los componentes del Equipo del W3C

trabajan en una de las tres instituciones que albergan al W3C: El MIT/CSAIL, en los Estados Unidos; el ERCIM, las oficinas centrales en Francia; y la Universidad de Keio, en Japón. Están coordinados por el Director Tim Berners-Lee, el Director de Operaciones Steve Bratt, y un Equipo de Dirección, los trabajadores del W3C:

- Se mantienen informados sobre las nuevas tecnologías, las fluctuaciones del mercado y las actividades de organizaciones relacionadas, con intención de orientar al W3C adecuadamente.
- Organizan las Actividades del W3C para, así, cumplir el mayor número de objetivos dentro de unos límites prácticos (tales como los recursos disponibles).
- Promueven la cooperación entre los Miembros, a la vez que buscan su diversidad, incentivan la innovación, y facilitan su activa participación.
- Divulgan los resultados del W3C a los Miembros y a la prensa, y promueven su aceptación en la comunidad Web; vea la lista de presentaciones públicas realizadas por el Equipo.
- Oficinas W3C (W3C Offices): El objetivo de las Oficinas del W3C es trabajar con las comunidades regionales para potenciar la adopción de las recomendaciones del W3C entre los desarrolladores, los creadores de aplicaciones, y los difusores de estándares, así como fomentar la inclusión de las organizaciones más importantes en la creación de futuras recomendaciones a través de su adscripción al Consorcio.

El concepto de web 2.0 nace para describir aquellas aplicaciones web que se centran en el usuario, en contraposición a la web tradicional (o 1.0) que simplemente actúa como una mera presentadora de datos. La web 2.0 fomenta la interacción con el usuario, la interoperabilidad, y busca compartir información y la colaboración. Este concepto se ha desarrollado principalmente mediante servicios web, redes sociales, wikis, blogs y sistemas de almacenamiento de vídeos, entre otros.

Se puede entender a la web 2.0 como la evolución de una serie de aplicaciones tradicionales en otras enfocadas al usuario, no tanto desde un punto de vista tecnológico, sino de concepto e intención. Así, a través del concepto 2.0, ciertas aplicaciones irán abandonando el escritorio e irán migrando a la web, siempre basándose en la colaboración con el usuario y la interactividad. Para permitir el uso del software online y otras aplicaciones multimedia, la extensión de la banda ancha ha sido un factor fundamental.

La web 2.0 es una actitud de compartir información, colaboración, interacción, cambio continuo y la creación de una plataforma global.

La web 2.0 es una reinterpretación de la web, que describe los pasos para llegar a un modelo de comunicación colectiva más participativa e innovadora. Uno de los principales cambios se producen en la gestión de los datos. En la web 1.0, es la empresa la que gestiona la información, a través de una red de expertos, o algún tipo de procesamiento artificial. Por su parte, la web 2.0 es un concepto colaborativo en el que se espera que los datos sean obtenidos por medio de los usuarios.

En 2004, O'Reilly Media realiza una conferencia sobre web 2.0, donde se comienza a sentar las bases de este concepto. Desde el primer momento se deja claro que no se trata de un cambio tecnológico en la web, sino un cambio en la forma en la que desarrolladores y usuarios utilizan la web. Sin embargo, Tim Berners-Lee, el creador de la World Wide Web, ha calificado el término 2.0 como “palabrería”, ya que, según él mismo, la web ya contenía esos valores desde un principio.

Con anterioridad al concepto 2.0, la mayoría de aplicaciones web eran portales estáticos, programados en lenguaje HTML (Hyper Text Markup Language), con una periodicidad de actualización baja, y con escasa interacción con el usuario, que simplemente “consumía” contenido. Una primera aproximación al concepto 2.0 fueron las aplicaciones webs dinámicas, en las que las páginas son servidas al usuario dinámicamente a partir de contenido obtenido de una base de datos.

Algunos expertos en la web 2.0 mantienen que la web debe enfocarse a la interacción y a las redes sociales, de modo que la web sea un punto de encuentro, y dependa del usuario.

Dale Dougherty, de O'Reilly Media, usa por primera vez el término “web 2.0” junto con Craig Cline, de MediaLive, para ilustrar la idea de que la web se estaba refundando. Una de las principales críticas del concepto 2.0 es que no existe una definición formal. Dougherty argumentó su punto de vista con ejemplos de aplicaciones web 1.0 y 2.0. Así, lo que DoubleClick era en la web 1.0, lo es Adsense en la 2.0. Y lo que era Ofoto en la web 1.0, lo es Flickr en la 2.0. Otros ejemplos son Google (que mide el impacto de los sitios web mediante el número de enlaces a páginas web, y no tanto por clicks absolutos) o la Wikipedia (un proyecto colaborativo a partir de una

gran cantidad de pequeños usuarios, en lugar de un reducido equipo de expertos).

De este modo, en octubre de 2004 se realiza la primera conferencia sobre web 2.0, con Dougherty, Cline y John Battelle. Un año después se realiza una segunda conferencia, donde se une Tim O'Reilly para resumir los principales rasgos de la web 2.0. Entre ellos, encontramos la innovación, el diseño para múltiples plataformas, y la importancia de la participación del usuario. En la web 2.0, la aplicación está orientada por el usuario, que es el que alimenta y modifica una base de conocimiento de la aplicación, a partir de un diseño interactivo y en red (tal y como definió Xavier Ribes en 2007).

Algunas de las características que definen la web 2.0 son:

- Permite el uso de aplicaciones online, sustituyendo potencialmente a las aplicaciones de escritorio, y desde un punto de vista multiplataforma. Las aplicaciones ejecutadas en web son independientes del navegador y del sistema operativo desde el que se ejecutan, lo que facilita el desarrollo.
- Permite la transferencia de información y contenidos entre aplicaciones web.
- Permite una experiencia de usuario simple, con una curva de aprendizaje rápida.
- Permite añadir nuevas funcionalidades de una manera simple e intuitiva.
- Permite gradualmente la virtualización de las estructuras sociales en el mundo online.



- Hace que el papel del usuario gane importancia, llevándolo hasta el rol de co-desarrollador, y fomentando un desarrollo colectivo. Se puede decir que un elevado número de usuarios pueden sustituir en algunos casos a un reducido grupo de expertos.
- Fomenta la interoperabilidad de las aplicaciones web, la incrustación de código externo y el uso de API's incluso por usuarios no expertos. En este punto es destacable la contribución de la tecnología RSS (Real Simple Syndication) que separa totalmente contenido y presentación de los datos. RSS permite conocer las actualizaciones de un portal web sin necesidad de visitarlo, y además hace posible crear sistemas de agrupación de información alimentado por distintas fuentes (los llamados agregadores, como Google Reader). RSS se basa en XML, que ha sido otro actor importante en la transmisión de información para la web 2.0.
- Fomenta la participación para la mejora continua de la aplicación.

## 23.13

### Wikis

Un wiki (del hawaiano “*wiki wiki*” = rápido) es un tipo de aplicación web que permite la edición de sus contenidos de forma concurrente, voluntaria y colaborativa por parte de usuarios, autenticados o no, a través de un navegador web, y con el objeto de acumular conocimiento de manera conjunta a modo de repositorio centralizado. Los wikis se basan en el esfuerzo democrático y compartido sobre una base de igualdad y facilidad: todo el mundo debe poder aportar nuevo contenido.

Cada página de contenido del wiki se corresponde con un nombre unívoco y simple que facilita su comprensibilidad, así como su enlazamiento desde

otras páginas de wiki y portales externos. Además, existe un lenguaje wiki que facilita la edición y creación de jerarquías, categorías, tesauros y taxonomías por medio de enlaces internos. Esto crea una estructura descentralizada que hace que la navegación por los wikis sea no lineal, ya que cada página contiene numerosos enlaces a otras páginas.

El origen de los wikis proviene de Ward Cunningham, quien desarrolló un servidor wiki como repositorio de patrones de diseño en Portland (Portland Pattern Repository) en 1995 (llamado WikiWikiWeb), y definió al wiki como 'la base de datos online más simple que podría funcionar'. Posteriormente, en 2001, Jimbo Wales y Larry Sanger usan un wiki como germen de su proyecto de enciclopedia Wikipedia, una enciclopedia libre y en red. Comienzan utilizando el software UseMod, aunque finalmente desarrollan un software propio, denominado Media Wiki, que se ha convertido en un estándar para otros wikis. Ha sido precisamente la Wikipedia, y otras enciclopedias colectivas, las que han colaborado en el auge de los wikis.

Cada artículo se co-escribe con el resto de la comunidad. Dentro de la colaboración múltiple, el wiki posibilita un historial de actualizaciones que actúa a modo de control de versiones temporal y por usuario. Al trabajar como un repositorio, los wikis permiten volver a versiones anteriores con facilidad. Normalmente no existe una supervisión, y la edición se basa en la negociación entre usuarios, pero la tendencia actual es que exista un reducido grupo de usuarios con un rol especial que revisa los contenidos y hace posible mantener la calidad en los contenidos y evitar incoherencias y sabotajes. En otros casos se requiere autenticación solamente para mantener el historial de cambios y firmar los contenidos.

Hoy en día, la versión inglesa de Wikipedia es el wiki más grande que existe. El resto de versiones en otros idiomas, y otras aplicaciones wiki más específicas, cuentan con menor número de usuarios debido a que sus comunidades son menos numerosas.

Lo que diferencia a un wiki de otras aplicaciones web de gestión de contenidos es su rapidez para crear y modificar páginas, así como la simplicidad y legibilidad de su interfaz. Esto ha fomentado el alto número de participantes en los proyectos wiki, destacablemente mayor que en otros proyectos colaborativos web. Si bien el usuario posee un alto grado de libertad para editar contenidos, existen plantillas para que las páginas wiki guarden coherencia entre sí, lo que hace aumentar la calidad del repositorio. Estas estructuras predefinidas facilitan que la edición de contenidos sea lo más simple posible. Además, los wikis tienen normalmente un diseño sencillo que no suele modificarse.

Los wikis tienen un componente de altruismo más notable que otras aplicaciones web. Si bien es colaborativo, y el usuario fomenta la pertenencia a una comunidad con intereses comunes, lo importante de la colaboración es el bien común, y no tanto el que un usuario destaque más o menos por sus contribuciones, ya que la autoría de los contenidos rara vez es exclusiva. No solo eso, los contenidos están en continua edición, de manera indefinida, lo que obliga a la reflexión y aun continua revisión de las ideas.

Es destacable la gran relación entre los wikis y el mundo educativo. Los wikis han cambiado radicalmente el modelo de consulta de conocimientos, pasando de las mastodónticas enciclopedias estáticas a los proyectos

colaborativos actuales basados en wikis donde el usuario juega un papel fundamental.

Los software más utilizados para el desarrollo de wikis son: MediaWiki, TikiWiki o CitiWiki (en PHP), y JSPWiki o XWiki (en Java), entre otros. La aplicación solamente define la presentación básica de los datos (estilos, entorno...), pero la edición de los datos corre por parte de los usuarios.

## **23.14**

### **Blogs**

Un blog (o weblog) es un sitio web de actualización frecuente que recopila artículos, presentando primero el más reciente, y que permite la interacción con los lectores por medio de comentarios sobre los artículos. El blog es cronológico (permite mantener una línea de tiempo de publicación), es colaborativo (pueden publicar varios autores), y es interactivo (los lectores pueden publicar sus comentarios, de modo que el autor pueda contestarles para conformar un diálogo). El administrador del blog puede tomar distintas opciones de diseño, como no permitir comentarios, y administrar los artículos (borrarlos o reordenarlos).

La temática del blog es variada, siendo su motivación primigenia el actuar a modo de diario personal o bitácora. Desde entonces, los hay corporativos, periodísticos, educativos, etc. El blog que se dedica esencialmente a la publicación de fotografías se denomina fotolog o fotoblog (y videoblog en el caso de vídeos). Una práctica habitual es proporcionar un gran número de enlaces que amplíen la información para cada entrada. En ocasiones las

entradas permiten que se les haga *trackback* (un enlace inverso) para conocer quién ha enlazado la entrada desde otro sitio web.

Las entradas del blog suelen agruparse en categorías, y es frecuente la práctica de indicar palabras clave o etiquetas para facilitar la búsqueda por contenido. Los blogs también proporcionan archivos e índices mensuales y anuales que permiten una navegación ordenada por fecha. Así mismo, es habitual que las entradas proporcionen facilidades para ser compartidas en otros blogs, o por correo electrónico, así como por sindicación de contenidos, mediante el uso de tecnología RSS o Atom.

Gran parte del auge de los blogs se ha debido a su facilidad de mantenimiento y a las numerosas alternativas gratuitas disponibles. No son necesarios grandes conocimientos técnicos para administrar un blog, y ni siquiera para crearlo, lo que los ha acercado al gran público. Además, como cualquier otro sitio web, un blog puede tener publicidad y generar ingresos.

Existen principalmente dos tipos de soluciones blog: las que proporcionan una solución completa de software y alojamiento web gratuito (como Blogger o LiveJournal), y las que simplemente proporcionan software que precisa ser instalado en un sitio web (como WordPress). Este último es un tipo específico de gestor de contenido (CMS).

## **23.15**

### **Comunidades Virtuales**

Una comunidad virtual es aquella cuyos vínculos, interacciones y relaciones no tienen lugar en un espacio físico, sino en un espacio virtual como Internet.

Las comunidades virtuales surgen con Internet y obtienen su modelo de las comunidades no informáticas, existentes desde mucho antes. La primera comunidad virtual data de los años 70, aunque su mayor desarrollo se produce en los 90, volviéndose, en este momento, accesibles al público en general, gracias al nacimiento de la World Wide Web (WWW) y la expansión de herramientas como los chats, correo electrónico o mensajería instantánea. Hasta este momento, las comunidades se hallaban restringidas al ámbito científico y a expertos en informática.

Los usuarios sin acceso a Internet, implementaron y popularizaron el uso del Sistema de Tablón de Anuncios (BBS o Bulletin Board System), que se trataba de un sistema que funcionaba mediante un acceso mediante módem telefónica a una central (el BBS), el cual podía basarse en una o varias líneas de teléfono. En los BBS se podían mantener conversaciones, intercambiar archivos, publicar comentarios, etc. En esta época las comunidades eran independientes, lo más habitual era que los usuarios particulares empleasen sus propios equipos domésticos para proporcionar servicio con hasta un único modem de entrada.

Actualmente, las comunidades virtuales han evolucionado, convirtiéndose en una herramienta muy útil desde el punto de vista empresarial. Esto se debe la mejora que ofrecen a las organizaciones en su dinámica de trabajo interno, en las relaciones con los clientes o en el incremento de la eficiencia de sus procedimientos.

Desde el punto de vista social, las comunidades virtuales permiten a los usuarios relacionarse con los demás, adquiriendo así un carácter socializador.

Se estima que en el año 2000 más de 40 millones de personas participaban en comunidades virtuales, las cuales podemos caracterizar de la siguiente forma:

- Asociación virtual de personas
- Existe un propósito determinado que es la razón de ser de la comunidad virtual
- Existe un gran deseo de interacción entre los usuarios para satisfacer unas necesidades o desempeñar unos roles concretos
- Existen sistemas que evalúan y miden las interacciones y favorecen la cohesión entre los miembros

El principal inconveniente al que se enfrenta el desarrollo de las comunidades virtuales es la problemática de la organización interna de las mismas, que suele ser muy difícil de establecer y gestionar. En muchos casos, es demasiado costoso crear la estructura de la comunidad con lo que se puede llegar a perder el verdadero propósito de la creación del mismo.

Una comunidad virtual queda definida desde 3 puntos de vista:

- Comunidad virtual como lugar: emplazamiento donde los individuos pueden mantener relaciones económicas o sociales.

- Comunidad virtual como símbolo: los miembros de una comunidad desarrollan un sentimiento de pertenencia a una estructura mayor.
- Comunidad virtual como virtual: a pesar de las similitudes entre las comunidades físicas y las virtuales, una comunidad virtual se desarrolla principalmente en un entorno virtual que no puede ser asimilable con una localización física.

## **23.16**

### **Redes Sociales**

El concepto de red social, está estrechamente ligado con el de comunidad virtual, entendiéndose una comunidad virtual como un caso más específico de red social.

Una red social es una estructura de nodos donde distintos actores, que pueden ser individuos u organizaciones, están conectados mediante un serie de relaciones basadas en propiedades comunes. Una red social se asienta sobre cierto tipo de relaciones, económicas, laborales, familiares, políticos, deportivos, etc.

Una red social es distribuida cuando su emplazamiento no está limitado a un sitio en concreto sino que se distribuye a nivel geográfico. En el caso de una red social, es lógico pensar que no todos los actores participantes en dicha red se encuentren localizados en un único espacio.

Una red social se apoya en el uso de alguna tecnología de comunicación que permite a los distintos usuarios interactuar entre si. En este caso, lo más habitual es emplear Internet como tecnología subyacente, sin embargo, también existen redes sociales basadas en tecnologías móviles e



incluso en tecnologías no digitales como el teléfono, el fax o el correo postal.

Actualmente las redes sociales se orientan alrededor de un sitio web que ofrece a los usuarios una serie de servicios, como son chat, mensajería instantánea, carga de imágenes, vídeos, grupos de debate, etc. Uno de los servicios que tienen una mayor importancia son los del “software social”, que abarca todas aquellas aplicaciones que simulan procesos sociales del mundo real. El ejemplo más habitual es la simulación del efecto “amigo de un amigo”, en este caso la aplicación localiza a los amigos de nuestros amigos para poder así facilitar el contacto con nuevos usuarios.

Dentro de las redes sociales más empleadas hoy en día, podemos nombrar a Facebook, Youtube, Twitter, Myspace, Orkut, Hi5 etc.

## **23.17** **Sindicación de contenidos**

Antes de definir qué es la sindicación de contenidos, es necesario aclarar que si bien la expresión correcta es “redifusión web”, el término “sindicación web” está muy expandido en su uso, especialmente en lo que se refiere a contenidos web, aunque esta redifusión puede llevarse mediante cualquier medio de comunicación.

La sindicación de contenidos (sindicación web o redifusión web) consiste en el reenvío o redistribución de contenidos desde un sitio web de origen, hasta otro sitio web receptor, el cual a su vez se puede ver como un emisor

de los contenidos, puesto que estos dejan de estar limitados a los usuarios del sitio web inicial. Esta redifusión de contenidos se hace habitualmente mediante una licencia o contrato entre los sitios web de origen y destino.

Los contenidos que se redistribuyen suelen codificarse en XML, aunque esto no es obligatorio y puede emplearse cualquier otro formato soportado por http.

Existen dos familias más destacadas en cuanto a formatos de redifusión web, que son RSS y Atom. De hecho, actualmente el término RSS (Really Simple Syndication) se ha empezado a usar indistintamente para referirse a cualquiera de los 2 formatos de fuentes web, el propio RSS o Atom.

Para poder leer una fuente web es necesario realizar una suscripción mediante un agregador, el cual muestra los nuevos contenidos que hayan sido publicados por el proveedor de la fuente web suscrita.

### **23.17.1**

#### ***Fuente Web***

Un canal web o fuente web (Web feed) es un medio de redistribución de contenidos web, que se emplea para suministrar información a los suscriptores de manera actualizada. Los cuales, deben contar con un programa “agregador” para acceder a todas las fuentes a las que están suscritos desde un mismo lugar.

Como ya comentamos anteriormente, los dos principales formatos de fuente web son RSS y Atom., ambos escritos en XML.

### **23.17.2**

#### ***Agregador de noticias***

Un lector RSS o agregador de noticias (eventualmente sólo agregador) es una aplicación que permite establecer una suscripción a fuentes de noticias en formatos Atom, RSS y otros derivados de XML/RDF. La función del agregador consiste en reunir todas las noticias y contenidos publicados en los sitios con redifusión escogidos y mostrarlas de manera unificada al usuario. De tal forma que el usuario pueda saber qué webs han incorporado o modificado contenidos desde la última lectura, y en cada caso, cual es el contenido de las mismas.

Los lectores RSS se han vuelto más populares con la implantación de XML y la web semántica, y hoy en día existe un gran número de blogs y sitios web que ofrecen sus actualizaciones, las cuales son administradas y agregadas en un solo lugar, gracias a herramientas como las de Google Reader, Netvibes, etc.

### **23.17.3**

#### ***Formato RSS***

RSS es un formato XML para syndicar contenidos web que se emplea para difundir información a los usuarios suscritos a una fuente de contenido. Este formato se caracteriza por permitir la distribución de contenidos sin necesidad de emplear un navegador, ya que se utiliza un agregador de contenidos RSS, aun así, es posible emplear el navegador para ver los contenidos RSS.

De hecho, las últimas versiones de los navegadores permiten visualizar los RSS sin necesidad de un agregador.

Este formato, se desarrollo específicamente para aquellos sitios que se actualizan de forma habitual y mediante al cual se puede compartir la información y ser empleada en otros sitios web.

#### **23.17.4**

#### ***Estándar Atom***

Atom hace referencia a 2 estándares relacionados entre si:

- Protocolo de Publicación Atom (AtomPub o APP): protocolo basado en http para crear o actualizar recursos en web.
- Formato de Redifusión Atom: fichero en formato XML usado para redifusión web.

Para crear un contenido, que pueda ser tratado con agregador o por otro sitio web que redifunde los contenidos de la fuente, el propietario del sitio web puede emplear un software específico como un sistema de gestión de contenidos, el cual publica una fuente web de artículos recientes en un formato estándar y legible por los ordenadores.

El formato Atom fue desarrollado como una alternativa a RSS. Atom surge por la incompatibilidad existente entre algunas versiones del protocolo RSS. El formato de redifusión Atom se publicó como un “estándar propuesto” de la IETF con RFC 4287, mientras que el protocolo de comunicación se publicó como RFC 5023.

## 23.18

### Podcast

Podcasting consiste en distribuir archivos multimedia, generalmente audio o vídeo, mediante un sistema de redifusión que permita a los usuarios establecer suscripciones, los cuales emplean un programa de descarga para poder visualizar el contenido en el momento que se desee. También existe la posibilidad de descargar los contenidos sin una suscripción previa. El podcasting es un tipo de sindicación donde los archivos que se redistribuyen son de contenido multimedia.

Al principio el podcasting se refería exclusivamente a las retransmisiones archivos de audio, aunque más tarde se ha extendido el concepto para hacer referencia tanto a audio como a vídeo de manera indistinta.

El contenido de los podcasts es muy diverso, sobre tecnología, política, noticias, contenidos educativos, etc. En función del productor del podcast su complejidad, número de participantes y estructura varían significativamente. Algunos se asemejan a programas de radio, con varios participantes y diversas opiniones, y otros se parecen más a comunicados o monólogos de una sola persona, cuya duración generalmente es más corta.

Los podcasts suelen ser accesibles desde el sitio web en que han sido colocados. Existen blogs que permiten realizar podcasting mediante el uso de plug-ins gratuitos. Además estos archivos también se pueden descargar.

### **23.18.1**

### ***Podcasting Vs. Streaming***

Antes de la aparición del podcasting, la forma habitual de transmitir contenidos multimedia era el streaming o webcasting. Mediante este sistema, proveedores de contenidos como cadenas de televisión o radios empleaban el streaming para emitir desde un servidor central.

Podcasting y streaming presentan ciertas diferencias, las más destacadas:

- Con streaming no se produce la descarga del fichero sino que este se reproduce en modo flujo mientras se está descargando. Cuando termina la reproducción, el fichero no se almacena en el equipo receptor. Esto presenta la ventaja del ahorro de espacio de almacenamiento, sin embargo, siempre que se quiera ver u oír nuevamente el archivo volverá a ser descargado y no puede ser reproducido si no existe una conexión a Internet.
- Con streaming, es necesario acceder al sitio Web donde está el canal deseado e indicar que la reproducción del contenido debe iniciarse mediante algún tipo de enlace, botón, etc. Sin embargo, con podcasting, cuando el contenido está disponible, este se descarga de manera automática y puede ser escuchado en cualquier momento.
- El streaming presenta más problemas de compatibilidad entre los distintos sistemas empleados que el podcasting.

- El streaming es más sensible a problemas de conexión a Internet o de sobrecarga del servidor ya que la descarga se produce mientras se está reproduciendo el archivo.

## **23.19**

### **Suites Ofimáticas en Web**

Un suite ofimática es un conjunto de herramientas que se utilizan habitualmente en entornos de oficina para el trabajo con documentos de cualquier tipo. Normalmente se incluyen en estos paquetes un editor de textos, un gestor de hojas de cálculo, un gestor de bases de datos, un programa de creación de diapositivas de presentación etc. Tradicionalmente todas las suites ofimáticas eran herramientas de escritorio. Actualmente existen distintas versiones de suites ofimáticas pero en web, a las cuales se accede mediante el uso de un navegador.

#### **23.19.1**

#### ***Feng Office***

Feng Office es una aplicación libre de tipo Web Office, antes conocida como OpenGoo. Se trata de una sistema completo que proporciona funcionalidades para crear, publicar, colaborar y compartir documentos.

Feng Office permite crear y trabajar entre otros, sobre:

- *Documentos*: permite alojar documentos de todo tipo y editar directamente algunos de ellos.

- *Listas de tareas*: permite la creación de listas de tareas asignadas a distintos usuarios, con opciones de notificación, categorización, etc
- *Correo electrónico*: permite centralizar la gestión de las distintas cuentas de correo.
- *Calendario*: permite establecer reuniones y una gestión de las actividades diarias.
- *Agenda*: permite realizar una gestión de contactos.

Esta aplicación puede funcionar bajo un modelo SaaS (Software as a Service), donde los servidores del proveedor a través de un navegador, nos permiten trabajar con la aplicación, pero también es posible realizar una instalación de la aplicación en un servidor propio, en este caso los requisitos de este sistema pasan por un servidor web Apache, PHP y MySQL como base de datos.

### **23.19.2**

### **Google Docs**

Google Docs & Spreadsheets, es un programa web gratuito que permite crear y trabajar sobre unos documentos de manera individual o en grupo.

Google Docs se compone de:

- Procesador de textos
- Hojas de cálculo
- Programa de presentación sencillo
- Editor de formularios



Entre las ventajas de Google Docs, se encuentra el hecho de que puede ser usado tanto online como offline. En esta modalidad offline, los cambios que se introduzcan en los documentos serán actualizados de forma automática en cuanto la conexión con internet se restablezca.

Además recientemente se ha incorporado compatibilidad entre Google Docs y los dispositivos móviles, de tal forma que se pueda no solo acceder a los documentos sino también editarlos.

### **23.19.3**

### ***Office Web Apps***

Office Web Apps es la solución de Microsoft para las suites ofimática en la web. Se trata de una versión gratuita basada en el conjunto de aplicaciones de Microsoft Office.

Office Web Apps, se compone de :

- Word Web App
- Excel Web App
- PowerPoint Web App
- OneNote Web App

Estas aplicaciones permite acceder a los documentos a través del navegador, así como compartir archivos y trabajar sobre ellos de forma colaborativa.

## **23.20**

### **Almacenamiento en web**

Un servicio de almacenamiento de archivos online (servicio de alojamiento de archivos o centro de medios online) es un servicio de alojamiento online cuyo propósito es facilitar el almacenamiento de contenido estático, como archivos, documentos, etc. Por norma general este tipo de servicios provee de accesos a través de diversas interfaces, web, ftp, etc.

### **23.20.1**

#### ***Dropbox***

Dropbox es un servicio de alojamiento de archivos en la nube, que permite almacenar y sincronizar archivos entre distintos ordenadores e incluso con distintos usuarios. Como característica principal cabe destacar que se trata de un sistema multiplataforma y que presenta versiones tanto gratuitas como de pago.

El funcionamiento es muy sencillo, cada ordenador cliente instala un software que permite a los usuarios desplazar cualquier contenido a una carpeta designada, la cual se integra en el sistema de archivos del sistema que se trate. Una vez se sitúa un archivo en esa carpeta, o se modifica, éste es sincronizado en la nube y con todos los demás ordenadores donde esté instalado el cliente Dropbox de ese usuario. El acceso a los archivos de la carpeta de Dropbox también puede realizarse a través de la Web e incluso ser compartido por varios usuarios. Aunque Dropbox funciona como un servicio de almacenamiento, su propósito se centra más en la sincronización y compartición de archivos.

#### **21.20.1.1 Funcionalidades de Dropbox**

- Historial de revisiones: los archivos borrados de la carpeta Dropbox pueden ser recuperados desde la web o desde cualquiera de los ordenadores sincronizados.
- Historial del documento: se puede acceder al historial de un documento, de tal forma que se puede trabajar sobre el mismo, sin que esto afecte a las versiones preexistentes.
- Optimización de la conexión: al modificar un archivo en una carpeta Dropbox, el sistema sólo cargará las partes del documento que han sido modificadas cuando se produzca la sincronización.

## **23.21**

### **Escritorios Virtuales**

Un escritorio virtual consiste en un servicio de virtualización aplicado sobre un escritorio tradicional. En este caso el escritorio del usuario se ejecuta en un servidor, donde las órdenes de dicho usuario se transmiten online al servidor, el cual envía de vuelta los resultados de dichas acciones.

La virtualización de escritorio es relativamente reciente y describe la separación del entorno que percibe el usuario, que engloba sus datos y programas, de la máquina física en la que éstos se almacenan y ejecutan. En este caso, el usuario puede tener un sistema completo y emplear adicionalmente el escritorio virtual para cierto tipo de tareas, o incluso, el usuario puede contar con un sistema sencillo tipo terminal, donde toda las tareas del usuario se realizan directamente contra el servidor.

### **23.21.1**

#### **eyeOS**

eyeOS es un sistema libre y multiplataforma que se basa en el estilo que tienen los escritorios en los sistemas operativos tradicionales, e incluye la estructura de un sistema operativo así como ciertas aplicaciones ofimáticas como procesador de textos, calendario, navegador, gestor de archivos, etc.

Este sistema se diferencia de otros, en que no necesita de ningún software adicional para poder usarlo, puesto que todo el acceso se realiza mediante un navegador web. Recientemente, el sistema ha sido adaptado para poder utilizarse en dispositivos móviles.

## **23.22**

### **Mashups**

Un mashup es una aplicación o página web que usa y combina funcionalidades y datos de una o más fuentes para crear nuevos servicios. Implica una integración rápida y sencilla, generalmente con APIs abiertos y fuentes de datos, para producir resultados enriquecidos. Se toman una serie de datos existentes y se transforman en otros con un valor añadido que son más útiles tanto a nivel personal como profesional. Como principales características de los mashup se puede destacar la visualización, la combinación y la agregación.

Los mashup se componen de 3 partes:

- Proveedor de contenidos o fuente de datos. Los datos están accesibles a través de un API y mediante distintos protocolos como RSS.

- Sitio mashup: aplicación web que ofrece un servicio a partir de distintas informaciones que no son tuyas.
- Navegador web: es la interfaz con la que el usuario interactúa con el mashup.

Un error muy frecuente es confundir los contenidos embebidos con los que existen en un mashup. Un sitio que permite embeber por ejemplo un video o un archivo de sonido, no es un mashup, ya que no ha existido ningún tipo de procesamiento en estos datos que permita incrementar el valor que estos tienen para el usuario.

Existen distintos tipos de mashups:

- *De consumidores*: es el más conocido. Se integran datos de diversas fuentes y se accede a través de una interfaz sencilla. Ejemplo: Google Maps.
- *De datos*: se mezclan datos de tipo similar de distintas fuentes. Ejemplo: combinación de múltiples feeds RSS en uno solo.
- *Empresariales*: integra datos de fuentes tanto externas como internas. Ejemplo: incorporar mayor información a un informe estratégico mediante datos existentes en algún registro oficial.
- *De negocio*: es una combinación de los 3 anteriores.

## 23.23

### P2P

Una red P2P (Peer-to-peer, red de pares o red punto a punto) es una red de ordenadores en la que todos o algunos de los aspectos funcionan sin que existan clientes ni servidores fijos, sino una serie de nodos que actúan

como iguales entre sí, donde cada uno de ellos es a la vez cliente y servidor.

Este tipo de redes permite el intercambio directo de información entre los ordenadores que están conectados. Habitualmente para compartir ficheros de cualquier tipo, aunque también se emplea para telefonía VoIP.

### **23.23.1**

### ***Características***

A continuación se detallan algunas características de las redes P2P:

- *Anonimato*: es importante que el autor de un contenido, su lector, editor y el servidor que lo almacena sean anónimos
- *Descentralización*: por definición los nodos P2P son iguales y la red descentralizada. Ningún nodo es imprescindible para el funcionamiento de la red.
- *Robustez*: al tratarse de redes distribuidas, la robustez también se ve incrementada ya que en caso de producirse un fallo, al existir una réplica de los datos en múltiples destinos, la información deseada siempre se puede encontrar, al no depender de un servidor central.
- *Seguridad*: consiste en identificar y evitar nodos maliciosos, así como el contenido potencialmente peligroso, etc. Los mecanismos de seguridad más destacados en este caso son: cajas de arena, reputación, comunicaciones seguras, comentarios sobre los ficheros, cifrado multiclave, etc.
- *Escalabilidad*: cuanto mayor número de nodos estén conectados a una red P2P mejor será el funcionamiento. Cuando se incorporan

nuevos nodos, con sus recursos, los recursos totales del sistema aumentan.

## **23.23.2**

### ***Tipos de redes P2P***

#### **21.23.2.1**

#### **Redes P2P centralizadas**

Este tipo de red se caracteriza por:

- Arquitectura monolítica donde todas las transacciones se hacen a través de un único servidor, el cual almacena y distribuye los nodos donde se almacenan los contenidos.
- Todas las peticiones dependen de la existencia del servidor.
- Administración dinámica.
- Privacidad de los usuarios limitada.
- Falta de escalabilidad.

#### **21.23.2.2 Redes P2P híbridas, semicentralizadas o mixtas**

Este tipo de redes se caracteriza por:

- Existe un servidor que atiende peticiones pero no almacena información.
- El servidor administra los recursos, enrutamientos y comunicación entre nodos.
- Los nodos son los encargados de almacenar la información.
- El servidor central reconoce la información que desea compartir cada nodo.

- Puede existir más de un servidor que gestione los recursos compartidos.
- Los nodos pueden seguir en contacto directo entre ellos en caso de que el servidor o servidores caigan.

### **21.23.2.3 Redes P2P puras o totalmente descentralizadas**

Este tipo de redes se caracteriza por:

- Son las más comunes y versátiles puesto que no necesitan de ningún tipo de gestión central.
- Se reduce la necesidad de usar un servidor central.
- Cada nodo es a la vez cliente y servidor.
- Las conexiones se establecen entre usuarios, con la ayuda de un tercer nodo que permite enlazar dicha conexión.
- No existe un enrutador central.

## **23.24**

### **Web semántica**

La web ha influido mucho en el modo de comunicación de los últimos tiempos y si bien tiene multitud de ventajas, como el acceso a millones de recursos independientemente de nuestra localización, también existen dificultades como son la sobrecarga de información y la heterogeneidad de las fuentes de información, lo que nos lleva a un problema de interoperabilidad.

Con la Web semántica estos problemas se solucionan, permitiendo que los usuarios deleguen ciertas tareas en el software. Gracias a la incorporación



de mayor “semántica” a la web, el software es capaz de procesar el contenido, combinarlo, realizar deducciones, etc.

### **23.24.1** ***Definición de web semántica***

La web semántica (semantic web) se basa en la idea de incorporar metadatos ontológicos y semánticos a la web. Esta información adicional, describe el significado, contenido y relación entre los datos. Además debe ser proporcionada de manera formal para que pueda ser evaluada automáticamente por equipos de procesamiento. Al enriquecer la web con más significado, se pueden obtener soluciones a problemas comunes en la búsqueda de información.

La Web se basa fundamentalmente en documentos HTML, lo cual no es demasiado versátil a la hora de categorizar los elementos que configuran el texto. La función de la web semántica es resolver estas deficiencias de tal forma que se puedan describir los contenidos de una web, mediante tecnologías como RDF, OWL, además de XML. Este tipo de tecnologías aporta descripciones explícitas de los distintos recursos incorporando una serie de etiquetas interpretables por los gestores de contenidos, de tal forma que sea posible la interpretación de los documentos, tratamiento de su información, etc.

### **23.24.2** ***RDF, SPARQL y OWL***

La Web semántica, para realizar una adecuada definición de los datos, emplea fundamentalmente RDF, SPARQL y OWL.

- *RDF*: proporciona información sobre los recursos de la web, de forma simple y descriptiva.
- *SPARQL*: es el lenguaje de consulta de RDF. Permite realizar búsquedas sobre los recursos de la web semántica.
- *OWL*: es un mecanismo que permite desarrollar vocabularios específicos que se puedan asignar a los recursos. Proporciona un lenguaje para definir ontologías que se pueden usar a través de distintos sistemas.

Las ontologías se encargan de definir los conceptos empleados para describir y representar un área de conocimiento, incluyen las definiciones de conceptos básicos y la relación entre los mismos.

## 23.25

### Bibliografía

- Jerri L. SEO: Optimización de Posicionamiento en Buscadores. Ledford Anaya Multimedia
- V. Canseco G. Gerónimo. Breve introducción a los sistemas colaborativos: Groupware& workflow. 1998.
- Ortega M. Velázquez Iturbide J.A. Paredes M., Fernández I. Escritura colaborativa y pdas: una propuesta de aprendizaje basada en resolución de problemas. 2003.
- Network Working Group. «[RFC 5321 - Simple Mail Transfer Protocol](#)»
- Mark Harrison (July 1995). *The USENET Handbook (Nutshell Handbook)*. O'Reilly. [ISBN 1-56592-101-1](#).
- Kate Gregory, Jim Mann, Tim Parker, and Noel Estabrook (June 1995). *Using Usenet Newsgroups*. Que. [ISBN 0-7897-0134-0](#).
- Bryan Pfaffenberger (1994-12-31). *The USENET Book: Finding, Using, and Surviving Newsgroups on the Internet*. Addison Wesley. [ISBN 0-201-40978-X](#).
- Kate Gregory, Jim Mann, Tim Parker, and Noel Estabrook (June 1995). *Using Usenet Newsgroups*. Que. [ISBN 0-7897-0134-0](#).
- Mark Harrison (July 1995). *The USENET Handbook (Nutshell Handbook)*. O'Reilly
- Videoconferencing and Videotelephony. Richard Schphorst. Editorial Artech House. Norwood, 1996.
- <http://www.wikipedia.es>
- <http://www.maestrosdelweb.com>

- <http://www.wikispaces.com>
- <http://www.w3c.com>

**Autor:** Francisco Javier Rodríguez Martínez

Subdirector de Sistemas Escola Superior Enxeñaría Informática Ourense

Colegiado del CPEIG

# **24. HERRAMIENTAS OFIMÁTICAS. PROCESADORES DE TEXTO, HOJAS DE CÁLCULO, BASES DE DATOS, PRESENTACIONES.**

## **Tema 24: Herramientas ofimáticas. Procesadores de texto, hojas de cálculo, bases de datos, presentación.**

---

### **ÍNDICE**

<b>24.1 Herramientas ofimáticas.....</b>	<b>2</b>
24.1.1 Definición de Suite Ofimática.....	2
24.1.2 OpenOffice.....	3
24.1.3 Microsoft Office.....	4
24.1.4 Suites ofimáticas en Web.....	5
<b>24.2 Procesadores de Texto .....</b>	<b>8</b>
24.2.1 Historia de los procesadores de textos.....	8
24.2.2 Procesadores o editores de textos.....	8
24.2.3 Ejemplos de procesadores de textos.....	9
<b>24.3 Hojas de cálculo.....</b>	<b>10</b>
24.3.1 Microsoft Excel.....	10
24.3.2 Openoffice.org Calc.....	10
<b>24.4 Bases de datos.....</b>	<b>11</b>
24.4.1 Características.....	11
24.4.2 Sistema de Gestión de Base de Datos (SGBD).....	11
24.4.3 Ventajas de las bases de datos.....	11
24.4.4 Desventajas de las bases de datos.....	13
24.4.5 Herramientas para la gestión de Bases de Datos presentes en suites ofimáticas.....	13
<b>24.5 Presentación.....</b>	<b>14</b>
24.5.1 Tipos de presentación.....	14

24.5.2Herramientas de presentación.....	14
<b>24.6BIBLIOGRAFIA.....</b>	<b>14</b>

## **24.1 HERRAMIENTAS OFIMÁTICAS**

El termino ofimática se puede definir como la automatización, mediante sistemas electrónicos, de las comunicaciones y procesos administrativos en las oficinas. Las actividades básicas de un sistema ofimático comprenden el almacenamiento de datos en bruto, la transferencia electrónica de los mismos y la gestión de toda la información(en formato digital) relativa al negocio. La ofimática ayuda a optimizar o automatizar los procedimientos existentes en la administración de una empresa.

Dentro de la clasificación tradicional del software, el software ofimático se encuadra dentro de la categoría software de aplicación.

Se pueden clasificar los sistemas ofimáticos según distintos factores o características:

- Por tipo de licencia.

Todo software tiene una licencia de uso, que conforma un contrato entre el propietario del software y el usuario que define qué es lo que se puede hacer con el mismo. Una primera clasificación que tiene en cuenta este hecho permite clasificar el software de ofimática en dos grandes bloques: software libre y software privativo.

El Software libre (en ingles FLOSS) es la denominación del software que provee al usuario de distintos niveles de libertad sobre el producto software adquirido y por lo tanto, una vez obtenido, puede ser usado, copiado, estudiado, modificado y redistribuido libremente.

El software no libre (también llamado software privativo, software propietario, software privado, software con propietario o software de propiedad) se refiere a cualquier programa en el que los usuarios están limitados en cuanto a las posibilidades de usarlo, modificarlo o redistribuirlo (con o sin modificaciones). Lo habitual en este tipo de software es que el código fuente no está disponible o el acceso a éste código fuente se encuentra restringido de alguna forma.



- Por función

Una segunda clasificación tiene en cuenta el objetivo del software, esto es, la función que cubre una aplicación dentro del esquema de funcionamiento administrativo de la organización. Los tipos o clases de herramientas ofimáticas según la función a la que están dirigidos son:

- o Sistemas gestores de bases de datos. □
- o Sistemas de trabajo en grupo. □
- o Sistemas de tratamiento, almacenamiento y archivo de textos y documentos. □
- o Hojas de calculo. □
- o Edición y diseño de gráficos.
- o Agendas y organizadores personales.
- o Correo electrónico y navegación por Internet.

### **24.1.1**

### ***Definición de Suite Ofimática***

Los fabricantes de software ofimático habitualmente organizan sus aplicaciones en suites o paquetes. Los diferentes sistemas comerciales o no comerciales proponen familias de aplicaciones bajo una misma estructura. El conjunto de estos programas que proporcionan una interfaz idéntica y tienen la posibilidad de intercambiar ficheros se denomina suite o paquete informático. Con estas aplicaciones se pretende satisfacer las necesidades de los usuarios en el terreno de la ofimática.

Un suite ofimática es un conjunto de herramientas que se utilizan habitualmente en entornos de oficina para el trabajo con documentos de cualquier tipo. Normalmente, tal como se veía en el apartado de clasificación por funciones, se incluyen en estos paquetes un editor de

textos, un gestor de hojas de cálculo, un gestor de bases de datos, un programa de creación de diapositivas de presentación etc. Tradicionalmente todas las suites ofimáticas eran herramientas de escritorio que se ejecutaban en el ordenador del usuario. Actualmente existen distintas versiones de suites ofimáticas en web, a las cuales se accede mediante el uso de un navegador, estando todos los archivos y datos del usuario guardados en entornos de cloud.

En la actualidad las suites ofimáticas dominantes en el mercado son, por parte del software privativo, Microsoft Office, la cual posee sus propios formatos cerrados de documentos (no existe una publicación del estándar que utilizan para su grabación) para cada uno de sus programas. Respecto al software libre, está OpenOffice.org, desarrollado por Sun Microsystems, también con un formato para cada programa, pero de código abierto.

Características de una suite:

- Las diferentes aplicaciones de un paquete informático tienen una interfaz de usuario con el mismo aspecto; de esta manera el usuario entiende la suite como un todo y sólo necesita conocer y aprender una interfaz.
- Las suites se crean de manera que sus aplicaciones puedan intercambiar ficheros entre ellas sin ninguna dificultad y sin pérdida de información.
- Se pueden generar archivos en una aplicación que serán insertados dentro de otra, y pueden ser modificados con la aplicación que los creó mediante un acceso directo.
- Algunas herramientas y utilidades son compartidas por todas las aplicaciones. El diccionario de revisión ortográfica es una de estas herramientas compartidas. Si agregamos una palabra al diccionario desde el procesador de textos, esta palabra estará disponible desde cualquier otra aplicación.

### 24.1.2

### *OpenOffice.*

Fue el primer paquete informático en ofrecer al mercado de usuarios software abierto con varias aplicaciones y herramientas, sin ningún coste económico. Ha sido traducido a más de 35 idiomas diferentes y está disponible para distintas plataformas, tales como Windows, Mac OS, GNU/Linux y Solaris.

La suite de OpenOffice.org está conformada por cinco módulos:

- Writer, el procesador de textos
- Calc, la hoja de calculo
- Impress, paquete de presentaciones
- Draw, editor de dibujos vectoriales
- Math, editor de formulas matemáticas

En la página web de OpenOffice.org en español podemos ver la diferencia principal con un software privativo:

La misión de OpenOffice.org: "Crear, en el entorno de una comunidad, el suite de oficina internacional —líder— que trabajará en todas las plataformas principales y permitirá un acceso a toda la funcionalidad y datos por medio de APIs basados en componentes abiertos y un formato de archivos XML."

#### 24.1.2.1

#### *Historia de OpenOffice.org*

Durante la pasada década de los 90's, la empresa Sun Microsystems intentó comercializar su serie de estaciones de trabajo Ultra, estos eran equipos de escritorio equipados con su sistema operativo Solaris, una versión de Unix. Entre las dificultades que encontraron para penetrar en el mercado fue que el usuario normal no podía crear o editar documentos de oficina como hojas de calculo o presentaciones.

En aquel entonces una de las suites de oficina disponibles para Unix era StarOffice, software desarrollado por la empresa alemana StarDivision. StarOffice era una suite ofimática madura y en una fase avanzada de desarrollo, en 1999 estaba en la versión 5.0, contaba con hoja de cálculo (StarCalc), procesador de texto (StarWriter), paquete de presentaciones (StarImpress), un editor de ecuaciones (StarMath) y un programa de dibujo vectorial (StarDraw). Además, StarOffice se integraba en un escritorio propio e incluía un navegador con el cual era posible ver páginas HTML de Internet. En 1998 Sun mostró interés en StarOffice y decidió la adquisición de StarDivision para hacer frente al problema de la falta de programas de oficina para Solaris.

Sun nunca tuvo la intención de competir con el Office de Microsoft y su motivación era darle un valor agregado a sus soluciones en workstations. Debido a ello la empresa ofreció el uso de StarOffice como freeware, es decir, que cualquiera lo podía usar sin pagar una licencia. De manera derivada, StarDivision también había desarrollado StarOffice para Windows y Linux.

StarOffice tuvo una discreta aceptación entre los usuarios de Windows, pero en Linux rápidamente se colocó como la suite de oficina preferida. Como única forma de crecer y desarrollarla, se decidió que en el año 2000 Sun Microsystems liberase el código de StarOffice bajo la licencia GPL al grupo de desarrollo de OpenOffice.org.

El código liberado de StarOffice 6.0 fue usado para lanzar en 2001 OpenOffice.org 1.0 y en 2003, —luego de miles de mejoras hechas por el equipo de OpenOffice.org—, se lanzó la versión 1.1..

En octubre de 2010 como consecuencia de la compra de Sun Microsystems (líder del desarrollo de OpenOffice.org) por Oracle y la orientación que esta empresa le daba al proyecto aparece LibreOffice. LibreOffice se creó como una bifurcación (una división de un proyecto de software libre en dos con orientaciones diferentes) de OpenOffice y está creada y mantenida por una comunidad liderada por la fundación The Document Foundation.

### 24.1.3

### *Microsoft Office.*

Es sin duda la suite más popular y utilizada de todas las disponibles, tanto en aplicaciones independientes como en conjunto. Existen diferentes versiones de esta suite diferenciándose principalmente en las aplicaciones incluidas y la licencia de uso de ciertas funcionalidades: estándar, para educación, para pequeñas empresas y profesionales. Sus programas principales son muy conocidos y considerados como estándar en algunos entornos.

Incluye una serie de aplicaciones que pueden adquirirse de manera independiente o con el paquete y son conocidas como:

- WORD
- EXCEL
- ACCESS
- POWERPOINT
- OUTLOOK
- FRONTPAGE
- PUBLISHER
- PROJECT

#### Historia de Microsoft Office

Microsoft siguió con Office la estela del paquete Lisa Office System de Apple que ya en 1983 ofrecía procesador de texto y hoja de cálculo entre sus siete aplicaciones, bajo un sistema operativo con ventanas, escritorio y papelera, 12 años antes del Windows 95. El paquete de office para Windows data de 1990. El término fue inicialmente usado en marketing para vender un set de aplicaciones, que previamente se vendían separadamente. El principal argumento de venta era que comprar el paquete completo resultaba más barato que comprar cada aplicación por

separado. La primera versión de Office contenía las aplicaciones Microsoft Word, Microsoft Excel y Microsoft PowerPoint. Adicionalmente, una llamada "versión profesional" de Office incluía también Microsoft Access y Schedule Plus.

Con el transcurso de los años las aplicaciones de Office han crecido sustancialmente desde un punto de vista técnico, incluso comparten funcionalidades, tales como: corrector ortográfico común, un integrador de datos OLE y el lenguaje de scripts de Visual Basic para Aplicaciones. Microsoft también posiciona a Office como una plataforma de desarrollo para la línea de software para negocios.

La versión actual es Office 2010 para Windows, presentada en 2010.

#### **24.1.4** ***Suites ofimáticas en Web***

##### **24.1.4.1** ***Feng Office***

Feng Office es una aplicación libre de tipo Web Office, antes conocida como OpenGoo. Se trata de una sistema completo que proporciona funcionalidades para crear, publicar, colaborar y compartir documentos.

Feng Office permite crear y trabajar entre otros, sobre:

- *Documentos*: permite alojar documentos de todo tipo y editar directamente algunos de ellos.
- *Listas de tareas*: permite la creación de listas de tareas asignadas a distintos usuarios, con opciones de notificación, categorización, etc
- *Correo electrónico*: permite centralizar la gestión de las distintas cuentas de correo.

- *Calendario*: permite establecer reuniones y una gestión de las actividades diarias.
- *Agenda*: permite realizar una gestión de contactos.

Esta aplicación puede funcionar bajo un modelo SaaS (Software as a Service), donde los servidores del proveedor a través de un navegador, nos permiten trabajar con la aplicación, pero también es posible realizar una instalación de la aplicación en un servidor propio, en este caso los requisitos de este sistema pasan por un servidor web Apache, PHP y MySQL como base de datos.

#### **24.1.4.2**

#### **Google Docs**

Google Docs&Spreadsheets, es un programa web gratuito que permite crear y trabajar sobre unos documentos de manera individual o en grupo.

Google Docs se compone de:

- Procesador de textos
- Hojas de cálculo
- Programa de presentación sencillo
- Editor de formularios

Entre las ventajas de Google Docs, se encuentra el hecho de que puede ser usado tanto online como offline. En esta modalidad offline, los cambios que se introduzcan en los documentos serán actualizados de forma automática en cuanto la conexión con internet se restablezca.

Además recientemente se ha incorporado compatibilidad entre Google Docs y los dispositivos móviles, de tal forma que se pueda no solo acceder a los documentos sino también editarlos.

#### **24.1.4.3**

#### **Office Web Apps**

Office Web Apps es la solución de Microsoft para las suites ofimática en la web. Se trata de una versión gratuita basada en el conjunto de aplicaciones de Microsoft Office.

Office Web Apps, se compone de :

- Word Web App
- Excel Web App
- PowerPoint Web App
- OneNote Web App

Estas aplicaciones permite acceder a los documentos a través del navegador, así como compartir archivos y trabajar sobre ellos de forma colaborativa.



## **24.2 PROCESADORES DE TEXTO**

### **24.2.1 *Historia de los procesadores de textos***

Las herramientas de procesamiento de texto fueron el primer tipo de aplicación que se le dio a los primeros ordenadores.

Inicialmente, los mecanismos de comunicación entre los programadores, que se encargaban de editar el comportamiento y operaciones, y las máquinas de proceso se realizaba fundamentalmente mediante el uso de tarjetas perforadas, o a través de codificaciones dificultosas y que implicaban una alta inversión de tiempo. Con el objeto de cubrir esta necesidad de interacción entre los programadores y la operativa de las máquinas, se diseñaron y construyeron las primeras aplicaciones que permitiesen programar las instrucciones de forma más legible, mediante la escritura de comandos en forma de texto. Esto dio origen a los primeros editores de textos.

A medida que los microcomputadores se van extendiendo fuera de ámbitos específicos y el uso de la informática se extiende a nivel general, la evolución de los editores de texto como herramientas de uso general va cobrando fuerza. Hoy en día los editores de texto son herramientas casi de propósito general que permiten la redacción de documentos, generación de informes, etc, en todos los ámbitos profesionales y personales. Este auge de este tipo de herramientas ha provocado que muchas empresas de diseño de software impliquen esfuerzos en la construcción y diseño de nuevos y mejores productos, actualmente integrados en suites ofimáticas que abarcan no sólo la edición de textos sino el manejo de otras operativas habituales, como la generación de gráficas o plantillas de cálculo.

### **24.2.2 *Procesadores o editores de textos***

Los editores o procesadores de textos son aplicaciones informáticas que se han convertido en las más usuales y utilizadas dentro del ámbito de la informática profesional y de usuario. Sobre el uso de las antiguas máquinas de escribir, los procesadores de texto digitales permiten la posibilidad de escribir documentos de cientos de páginas, realizar correcciones y/o

modificarlos, automatizar tareas de edición, corrección, y finalmente imprimirlos.

Son programas que permiten realizar todas las operaciones necesarias para editar, guardar, recuperar, modificar e imprimir un texto. Al escribir con un procesador de textos, no hay que preocuparse del final de la línea, ni tampoco del fin de página, ya que el programa salta automáticamente a la línea siguiente o a la hoja siguiente cuando una esté completa. El propio procesador delimitará el tamaño de la hoja, líneas por página.

Mediante sencillos procedimientos podemos mover, borrar, subrayar o repetir párrafos, frases y palabras dentro de un texto. Una vez finalizada la manipulación del documento, podemos guardarlo en un soporte magnético u óptico, imprimirlo o ambas cosas. (Cuando se edita o crea un texto, éste reside en la memoria interna, luego solo permanece temporalmente, perdiéndose al desconectar el equipo. Para evitar la pérdida, antes debe guardarse en un soporte, si deseamos conservarlo).

Los procesadores tienen acceso y comunicación con otros programas: bases de datos, hojas de cálculo, diccionarios, otros archivos, etc, aunque esto ya no es imprescindible en los entornos Linux o Windows, dadas las facilidades que ofrecen para la interrelación entre programas.

Todos los procesadores de textos permiten establecer cabeceras y piés de página, definir la anchura del documento, el número de caracteres por línea, definir longitud de páginas, marcar márgenes y tabuladores, mover, copiar o borrar bloques de textos, definir atributos de texto (negrita, subrayado ...). También ofrecen la posibilidad de crear de forma sencilla tablas, gráficos, dibujos e insertar incluso imágenes. Es normal la la posibilidad de visualizar el documento en pantalla en su formato definitivo, es decir tal y como va a imprimirse. Esta muestra previa es muy interesante para comprobar su aspecto final sin necesidad de imprimirlo; se conoce como WYSIWYNG (se obtiene lo que se ve). Un programa que no es WYSIWYNG, aunque es el más potente de todos es Tex, escrito por Donald Knuth de la Universidad de Stanford (EE.UU.), muy utilizado por los matemáticos y científicos en general, contando con muchos defensores.

Respecto a la seguridad, guardan automáticamente una copia del documento anterior; otros tienen claves de acceso (PASSWORD) que protegen el texto, permitiendo su manipulación solamente a los usuarios que conozcan la contraseña. Sin embargo Microsoft WORD tiene niveles de seguridad muy deficientes.

Los procesadores de texto tienen la posibilidad de disponer de los tipos de letras del respectivo sistema operativo, aunque es el tipo de impresora el factor limitativo de la calidad de los resultados obtenidos.

Los procesadores actuales cuentan con programas auxiliares como los diccionarios ortográficos, de sinónimos o bilingües. Los ortográficos sirven para revisar el documento completo detectando los errores de mecanografía y faltas de ortografía. Los de sinónimos (tesauros) permiten consultar posibles alternativas a una palabra. Los diccionarios bilingües permiten buscar la palabra en otro idioma. Otras posibilidades hasta hace poco tiempo consideradas como avanzadas son: editores de fórmulas, posibilidad de definir macros, sombreados de marcos, escritura en columnas. Otros programas interesantes son los comprobadores de estilo que comparan los textos con una serie de reglas gramaticales. Detectan errores de puntuación, mayúsculas, palabras repetidas, palabras en desuso.

Es también muy importante la existencia de una ayuda lo más sencilla y completa posible que evite la consulta constante del manual. Algunos programas incluyen discos tutores y libros de aprendizaje con ejemplos.

El procesamiento de textos no es escribir, aunque se piense que el ser un buen mecanógrafo es importante para utilizar un procesador de textos, hay algunas aptitudes de la mecanografía que son contraproducentes. Seguidamente se listan los nuevos hábitos a tener en cuenta al pasar al uso de un editor de textos:

- Utiliza la tecla intro (o retorno de carro o fin de línea) sólo cuando debas. Los ajustes de línea se producen automáticamente.
- Utiliza guías de tabulación y márgenes, no la barra espaciadora para alinear columnas
- No subrayes. Utiliza cursivas y negritas para enfatizar el texto.

- Utiliza sólo un espacio después de un punto. Se deben evitar los espacios dobles
- Benefíciate de los caracteres especiales. Caracteres no comunes de las máquinas de escribir hacen que los escritos parezcan más profesionales.
- Sistemas de tratamiento de textos y documentos.

### **24.2.3 Ejemplos de procesadores de textos**

#### **24.2.3.1 Microsoft Word:**

Es uno de los principales procesadores de texto existentes hoy en día. Su popularidad se debe fundamentalmente a la gran difusión de las plataformas de Microsoft, y a la facilidad de uso tanto del propio editor como de la suite ofimática en la que está integrado. El del paquete Microsoft Office se ha convertido casi en un estándar de referencia precisamente dado el elevado porcentaje de usuarios que lo han instalado en su ordenador y lo utilizan.

#### **24.2.3.2 Lotus Word Pro:**

Una de las alternativas más populares a Microsoft Word es este procesador de textos, incluido en el paquete de software de escritorio de Lotus. Lotus es otra firma comercial que distribuye sus productos al igual que Microsoft bajo licencia comercial, y actualmente representa una de las alternativas más robustas y fiables del mercado.

#### **24.2.3.3 Word Perfect:**

A mediados de la década de los 90 Word Perfect representaba el editor de moda, aunque actualmente se encuentre en cierta decadencia a favor de herramientas comerciales ampliamente más extendidas como los paquetes de Microsoft o Lotus, o bien por alternativas libres como OpenOffice. Este procesador de textos presenta un despliegue de innovadoras características que demuestran el interés por parte de sus promotores en volver a hacer de este programa un producto puntero capaz de competir en el mercado con Microsoft Word.

#### **24.2.3.4**

#### **Word Pad:**

Instalado por defecto en todas las versiones de los sistemas operativos de Windows, se le podría considerar el "hermano pequeño" de Microsoft Word. Es la opción ideal para aquellos usuarios que necesitan de forma esporádica un procesador con el que dar cierto estilo estético a sus escritos, sin buscar un acabado de apariencia profesional ni excesivas florituras ornamentales.

#### **24.2.3.5**

#### **Block de Notas:**

También presente por defecto en todas las instalaciones del sistema de Windows, independientemente de la versión, este programa se muestra como la opción ideal para usuarios austeros. Al tratarse de una aplicación de posibilidades reducidas, no tendrán que familiarizarse con un complejo entorno lleno de funciones que nunca van a utilizar.

#### **24.2.3.6**

#### **OpenOffice Writer:**

Es el equivalente a Word en software libre. Sus posibilidades y forma de utilizarlo es enormemente parecida, con la ventaja de que al tratarse de software libre podemos descargarlo gratuitamente de Internet y actualizarlo cuantas veces queramos sin pagar derechos de autor.

#### **24.2.3.7**

#### **LibreOffice Writer:**

Es el procesador de textos de la suite ofimática LibreOffice y ofrece características muy similares al de OpenOffice, puesto que comparten una base común. Puede proteger documentos con contraseña, guardar versiones del mismo documento, insertar imágenes, objetos OLE, admite firmas digitales, símbolos, fórmulas, tablas de cálculo, gráficos, hiperenlaces, marcadores, formularios, etc.

Writer permite exportar archivos de texto a los formatos PDF y HTML sin software adicional, lo que permite que pueda ser utilizado como un editor WYSIWYG para crear y editar páginas web.

## **24.3**

## **HOJAS DE CÁLCULO**

Con el avance de la computación y el diseño de nuevos ordenadores más potentes y compactos, las tendencias de los grandes sistemas de procesamiento fueron migrando hacia la adquisición microordenadores con alta capacidad de proceso. El proceso de adaptación de los sistemas de planificación utilizados en los antiguos sistemas de proceso dieron paso a nuevas herramientas de cálculo y gestión más dinámicos, que integrasen herramientas sencillas de cálculo y que se asemejasen a las tablas de cálculos que habitualmente se utilizaban en la gestión a base de lápiz y papel.

Como solución a esta tendencia, nacen las hojas de cálculo, también llamadas hojas electrónicas de cálculo. Las hojas de cálculo son sistemas software, habitualmente integrados dentro de alguna suite ofimática, que representan tablas con celdillas en un formato que resulta cómodo para realizar operaciones contables y generación de informes o cálculos estadísticos. Combinan además herramientas de cálculo y diferentes funcionalidades añadidas. En la actualidad, las hojas de cálculo conforman una herramienta de gran importancia en el mundo empresarial.

En 1961 se desveló el concepto de una hoja de cálculo electrónica en el artículo "Budgeting Models and System Simulation" de Richard Mattessich. Pardo y Landau merecen parte del crédito de este tipo de programas, y de hecho intentaron patentar (patente en EE.UU. número 4398249) algunos de los algoritmos en 1970. La patente no fue concedida por la oficina de patentes por ser una invención puramente matemática. Pardo y Landau ganaron un caso en la corte estableciendo que "algo no deja de ser patentable solamente porque el punto de la novedad es un algoritmo". Este caso ayudó al comienzo de las patentes de software. La primera hoja de cálculo comercializada masivamente (Visicalc, posteriormente comprada por Lotus) nació en 1977 de una idea de un estudiante, Dan Bricklin, de administración de empresas de la Universidad de Harvard (EE.UU.).

Los cálculos empresariales manejan unas cantidades de datos que suelen variar dependiendo de los supuestos, por lo que hasta la invención de la hoja electrónica se había desechado aplicar la informática a ese entorno de trabajo, donde cada caso originaría un programa distinto.

La hoja de cálculo se presenta como una tabla o matriz de dos dimensiones (actualmente existen de tres) que consta de un número de filas y columnas variable según el modelo del que se trate. (Ej. en LOTUS 123, una de las primeras en el entorno IBM PC, tenía 8192 filas y 256 columnas, en total más de 2000000 de celdas). Con los datos se pueden representar variados tipos de gráficos, de gran utilidad en ambientes empresariales. Las filas son los datos horizontales y las columnas los verticales. El lugar donde se produce la intersección de una fila y una columna se denomina celda. Las columnas habitualmente se nombran con letras (A,B, ... AA, AB, ....) y las filas desde 1 en adelante.

Una hoja de calculo es un programa que permite manipular datos numéricos y alfanuméricos dispuestos en forma de tablas (la cual es la unión de filas y columnas). Habitualmente es posible realizar cálculos complejos con fórmulas y funciones y dibujar distintos tipos de gráficos.

### **24.3.1**

### ***Microsoft Excel***

Microsoft Excel es una aplicación integrada dentro de la suite ofimática Microsoft Office, distribuido bajo licencia comercial en todas sus versiones. Está fundamentalmente orientado al manejo de hojas de cálculo y se utiliza normalmente en tareas financieras y contables.

Microsoft Excel ofrece una interfaz de usuario ajustada a las principales características de las hojas de cálculo generales. El programa muestra las celdas organizadas en filas y columnas. Cada celda contiene datos o fórmulas con referencias relativas, absolutas o mixtas a otras celdas.

Como añadido inicia, Microsoft Excel destaca por haber sido la primera hoja de cálculo que permite al usuario definir la apariencia del documento final, relacionado con las fuentes de letra, atributos de los caracteres, celdas, etc. También integra facetas como el recálculo inteligente de

celdas, consistente en que las celdas cuyo valor es dependiente de otras que se modifican, se actualizan al instante.

Además de estas facetas, Microsoft Excel integra un módulo gráfico que permite el diseño de una gran cantidad de gráficas representativas de los valores con los que se trabaja en las tablas, e integra además un motor de solver propio con capacidad para la resolución de problemas de optimización en el ámbito de la programación lineal.

Microsoft Excel incluye también Visual Basic para Aplicaciones (VBA), que es un lenguaje de programación basado en Visual Basic, y que añade la capacidad para la automatización de tareas. Además permite al usuario definir funciones para su uso en las hojas de trabajo. En las versiones más recientes del paquete Office, Microsoft Excel cuenta ya con un entorno completo integrado de edición de VBA para el desarrollo de funciones y automatización de tareas. Así mismo, la generación de macros puede producir código VBA para repetir las acciones del usuario, lo que permite la automatización de simples tareas. VBA permite la creación de formularios y controles en la hoja de trabajo para comunicarse con el usuario. Admite el uso del lenguaje (pero no la creación) de las DLL de ActiveX (COM); versiones posteriores añadieron soporte para los módulos de clase permitiendo el uso de técnicas de programación básicas orientadas a objetos.

Como contrapartida a la funcionalidad proporcionada por VBA y la automatización de tareas, se encuentra la vulnerabilidad provocada por el ataque de virus en forma de macro.

### **24.3.2**

### ***Openoffice.orgCalc***

OpenOffice.org Calc es un software de creación y gestión de hojas de cálculo integrado en la suite ofimática OpenOffice.org. Se trata de una hoja de cálculo Open Source y software libre compatible con Microsoft Excel. OpenOffice.org, Calc es perfectamente compatible con prácticamente todas las plataformas existentes, abarcando Mac OS X, Windows, GNU/Linux, FreeBSD y Solaris, y se encuentra disponible bajo licencia LGPL.



OpenOffice.org Calc es una hoja de cálculo similar a Microsoft Excel, con un conjunto de características equivalentes a nivel de funcionalidad. Sin embargo su tamaño es menor y proporciona un número de características no presentes en Excel, incluyendo un sistema que automáticamente define series para representar gráficamente, basado en la disposición de los datos del usuario. Calc también es capaz de exportar hojas de cálculo como archivos PDF, cuenta con filtros y además puede realizar agrupaciones en tablas dinámicas.

En cuanto a la automatización de tareas y programación de macros, Calc no es compatible con el modelo de objetos de Excel, y esto supone una fortaleza frente a la gran cantidad de virus basados en macros.

## **24.4**

## **BASES DE DATOS**

Se define una base de datos como “una serie de datos organizados y relacionados entre sí, los cuales son recolectados y explotados por los sistemas de información de una empresa o negocio en particular”.

Es decir, una base de es un conjunto de datos que pertenecen al mismo contexto almacenados sistemáticamente para su posterior uso, un “almacén” que nos permite guardar grandes cantidades de información de forma organizada para que luego podamos encontrar y utilizar fácilmente.

El término de bases de datos fue escuchado por primera vez en 1963, en un simposio celebrado en California, USA. Una base de datos se puede definir como un conjunto de información relacionada que se encuentra agrupada ó estructurada.

Desde el punto de vista informático, la base de datos es un sistema formado por un conjunto de datos almacenados en discos que permiten el acceso directo a ellos y un conjunto de programas que manipulen ese conjunto de datos.

Cada base de datos se compone de una o más tablas que guarda un conjunto de datos. Cada tabla tiene una o más columnas y filas. Las columnas guardan una parte de la información sobre cada elemento que queramos guardar en la tabla, cada fila de la tabla conforma un registro.

### **24.4.1**

### **Características**

Entre las principales características de los sistemas de base de datos podemos mencionar:

- Independencia lógica y física de los datos.
- Redundancia mínima.
- Acceso concurrente por parte de múltiples usuarios.

- Integridad de los datos.
- Consultas complejas optimizadas.
- Seguridad de acceso y auditoría.
- Respaldo y recuperación.
- Acceso a través de lenguajes de programación estándar.

### **24.4.2    *Sistema de Gestión de Base de Datos (SGBD)***

Los Sistemas de Gestión de Base de Datos (en inglés DataBase Management System) son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta.

### **24.4.3                    *Ventajas de las bases de datos***

#### **24.4.3.1            *Control sobre la redundancia de datos:***

Los sistemas de ficheros almacenan varias copias de los mismos datos en ficheros distintos. Esto hace que se desperdicie espacio de almacenamiento, además de provocar la falta de consistencia de datos.

En los sistemas de bases de datos todos estos ficheros están integrados, por lo que no se almacenan varias copias de los mismos datos. Sin embargo, en una base de datos no se puede eliminar la redundancia completamente, ya que en ocasiones es necesaria para modelar las relaciones entre los datos.

#### **24.4.3.2                    *Consistencia de datos:***

Eliminando o controlando las redundancias de datos se reduce en gran medida el riesgo de que haya inconsistencias. Si un dato está almacenado una sola vez, cualquier actualización se debe realizar sólo una vez, y está disponible para todos los usuarios inmediatamente. Si un dato está duplicado y el sistema conoce esta redundancia, el propio sistema puede encargarse de garantizar que todas las copias se mantienen consistentes.

#### **24.4.3.3** **Compartición de datos:**

En los sistemas de ficheros, los ficheros pertenecen a las personas o a los departamentos que los utilizan. Pero en los sistemas de bases de datos, la base de datos pertenece a la empresa y puede ser compartida por todos los usuarios que estén autorizados.

#### **24.4.3.4** **Mantenimiento de estándares:**

Gracias a la integración es más fácil respetar los estándares necesarios, tanto los establecidos a nivel de la empresa como los nacionales e internacionales. Estos estándares pueden establecerse sobre el formato de los datos para facilitar su intercambio, pueden ser estándares de documentación, procedimientos de actualización y también reglas de acceso.

#### **24.4.3.5** **Mejora en la integridad de datos:**

La integridad de la base de datos se refiere a la validez y la consistencia de los datos almacenados. Normalmente, la integridad se expresa mediante restricciones o reglas que no se pueden violar. Estas restricciones se pueden aplicar tanto a los datos, como a sus relaciones, y es el SGBD quien se debe encargar de mantenerlas.

#### **24.4.3.6** **Mejora en la seguridad:**

La seguridad de la base de datos es la protección de la base de datos frente a usuarios no autorizados. Sin unas buenas medidas de seguridad, la integración de datos en los sistemas de bases de datos hace que éstos sean más vulnerables que en los sistemas de ficheros.

#### **24.4.3.7** **Mejora en la accesibilidad a los datos:**

Muchos SGBD proporcionan lenguajes de consultas o generadores de informes que permiten al usuario hacer cualquier tipo de consulta sobre los datos, sin que sea necesario que un programador escriba una aplicación que realice tal tarea.

#### **24.4.3.8** **Mejora en la productividad:**

El SGBD proporciona muchas de las funciones estándar que el programador necesita escribir en un sistema de ficheros. A nivel básico, el SGBD proporciona todas las rutinas de manejo de ficheros típicas de los programas de aplicación.

El hecho de disponer de estas funciones permite al programador centrarse mejor en la función específica requerida por los usuarios, sin tener que preocuparse de los detalles de implementación de bajo nivel.

#### **24.4.3.9** **Mejora en el mantenimiento:**

En los sistemas de ficheros, las descripciones de los datos se encuentran inmersas en los programas de aplicación que los manejan.

Esto hace que los programas sean dependientes de los datos, de modo que un cambio en su estructura, o un cambio en el modo en que se almacena en disco, requiere cambios importantes en los programas cuyos datos se ven afectados.

Sin embargo, los SGBD separan las descripciones de los datos de las aplicaciones. Esto es lo que se conoce como independencia de datos, gracias a la cual se simplifica el mantenimiento de las aplicaciones que acceden a la base de datos.

#### **24.4.3.10** **Aumento de la concurrencia:**

En algunos sistemas de ficheros, si hay varios usuarios que pueden acceder simultáneamente a un mismo fichero, es posible que el acceso interfiera entre ellos de modo que se pierda información o se pierda la integridad. La mayoría de los SGBD gestionan el acceso concurrente a la base de datos y garantizan que no ocurran problemas de este tipo.

#### **24.4.3.11** **Mejora en los servicios de copias de seguridad:**

Muchos sistemas de ficheros dejan que sea el usuario quien proporcione las medidas necesarias para proteger los datos ante fallos en el sistema o

en las aplicaciones. Los usuarios tienen que hacer copias de seguridad cada día, y si se produce algún fallo, utilizar estas copias para restaurarlos.

En este caso, todo el trabajo realizado sobre los datos desde que se hizo la última copia de seguridad se pierde y se tiene que volver a realizar. Sin embargo, los SGBD actuales funcionan de modo que se minimiza la cantidad de trabajo perdido cuando se produce un fallo.

#### **24.4.4            *Desventajas de las bases de datos***

##### **24.4.4.1                            Complejidad:**

Los SGBD son conjuntos de programas que pueden llegar a ser complejos con una gran funcionalidad. Es preciso comprender muy bien esta funcionalidad para poder realizar un buen uso de ellos.

##### **24.4.4.2                            Coste del equipamiento adicional:**

Tanto el SGBD, como la propia base de datos, pueden hacer que sea necesario adquirir más espacio de almacenamiento. Además, para alcanzar las prestaciones deseadas, es posible que sea necesario adquirir una máquina más grande o una máquina que se dedique solamente al SGBD. Todo esto hará que la implantación de un sistema de bases de datos sea más cara.

##### **24.4.4.3                            Vulnerable a los fallos:**

El hecho de que todo esté centralizado en el SGBD hace que el sistema sea más vulnerable ante los fallos que puedan producirse. Es por ello que deben tenerse copias de seguridad (Backup).

#### **24.4.5   *Herramientas para la gestión de Bases de Datos presentes en suites ofimáticas***

##### **24.4.5.1                            LibreOffice Base**

LibreOffice Base es un programa de base de datos. LibreOffice Base permite la creación y manejo de bases de datos, elaboración de formularios e informes que proporcionan a los usuarios finales un acceso fácil a los

datos. Al igual que Microsoft Access, es capaz de trabajar como un front-end para diversos sistemas de bases de datos tales como el de Access (JET ), fuente de datos ODBC y MySQL/PostgreSQL. Base forma parte de la suite ofimática desde la versión 2.0 de OpenOffice (de la cual se deriva LibreOffice).

#### **24.4.5.2**

#### **Microsoft Access**

Microsoft Access es un sistema de gestión de bases de datos relacionales para los sistemas operativos Microsoft Windows, desarrollado por Microsoft y orientado a ser usado en un entorno personal o en pequeñas organizaciones. Es un componente de la suite ofimática Microsoft Office. Permite crear ficheros de bases de datos relacionales que pueden ser fácilmente gestionadas por una interfaz gráfica simple. Además, estas bases de datos pueden ser consultadas por otros programas. Este programa permite manipular los datos en forma de tablas (formadas por filas y columnas), crear relaciones entre tablas, consultas, formularios para introducir datos e informes para presentar la información.

## **24.5** **PRESENTACIÓN**

Un programa de presentación es un paquete de software usado para mostrar información, normalmente mediante una serie de diapositivas.

Típicamente incluye tres funciones principales: un editor que permite insertar un texto y darle formato, un método para insertar y manipular imágenes y gráficos y un sistema para mostrar el contenido en forma continua.

### **24.5.1** ***Tipos de presentación***

Hay muchos tipos de presentaciones, para educación, o para comunicar noticias en general. Los programas de presentación pueden servir de ayuda o reemplazar a las formas tradicionales de dar una presentación, como por ejemplo panfletos, resúmenes en papel, pizarras, diapositivas o transparencias.

Un programa de presentación permite colocar texto, gráficos, películas y otros objetos en páginas individuales o "diapositivas". El término "diapositiva" es una referencia al proyector de diapositivas, un dispositivo que se ha quedado obsoleto para estos fines desde la aparición de los programas de presentación.

Las diapositivas se pueden imprimir en transparencias y ser mostradas mediante un proyector de transparencias, o ser visualizadas directamente en la pantalla del ordenador (o en una pantalla normal usando un proyector de vídeo) bajo el control de la persona que da la presentación. La transición de una diapositiva a otra puede ser animada de varias formas, y también se puede animar la aparición de los elementos individuales en cada diapositiva.



## **24.5.2** *Herramientas de presentación*

### **24.5.2.1** *Microsoft PowerPoint*

Microsoft PowerPoint es un software incluido dentro de la suite ofimática de Microsoft Office orientado a la realización de presentaciones digitales que permite integrar diferentes medios multimedia en la elaboración. Está disponible para plataformas Windows y MacOS y es ampliamente utilizado en distintos campos como la enseñanza, negocios, etc.

El objetivo principal es la realización de presentaciones esquemáticas que combinan texto, imágenes, animaciones, sonidos e incluso vídeos. El objeto es realizar contenidos elegantes y fácilmente comprensibles, pudiendo hacer uso de las funcionalidades del programa para la completa edición de todos los elementos que se utilizan, aplicando diferentes diseños de fuentes y tipos de letra, plantillas predefinidas, animaciones, e incorporación de imágenes prediseñadas o bien importadas desde bibliotecas externas. Este tipo de presentaciones suele ser muy llamativo y mucho más práctico que los de Microsoft Word.

La facilidad de uso de este software, así como la popularidad de las plataformas y familiar de productos de Microsoft, ha provocado que PowerPoint sea uno de los programas de presentación más extendidos actualmente. Viene integrado en el paquete Microsoft Office como un elemento más, que puede aprovechar las ventajas que le ofrecen los demás componentes del equipo para obtener un resultado óptimo.

### **24.5.2.2** *OpenOffice Impress*

Los usuarios de OpenOffice.org Impress también pueden instalar la Open Clip Art Library (Biblioteca Abierta de Clip Art), que agrega una enorme galería de banderas, logos, iconos, estandartes y pancartas para presentaciones generales y proyectos de dibujo. Algunas distribuciones Linux, como Debian, Mandriva Linux y Ubuntu han proporcionado un paquete llamado openclipart listo para usar y fácil de bajar desde sus repositorios, instalando una galería de imágenes y sonidos para el OpenOffice.org.

OpenOffice.org Impress es un programa de presentación similar a Microsoft PowerPoint que viene integrado en la suite de OpenOffice.org, desarrollada por Sun Microsystems. Entre las principales ventajas añadidas que vienen integradas en este software, destaca la posibilidad de exportar las presentaciones a formatos de archivo SWF, utilizado por Adobe Flash. Esto permite cierta portabilidad permitiendo que dichas presentaciones puedan ser ejecutadas en cualquier ordenador que disponga de una versión del reproductor de Adobe Flash Player instalado. Como en el resto de herramientas del paquete de OpenOffice, Impress permite también la creación de archivos PDF de forma directa. Como punto negativo, Impress no dispone del repertorio de diseños predefinidos con los que cuentan otras herramientas como el caso de su competidor directo de Microsoft, PowerPoint. De todas formas, esto no supone un gran inconveniente dado que pueden obtenerse sin dificultad plantillas elaboradas por terceros a través de Internet, proporcionadas por la creciente comunidad de usuarios.

### **24.5.2.3** **LibreOffice Impress**

LibreOffice Impress, derivado de OpenOffice Impress es otro de los programas orientados al desarrollo de presentaciones multimedia basadas en transparencias o diapositivas. El formato nativo de las presentaciones es ODP, pero también tiene la capacidad de leer y escribir en el formato de archivos de Microsoft PowerPoint (ppt), así como la capacidad de exportar las presentaciones a archivos PDF. Puede también exportar las presentaciones al formato SWF, permitiendo que se reproduzcan en cualquier computadora con un reproductor de Flash instalado.

## **24.6** **BIBLIOGRAFIA**

- John L. Hennessy, David A. Patterson Computer architecture: a quantitative approach, Elsevier, Morgan Kaufmann, 2007.
- Carl Hamacher, Zvonko Vranesic and Safwat Zaky. Organización de Computadores, 5ª edición. Ed. McGraw Hill, 2002.
- Fundamentos de sistemas de información. Madrid: Prentice Hall. Edwards, C.; Ward, J.; Bytheway, A. (1998).

- Essentials of Management Information Systems. Organisation and Technology. Englewood Cliffs: Prentice Hall. Laudon, K.C.; Laudon, J.P. (2002).
- Administración de los Sistemas de Información. Prentice-Hall. Laudon, K.C. y Laudon, J.P. (2002)
- PCWORLD Marzo 2010.
- <http://es.wikipedia.com>
- <http://www.maestrosdelweb.com/principiantes/%C2%BFque-son-las-bases-de-datos/>

**Autor:** Francisco Javier Rodríguez Martínez

Subdirector de Sistemas Escola Superior Enxeñaría Informática Ourense

Colegiado del CPEIG

# **ARQUITECTURA DE LOS SISTEMAS DE INFORMACIÓN.**



# **25. ARQUITECTURA DE LAS REDES INTRANET E INTERNET: CONCEPTO, ESTRUCTURA Y CARACTERÍSTICAS. SU IMPLANTACIÓN EN LAS ORGANIZACIONES.**

## **TEMA 25. ARQUITECTURA DE LAS REDES INTRANET E INTERNET: CONCEPTO, ESTRUCTURA Y CARACTERÍSTICAS. SU IMPLANTACIÓN EN LAS ORGANIZACIONES.**

### **25.1. INTRODUCCIÓN Y CONCEPTOS**

### **25.2. INTERNET**

### **25.3. INTRANET/EXTRANET**

### **25.4. IMPLANTACIÓN DE REDES EN ORGANIZACIONES**

### **25.5. ESQUEMA**

### **25.6. REFERENCIAS**

### **25.1. INTRODUCCIÓN Y CONCEPTOS**

Una red son dos o más nodos comunicados entre sí. A partir de ahí, la red puede aumentarse en cualquier número de nodos y conectarse a otras redes. **Internet** es una red de alcance mundial que conecta las diferentes redes físicas de una manera descentralizada como una red lógica única.

En el mundo de la informática un nodo puede ser cualquier componente de una red, desde dispositivos de interconexión a equipos o estaciones de trabajo, o cualquier otro tipo de cliente como equipos portátiles y dispositivos móviles.

Por debajo de estas redes además tendremos diferentes tipos de redes físicas, que tomarán diferentes medios y tecnologías. Internet proporcionará un mecanismo de comunicación común basado en la familia de protocolos TCP/IP, de manera que cualquiera de estas redes que implemente o acepte esta familia de protocolos podrá comunicarse con las demás.

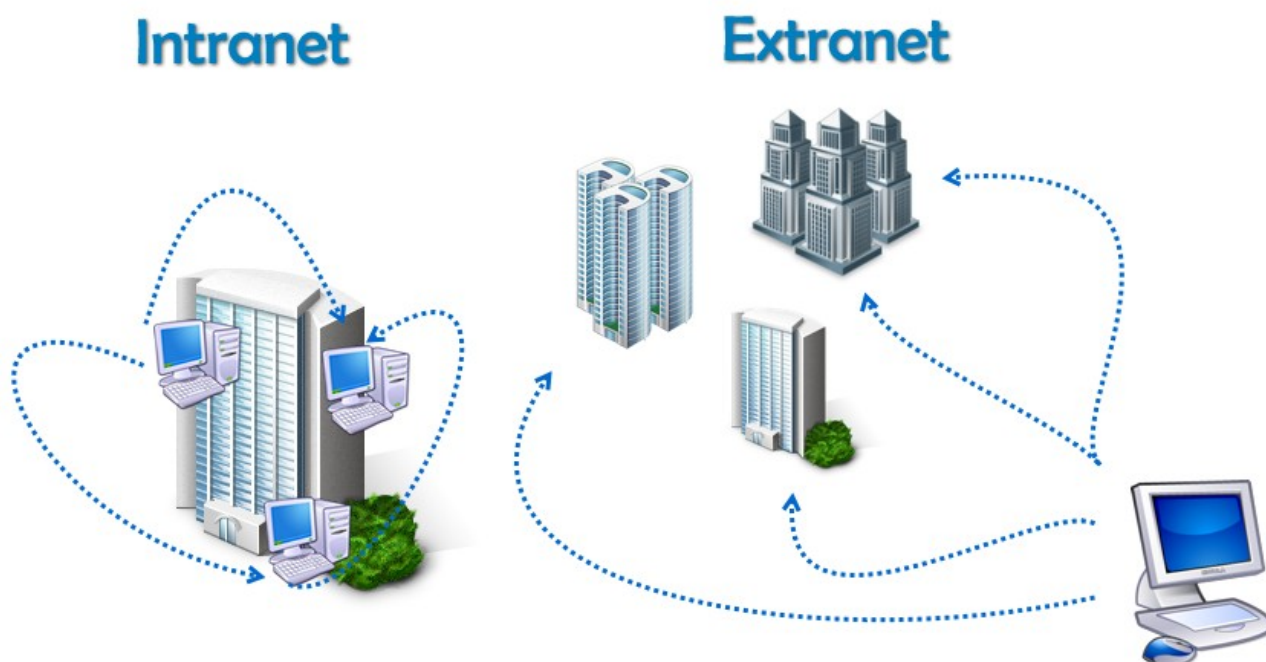
De entre todos los servicios que proporciona Internet el buque insignia es el World Wide Web (WWW, o la Web) el conjunto de protocolos que permite la

consulta de archivos de hipertexto o páginas web emplazados en diferentes sitios de alojamiento o sitios web.

Una **Intranet** es una red interna a una organización o institución, que tiene por objeto proporcionar un conjunto de servicios accesibles exclusivamente desde la red local o un conjunto de redes aisladas del exterior a través de Internet.

La idea principal de una Intranet es que sus servicios sean sólo accesibles para los usuarios de la organización o institución, de una manera personal. Estos servicios pueden incluir servidores web, servidores de correo electrónico, sistemas de gestión de archivos, contenidos y utilidades de comunicación o mensajería.

Extendiendo este concepto a Internet, cuando los servicios están disponibles hacia fuera, pero sólo para los usuarios de la organización o institución se estará hablando de una **Extranet**. En el caso de la Extranet se establece un mecanismo de seguridad o autenticación de los usuarios para garantizar que pertenecen a la organización o institución. En consecuencia, una Extranet no será ni una Intranet ni un sitio de Internet sino la publicación de los servicios de una Intranet a través de la Internet mediante un sistema de autenticación de los usuarios de la organización o institución.



***Figura 1: Intranet y Extranet***

## **25.2. INTERNET**

### **25.2.1. CARACTERÍSTICAS BÁSICAS**

Internet tiene sus orígenes a finales de la década de los 60, siendo una evolución de la red experimental ARPANET (Red de la Agencia de proyectos de investigación avanzada), desarrollada por el departamento de Defensa de los EUA.

La idea original era disponer de una red en la que en caso de producirse daños o la desaparición de algún nodo o punto de la misma, la red permanecería activa entre los nodos o elementos restantes, garantizando así la supervivencia de la información y el funcionamiento del medio de comunicación. A partir de este concepto puede entenderse el funcionamiento distribuido y completamente descentralizado que posee el sistema actualmente, de suerte que cada nodo individual tiene la misma



importancia y peso en el conjunto a la hora de dar servicio o comunicarse con los demás.

Posteriormente se desarrolló sobre la red un software básico de control de la transmisión de información que terminaría por dar lugar a la **familia de protocolos TCP/IP**. Esta familia de protocolos representa un conjunto de normas y estándares que definen el mecanismo de comunicación entre los diferentes nodos de la red. Cualquier red física que implemente o dé soporte a este conjunto de protocolos podrá comunicarse con otras redes que también lo hagan. A partir de uno de estos protocolos, podemos especificar otro de los factores fundamentales que explican el funcionamiento de esta red: el concepto de Dirección **IP** (Protocolo de Internet). Esta dirección representa la dirección o nombre de cada nodo de la red, siendo un identificador único para cada uno de ellos. Las direcciones IP se componen de cuatro cifras numéricas separadas por puntos que toman valores entre 0 y 255. Por ejemplo: 192.168.1.1. Debido al aumento del rango de direcciones se diseñó el **IPv6** que pasa a valores de 128 bits, con ocho grupos de cuatro dígitos hexadecimales, por ejemplo: FEDC:BA98:7654:3210:FEDC:BA98:7654:3210.

Dirección IP	Significado
::	Ausencia de dirección
0:0:0:0:0:0:0:0	Ausencia de dirección
0	
::1	Loopback
::1.2.3.4	Compatible con IPv4
::ffff:0:0	Dirección Ipv4 mapeado
Ff00::	Multicast
FF01:0:0:0:0:0:0:0:101	Multicast

**Tabla 1: Ejemplos de direcciones Ipv6.**

Como este tipo de identificación puede resultar difícil de recordar se emplea en conjunción el Sistema de Nombres de Dominio (**DNS**). En este sistema diferentes nodos de la red hacen las funciones de traductores entre direcciones IP y nombres de Dominio, siendo estos varias palabras separadas por puntos, por ejemplo [www.xunta.es](http://www.xunta.es), indicando en última instancia la zona o tipo de organización a la que pertenece el sitio, en este caso España, con el acrónimo 'es', a continuación la organización, institución o mnemotécnico, en este caso 'xunta', y por último el usuario o protocolo, en este caso 'www'.

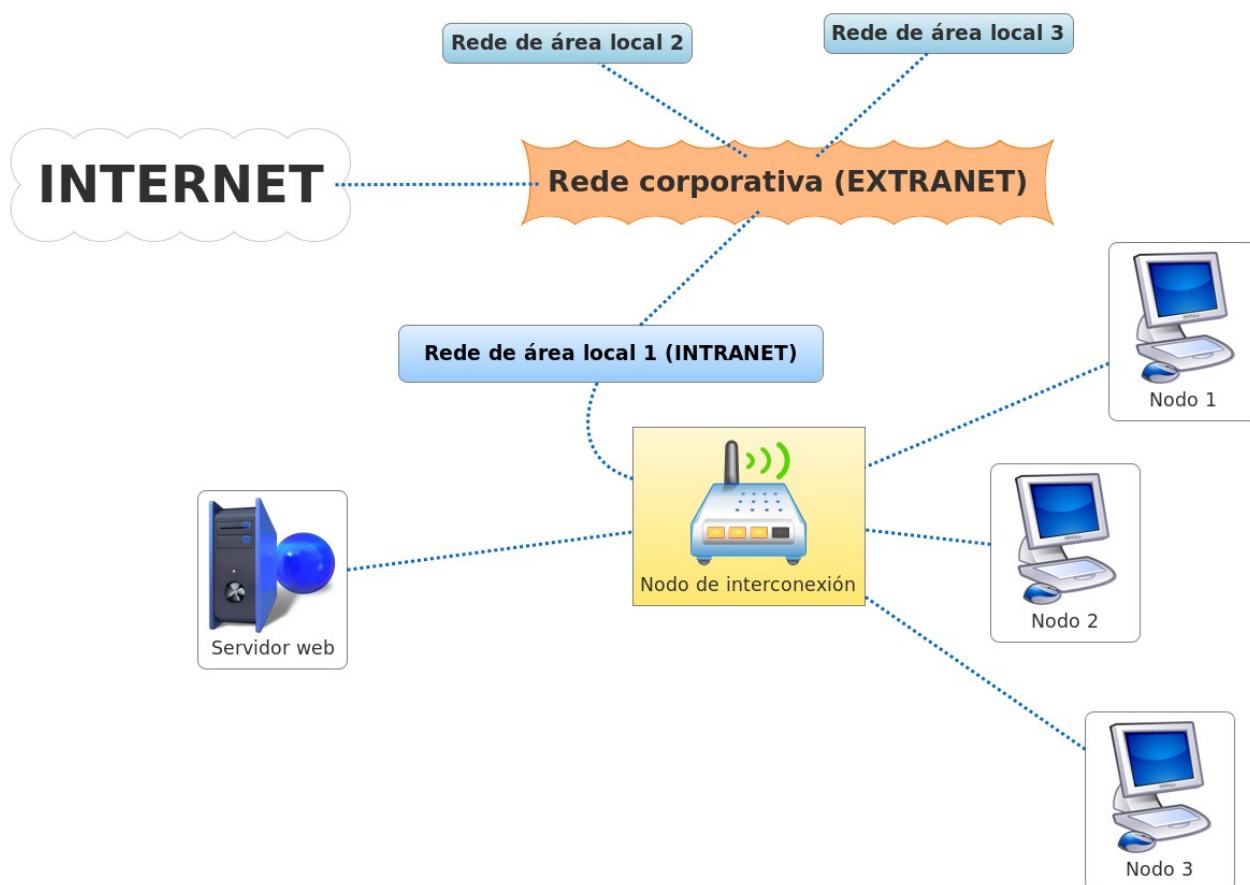
Otro concepto fundamental es el de clientes **y servidores**. El objetivo de la red será doble, comunicar y dar servicios. En este caso podemos distinguir tres tipos de nodos:

1. **Servidores**. Proporcionan servicios a la red, tales como contenidos web, correo electrónico, video, gestión de las comunicaciones, seguridad, etc.
2. **Clientes**. Nodos que representan el equipo de trabajo de un usuario final, el cual hace uso de uno de los servicios de la red que le proporciona un servidor.
3. **Elementos de interconexión**. Son nodos específicos de comunicación, se encargan de gestionar las comunicaciones, retransmitir y dirigir los mensajes.

El modelo de Internet puede aplicarse sobre redes más pequeñas, de menos equipos y una extensión menor. La idea de Internet es una red global, con servicios y comunicación a escala mundial. Abstrayendo funcionamiento y protocolos, pueden hacerse redes más pequeñas con un servicio reducido a su ámbito. A partir de esto tenemos la clasificación habitual de las redes, que incluye:

1. **Redes de área local.** (En inglés *Local Area Network* o LAN). Interconexión de varias computadoras y elementos de interconexión limitada físicamente a un edificio o entorno de alrededor de 200 metros – 1 Kilómetro. Ejemplos de estas redes serían las redes corporativas o institucionales dentro de un mismo edificio, como puede ser la red interna de una Consellería, y por norma general incluyen además servicios de Intranet.
2. **Redes de área metropolitana.** (En inglés *Metropolitan Area Network* o MAN). Interconexión de varias computadoras y elementos de interconexión en un área extensa, como puede ser una ciudad, provincia o comunidad autónoma. Ejemplo de este tipo de redes sería la red corporativa de la Xunta de Galicia. Por norma general este tipo de redes incorporan servicios de Intranet/Extranet.
3. **Redes de área amplia.** (En inglés *Wide Area Network* o WAN). Interconexión de varias computadoras y elementos de interconexión en distancias de 100-1000 Kilómetros. Ejemplo de este tipo de redes sería la propia red Internet.

En la siguiente figura, podemos ver un ejemplo de red de área local, con algunos elementos básicos. Diferentes equipos de trabajo, conectados por nodos de interconexión y con algún servicio como el proporcionado por el servidor web. Esta pequeña red puede encontrarse integrada en una red de mayor alcance con servicios de Intranet y como medio de comunicación con el resto del mundo a través de Internet.



**Figura 2: Ejemplo de red de área local conectada a una red corporativa y a Internet.**

TRADUCCIÓN FIGURA 2: Red corporativa. Red de Área Local

### 25.2.2. TIPOS DE CONEXIÓN A INTERNET

Para conectar otra red o equipo cliente, ya sea un ordenador de sobremesa, portátil, teléfono móvil, PDA , etc., a la red Internet el primer paso será disponer de un **proveedor de acceso o ISP** (en inglés *Internet Service Provider*, proveedor de servicios de Internet). Se trata de empresas que proporcionan y gestionan la conexión a la red a sus clientes, empleando diferentes tecnologías. Por norma general los ISP proporcionan un hardware de conexión a la red específico y puede que un software para gestionarlo.

Entre las tecnologías de conexión más empleadas hoy en día disponemos de:

1. **RTC.** La red telefónica conmutada, que emplea la misma red que los teléfonos fijos en Galicia. En este caso se trata de un soporte analógico por lo que para enviar datos digitales habrá que cambiarlos empleando un dispositivo denominado **Módem** (modulador - demodulador), o variantes más avanzadas con mayores características como la enrutación al estilo de los **Módem-Routers**. De este modo un usuario que quiera acceder a Internet precisará disponer de una línea telefónica y un Módem o Módem-Router. Estos dispositivos pueden ser internos, como sucede normalmente en los dispositivos portátiles o externos. En este último caso la conexión con el equipo de trabajo se realizará conectando el dispositivo por un cable/puerto (P.ej.: USB) o con conectividad sin hilos. Actualmente esta tecnología se encuentra en un estado prácticamente obsoleto, debido a que no puede transmitir datos y voz a la vez y su velocidad máxima es muy baja (alrededor de 56 Kbps).
2. **ADSL.** La línea de abonado digital asimétrica convierte la línea telefónica en una línea de alta velocidad debido a que aprovecha toda la potencia de los hilos estableciendo tres canales independientes:
  - a) Canal de alta velocidad para transmitir datos.
  - b) Canal de alta velocidad para recibir datos.
  - c) Canal de alta velocidad para voz.

De este modo se permite que a través de la misma línea se envíen datos y voz a la vez. El concepto de asimétrica viene de que las velocidades de subida y bajada de datos son diferentes siendo más altas las velocidades de bajada, en una interpretación de que las necesidades de los usuarios van en este sentido. El hardware empleado en este caso serán Módem-

Routers, proporcionados por un ISP. Las velocidades de descarga conseguidas son muy superiores al RTC, yendo de 512 Kbps a un máximo teórico para VSL (una evolución de la ADSL de muy alta tasa de transferencia) de 55 Mbps, si bien los proveedores en Galicia proporcionan bastante menos.

3. **Sin hilos.** Aunque en origen las redes sin hilos fueron diseñadas para redes de área local actualmente también se emplean para posibilitar accesos a Internet. Basados en el conjunto de estándares Wi-Fi (en inglés *Wireless Fidelity*) llegan a conseguir velocidades de alrededor de 54 Mbps llegando al máximo teórico de 600Mbps. El hardware necesario en este caso será un Router Wi-Fi sin hilos que haga las funciones de punto de acceso (en inglés *hotspot*) y en el equipo de trabajo una antena receptora integrada en una tarjeta de red (interna o externa).
4. **Cable.** Redes basadas en tecnologías de fibra óptica lo que implica que precisa una línea de transmisión de esta tecnología. El hardware empleado es similar al de la ADSL, pero en este caso será un Cable-Módem el encargado de gestionar la comunicación, aunque el término Cable-Router sería más idóneo en este caso, pues la gestión es más avanzada que en el caso del Módem. Las velocidades son muy elevadas, esta tecnología también resulta muy cara en contrapartida, llegando a 10 Gbps de máximo teórico con 1 Gbps habituales. Con respecto al usuario y en Galicia, el ancho de banda es mucho menor, los proveedores más habituales acostumbran a proporcionar velocidades similares a las de la ADSL.
5. **Satélite.** La conexión vía satélite se emplea en emplazamientos con poca infraestructura donde no es posible aplicar las tecnologías anteriores, como ADSL o Cable. En Galicia se recurre a este tipo de

tecnologías en zonas del entorno rural o zonas de alta montaña. Esta tecnología tiene un coste muy alto, pero presenta una amplia cobertura. El hardware necesario requiere la instalación de una antena parabólica y en la oferta habitual de los ISP proporcionan 2 Mbps de subida y bajada.

6. **Módem Móvil.** Las últimas tecnologías desarrolladas para teléfonos móviles como GSM, GPRS, o UTMS/3G permiten que los operadores ofrezcan a los usuarios servicios de Internet bien directamente desde **el dispositivo móvil** o bien conectando otro equipo de trabajo a la red a través del mismo. Emplean un protocolo específico denominado WAP (en inglés *Wireless Application Protocol*) y las velocidades de conexión varían dependiendo de la tecnología de 56 Kbps a 2 Mbps con las tecnologías de última generación. El hardware básico es un teléfono móvil que soporte estas tecnologías, pudiendo precisar algún elemento de conexión extra para conectarlo con otros equipos de trabajo.
7. **PLC.** (Del inglés *Power Line Communication*) Esta tecnología ofrece conexión a Internet a través de la red eléctrica. Como la ADSL esta tecnología aprovecha una infraestructura de cableado ya existente para ampliar los canales empleando medias y altas frecuencias. Requiere hardware específico, los denominados **Módem PLC**. Consigue velocidades de hasta 134 Mbps, y a pesar de que el ancho de banda es incluso superior al de la ADSL y la infraestructura de cableado eléctrico puede ser superior al telefónico, en Galicia el uso de esta tecnología está muy poco extendido.

### **25.2.3. SERVICIOS DE INTERNET**

El fin último de acceder la Internet o a otra red es hacer uso de los

**servicios** que se encuentran en ella, y que veremos a continuación:

## **1. WWW**

En el caso de Internet el servicio más empleado es la Red global mundial o WWW (siglas en inglés de *World Wide Web*), se trata de un sistema de publicación e intercambio de información distribuido que relaciona unos contenidos con otros a través de vínculos. Este sistema se extendió rápidamente gracias a su facilidad de uso.

En este contexto surge el **Hipervínculos**, que viene siendo un texto u otro objeto que contiene un vínculo en el que clicando se accede a otra información emplazada en otra zona del documento o en otro documento distinto. Esta funcionalidad permite relacionar unos documentos con otros, o lo que es lo mismo unos nodos con otros formando un red denominada “tela de araña” (en inglés *web*), de ahí que cada documento haya pasado a denominarse página web. Cuando se trata de texto los vínculos suelen aparecer resaltados en color azul y subrayados, e incluso puede cambiar el estilo del puntero del ratón para que no pasen desapercibidos.

Los documentos denominados páginas web, son documentos en lenguajes estándar como HTML o XML que pueden incluir diferentes tipos de información: texto, hipervínculos, gráficos y otros elementos multimedia. Estas páginas web se alojan en servidores web distribuidos por todo el mundo en lo que se conoce como sitios **web**. Cuando el servidor se encuentre conectado a la red la conjunción de direcciones IP y nombres de dominio permitirá acceder a los documentos del sitio y visualizarlos mediante unos programas denominados **navegadores** de Internet. Este tipo de programas implementan el protocolo HTTP que funciona sobre la familia de protocolos TCP/IP encargándose de gestionar la comunicación entre el cliente y el servidor web. Para acceder a la dirección de una página



web podemos hacerlo tanto mediante vínculos como directamente desde la barra de direcciones del navegador sin más que insertar directamente en ella el nombre o dirección web del sitio.

Las direcciones web sirven para identificar los recursos de la red, y se denominan **URL** (en inglés *Uniform Resource Locator*) o localizador uniforme de recurso. Las URL pueden ser de la forma: <https://www.xunta.es:80/ruta/index.htm> teniendo los siguientes componentes:

- a) El protocolo de la red que se emplea para recuperar la información del recurso especificado, en este caso 'https' siendo uno de los más habituales junto a 'http', 'ftp', 'mailto', 'file', o 'ldap'. Normalmente el protocolo HTTP es opcional en la mayoría de los navegadores ya que se trata del protocolo más utilizado.
- b) El nombre de dominio, o servidor con el que se comunica, en este caso '[www.xunta.es](http://www.xunta.es)'.
- c) El puerto de comunicación que emplea ese protocolo en el servidor, en este caso ':80', siendo éste opcional pues los protocolos acostumbran a llevar un puerto asociado por defecto.
- d) La ruta del recurso en el servidor, (en inglés *path*), en este caso '/ruta'.
- e) El nombre del archivo alojado en esa ruta o directorio, en este caso '*index.htm*'.
- f) Otros campos como parámetros o propiedades propias de determinados protocolos.

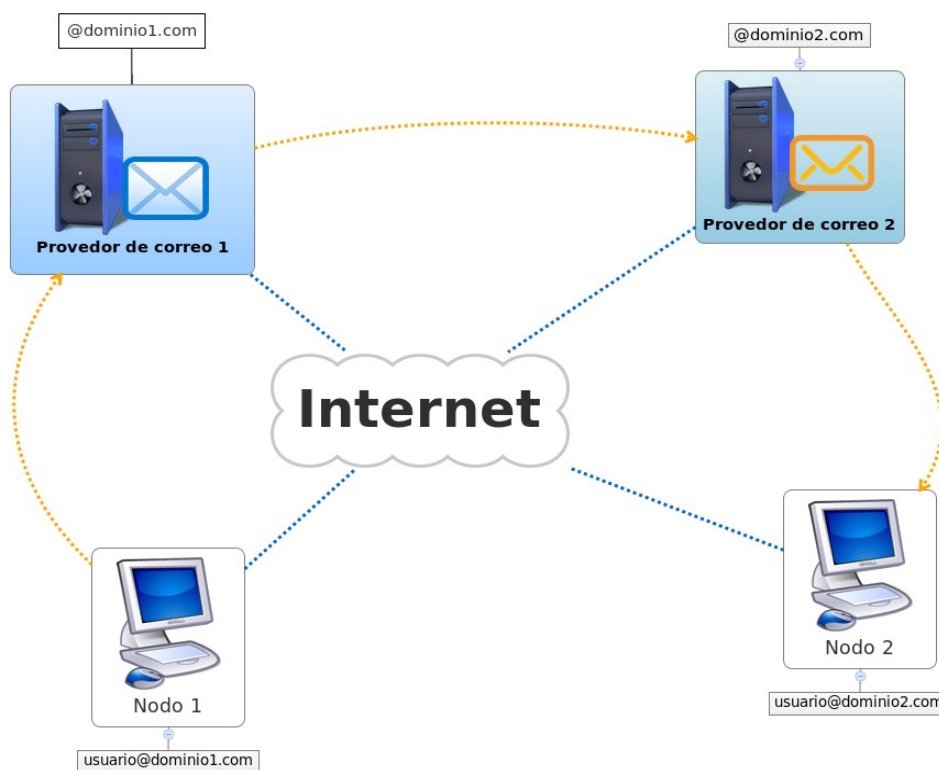
## **2. Correo electrónico**

El servicio de correo **electrónico** (e-Correo) proporciona los mecanismos para facilitar el envío y recepción de mensajes que pueden incluir texto y otras aportaciones a modo de archivos multimedia. En este servicio se

identifica cada usuario con una cuenta que llevará su nombre de usuario para el dominio de ese servidor de correo seguido del símbolo '@' (arroba) y el nombre de dominio (DNS) de ese servidor. Como por ejemplo: [usuario@xunta.es](mailto:usuario@xunta.es). El acceso al correo electrónico puede hacerse con dos sistemas diferentes, o bien accediendo con un correo web o bien con un cliente de correo electrónico. En el **correo web** (en inglés *webmail*) se accede desde un navegador a una página de administración del correo, que requiere la autenticación del usuario y permite hacer las operaciones como en cualquier otro sitio web. En este caso es idéntico a cualquier otro servicio www, es decir emplea los protocolos HTTP o HTTPS según el nivel de seguridad del servidor. No requiere software adicional y cualquier equipo que haya instalado un navegador permitirá acceder a un servidor de correo remoto.

La alternativa es emplear software a modo de clientes de correo electrónico específicos que permiten conectar un software de gestión de correo con el servidor a través de los protocolos de correo **POP3** o **IMAP**. Estos dos protocolos permiten obtener y enviar mensajes de correo desde y hacia un servidor remoto. La diferencia entre ambos protocolos es que POP3 se encuentra más orientado hacia recepción de correo que para el envío, con lo cual al conectarse descarga todos los mensajes al equipo cliente y los elimina del servidor, mientras que el protocolo IMAP los mantiene. En líneas generales IMAP proporciona más funcionalidades que POP3, siendo un poco más complejo por lo que no se encuentra implantado en todos los servidores de correo. Para especificar cómo deben encaminarse los correos se emplean los **Registros MX** (en inglés *Mail eXchange Record*), recursos DNS que indican los servidores de correo por prioridad. El MTA (en inglés *Mail Transfer Agent*) solicita el Registro MX ante una petición DNS encaminando posteriormente el envío. Existen muchos riesgos de seguridad asociados a los correos, además de la posibilidad de envío de virus, Hoax o troyanos, algunos servidores permiten el envío abierto u Open

Relay.



**Figura 3: Funcionamiento del correo electrónico.**

TRADUCCIÓN FIGURA 3: Proveedor de correo 1 y 2.

### **3. Transferencia de archivos (FTP)**

El servicio de transferencia de archivos o FTP (en inglés *File Transfer Protocol*) es un protocolo que define los estándares para el servicio de transferencia de archivos a través de Internet. Se trata de un sistema cliente-servidor al estilo de los anteriormente comentados donde un equipo cliente se puede conectar con un servidor de archivos remoto para descargar o enviar uno o más ficheros independientemente del sistema operativo del equipo cliente. Como sucedía con el correo electrónico puede gestionarse desde un navegador empleando el servicio www, o bien con un cliente FTP que haga transparentes y usables las diferentes funcionalidades del servicio. Una cuenta de usuario especial es la que tiene como usuario y contraseña 'anonymous' que se emplea para acceder a servidores FTP anónimos o públicos, se trata de un estándar de facto para permitir

acceder a cualquier persona a los contenidos de un directorio público de un servidor FTP. Ampliaciones de este protocolo en el campo de la seguridad dan lugar a la evolución a SCP (en inglés *Secure Copy*) y SFTP (en inglés *SSH File Transfer Protocol*) ambos añaden la seguridad **SSH** (en inglés *Secure Shell*) en el primero limitado a transferencia de archivos y en el segundo con más opciones.

#### **4. Conexión o acceso remoto (Telnet)**

Este servicio permite el acceso remoto a otro equipo a través de la red y trabajar con ella desde nuestro equipo a través de una consola como si estuviésemos conectados directamente a ella, como un usuario de esa máquina. **Telnet** es el protocolo de red que permite realizar este tipo de comunicaciones, que precisan que en el servidor remoto esté activado el servicio de Telnet para aceptar las comunicaciones. Requiere una cuenta de usuario y contraseña para el servidor de Telnet, que en muchos casos puede coincidir con un usuario del equipo remoto. Los problemas de seguridad de las versiones iniciales del Telnet se arreglaron con su evolución a SSH una nueva versión del sistema con técnicas de cifrado y con nuevas funcionalidades. Como ocurría con el Telnet, SSH es tanto el nombre del protocolo como el del programa que lo implementa, y como sucedía con el FTP y con el correo electrónico existe software de gestión que facilita al usuario la conexión vía Telnet o SSH. La posibilidad de estar en una computadora mientras se trabaja en otra resulta muy útil para tareas administrativas, sobre todo para los administradores de red o para situaciones de teletrabajo.

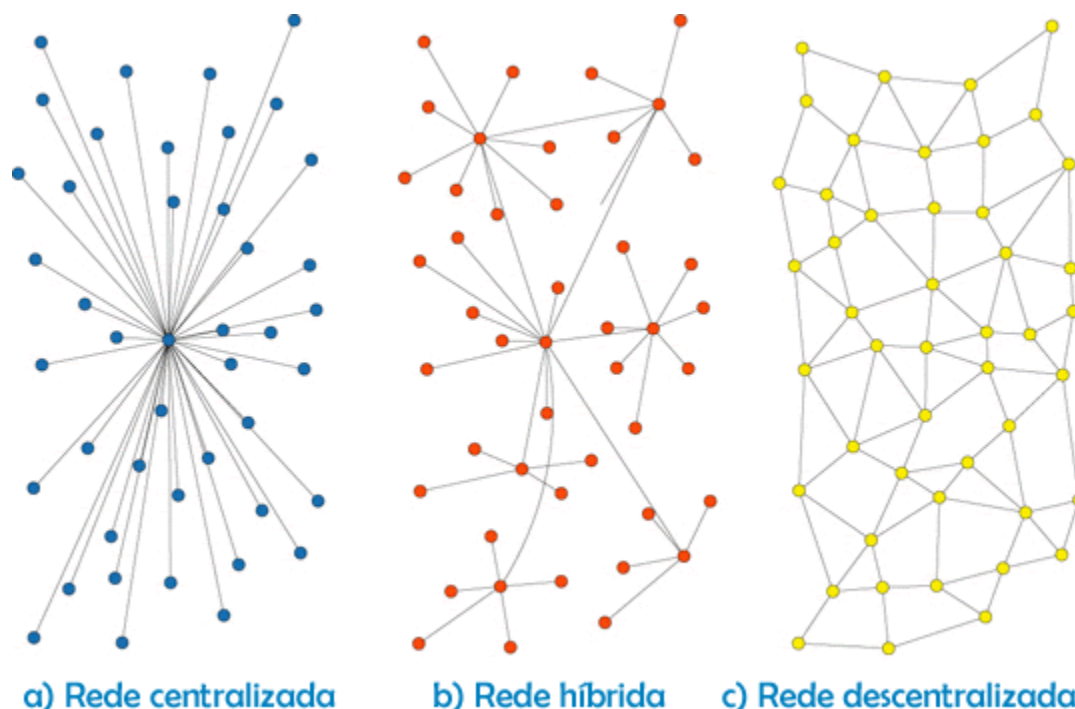
Un paso más allá, los **terminales en modo gráfico** permiten además de texto mostrar imágenes, con lo cual accederíamos desde nuestro equipo a un escritorio idéntico, cómo lo veríamos si nos encontrásemos físicamente

en el equipo remoto. Los clientes de este servicio emplean los protocolos RDP o X11 según el sistema operativo, para sistemas Windows y Unix/Linux respectivamente.

## **5. P2P**

Los servicios P2P tienen su origen en el concepto de las redes entre iguales (en inglés *peer-to-peer*). La característica principal de este tipo de redes es que todos los nodos que participan en la red tienen el mismo peso en la misma, todos actúan como clientes y como servidores. Se trata de subredes dentro de la Internet establecidas a partir de un determinado software de gestión para P2P. En un principio estas redes podían tener nodos centrales para gestionar las comunicaciones si bien la base del intercambio de archivos seguía siendo distribuida. Cada nodo, equivalente a un usuario conectado a la red P2P, comparte sus recursos con todos los demás nodos, si bien la manera más habitual es la de compartir archivos, en ocasiones permiten realizar cálculos de coste elevado o procesamientos de datos masivos con orientación científica. Según dispongan de nodos centrales podemos hablar de los siguientes tipos de redes:

- a) Redes P2P centralizadas, en forma de estrella con un servidor central que monopoliza la gestión y administración de la red.
- b) Redes P2P híbridas, donde además del nodo central existen nodos de segundo nivel que centralizan la gestión de subredes.
- c) Redes P2P descentralizadas, donde todos los nodos son clientes y servidores con el mismo peso.

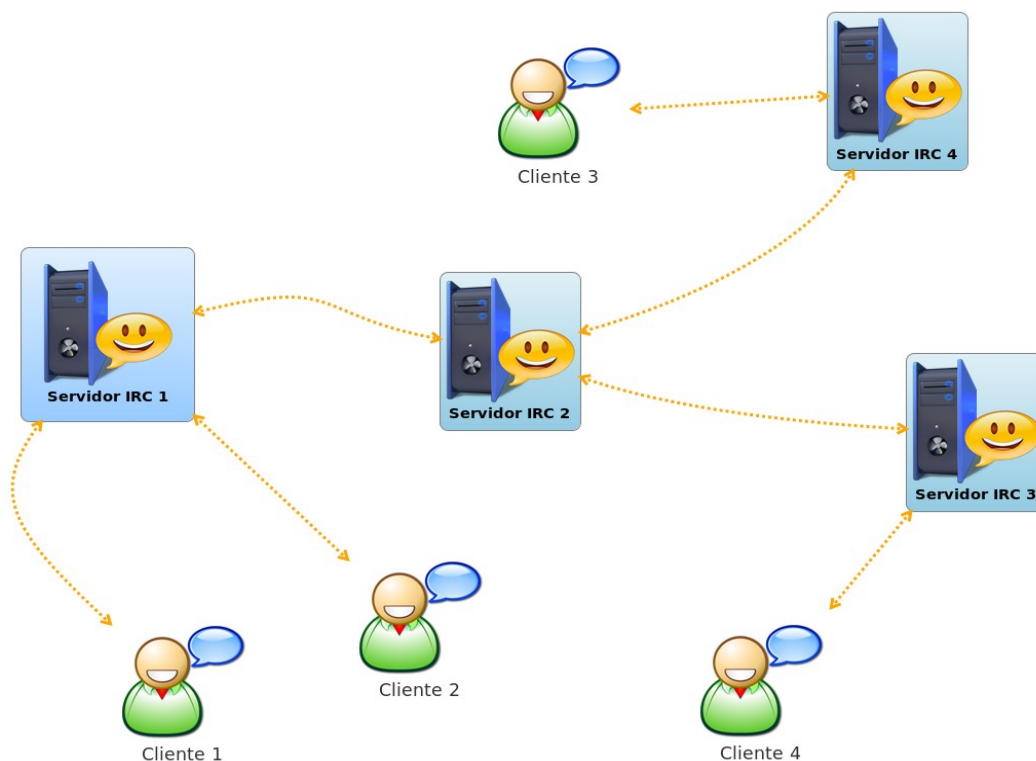


***Figura 4: Topologías habituales de las redes P2P.***

***FIGURA 4: Topologías habituales de las redes P2P***

## 6. Conversación (Chat)

Este servicio permite que dos o más usuarios conectados simultáneamente a Internet sostengan conversaciones interactivas en tiempo real. El IRC (en inglés *Internet Relay Chat*) es el protocolo de comunicación basado en texto que sustenta el servicio. Las conversaciones tienen lugar en los denominados canales de IRC de manera que cada canal puede sostener una conversación paralela entre dos o más nodos cualesquiera de la red. Existen múltiples clientes que como ocurría con los servicios anteriores facilitan el uso del servicio a los usuarios. En sus orígenes permitía sólo el envío de mensajes de texto, pero evolucionó hasta permitir el envío de archivos, transmisión de voz y video e incluso conexión de escritorio remota.



**Figura 5: Ejemplo de una red IRC.**

#### **25.2.4. MOTORES DE BÚSQUEDA**

Otra característica de Internet fruto de la gran cantidad de información que almacena sería la existencia de unas herramientas denominadas **Motores de búsqueda** (en inglés *browser*). Estas herramientas buscan los archivos almacenados en los servidores web y los indexan para poder proporcionar resultados de búsquedas de palabras llave en los mismos en un tiempo excelente. Los motores de búsqueda emplean un **robot** (o simplemente *bot*) que hace las funciones de rastreador de la web. Periódicamente este robot recoge información sobre los sitios y páginas web que recorre desde un punto de partida a los vínculos de cada documento que recorre. De este modo puede descubrir nuevos documentos en un sitio siempre y cuando estén vinculados desde otros documentos ya encontrados del sitio. En determinadas ocasiones podemos no desear que el documento sea incorporado a los buscadores, por lo que podremos mediante código HTML

indicarle al robot que salte ese documento. La información que recoge un robot incluye el texto y parte del código de la página web, no pudiendo interpretar imágenes, animaciones o videos no siendo a través de su descripción. Para que un sitio web pase a existir hace falta darlo de alta en por lo menos uno de los buscadores, siendo suficiente con incluir la página principal del sitio siempre y cuando el resto de páginas estén ligadas desde ella.

A partir de la información recogida por el robot se elabora un **índice** o catálogo de documentos orientado a facilitar la búsqueda de información. Con cada nueva búsqueda la información de rastreo deberá actualizarse y consecuentemente también el índice o catálogo, incorporando las nuevas páginas descubiertas, eliminando las que fueron borradas así como los cambios de cada documento.

De cara al usuario el motor de búsqueda proporcionará una **interfaz de búsqueda** vía web o cliente software donde a partir de un término insertado le dará como resultado los vínculos encontrados que mejor se correspondan por orden de importancia. Los factores llave del resultado de un buscador serán por tanto el tiempo de respuesta, optimizado gracias al índice y la importancia o adaptación de los resultados a los términos empleados en la búsqueda.

Por norma general los buscadores implementan sus propios algoritmos de importancia o posicionamiento, que establecen un peso para cada página en función del número de visitas, número de páginas que la enlazan, aspectos comerciales, valoración de los usuarios y un largo etcétera. En el resultado de una búsqueda aparecerán primero las páginas que tengan un mayor posicionamiento o importancia.

### **25.3. INTRANET/EXTRANET**



En líneas generales una Intranet se comporta igual que Internet, siendo una Internet limitada al ámbito de la organización para la que da servicio, es decir una Internet personal. Una Intranet sería una Internet que restringe el acceso a los sistemas de información. A efectos de alcance y servicios podremos disponer de las mismas posibilidades en cada tipo de red. En lo relativo a su funcionamiento también es idéntica a la de Internet, cada equipo o nodo también dispondrá de una dirección IP, pero en este caso no se corresponderá con las direcciones IP de Internet sino que será una dirección IP personal, para uso interno. Si parte de los equipos se encuentran abiertos a Internet pasaremos a hablar de Extranet, pudiendo convivir ambas en la misma organización.

En otra variante una Extranet puede comunicar dos Intranets con distinta localización geográfica estableciendo por ejemplo una Red personal virtual o VPN (en inglés *Virtual Private Network*) que define una red personal lógica sobre una red pública. Existen varias arquitecturas de VPN:

- 1) **VPN de acceso remoto.** Conecta directamente los usuarios a la red a través de Internet siempre que el usuario se autentifique de manera correcta.
- 2) **VPN punto a punto (Tunneling).** Requiere un servidor VPN que responde a las conexiones a través de Internet y crea un túnel VPN, que consiste en enmascarar un protocolo de red sobre otro. De este modo se pueden transmitir los paquetes con protocolos cifrados como SSH.
- 3) **VPN LAN.** En esta solución no se emplea Internet para el acceso remoto sino que se hace sobre la propia red de la organización. En redes sin hilos, permite establecer un nivel de seguridad añadido donde además de los protocolos de seguridad de la Wi-Fi se incluyen las credenciales de seguridad del túnel VPN.

Particularizando y concretando los servicios que ofrece Internet podemos definir una serie de servicios básicos que puede proporcionar una Intranet/Extranet:

#### **a) Acceso a sistemas de información**

- ✓ Acceso a documentación: manuales, publicaciones, guías y formularios internos.
- ✓ Acceso a sistemas de información y bases de datos corporativas.
- ✓ Consulta y edición de informes, formularios y listas.
- ✓ Agenda, calendarios y planificaciones de trabajo en grupo.
- ✓ Acceso a información de contacto de la organización.
- ✓ Páginas de noticias y vínculos de interés.

#### **b) Recursos compartidos**

- ✓ Acceso a recursos compartidos: conexión a Internet, impresoras, escáneres, etc.
- ✓ Acceso a sistemas de intercambio de archivos.
- ✓ Buscadores de recursos e información.

#### **c) Flujos de trabajo**

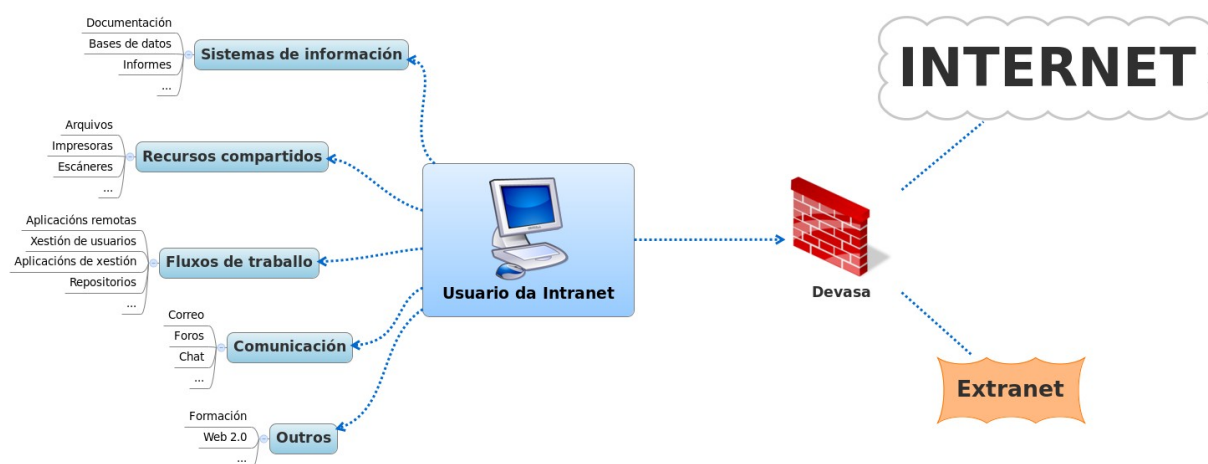
- ✓ Gestión de usuarios y perfiles.
- ✓ Acceso a aplicaciones/equipos remotos.
- ✓ Acceso a repositorios de versiones.
- ✓ Acceso a aplicaciones de gestión y control de incidentes.
- ✓ Soporte a trabajadores móviles o tele-trabajadores.

#### **d) Comunicación**

- ✓ Servicios de mensajería interna, correo electrónico, foros y videoconferencia.

### y) Otros

- ✓ Realización de actividades de formación.
- ✓ Acceso a herramientas de la Web 2.0: portales, blogs, wikis, redes sociales, etc.



**Figura 6: Servicios básicos de una Intranet/Extranet.**

TRADUCCIÓN FIGURA 6: Archivos. Gestión de usuarios. Aplicaciones de gestión. Flujos de trabajo. Cortafuegos. Otros

Para poder soportar esta larga lista de servicios una Intranet debería estar dotada de los siguientes **componentes** básicos:

### 1. Soporte e infraestructura de red.

El modelo más sencillo de Intranet sería dos equipos de una organización conectados en red. A partir de ahí la red puede crecer tanto como requiera la organización, pudiendo incluir cualquier número de redes, subredes, equipos y elementos de interconexión. La implantación equivaldría a la de una red LAN, siendo preciso definir una topología, un conjunto de tecnologías (Ethernet, Fibra óptica, Wi-Fi, etc.), dispositivos de interconexión y de seguridad y una política de asignación de direcciones IP

y nombres de dominio DNS. En un paso más allá habría que extender la red hacia el exterior en caso de que se quiera definir parte de la misma como Extranet, teniendo en cuenta también este conjunto de características. A este respecto hay que tener en consideración el Plan de Direccionamiento e Interconexión de Redes de Área Local en la Administración (2010), el cual establece los rangos de asignación de direcciones IP. Para la Xunta de Galicia estable el siguiente **rango**:

<b>10.179.0.0 - 10.180.0.0</b>	<i>Xunta de Galicia</i>
------------------------------------	-------------------------

Así como otras **recomendaciones**, tales como:

- ✓ Cada entidad u organismo puede gestionar independientemente sus planes de numeración IP pero siguiendo el plan para evitar direcciones duplicadas con los otros organismos.
- ✓ Emplear máscaras de red de 24 bits, para tener redes de 254 nodos por segmento, con lo cual tendríamos la máscara 255.255.255.0 independientemente de la situación física del nodo.
- ✓ La asignación del grupo de bits para el *Host* se realizará de manera ascendente para permitir subredes en las zonas aun no asignadas.
- ✓ Emplear valores de direcciones IP bajos para servidores y equipos de comunicaciones.
- ✓ Valores por encima de los anteriores para equipos de usuarios, ordenadores personales y estaciones de trabajo.
- ✓ Seguir el plan de numeración en cada subred.
- ✓ Mantener al día la documentación de los cambios que se producen en el mismo.

## **2. Servidores.**

Los servidores serán los proveedores de servicios en la Intranet/Extranet encontrándonos diferentes requerimientos hardware y software según la función que van a desempeñar. Según su perfil dentro de la red encontramos dos tipos:

- a) **Servidores dedicados.** Invierten toda su potencia en dar servicio a la red, dedicando todos sus recursos a tal función.
- b) **Servidores no dedicados.** Funcionan tanto como servidor como estación de trabajo, repartiendo sus recursos en las dos funciones.

Por otra parte, atendiendo al tipo de servicio que proporcionan, podrían clasificarse de la siguiente manera:

- 1) **Servidores de archivos.** Almacenan archivos y directorios y gestionan el acceso a los mismos por parte de los usuarios de la Intranet. En sistemas avanzados proporcionan información de versiones, permisos y servicios de transferencia, sincronización, replicación y soporte de protocolos SMB/NetBIOS, CIFS, NFS y FTP así como funcionalidades de integración del estilo de Samba.
- 2) **Servidores de impresión.** Controlan las impresoras, fax o escáneres en red, realizando tareas de gestión de colas, asignación de prioridades y detección de errores, con soporte para protocolos IPX/SPX, LDP, IIP, CUPS o vía Socket.
- 3) **Servidores de comunicaciones.** Realizan la gestión de las comunicaciones de telefonía, voz sobre IP (VoIP) o videoconferencia. Sistemas avanzados incluyen contestadores automáticos y sistemas de respuesta robótica paralela automática. Deben soportar diferentes protocolos como TCP/IP, IPX, PPP, SLIP/CSLIP, SNMP, LAT o NetBEUI .
- 4) **Servidores de correo.** Almacenamiento y gestión de mensajes exclusivo para usuarios de la Intranet/Extranet, con soporte para SMTP, IMAP, POP3 y seguridad SSL/TLS.
- 5) **Servidores de mensajería instantánea.** Gestionan las

comunicaciones de Chat o conversación instantánea entre los usuarios de la Intranet/Extranet con soporte IRC, MUC, SIMPLE, MNP o XMPP.

- 6) **Servidores de red.** Realizan funciones de interconexión de las redes y subredes que forman la Intranet/Extranet. Gestiones de cachés (en inglés *proxy*), encaminamiento, servicios de cortafuegos (en inglés *firewall*), NAT, DHCP, etc.
- 7) **Servidores de acceso remoto.** Gestionan la conexión remota de equipos desde otras localizaciones con protocolos XDMCP, NX, RFB o RDP. Optimizan la elevada carga del uso de aplicaciones y escritorios de manera remota e incorporan mecanismos de autenticación avanzados.
- 8) **Servidores de aplicaciones.** Permite que los clientes trabajen con aplicaciones de coste de implantación/configuración elevado o con una alta demanda de recursos de manera remota. Las soluciones más habituales se basan en las plataformas JEE, .NET, PHP y Coldfusion.
- 9) **Servidores de copias de seguridad.** Permiten mantener un sistema de control de almacenamiento de copias de seguridad de datos o servidores en discos duros redundantes o cintas, en ocasiones en otras localizaciones pero dedicados o SAN. El objetivo de estos sistemas es restaurar el sistema a un estado funcional y seguro después de un error, caída o desastre que provoque la pérdida de la funcionalidad de la red, convirtiendo las redes locales en NAS. Actualmente con el avance de las conexiones van ganando fuerza los *backups* en la nube.
- 10) **Servidores de Bases de datos.** Proporcionan los servicios de acceso a las Bases de datos así como la gestión de las mismas desde ordenadores con más recursos que las estaciones de trabajo. Resulta habitual su comunicación con otros servidores para proporcionar servicios conjuntos. Algunos de los ejemplos más representativos son Oracle, DB2, SQL Server, MySQL y PostgreSQL.

- 11) **Servidores web.** Soportan el servicio de contenidos web a nivel interno controlando el acceso a las páginas y documentos HTML y XML. Los dos ejemplos más representativos son Apache e IIS.
- 12) **Otros.** Cualquier otro servicio de importancia para la Intranet/Extranet debería tener un servidor dedicado especializado que destinase todos sus recursos a la gestión y soporte de ese servicio. Algunos de estos servicios podrían ser el control y gestión de usuarios (Servidores LDAP), servidores de informes, control de versiones, etc.

### **3. Control de Seguridad**

En todas las redes y sistemas de comunicación es importante dedicar recursos al control de la seguridad, principalmente en las partes visibles desde fuera, es decir las partes de la Extranet, pero tampoco hay que olvidar las partes propias de la Intranet. La mayor parte de la seguridad recae sobre los **cortafuegos**, que filtran las comunicaciones con el exterior, restringen aplicaciones y controlan las direcciones IP y físicos de las máquinas según una serie de reglas y filtros de control. Asimismo disponen de herramientas de monitorización y registro que permiten hacer seguimientos y auditorías de la red.

Por otra parte, se puede restringir la comunicación con el exterior empleando equipos de interconexión puente que den servicio al resto de la red. Estos dispositivos de **gestión de caché**, hacen la función de repetidores en la red, pero aíslan a los equipos internos y permiten centralizar y reforzar la seguridad y el control en este equipo, en lugar de en toda la red. Los equipos pasarela deberían incluir todos los servicios básicos como Web, FTP o mensajería instantánea.

La seguridad debe contemplarse también en los **clientes**, aunque una correcta gestión en los servidores protege por extensión a los equipos de trabajo. Hace falta prestar especial atención al control de los usuarios de cada equipo, controlar accesos físicos, gestión de contraseñas, y dotarlos de un software antivirus idóneo. En ocasiones puede ser necesario controlar el acceso de los usuarios al mismo equipo distinguiendo en diferentes perfiles usuario/administrador o más según las necesidades de la organización.

#### **4. Administración de la red.**

El papel de administrador de la red resulta fundamental para asegurar el correcto funcionamiento y seguridad del sistema, además de para dar soporte y participar de la resolución de incidentes.

Mención especial merece el control de las comunicaciones que se realizan entre los usuarios, teniendo especial cuidado con temas como correos masivos o SPAM, envío masivo o no autorizado fuera de la organización. Entre las labores o funciones del administrador o administradores encontraríamos:

- ✓ Establecimiento y mantenimiento de políticas de gestión de usuarios y papeles, permisos y accesos.
- ✓ Mantenimiento y soporte físico del hardware de la red.
- ✓ Configuración y mantenimiento de cortafuegos, antivirus y cachés, así como cualquier otro equipamiento o software de conexión de la red.
- ✓ Evaluación de la calidad del servicio.
- ✓ Realización de auditorías periódicas de control y evaluación de la seguridad y rendimiento.
- ✓ Atención a los usuarios, soporte y resolución de incidentes.
- ✓ Documentación del diseño y descripción de la red, configuraciones de



servidores y protocolos de restauración/recuperación de la red en caso de error o desastre.

## **25.4. IMPLANTACIÓN DE REDES EN ORGANIZACIONES**

Cuando las organizaciones conectan su red y servicios con Internet tienen varias alternativas:

**1) Integrar por completo la red corporativa en Internet.** De este modo cada equipo de la organización pasa a ser un nodo de Internet, con direcciones IP de Internet. Desde cualquier localización se podrá tener acceso a los servicios y equipos de la red directamente, sin limitaciones. Esta solución plantea riesgos de seguridad, ya que el conjunto de la red queda expuesto a ataques desde el exterior, y cada nodo se convierte en un potencial punto débil.

**2) Integrar parcialmente la red y equipos.** En este tipo de soluciones la mayoría de la red aparece oculta al exterior, fuera de Internet, para evitar los riesgos de seguridad. Desde el exterior se pueden ver algunos servidores de la organización y del mismo modo desde la red se puede acceder a servidores externos, pero restringidos los servicios y comunicaciones. Cuando la integración es parcial se puede hablar de redes de los tipos Intranet y Extranet. Partiendo de esto surgen otras muchas cuestiones, número de usuarios, distribuciones físicas que abarcará la red, servicios que se implantarán, y un largo etcétera. Resulta obvio que el primer paso de la implantación de una Intranet/Extranet será la planificación. En la planificación se abordarán los siguientes puntos:

### **1. Planteamiento de objetivos.**

Los **objetivos** de la red quedarán definidos por su alcance. Habría que realizar tareas tales como:

- ✓ Estimación del número de usuarios y su posible evolución.
- ✓ Determinar el emplazamiento de la red y posibles subredes, situación de posibles redes externas y las necesidades de comunicación con las mismas.
- ✓ Considerar los sistemas de información y necesidades de acceso a los mismos, teniendo en cuenta también los flujos y procesos internos.
- ✓ Definir los servicios que se proporcionarán en la Intranet/Extranet detalladamente, con estimaciones de carga, predicciones de su evolución futura, etc.

A partir de los objetivos puede elaborarse la lista de requisitos de la Intranet/Extranet, como paso previo al diseño de la red. La documentación en ambos puntos debería ser lo más completa posible.

## **2. Selección de tecnologías**

En un segundo paso tomando los objetivos y requisitos habría que seleccionar las tecnologías más adecuadas tanto a nivel de hardware, físico, como de software. A nivel físico acostumbra a optarse por redes Ethernet, pero pueden ser necesarias redes sin hilos, asimismo cada red precisaría diferentes elementos de interconexión dependiendo de la tecnología y lo mismo para clientes y servidores. Desde sistemas operativos a software de gestión y control de cada servicio, gestores de contenidos y otros propósitos, debería seleccionarse atendiendo a las necesidades especificadas por los objetivos y requisitos sin olvidar otros aspectos como costes, la existencia y disponibilidad de soporte para cada tecnología y complejidad de instalación, configuración y mantenimiento.

## **3. Definición de los recursos necesarios.**

Una vez seleccionadas las tecnologías que tomarán parte en el diseño de la

selección, habría que definir el número de recursos necesarios para la implantación. Esto incluye, número, tipo y software de los equipos clientes y lo mismo para los equipos de interconexión de la red y servidores. Este paso sería el **diseño** de la red en sí, desde el cableado a la lista completa de software necesario. Según las particularidades de la Intranet/Extranet podría ser preciso el desarrollo de software a medida, lo cual habría que incluir también en esta fase del diseño.

#### **4. Definición de políticas de seguridad.**

Paralelamente habrá que establecer y documentar los protocolos y políticas de seguridad como:

- ✓ La asignación de cuentas de usuario y contraseñas y usuarios de los equipos y servidores, así como caducidad y revisión del cambio de contraseñas en las mismas.
- ✓ Equipos que comunican con el exterior y que precisan más seguridad y equipos que pertenecerán a la DMZ (zona desmilitarizada).
- ✓ Filtros de aplicaciones, de direcciones IP y direcciones físicas.

A continuación vendría la implantación en sí, siendo lo recomendable establecer un período de prueba y realimentación previo para ofrecer un producto de mayor calidad.

#### **1. Período de prueba.**

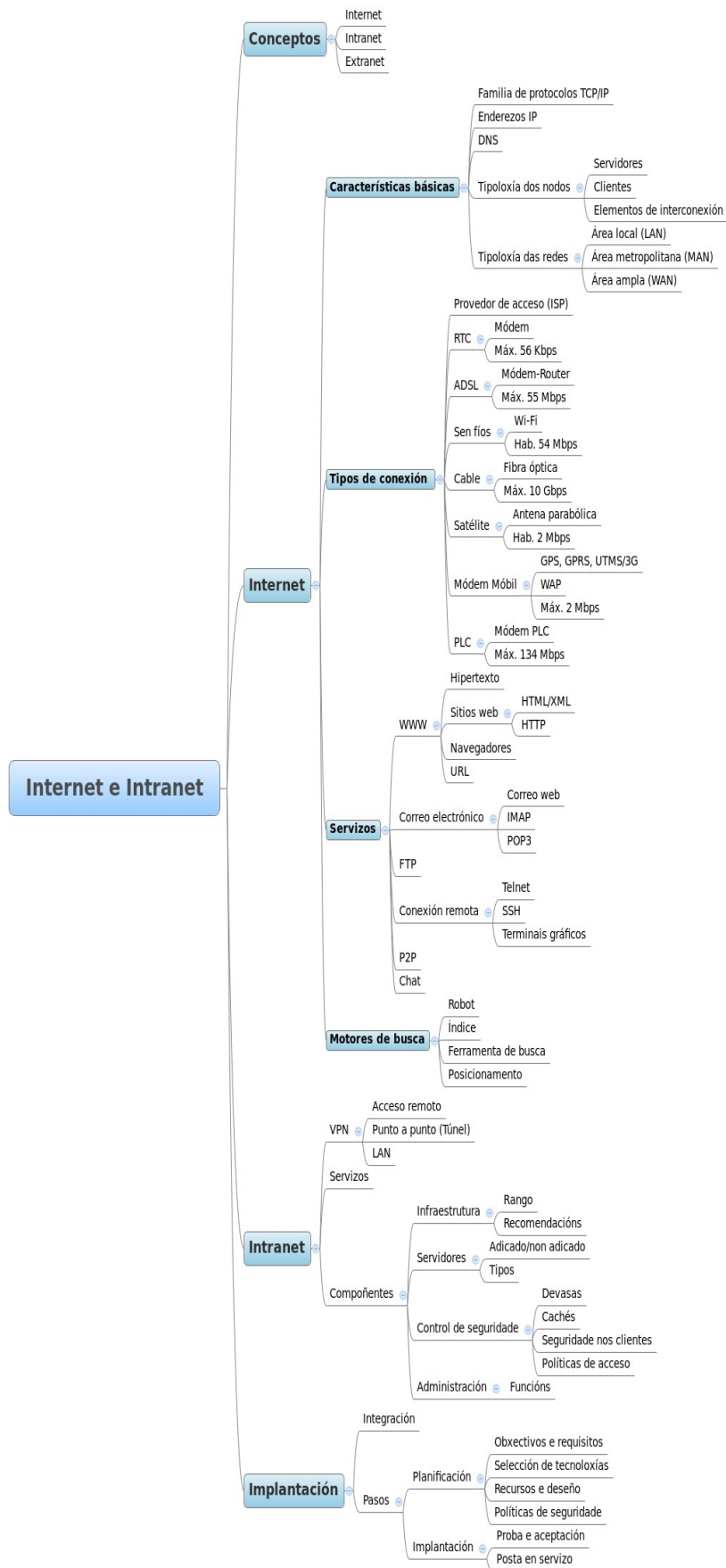
Durante este período se realizarán pruebas completas del funcionamiento de la Intranet/Extranet. Deberían incluirse casos de prueba para los diferentes servicios y las comunicaciones entre las diferentes subredes y redes externas. Con la calidad como objetivo habría que realizar pruebas más complejas como pruebas de carga, buscando los picos de demanda de recursos, y casos de ataques de seguridad controlados. Todo ello mientras se realiza la monitorización y posteriores pruebas de auditoría del sistema

que permite elaborar un informe completo de los límites y deficiencias de la red, útil para detectar puntos débiles que arreglar antes de la puesta en servicio.

## **2. Puesta en servicio.**

Después de las sucesivas pruebas y una vez obtenidos resultados de aceptación puede llevarse a cabo la puesta en servicio definitiva de la Intranet/Extranet, abriéndola a todos los usuarios. En los primeros momentos de la puesta en servicio hace falta realizar la monitorización y auditoría de los sistemas al igual que se hizo durante el período de prueba, pues en este primer momento podrán detectarse problemas y debilidades reales que pudieron pasar desapercibidas durante las pruebas controladas realizadas anteriormente.

### **25.5. ESQUEMA**



**TRADUCCIÓN TEXTO ESQUEMA:** Direcciones IP. Tipología de los nodos. Tipología de las redes. Proveedor de accesos. Servicios. Terminales gráficos. Motores de búsqueda. Herramienta de búsqueda. Posicionamiento. Infraestructura. Recomendaciones. Dedicado. Componentes. Control de Seguridad. Cortafuegos. Objetivos y requisitos. Selección de tecnologías. Recursos y diseño. Políticas de seguridad. Puesta en servicio.

## **25.6. REFERENCIAS**

Abel Rodríguez Ávila.

Iniciación a la red Internet. Concepto, funcionamiento, servicios y aplicaciones de Internet. (2007).

Irene Rodil y Camino Pardo.

Operaciones auxiliares con tecnologías de la información y la comunicación. (2010).

Ministerio de la Presidencia

Plan de direccionamiento e interconexión de redes en la Administración. (2010).

Ralph Stair y George Reynolds.

Principios de Sistemas de información. Enfoque administrativo. (1999).

**Autor: Juan Marcos Filgueira Gomis**

**Asesor Técnico Consellería de Educación e O. U.**

**Colegiado del CPEIG**



## **26. MODELO DE CAPAS: SERVIDORES DE APLICACIONES, SERVIDORES DE DATOS, GRANJAS DE SERVIDORES. SCRIPTS DEL CLIENTE.**

## **TEMA 26. MODELO DE CAPAS: SERVIDORES DE APLICACIONES, SERVIDORES DE DATOS, GRANJAS DE SERVIDORES. SCRIPTS DEL CLIENTE.**

### **26.1. INTRODUCCIÓN Y CONCEPTOS**

### **26.2 MODELO DE CAPAS: SERVIDORES DE APLICACIONES, SERVIDORES DE DATOS, GRANJAS DE SERVIDORES.**

### **26.3 SCRIPTS DEL CLIENTE.**

### **26.4. ESQUEMA**

### **26.5. REFERENCIAS**

### **26.1. INTRODUCCIÓN Y CONCEPTOS**

En una red formada por equipos informáticos los **nodos** acostumbran a realizar tres **funciones** diferenciadas:

1. Facilitar la comunicación e interconexión de los nodos de la red.
2. Proporcionar servicios o información a otros nodos.
3. Realizar funciones de equipo de trabajo, haciendo uso de las comunicaciones, servicios e información disponible.

En este contexto los nodos o equipos que realizan las funciones de proporcionar servicio o información al resto se denominan **Servidores** y los que hacen uso de los servicios **Cientes**, formando en su conjunto lo que se da en denominar **Arquitectura Cliente-Servidor**. Se trata de una de las arquitecturas más extendidas en los entornos distribuidos, permitiendo la heterogeneidad en los clientes y un acceso transparente a la información. Los servidores permanecen a la escucha de la red en todo momento para atender las solicitudes o demandas de los clientes.



El esquema de funcionamiento **básico** seguiría el siguiente modelo:

1. El cliente solicita un servicio al servidor a través de la red.
2. El servidor a la escucha recibe la petición del servicio y la pone en la cola de demanda.
3. El servidor obtiene el resultado de la petición.
4. El servidor envía la respuesta de la petición al cliente a través de la red.
5. El cliente obtiene el resultado y lo procesa.

A partir de estos conceptos básicos podemos extraer las **características básicas** de la arquitectura Cliente-Servidor:

- a) **Servicios.** Son la base de las peticiones entre los clientes y los servidores, se trata de cualquier entidad susceptible de ser demandada por uno o más clientes.
- b) **Recursos compartidos.** Elementos y servicios de la red, tanto lógicos (software, datos e información), como físicos (hardware, impresoras, unidades en red, etc.).
- c) **Comunicación asíncrona basada en el envío de mensajes.** Este tipo de arquitecturas emplean protocolos de comunicación asimétricos donde los clientes inician conversaciones y los servidores esperan que se establezca la comunicación escuchando la red. Toda la comunicación se realiza mediante el envío de mensajes y respuestas.
- d) **Transparencia.** La localización, la organización lógica y física, así como la implementación de los servicios resulta transparente a los clientes. El uso de los mismos se limita a hacer una petición a la red y obtener la respuesta.
- e) **Escalabilidad.** Horizontal en los clientes a la hora de permitir añadir nuevos nodos sin más que añadirlos a la red y vertical en los servidores, de suerte que administrando un único punto puede

mejorarse la potencia, el rendimiento, el mantenimiento y la recuperación de errores.

Fruto de la escalabilidad de esta arquitectura surgen las **granjas de servidores**, consistentes en emplear varios servidores a la vez suministrando el mismo servicio y repartiéndose las peticiones o carga del sistema. La gestión de una granja de servidores será compleja debido a la necesidad de balancear la carga para obtener el mayor rendimiento posible.

Uno de los servicios más extendidos actualmente es el web o WWW, (siglas en inglés de *World Wide Web*), se trata de un sistema de publicación e intercambio de información distribuido que relaciona unos contenidos con otros a través de vínculos. En este servicio los clientes solicitan información a modo de páginas web, tratándose de documentos en lenguajes estándar como HTML o XML que incluyen diferentes tipos de información: texto, hipervínculos, y elementos multimedia. Entre estos elementos multimedia encontramos:

- a) **Texto.** Distinguiendo entre sin formato, con formato o enriquecido (tipo de letra, tamaño, color, color de fondo, etc.) e hipertexto, texto con un vínculo a otro texto o documento.
- b) **Sonido.** Digitalización del habla, la música u otros sonidos.
- c) **Gráficos.** Representan esquemas, planos, dibujos vectoriales, etc. son documentos que se construyen a partir de una serie de primitivas: puntos, segmentos, elipses, etc. aplicándoles a continuación todo tipo de transformaciones o funciones: giro, cambio de atributos, escalado, efectos, etc.
- d) **Imágenes.** Representaciones fieles de la realidad, como fotografías. Son documentos formados exclusivamente por píxeles, punto a punto y por tanto no se estructuran o dividen en primitivas.

- e) **Animación.** Representación de una secuencia de gráficos por unidad de tiempo, para ofrecer la sensación de movimiento. Asimismo ofrece posibilidades de interacción ante eventos.
- f) **Video.** Representación de una secuencia de imágenes por unidad de tiempo, para ofrecer la sensación de movimiento.

Documentos complejos agrupan diversos componentes multimedia en una misma página o documento. Los sistemas de publicación actuales permiten que la multimedia **digital on line** pueda transmitirse **en flujo** (en inglés *streaming*), que se encuentra disponible tanto on line en tiempo real como bajo demanda. En este modelo no es necesario descargar o acceder a la totalidad del documento para acceder a los contenidos sino que se proporciona acceso directo a cualquier parte del flujo y reproducción desde ese punto.

Otra característica de las páginas web o documentos HTML/XML es que pueden incluir **código de script para los clientes**. Este código representa un guión o secuencia de instrucciones a manera de un programa sencillo. Este programa puede ser interpretado por el navegador del equipo cliente con unos permisos limitados en el equipo y focalizados principalmente en la página o documento web en el que se encuentran incrustados o desde el que son llamados. Existen diferentes tecnologías para estos lenguajes de script siendo las más conocidas: Javascript, Visual Basic Script, Flash, y la evolución de Javascript: AJAX, aceptados con mayor o menor fortuna por los navegadores actuales, muchas de ellas denominadas tecnologías RIA en un acercamiento de las aplicaciones web a las aplicaciones de escritorio.

## **26.2 MODELO DE CAPAS: SERVIDORES DE APLICACIONES,**

## **SERVIDORES DE DATOS, GRANJAS DE SERVIDORES**

La distribución de los sistemas de información fue evolucionando a lo largo del tiempo en función de las demandas y crecimiento de las redes y el aumento de la complejidad de las arquitecturas de red.

### **26.2.1. Arquitectura en una capa: Superordenador central.**

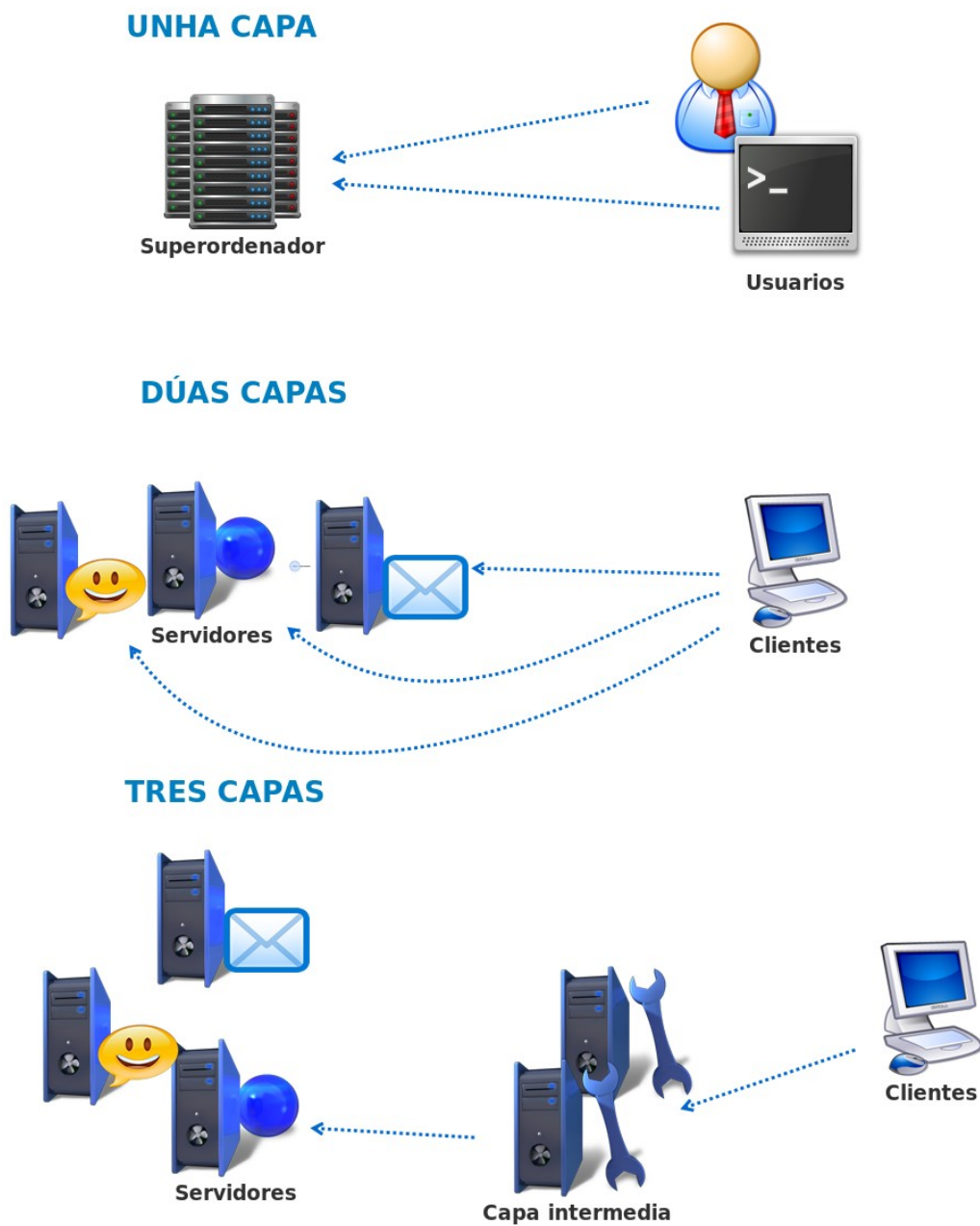
La arquitectura más simple estaría formada por **un superordenador central** (en inglés *mainframe*) que centraliza toda la capacidad de procesamiento y almacenamiento de la red, también denominada monolítica. En este modelo el acceso a la información se hace directamente a través de la computadora principal o bien a través de clientes ligeros que se limitan a hacer las funciones de terminales. En esta arquitectura centraliza todo el coste de administración y mantenimiento se dedica al servidor central. Los terminales carecen de programas propios y tienen recursos de memoria o disco mínimos, pudiendo incluso carecer de disco. Cualquier instalación o avance en el servidor repercute al punto en la red de suerte que cualquier programa instalado estará disponible para todos los clientes. Por contra, si tenemos en cuenta la sostenibilidad del sistema en caso de caída o errores en el servidor central toda la red se ve afectada, al igual que si un cliente sobrecarga el sistema todos los demás se verán afectados en cuanto a rendimiento. A su vez, los mainframes se organizan según arquitecturas paralelas tipo **SNA** (en inglés *Systems Network Architecture*) con un diseño de red con comunicación P2P a través de APPN (en inglés *Advanced Peer-to-Peer Networking*).

### **26.2.2. Arquitectura en dos capas: Modelo Cliente-Servidor.**

En este modelo el sistema se estructura en dos capas, una capa a nivel de usuario que almacena y procesa parte de la información y otra capa remota

a nivel de servicios que almacena y da funcionalidad a la totalidad de los clientes de la red. De este modo se consigue descargar de parte de la carga de la red a los servidores centrales y mantiene la capa de servicios transparente a los usuarios con la posibilidad de escalar el sistema mejorando o aumentando el número de servidores sin que estos lleguen a notar el cambio en algo más que el rendimiento. Por contra, un modelo más distribuido en lo relativo a los clientes obliga a un mayor mantenimiento de los mismos por parte de los administradores. Otro de los puntos a tener en cuenta es la consistencia de los datos entre cliente y servidor, de suerte que hace falta coordinar cada servicio por separado.

En esta línea el uso de protocolos de comunicación soporta el uso efectivo por parte de los clientes de los servicios de la red permitiendo la heterogeneidad de los clientes siempre y cuando los implementen.



**Figura 1: Arquitecturas en capas**

### **26.2.3. Arquitectura en tres capas: Granjas de servidores.**

Debido a los inconvenientes de los sistemas de una única capa, que obligan a mantener un servidor central de tamaño demasiado grande para un mantenimiento y rendimiento eficientes, y de dos capas, que obligan a

mantener cada servidor independiente del resto para un único servicio, se optó por el establecimiento de una capa más entre las de cliente y servidor. En esta capa se agrupan varios servidores en una DMZ soportando el mismo servicio dando lugar a una redundancia que tiene como ventajas una mayor tolerancia a fallos y un avance de rendimiento. A efectos de la red las granjas de servidores proporcionan un único servicio lógico o virtual integrado por cualquier número de servidores físicos. Cada servidor de la granja debe ser una réplica exacta del servidor lógico en cuanto a datos y software instalado. Para escalar el sistema se añade a la granja un nuevo servidor réplica del virtual y aumenta la disponibilidad de recursos. El ejemplo más habitual de granja de servidores es un servicio web, donde si por ejemplo un servidor atiende a mil usuarios y tenemos previstos picos de diez mil usuarios simultáneos pondremos una granja de diez servidores, para atender el servicio y otros dos más en previsión de caída de alguno o picos puntuales aún más altos. La escalabilidad de las granjas en función de los servicios ofertados define tipologías básicas como *Datacenters*, servidores de aplicaciones, de importación, de *front-end* existiendo elementos específicos para control de carga, Teredo o de dominio, entre otros.

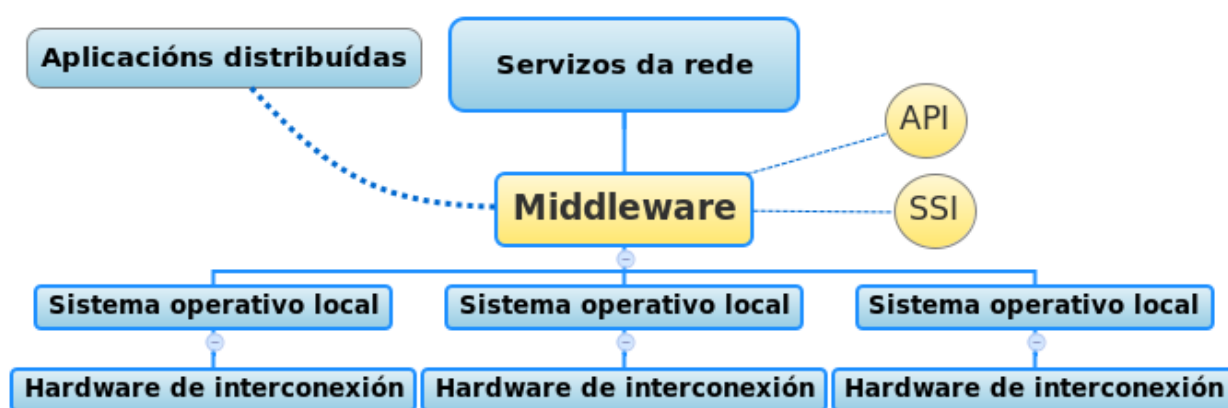
#### **26.2.3.1 Componentes intermedios: *Middleware***

Para dar el efecto de transparencia a los clientes, ese sistema requiere de una serie de componentes intermedios, es decir que se encuentran “por el medio” (en inglés *middleware*) de las capas principales. Estos componentes se encargan de recibir y repartir las peticiones de los clientes entre los servidores de la granja, cuidar el balanceo de carga, el mantenimiento de la sesión, etc. El ***middleware*** se describe como un conductor o intermediario entre sistemas, dirigiendo las peticiones de datos y servicios a otros nodos de la red. Entre sus principales características destacarían:

- a) Simplificar el desarrollo de aplicaciones al capsular comunicaciones

entre sistemas.

- b) Facilitar la interconexión de los sistemas de información con independencia de la red física.
- c) Mejorar la escalabilidad del sistema, aumentando la capacidad sin pérdida de funcionalidad.
- d) Mejorar la tolerancia a fallos del sistema, fiabilidad.
- e) Aumentar la complejidad de administración y soporte.



**Figura 2: El middleware en un sistema distribuido.**

TRADUCCIÓN TEXTO FIGURA 2: Aplicaciones distribuidas. Servicios de la red.

En un sistema distribuido el *middleware* es el software de conectividad que permite disponer de un conjunto de servicios sobre plataformas distribuidas heterogéneas. Actúa como una capa de abstracción de las funciones del sistema distribuido haciendo transparente en la red los sistemas operativos y el hardware de interconexión de las redes de comunicaciones. Proporciona una Interfaz de Programación de Aplicación (**API**) para la comunicación y acceso a aplicaciones y servicios distribuidos. Por otra parte proporciona una interfaz única de acceso al sistema denominada **SSI** (del inglés *Single System Image*), la cual da al cliente la sensación de acceder a un único servidor, el virtual.



Para garantizar la heterogeneidad en la comunicación de los sistemas, el *middleware* se estructura en tres **capas o niveles de comunicación** separados:

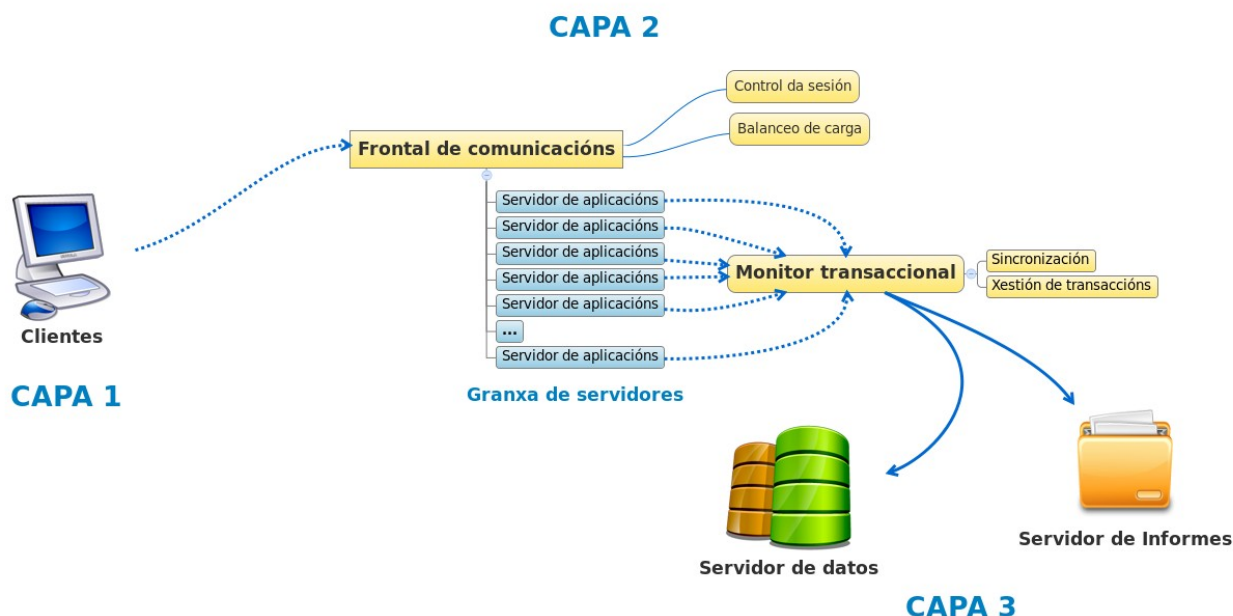
- 1) **Protocolo de transporte.** Protocolos de comunicaciones comunes a la capa de transporte de la red, como TCP o UDP. Establecen niveles de seguridad, control de sesiones, etc.
- 2) **Sistema Operativo en Red o NOS** (en inglés *Network Operating System*). Extensión del sistema operativo de los clientes que captura las peticiones y las dirige hacia el servidor adecuado para devolver a continuación la respuesta del mismo al cliente.
- 3) **Protocolo de servicio.** Protocolo específico del servicio o aplicación en el sistema Cliente-Servidor.

Los *middleware* acostumbran a clasificarse según el tipo de comunicación que realizan en el Sistema Operativo en Red y a los parámetros que comunican (infraestructura, acceso a datos, aplicaciones, etc.), los **tipos de middleware** más habituales serían:

1. **Llamadas a procedimientos remotos** (en inglés *Remote Procedure Call* o RPC). Los Clientes invocan directamente procedimientos o funciones de procesos que se ejecutan en servidores remotos, permitiendo distribuir la lógica de la aplicación remota a través de la red. Las llamadas pueden realizarse de manera asíncrona o síncrona. Mantiene al mínimo la información de la sesión y en caso de ruptura de la misma el cliente reinicia la comunicación de cero.
2. **Publicación/suscripción.** Este *middleware* realiza una monitorización del sistema detectando los servicios y procesos activos. Los componentes registran su interés en determinados eventos, en cuanto estos eventos son detectados por el monitor

envía esa información a los suscriptores. La interacción es asíncrona recayendo por completo en el servicio de notificación/monitorización.

3. **Middleware orientado a mensajes** (en inglés *Message Oriented Middleware* o MOM). La comunicación se basa en el envío de mensajes asíncronos por parte de los nodos (cliente, servidor, servicio o aplicación). Los mensajes se recogen en colas priorizadas en el nodo destino y se almacenan hasta que pueden responderse. El funcionamiento del sistema es análogo a cómo funciona un servicio de correo electrónico.
4. **Middleware basado en objetos** (en inglés *Object Request Broker* u ORB). Incorpora a RPC los paradigmas de orientación a objetos. Define una arquitectura cliente servidor donde los servicios devuelven objetos, siendo estos la unidad de comunicación. Los nodos piden los objetos por el nombre siendo estos entregados por un servicio de resolución de nombres. Ejemplos de implementaciones de este *middleware* serían: CORBA, RMI, COM, .NET Remoting, etc.
5. **Middleware de acceso a datos** (en inglés *Oriented Data Access Middleware*). Proporcionan la API transparente de acceso a datos agrupando las operaciones de manejo de la conexión con las bases de datos. Ejemplos de este tipo de API serían JDBC y ODBC. Por norma general realizan conexiones síncronas y operaciones transaccionales.
6. **Arquitecturas orientadas a servicios** (en inglés *Service Oriented Architecture* o SOA). Las funcionalidades o procedimientos se publican desde cualquier servidor a modo de servicios. Los servidores publican el servicio y permanecen a la escucha hasta que llega una petición, la procesan y devuelven una respuesta al cliente del servicio. Ejemplos de este *middleware* son los Servicios Web y los Servicios CORBA.



**Figura 3: Granja de servidores.**

TRADUCCIÓN TEXTOS FIGURA 3: Servidor de aplicaciones. Gestión de transacciones. Granja de servidores.

Las granjas de servidores se completan con dos componentes fundamentales, funciones que de manera general realiza el *middleware*:

1. **El frontal de comunicaciones.** El frontal de comunicaciones (en inglés *front-end*) es el punto de acceso único a la granja de servidores, simulando un único servidor lógico. Aunque cada servidor de la granja pueda tener su propia dirección IP lo normal es que el frontal tenga uno propio y sea ésta la forma en que los clientes accedan a él, o bien directamente o bien a través de un nombre de dominio (DNS).
- ✓ **Balanceo de carga.** Consiste en dividir la carga de trabajo de los clientes entre los servidores de la granja. Puede implementarse vía hardware, software o una combinación de ambas. Para hacerlo vía hardware el frontal de comunicaciones debe disponer de un

equipo específico, aunque hay enrutadores que permiten esta funcionalidad.

- ✓ **Control de la sesión.** Si debido al balanceo de la carga diferentes peticiones de un mismo cliente van hacia servidores diferentes hace falta coordinar a los servidores en el seguimiento de una sesión o que puedan compartirla.
- ✓ **Priorización.** En caso de tener peticiones simultáneas el frontal debe ser capaz de atender primero a los clientes críticos o de mayor prioridad así como de asignar el procesamiento de sus tareas a aquellos servidores dotados de más recursos.

2. **Los monitores transaccionales.** Los monitores transaccionales son los encargados de mantener en las redes la consistencia de los datos y los procesos que se realizan simultáneamente en los servidores de la granja. Tiene que garantizar que una modificación de datos fruto de una petición se realiza como una transacción, es decir, o se realiza completamente o no se realiza en absoluto. Cada transacción tiene que tener lugar independientemente de que tenga lugar otra simultánea, deben procesarse de manera aislada. Las principales funciones serán:

- ✓ **La gestión de las transacciones.** Se encarga de controlar la atomicidad y secuencialidad de las transacciones, para garantizar la consistencia de datos y operaciones. La gestión de transacciones debe garantizar el correcto funcionamiento del sistema cuando la carga de trabajo o el número de usuarios son muy elevados. Debe contemplar la posibilidad de errores en las aplicaciones y caídas de elementos del sistema durante las transacciones, permitiendo operación de vuelta atrás. Vista detenidamente la gestión de transacciones constará de:

1. **Gestor de transacciones** (en inglés *Transaction Manager*).

Controla el inicio de transacción, registra los recursos que precisa y gestiona las operaciones de confirmación de la transacción (en inglés *commit*) y de vuelta atrás y recuperación del estado inicial de la transacción (en inglés *rollback*).

2. **Gestor de registro** (en inglés *Log Manager*). Guardar los estados de los recursos que están en uso por parte de las transacciones, elaborando un historial de versiones de los mismos. Esta información es compartida por los distintos gestores de transacciones siendo lo que permite garantizar la consistencia de los recursos empleados.
  3. **Gestor de bloqueos** (en inglés *Lock Manager*). Gestiona el acceso simultáneo por parte de varios procesos a los recursos, permitiendo bloquearlos para evitar que dos o más accesos a la vez den lugar a inconsistencias. Asimismo lleva a cabo la detección del momento en que se libera un recurso y envía una notificación al gestor de transacciones.
- ✓ **Sincronización.** La sincronización de las comunicaciones resulta compleja en este modelo, ya que un cliente puede acceder a un servicio empleando diferentes servidores de la capa intermedia, incluso simultáneamente. Según el comportamiento del servicio podemos encontrar soluciones síncronas, donde se espera siempre a la respuesta del servidor, simple pero con riesgo de bloqueo. Frente a las asíncronas donde se envía la petición y ya llegará la respuesta, con lo cual no hay bloqueos, pero la respuesta podría no llegar nunca sin más opción que detectarlo a través de tiempos de espera agotados.

#### **26.2.6. Arquitecturas en n-Capas.**

Las arquitecturas en tres capas pueden extenderse a n-capas cuando en la

capa intermedia se incorporan otros elementos de interconexión como distribuidores o sistemas de cortafuego. También puede dividirse la capa de servidores por servicios o diferentes capas de acceso a datos o presentación de información. La separación en capas es una organización lógica del sistema con el que se puede establecer cualquier número de capas según las necesidades del mismo. Cualquier especialización de servidores que se quiera hacer en la red y provoque un nuevo agrupamiento podemos identificarla con una nueva capa.

#### **26.2.7. Arquitecturas para Red entre iguales (P2P).**

En los modelos distribuidos de igual a igual o P2P (en inglés *Peer-to-Peer*), todos los equipos (excepto los elementos de interconexión) tienen el mismo papel doble de cliente/servidor en la red. Hacen uso de servicios y los proporcionan. En esta arquitectura por tanto no se pueden agrupar los nodos y pierde sentido hablar de capas. Estas redes no resultan óptimas para todo tipo de servicios, en muchos casos, por ejemplo a la hora de funcionar como servidor web, requerirían un coste muy alto para control de la consistencia del sitio web, mantenimiento y configuración del servidor, etc. Por el contrario, en situaciones donde los nodos caen a menudo, por ejemplo un servidor atacado continuamente, la redundancia de nodos garantiza que el sistema siga funcionando. Otros servicios como el intercambio de archivos, o el procesamiento compartido presentan más ventajas a la hora de emplear una arquitectura de este tipo como solución de implantación. Los modelos más habituales son centralizados, puros, descentralizados o híbridos, según el peso de cada nodo individual en la red o de la existencia de servidores con responsabilidad de control y gestión en el modelo. La adaptación de este modelo por parte de los ISP da lugar a P2P híbridas de servicio denominadas P4P (en inglés *Proactive network Provider Participation for P2P*). Otros modelo similares serían los

P2M, que actúan en arquitecturas híbridas empleando el correo electrónico como soporte del envío de datos.

La gestión de este tipo de redes se realiza fundamentalmente vía software. En este caso el software deberá realizar las mismas funciones ya vistas para la arquitectura de tres o más capas: frontal de comunicaciones y gestión de transacciones. Por debajo acostumbran implementar servidores propios como Kademia, eDonkey, Gnutella, FastTrack, BitTorrent u OpenNap entre otros.

#### **26.2.8. SERVIDORES.**

##### **26.2.8.1. Servidor web.**

Se trata de servidores que proveen el servicio WWW a través del protocolo HTTP. En esencia se trata de una aplicación ejecutándose en un servidor a la espera de peticiones HTTP por parte de un cliente respondiendo con los documentos solicitados generalmente páginas web y los objetos que enlazan: imágenes, archivos de script, animaciones, etc. En funciones más avanzadas estos servidores añaden seguridad a través de conexiones encriptadas con protocolos tipo HTTP Seguro o HTTPS. Por regla general, los servidores de aplicaciones se integran en arquitecturas de mínimo tres **capas**:

- 1) **Primera capa.** Capa de interacción con los usuarios, principalmente a través de navegadores web.
- 2) **Capa intermedia.** Capa de los servidores web, que pueden estar distribuidos en un modelo de granja de servidores. Cada servidor incorporaría los módulos necesarios para seguridad, lenguajes de servidor interpretados, correo, mensajería, acceso a datos y otras funcionalidades.
- 3) **Tercera capa.** Capa de servidores de acceso a datos, como

servidores de archivos, base de datos o informes.

Entre los **servidores web de uso más extendido** actualmente se encontrarían:

- ✓ **Apache.** Uno de los más utilizados, por ser un servidor libre que ofrece prestaciones a nivel de otras soluciones propietarias además de una gran facilidad de uso y configuración.
- ✓ **Internet Information Server (IIS).** Servidor propietario con soporte para aplicaciones .NET o ASP entre otras.
- ✓ **Otros:** Java Web Server, AOLServer, Cherokee, Tomcat, lightHttpd, etc.

Los servidores web pueden disponer de módulos para la ejecución de programas de servidor interpretados, como son los de las tecnologías Python, PHP, ASP, JSP, Tcl, etc.

#### **26.2.8.2. Servidor de aplicaciones.**

Los servidores de aplicaciones son servidores web con capacidad de procesamiento ampliada, pudiendo ejecutar aplicaciones y componentes de lógica de negocio y recursos relacionados como el acceso a datos. Debido a esto permiten realizar el procesamiento de aplicaciones de cliente en el propio servidor. Proporcionan soporte como middleware o software de conectividad y para diferentes tecnologías de servidor. Por regla general, los servidores de aplicaciones se integran en arquitecturas de mínimo tres **capas**:

- 1) **Primera capa.** Capa de interacción con los usuarios, principalmente a través de navegadores web.
- 2) **Capa intermedia.** Capa de los servidores de aplicaciones, que pueden estar distribuidos en un modelo de granja de servidores. Un subconjunto de los servidores de aplicaciones darán servicio a los



usuarios/clientes mientras otro grupo se encargará de soportar la operativa común del dominio, como librerías o aplicaciones y servicios web de los que hagan uso las aplicaciones para usuarios/clientes.

- 3) **Tercera capa.** Capa de servidores de acceso a datos, como servidores de archivos, base de datos o informes.

El servidor de aplicaciones acostumbra a tener integrado un servidor web, para gestionar de manera independiente el servicio WWW a través del protocolo HTTP.

Además de este servicio presenta un amplio conjunto de herramientas:

- ✓ Servidor web integrado.
- ✓ Contenedor de programas de servidor (en inglés *servlets*).
- ✓ Contenedores de objetos de lógica de negocio (como por ejemplo EJBs).
- ✓ Sistemas de mensajería.
- ✓ Software de conectividad con bases de datos.
- ✓ Balanceo de carga.
- ✓ Gestión de límites y colas de conexiones (en inglés *Pool*) para bases de datos y objetos.
- ✓ Etc.

En esencia un servidor de aplicaciones realiza las mismas funciones que un servidor web, pero cuando la demanda de uso es grande y estamos ante un sistema complejo la solución pasa por emplear un servidor de aplicaciones que ofrezca las siguientes **ventajas**:

- ✓ **Centralización.** Centraliza en los servidores la administración y configuración de la lógica de negocio de las aplicaciones, de manera que aspectos como el mantenimiento de los accesos a base de datos pueden realizarse de manera centralizada. Asimismo cambios

derivados de actualizaciones, migraciones o recuperaciones ante errores tienen lugar desde un único punto.

- ✓ **Seguridad.** Al existir un único punto de acceso a datos puede reforzarse la defensa y los sistemas de control de errores en ese punto, mejorando su gestión y protección.
- ✓ **Rendimiento.** Como punto intermedio permite gestionar las peticiones de los clientes a la Base de datos.
- ✓ **Escalabilidad.** Un mismo servidor de aplicaciones puede dar servicio a varios clientes, y por tanto aumentando el número de servidores se mejora el rendimiento del sistema.

Entre los **servidores de uso más extendido** actualmente se encontrarían:

- ✓ **Jboss, Glassfish.** Servidores de aplicaciones libres bajo licencia GPL.
- ✓ **BEA Weblogic, IBM Websphere, Oracle Application Server.** Alternativas propietarias integradas en paquetes de aplicaciones con funcionalidades de gestión y monitorización extendidas.
- ✓ **Tomcat, Internet Information Server (IIS), Jetty.** Proporcionan funciones parciales de servidores de aplicaciones, con lo cual en ocasiones se definen más bien como contenedores de programas de servidor.

#### **26.2.8.3. Servidor de acceso a datos.**

Los servidores de acceso a datos ocuparían la última capa de los sistemas de información encargándose del acceso directo a los datos, existiendo diferentes tipos según el sistema de información empleado para su almacenamiento o publicación:

- ✓ **Servidores de archivos.** En este tipo de servidores la información se almacena directamente en archivos, por tanto la función de estos equipos será la de permitir el acceso remoto a los mismos desde los

clientes u otros servidores. Los protocolos más habituales ofrecen servicio solo desde redes locales pero en sistemas avanzados pueden proporcionar servicios como FTP o WebDAV para conexión remota a través de Internet. Actualmente el término empleado para referirse a estos servidores es NAS (en inglés *Network-Attached Storage*), pero ésta tan sólo sería la tecnología más habitual frente a otras como DAS (en inglés *Direct Attached Storage*), basada en SCSI o SAN (en inglés *Storage Area Network*) basada en fibra óptica. No requieren un software muy específico como otros tipos de servidores sino más bien soporte para diferentes protocolos y tecnologías. Por norma general acostumbran a estar dispuestos en [RAID](#) (en inglés *Redundant Arrays of Independent Disks*), equipos de almacenamiento redundante.

- ✓ **Servidores de bases de datos.** Albergan uno o más sistemas de gestión de bases de datos (en [inglés](#) *database management system*, o DBMS), software de gestión que se encarga de la comunicación entre las aplicaciones y las bases de datos. Permiten realizar operaciones de definición, manipulación y seguridad de los datos a través de una API de comunicación con las aplicaciones y un lenguaje estructurado de consulta como el SQL. Permiten accesos simultáneos a los datos, seguridad y gestión de transacciones.

Estos sistemas suelen presentar además programas o consolas de administración avanzadas para realizar las tareas generales de gestión de la base de datos.

- ✓ **Servidores de informes.** Pueden considerarse una capa intermedia entre los servidores de datos y los de aplicación, donde se establecen servidores o granjas de servidores que sirven los datos en documentos predefinidos multiformato: hojas de cálculo, PDF, XML, HTML, etc. El software de este tipo de servidores acostumbra a incorporar software de gestión para el servidor, y software de auto-edición de informes, para definir modelo de informes compuestos de

cabeceras, imágenes, fórmulas, subinformes, etc. que se generarán dinámicamente los informes a partir de consultas sobre los datos.

#### CAPA DE USUARIO

#### CAPA DE SERVIDOR DE APLICACIONES



#### CAPA DE ACCESO A DATOS

**Figura 4: Arquitectura en 3 capas con servidor web, de aplicaciones y base de datos.**

TRADUCCIÓN FIGURA 4: Capa de servidor de aplicaciones. Granja de servidores. Lenguajes interpretados. Seguridad. Contenedor de programas de servidor. Contenedor de objetos. Sistemas de mensajería. Pool de conexiones.

Entre los **servidores de uso más extendido** actualmente se encontrarían:

- ✓ **Servidores de archivos.** No requieren gestores especializados pero sí soporte software a los protocolos: CIFS, NFS, SMB, FTP, WebDAV, etc. Así como utilidades tipo Samba o FreeNAS.
- ✓ **Servidores de bases de datos.**
  - ✓ De licencia libre: PostgreSQL, MariaDB, Firebird, SQLite, Apache derby,...
  - ✓ Dual, dependiendo de su uso: MySQL.

- ✓ Software propietario: SQLServer, Oracle, Access, Paradox, Informix, DBase, etc.
- ✓ **Servidores de informes.** Jasper Reports, Jreports, Crystal Reports, Oracle Reports etc.

### **26.3 SCRIPTS DEL CLIENTE.**

Los *scripts* del cliente son programas interpretados diseñados para ejecutarse en los navegadores con el objetivo de dotar a las páginas de mayor interactividad con el usuario y dinamismo en una aproximación a las aplicaciones de escritorio. El **funcionamiento básico** de un *script* consiste en interpretar una serie de comandos a través de los cuales puede modificar y manipular objetos y reaccionar ante eventos de la interfaz como respuestas a periféricos (ratón, teclado, etc.), o cambios en los elementos del documento (botones, elementos de formularios, etc.). Sus usos básicos son validaciones, manipulación de formularios, procesamiento de funciones y carga asíncrona de datos.

Los *scripts* de cliente proporcionan las siguientes **ventajas**:

- ✓ Modificar el contenido de la página sin recargarla del servidor en función de las interacciones con el usuario.
- ✓ Modificar parámetros de configuración del navegador y otros elementos de la página web.
- ✓ Mejorar la interacción entre el usuario y el documento, en general la usabilidad.

Por el contrario, presentan una serie de inconvenientes o desventajas:

- ✓ Problemas de accesibilidad, pues se complica la posibilidad de

presentar alternativas a usuarios que no soporten la tecnología de script.

- ✓ Problemas de seguridad, pues toda la lógica de interacción aparece sin protección descargada en el equipo del usuario, con lo cual disponen del código fuente del programa de script.

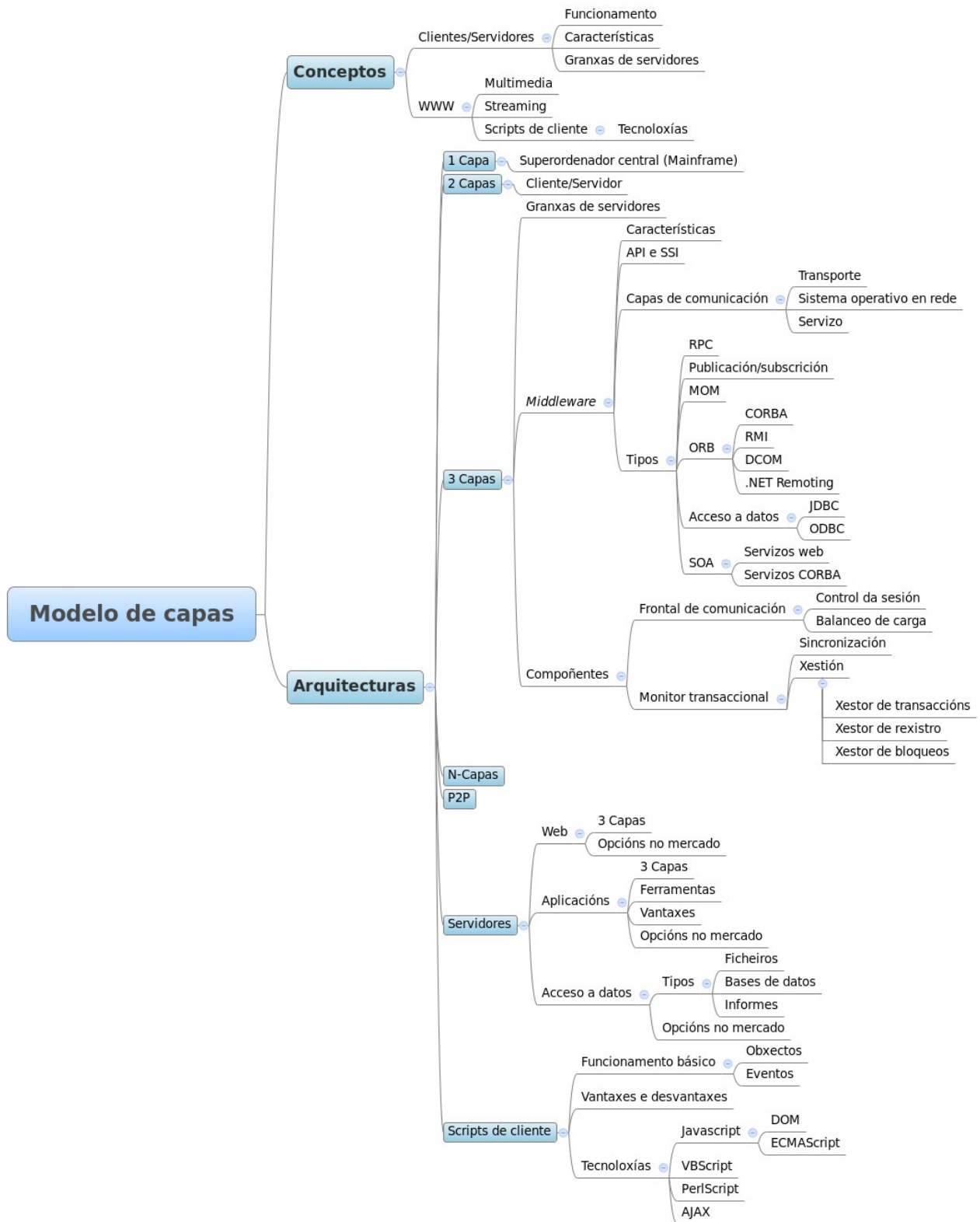
Las **tecnologías de script** más empleadas son Visual Basic Script, Javascript, con su evolución AJAX y PerlScript. El principal problema a la hora de seleccionar una tecnología cuando se diseña una página web es el soporte que recibirá por parte de los navegadores, pues hay que recordar que los lenguajes de *script* serán en última instancia interpretados en el navegador.

- a) **Javascript.** Basado en el lenguaje Java, es una de los lenguajes de script de uso más extendido. Hay que señalar que tiene aplicación con otras tecnologías además de la web, como en documentos PDF o aplicaciones de escritorio. Para permitir la interacción con los elementos de un documento web este lenguaje dispone de una API que implementa el DOM (en inglés *Document Object Model*) o Modelo de Objetos para la Representación del Documento, estandarizado por el W3C (en inglés *World Wide Web Consortium*). En un intento por estandarizar este lenguaje surge el ECMAScript una especificación del lenguaje aceptado como estándar ISO.
- b) **Visual Basic Script.** Similar al Javascript en lo relativo a funcionamiento y estructura pero basado en Visual Basic. Tiene menor soporte dentro de los diferentes navegadores, excepto en el Internet Explorer.
- c) **PerlScript.** Basado en lenguaje C su uso no está tan extendido como las tecnologías anteriores aunque tiene menos limitaciones. Debido a esto fue derivando hacia el lenguaje de servidor.
- d) **AJAX.** Acrónimo de Javascript Asíncrono y XML (en inglés

*Asynchronous Javascript And XML*). Esta es una tecnología de *script* de cliente asíncrona, de suerte que puede realizar cargas de datos sin que afecten a la recarga de la página. AJAX es un conjunto de tecnologías que hace uso de:

- 1) XHTML y hojas de estilo en cascada (CSS) para la estructura y diseño de los contenidos.
- 2) El lenguaje Javascript como lenguaje de programación para funciones y definición del programa cliente.
- 3) El objeto *XMLHttpRequest* para intercambio de información asíncrona con el servidor. Por tanto, hace falta que el navegador soporte este objeto, siendo empleado en ocasiones el objeto *Iframe*.
- 4) XML y DOM como estándares asociados para intercambio de datos y manipulación del documento.

#### **26.4. ESQUEMA**





**TRADUCCIÓN ESQUEMA:** Granjas de servidores. Tecnologías. Sistema operativo en red. Publicación /Suscripción. Servicios web. Servicios CORBA. Componentes. Gestión. Gestor de transacciones. Gestor de registro. Gestor de bloqueos. Opciones en el mercado. Aplicaciones. Herramientas. Ventajas. Ficheros- Funcionamiento básico. Objetos. Ventajas y desventajas. Tecnologías

## **26.5. REFERENCIAS**

José Antonio Mañas.

Mundo IP. Introducción a los secretos de Internet y las redes de datos. (2004).

Andrew S. Tanenbaum.

Redes de computadoras. (2003).

Sergio Luján Mora.

Programación de aplicaciones web: historia, principios básicos y clientes web. (2003).

**Autor: Juan Marcos Filgueira Gomis**

**Asesor Técnico Consellería de Educación e O. U.**

**Colegiado del CPEIG**



## **27. ARQUITECTURAS .NET. ARQUITECTURAS J2EE.**

## **TEMA 27. ARQUITECTURAS .NET. ARQUITECTURAS J2EE.**

### **27.1. INTRODUCCIÓN Y CONCEPTOS**

### **27.2 ARQUITECTURA WEB EN .NET**

### **27.3 ARQUITECTURA WEB EN J2EE**

### **27.4. ESQUEMA**

### **27.5. REFERENCIAS**

### **27.1. INTRODUCCIÓN Y CONCEPTOS**

El desarrollo de aplicaciones, servicios web y otros componentes software tiene hoy en día dos vertientes principales, la plataforma .NET y la plataforma J2EE, o JEE nombre con el que es conocida actualmente al cambiar de versión. La competencia entre estas dos tecnologías es muy fuerte, pues proporcionan soluciones similares fuertemente soportadas por las compañías de uno y del otro bando.

**.NET** es la plataforma de desarrollo propuesta por la empresa Microsoft para el mundo de los servidores de aplicaciones que funciona como herramienta de diseño y programación, además de proporcionar un amplio conjunto de utilidades extendidas de apoyo al desarrollo en este tipo de entornos. Por su parte **JEE**, (en inglés *Java Platform, Enterprise Edition*) o Java EE, es una evolución de la plataforma Java para desarrollar y soportar componentes software según un conjunto de especificaciones de modo que puedan operar en un servidor de aplicaciones, incluyendo también herramientas de diseño y programación.

A pesar de que ambas plataformas persiguen el mismo objetivo, tienen una serie de particularidades o diferencias que se ven acentuadas por las guerras comerciales entre las empresas que las soportan. Mientras JEE

tiene soporte multiplataforma, .NET funciona sólo bajo la familia de Sistemas Operativos Windows. Mientras JEE se basa exclusivamente en el lenguaje Java, en .NET se permiten muchos lenguajes de alto nivel, aunque en la práctica los principales sean C# y VB .NET. JEE lleva más años de experiencia en el mercado, mientras que .NET es más reciente. Asimismo JEE presenta mayor soporte en cuanto a soluciones y posibilidades de software libre, que son muy escasas y de poca calidad en .NET. Con JEE se puede instalar una infraestructura de alto rendimiento de manera completamente gratuita.

## **27.2 ARQUITECTURA WEB EN .NET**

**.NET** es, según la empresa Microsoft, una plataforma para el desarrollo de servidores, clientes y servicios. Representa un conjunto de tecnologías que tienen como núcleo principal el .NET Framework, un marco de desarrollo y componente software que puede instalarse en Sistemas Operativos de la familia Windows (Windows 2003, Vista, Windows 7, ...). Existe una versión adaptada para móviles disponible en Windows Mobile. La norma ISO/IEC 23271 recoge un conjunto funcional mínimo que deben cumplir los productos software desarrollados para que puedan funcionar dentro del marco de trabajo. Ésta y más normas se recogen en los estándares:

- a) **Estándar ECMA-334.** Especificación del lenguaje C#. (2006).
- b) **Estándar ECMA-335.** Especificación del lenguaje de infraestructura común (CLI). (2010).

Por el contrario, otros componentes como ASP .NET, *Windows Forms* o ADO .NET no se encuentran estandarizados. Paralelamente, una vez publicados los documentos de especificación de la arquitectura .NET apareció el **Proyecto Mono** con el objetivo de implementar el marco de trabajo .NET Framework empleando código abierto, para a partir de ahí desarrollar aplicaciones para sistemas UNIX/Linux.

### **27.2.1 .NET Framework**

Marco de trabajo que proporciona el conjunto de herramientas y servicios para el desarrollo de componentes software, aplicaciones, servidores y servicios web. Puede dividirse en tres bloques principales:

- 1) El **Entorno de Ejecución Común** (en inglés *Common Language Runtime* o CLR). Se encarga de la gestión de código en ejecución, control de memoria, seguridad y otras funciones relacionadas con el Sistema Operativo.
- 2) La Biblioteca **de Clases Base** (en inglés *.NET Framework Base Classes*). Realizan la función de API de servicios a disposición de los desarrolladores para tareas como gestión de ficheros, mensajería, procesos en varios hilos, acceso a datos, encriptación, etc.
- 3) **Control de acceso a datos**, que permite realizar las operaciones de acceso a datos a través de clases y objetos del *framework* incluidos en el componente ADO.NET.
- 4) El **Motor de Generación de la Interfaz de Usuario**, que permite crear interfaces para aplicaciones de escritorio o web empleando componentes específicos como ASP.NET para web, *Web forms* para aplicaciones de escritorio o *Web services* para servicios web.

### **27.2.2 Entorno de Ejecución Común (CLR)**

El Entorno de ejecución común o CLR es el encargado de gestionar el código en tiempo de ejecución. De manera análoga a la Máquina virtual de Java este entorno permite ejecutar aplicaciones y servicios web o de escritorio en cualquier cliente o servidor que disponga de este software. A diferencia de la Máquina virtual de Java el soporte de .NET es multilenguaje permitiendo C++, C#, ASP .NET, Visual Basic, Delphi, y muchos otros. Además permite integrar y heredar componentes entre diferentes lenguajes, con mayor o menor fortuna a la hora de sacar beneficio de lenguajes antiguos.

El entorno de desarrollo compila el código fuente en cualquiera de los lenguajes soportados a un código intermedio denominado **CIL** (en inglés *Common Intermediate Language*) de manera análoga al BYTECODE de Java. A este lenguaje intermedia se llega empleando la especificación CLS (en inglés *Common Language Specification*) donde se especifican unas reglas necesarias para crear el código intermedio CIL compatible con el CLR. Asimismo, el CLR dispone de compiladores como JIT (en inglés *Just InTime*) o AOT (en inglés *Ahead Of Time*) adaptados a cada lenguaje.

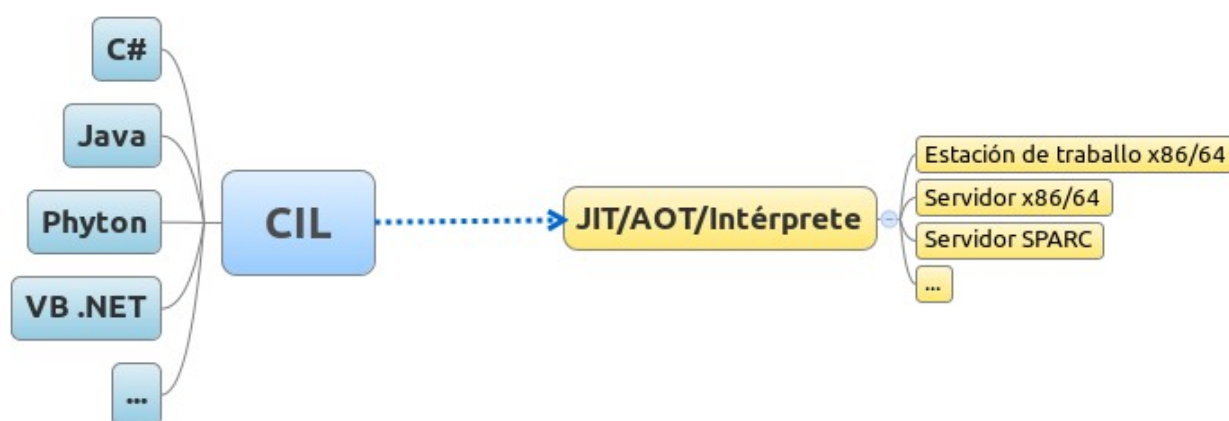
**JIT** genera el código máquina real en cada máquina a partir de ese código intermedio consiguiendo independencia del hardware. Esta compilación se hace en tiempo de ejecución a medida que la aplicación o servicio invoca métodos o funciones. Para agilizar el procesamiento este código máquina obtenido en tiempo de ejecución se guarda en la memoria caché actualizándose tan sólo cuando se produce algún cambio en el código fuente, momento en el que se repite el proceso. Por el contrario, **AOT** compila el código antes de ejecutarse con lo cual logra un mayor rendimiento en ejecución pero menos independencia de la plataforma. En lo tocante a JIT normalmente se distingue entre:

- 1) **Jitter estándar.** Compila el código CIL a nativo bajo demanda.
- 2) **Jitter económico.** No optimiza, traduce cada instrucción así precisa menos tiempo y memoria de compilación.
- 3) **Prejitter.** Realiza una compilación estática de un componente software completo.

Las principales **ventajas** de este modelo de compilación son:

- ✓ La **reutilización** de componentes escritos en diferentes lenguajes en una misma aplicación o servicio web.
- ✓ **Modularidad**, gracias a la implementación del patrón Interfaz para cada componente o librería ya que será accesible desde cualquier lenguaje a través de su API (ASP, C#, Java, Phyton, etc.)

- ✓ **Integración multilenguaje**, ya que cada lenguaje con un compilador a CIL puede integrarse en la plataforma, con lo cual cada componente en ese lenguaje puede integrarse una aplicación o servicio web .NET.
- ✓ **Seguridad**. Por el aislamiento del código de usuario respecto de los accesos a datos y otras partes críticas del Sistema Operativo.



**Figura 1: Estructura multilenguaje del CLR**

**Traducción Texto Figura 1:** Estación de trabajo

El CLR cumple además la función de proporcionar una amplia gama de servicios a las aplicaciones. A través de la API de cada servicio los componentes web pueden tener acceso a funcionalidades comunes, como:

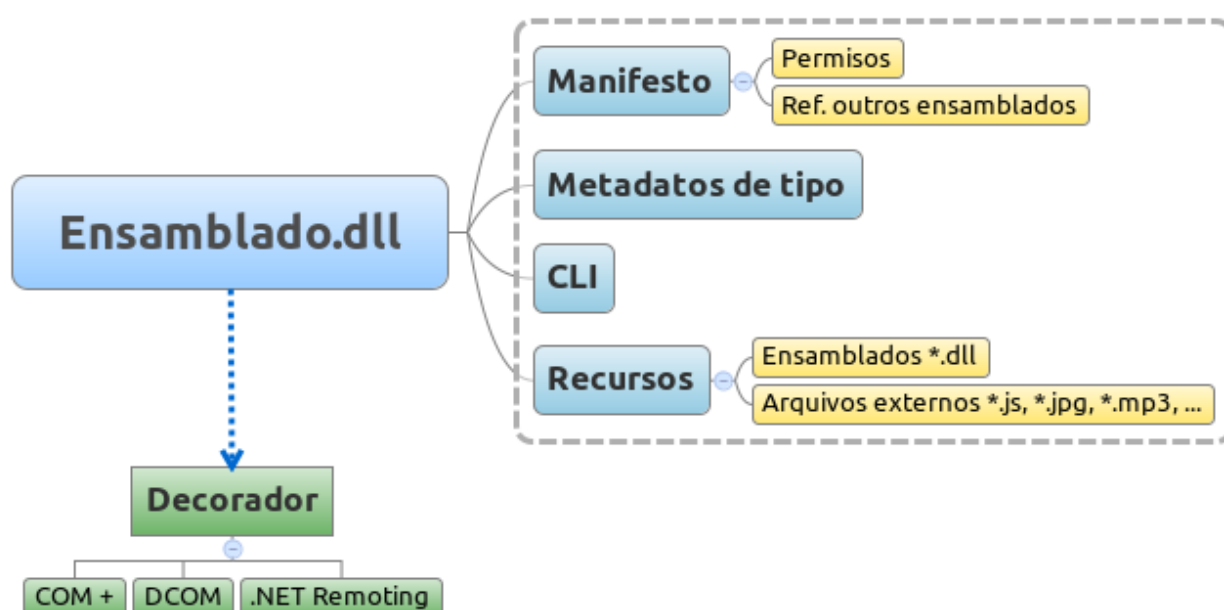
- a) **Seguridad acceso al código o CAS** (en inglés *Code Access Security*). Controla qué tipo de operaciones puede realizar un código según se identifique en la firma del ensamblado o en el origen del código. Asimismo reconoce directivas de administración del sistema para componentes o a nivel de *host*. Cuando un componente trate de acceder a recursos protegidos del sistema se lanzará el CAS para comprobar los permisos, pero a este nivel no se pueden establecer comprobaciones dinámicas, por ejemplo contra Bases de datos.
- b) **Atributos de protección del host o HPA** (en inglés *Host Protection Attributes*). Mantiene una lista de atributos protegidos, denegando el

acceso o modificación de los mismos. Algunos de estos atributos serían *SharedState*, para estados compartidos, *Synchronization*, para permitir la capacidad de sincronizar procesos en el *host* o *ExternalProcessment* que indica si los procesos en el *host* se pueden controlar externamente a través de la API.

- c) **Dominios de aplicación.** Definen dominios aislados de código para restringir el acceso de los componentes software, creando una zona reservada para un subproceso. Por norma general el CLR crea para cada aplicación un dominio en tiempo de ejecución pero puede precisar dominios específicos para componentes DLL o externos.
- d) **Comprobación de la seguridad de tipos.** Reserva espacios de memoria para cada objeto según lo especificado para su tipo. Cuando el espacio es aleatorio o queda algún hueco fuera del espacio del objeto, quiere decir que ese objeto no tiene seguridad de tipos. Con la compilación JIT se realiza una comprobación en tiempo de ejecución para verificar si cada objeto tiene seguridad de tipos. Del mismo modo impide variables sin valores iniciales o *cast* no seguros.
- e) **Cargador de clases.** Permite cargar en memoria clases y tipos de datos a partir de la interpretación de los metadatos. Existe además la posibilidad de crear cargadores personalizados, aunque sólo para Java, ya que por motivos de rendimiento resulta mucho mejor emplear un ensamblado con otros lenguajes. En la compilación el cargador realiza la función de evitar código innecesario a través de funciones *stubs* que sustituye con el código correcto bajo demanda.
- f) **Recolección de basura** (en inglés *Garbage Collector*). Este servicio se ejecuta de manera continua para buscar y eliminar de memoria los objetos que no sean referenciados o terminen el tiempo de espera de utilización.



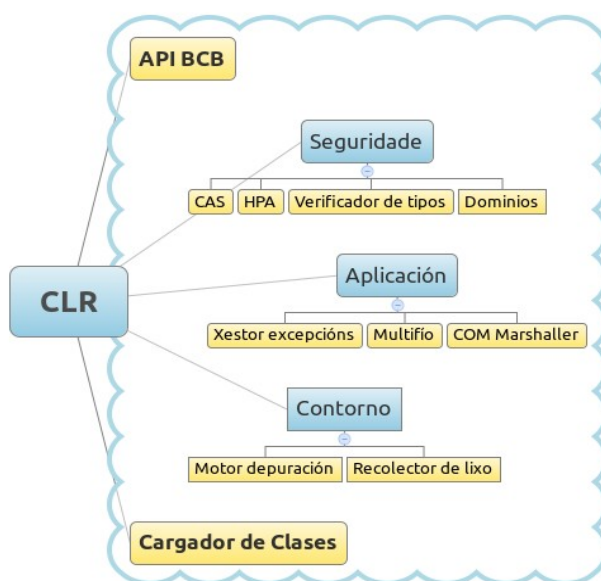
- g) **Motor de interacción COM.** Realiza funciones de conversión de datos y mensajes o *marshaling* desde y hacia objetos COM, lo que permite la integración con aplicación *Legacy*.
- h) **Motor de depuración.** Permite realizar un seguimiento de la ejecución del código aunque mezcle diferentes lenguajes.
- i) **API multihilo** (en inglés *multithread*). Proporciona una API y las clases necesarias para gestionar la ejecución de hilos paralelos.
- j) **Gestor de excepciones.** Realiza la gestión estructurada e integración con *Windows Structured Exception Handling* de excepciones aunque el error provenga de diferentes lenguajes en un mismo componente e incluso en el código aún no ejecutado. Este código puede incluir excepciones SHE del tipo C++ o resultados HRESULTS típicos de COM.
- k) **API de la Biblioteca de Clases Base (BCB).** Interfaz con la BCB del marco de trabajo que realiza la integración del código con el motor de ejecución.



**Figura 2: Ensamblados.**

## **TRADUCCIÓN TEXTO FIGURA 2:** Manifiesto. Otros ensamblados. Archivos externos.

En el .NET Framework cuando se compila un programa o aplicación web se genera un archivo denominado **ensamblado** que contiene el código compilado al lenguaje intermedia CLI y un manifiesto con permisos y referencias a otros ensamblados, componentes software o servicios web. Son paquetes o librerías EXE o DLL destinadas al control de versiones, seguridad y comprobaciones de implementación al por menor. Los ensamblados llevan una indicación que define el entorno de ejecución en el que se debe lanzar: COM+, DCOM, .NET Remoting,...



**Figura 3: Servicios del CLR**

**TRADUCCIÓN TEXTO FIGURA 3:** Seguridad. Gestor de excepciones.  
Multihilo. Entorno. Recogedor de basura.

### **27.2.3 Biblioteca de Clases Base (BCB)**

La Biblioteca de Clases Base es una API de alto nivel para permitir acceder a los servicios que ofrece el CLR a través de objetos en una jerarquía denominada **espacio de nombres**. Agrupa las funcionalidades de uso

frecuente permitiendo su redefinición. Se encuentra implementada en CIL por lo que puede integrarse en cualquier otro lenguaje. Es un conjunto de clases, interfaces y tipos valor que son la base sobre la que se crearán las aplicaciones, componentes y controles del .NET Framework. Permite realizar operaciones como: soporte para diferentes idiomas, generación de números aleatorios, manipulación de gráficos e imágenes, operaciones sobre fechas y otros tipos de datos, integración con APIS antiguas, operaciones de compilación de código adaptada a los diferentes lenguajes de .NET, elementos para interfaces de usuario, tratamiento de excepciones, acceso a datos, encriptación, administración de memoria, control de procesos, etc.

<b>Espacio de nombres</b>	<b>Utilidad y objetos</b>
<b>System</b>	Tipos básicos, tablas, excepciones, fechas, recolector de basura, etc.
<b>System.Collections</b>	Manipulación de colecciones como pilas, colas, <i>hash</i> , etc.
<b>System.Data</b>	Arquitectura ADO.NET (Objetos <i>DataSet</i> , <i>DataTable</i> , <i>DataRow</i> , <i>DataView</i> , ...)
<b>System.IO</b>	Manipulación de Y/S archivos y otros orígenes de datos
<b>System.Net</b>	Gestión de comunicaciones de red (TPC/IP, <i>Sockets</i> , ...)
<b>System.Security</b>	Gestión de las políticas de seguridad del CLR
<b>System.XML</b>	Acceso y manipulación de datos en documentos XML con compatibilidad con el W3C (Transformaciones en <i>System.Xml.Xls</i> y serialización para servicios web en <i>System.XML.Serialization</i> )
<b>System.Web</b>	Servicios para gestión de caché, seguridad y configuración para Servicios Web, estado de las sesiones e interfaces de usuario
<b>System.Web.Services</b>	Gestión de los requerimientos de Servicios Web
<b>System.Web.UI</b>	Controles para interfaces de usuario <i>HTMLControl</i> para mapeo de etiquetas HTML y <i>WebControl</i> para estructurar controles de usuario avanzados como <i>DataGrids</i>

<b>System.Windows.Forms</b>	Creación de la IU del cliente
<b>System.Drawing</b>	Acceso a funcionalidades gráficas básicas de la GDI+ (Funcionalidades avanzadas en <i>System.Drawing.Imaging</i> , <i>System.Drawing.Text</i> y <i>System.Drawing.Drawing2D</i> )
<b>System.Reflection</b>	Acceso a metadatos sobre los ensamblados, módulos, miembros, parámetros y otras entidades del código administrado
<b>System.JSON</b>	Proporciona compatibilidad basada en estándares JSON, nota de objetos JavaScript (en inglés <i>JavaScript Object Notation</i> )
<b>System.Threading</b>	Manipulación de procesos e hilos de ejecución
<b>System.Text</b>	Proporciona clases para manipular la codificación de caracteres UNICODE y UTF-8 conversión de bloques de caracteres en bloques de <i>bytes</i> y viceversa
<b>System.Transactions</b>	Contiene clases que permiten crear y administrar transacciones, admitiendo participantes distribuidos, notificaciones de fase e inscripciones duraderas
<b>System.Resources</b>	Proporciona clases e interfaces que permiten crear, almacenar y administrar recursos de localización
<b>System.Runtime.Remoting</b>	Proporciona la interfaz para acceso remoto y marco para la implantación de sistemas de componentes distribuidos
<b>Microsoft.CSharp</b>	Clases para realizar la compilación y ejecución de código en C# (Lo mismo para otros lenguajes)

**Tabla 1: Principales espacios de nombres.**

#### **27.2.4 Control de acceso a datos.**

El control de acceso a datos, documentos XML y servicios de datos en el marco .NET Framework se recoge en la arquitectura ADO .NET, como evolución del *ActiveX Data Objects*. Su orientación principal es el acceso a datos del gestor de base de datos relacional *SQL Server*, orígenes XML y orígenes de datos vía objetos OLE DB y ODBC. Las **conexiones** se realizan identificando los proveedores de datos a través de los objetos (Connection, Command, DataReader y DataAdapter). Una vez establecida la conexión

entra en escena el principal elemento del marco, el *Dataset*, que recoge los **resultados** cargados a partir de un origen. A su vez, puede particularizarse con otros elementos de la base de datos con objetos como: *DataTable*, *DataRow*, *DataColumn* o *Constraint*. Los **objetivos** de diseño principales de este marco son:

- ✓ Soporte a la tecnología ADO previa.
- ✓ Integración con tecnologías basadas en XML.
- ✓ Soporte a modelos de arquitectura multicapa.

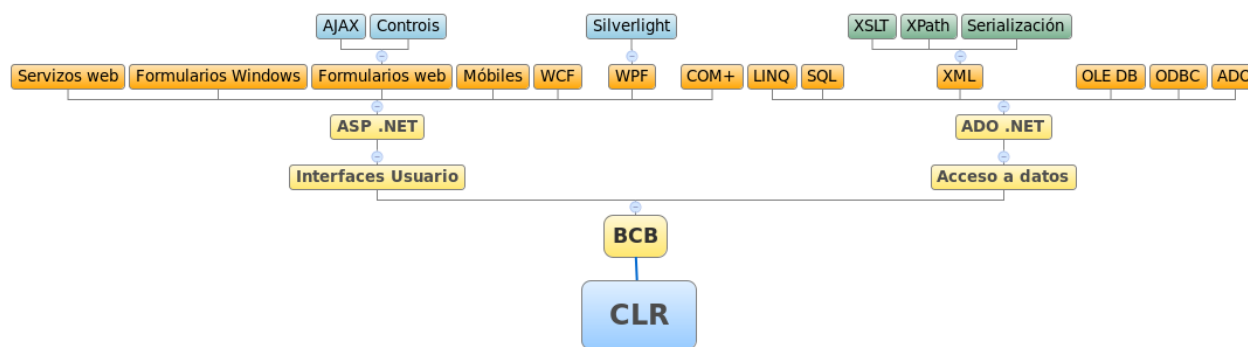
En las últimas versiones se incorpora el **Marco de Entidades** (en inglés *Entity Framework*) que permite realizar Consultas Integradas en los Lenguajes (en inglés *Language Integrated Query*) o **LINQ**.

Este marco permitirá emplear LINQ sobre muchos componentes de acceso a datos nuevos: *LINQ to SQL*, *LINQ to DataSet* y *LINQ to Entities*. Asocia una llave lógica a las entidades, dotando al modelo relacional o conceptual de orientación a objetos. Utilidades del marco de trabajo como **SQLMetal** permiten la generación automática de clases a partir de la base de datos o documentos XML.

#### **27.2.5 Motor de Generación de Interfaces de Usuario**

La parte del marco de trabajo encargado de la generación de interfaces de usuario y servicios web sería la arquitectura ASP .NET. El conjunto de clases correspondientes se agrupan en los espacios de nombres: System.Web, System.Web.Services, System.WebUI. Las páginas web desarrolladas con ASP .NET tienen la extensión **ASPX** y son conocidas con **Formularios Web** (en inglés *Web Forms*). Paralelamente a esta tecnología de presentación encontraríamos **Formularios Windows** (en inglés *Windows Forms*) para aplicaciones de escritorio y tecnologías para Móviles, siendo la primera la más empleada para aplicaciones de servidor. En el espacio System.Web.UI

se recogen las dos clases principales de controles los HTML para acceso directo a las etiquetas estáticas de estos lenguajes y los controles web que incorporan código de servidor dinámico. La arquitectura recomendada emplea el modelo **Code-Behind** en el que se crea un archivo separado con el código de servidor, a diferencia de la arquitectura DNA para ASP anterior. Los **Controles de usuario** (en inglés *User Controls*) siguen la misma estructura que los Formularios Web, pero derivan del espacio `System.Web.UI.UserControl`, y se guardan en archivos **ASCX**, que deberían seguir también el modelo Code-Behind. El marco incorpora también elementos para control del estado y la sesión así como otros para seguridad, autenticación de usuarios, uso de papeles, uso del servicio Indigo o WCF (en inglés *Windows Communication Foundation*). Asimismo, permite la integración con AJAX, a través de la incorporación de un **Toolkit** en la aplicación, o WPF (en inglés *Windows Presentation Foundation*) basado en XALM y marco para Silverlight.



**Figura 4: Arquitectura general**

### 27.2.6 Niveles lógicos.

En los modelos y soluciones que propone esta estructura se establecen una serie de servicios genéricos presentes en la mayoría de las aplicaciones corporativas actuales. Esta división permite definir un diseño y una arquitectura específicos para cada nivel, facilitando el desarrollo y soporte de la aplicación.

1. **Servicios de usuarios.** Se encuentran en la primera línea de interacción con los usuarios y proporcionan la interfaz de acceso al sistema que deriva en llamadas a los componentes del nivel de Servicios corporativos. En ocasiones se consideran dentro de este nivel procesos fuera de las interfaces de usuario, como procedimientos de control o automatizados que no requieren la presencia de un usuario.
2. **Servicios corporativos.** Encapsulan la lógica corporativa proporcionando una API de las funcionalidades básicas del sistema. Esto permite abstraer los servicios de usuario de la lógica corporativa y mantener diferentes servicios de usuario a partir de las mismas funcionalidades. Cada funcionalidad puede precisar disponer de varios servicios corporativos.
3. **Servicios de datos.** Sería la parte más aislada del usuario, proporcionando el acceso a datos y a otros sistemas o servidores. Establecen diferentes API genéricas de las que pueden hacer uso los Servicios corporativos. Contienen una amplia gama de orígenes de datos y sistemas de servidor, encapsulando reglas de acceso y formatos de datos.

#### **27.2.6 Soluciones de integración.**

Con la tecnología .NET existen tres principales planteamientos de soluciones arquitectónicas:

- 1) **SOA** (en inglés *Service Oriented Architecture*). Entiende la comunicación entre aplicaciones y componentes como servicios, no necesariamente servicios web, demandados por clientes o suscriptores y proporcionados y publicados por proveedores.
- 2) **MOA** (en inglés *Message Oriented Architecture*). La comunicación se

realiza por paso de mensajes forzando un modelo SOA distribuido.

- 3) **EAI** (en inglés *Enterprise Integration Application*). Especifica una serie de requerimientos de integración y comunicación en sistemas regulados por los patrones de integración Mediación y Federación, donde un sistema EAI hace funciones de *Hub* o *bus* de comunicaciones.

## **27.3 ARQUITECTURA WEB EN J2EE**

JEE representa un conjunto de especificaciones para plataformas de desarrollo basadas en lenguaje Java para servidores de aplicaciones en arquitecturas de múltiples capas. El **JCP** (en inglés *Java Community Process*) es el organismo encargado de validar los requisitos de conformidad para cada plataforma y aceptarla. Las plataformas JEE constan de los siguientes componentes:

- 1) Un conjunto de especificaciones.
- 2) Un test de compatibilidad o CTS (en inglés *Compatibility Test Suite*)
- 3) Una implementación de referencia para cada especificación.
- 4) Un conjunto de guías de desarrollo y buenas prácticas denominadas JEE Blueprints.

Las principales **especificaciones** que incluye JEE dan soporte a Servicios web, RPC basado en XML, Mensajería XML, despliegues, servicios de autorización, conexión remota RMI, JSP, JSF, JSTL, Servlets, Portlets, Applets, JavaBeans, esquemas XML, acceso a datos JDBC, documentación Javadoc, transformaciones XSL, etc.

El soporte multiplataforma del Java tiene su base en la **Máquina Virtual** (VM/JVM o KVM/CVM para móviles), una plataforma lógica capaz de instalarse en equipos con diferente hardware y Sistema operativo e interpretar y ejecutar instrucciones de código **Java bytecode**. La especificación de la VM también se recoge como especificación por la JCP y



del mismo modo están disponibles test de compatibilidad. La forma más habitual para la VM es mediante un compilador JIT pero también permite interpretación. Del mismo modo se permite ejecución segura mediante el modelo de las Java Applets. Programas de cliente que se ejecutan en una VM dentro del navegador después de descargar vía HTTP código del servidor, que se ejecuta en una *Sandbox* muy restringida.

Los componentes software web y de negocio dentro de esta tecnología se despliegan a través de **Contenedores** (en inglés *containers*). Los contenedores son implementaciones de arquitecturas JEE que proporcionan los servicios del servidor de aplicaciones a los componentes, incluyendo seguridad, acceso a datos, manejo de transacciones, acceso a recursos, control de estados, gestión del ciclo de vida y comunicaciones entre otros. Antes de ejecutarse un componente software debe configurarse como un servicio JEE y desplegarse dentro de un contenedor. Los principales serían los contenedores web para Servlets y JSP, los contenedores EJB para componentes de la lógica de negocio, y contenedores de aplicaciones cliente y contenedores de Applets para los programas de cliente y código de cliente para el navegador respectivamente.

Los principales **servicios** que proporciona JEE junto con sus respectivas API serían:

- 1) **HTTP y HTTPS**. Para control de las comunicaciones web y SSL a través de estos protocolos. Las API de servidor vienen dadas por los paquetes de clases Servlets y JSP y la de clientes en el paquete Java.Net.
- 2) **JDBC** (en inglés *Java Data Base Connection*). API de acceso a datos en sistemas gestores de bases de datos relacionales vía SQL. Por un lado aporta la interfaz para ser empleada por los componentes software y por otra la interfaz para que los proveedores puedan desarrollar los controladores específicos. Las versiones más recientes

- son las JDBC 3.0 y 4.0, que incluyen los paquetes *java.sql* y *javax.sql*.
- 3) **JSTL** (en inglés *Java Server Pages Standard Tag Library*). Proporciona las funcionalidades para etiquetas en las páginas JSP.
  - 4) **RMI-IIOP** (en inglés *Remote Method Invocation-Internet Inter-ORB Protocol*). Proporciona la API para permitir comunicaciones en aplicación distribuidas a través de JAVA RMI, por ejemplo para acceder a componentes EJB. Los protocolos más habituales son JRMP, de RMI e IIOP, de CORBA.
  - 5) **IDL** (en inglés *Java Interfaz Definition Language*). Permite la comunicación de clientes con servicios CORBA a través del protocolo IIOP, servicios SOAP o RPC.
  - 6) **JNDI** (en inglés *Java Naming and Directory Interfaz*). Proporciona el servicio de nombres y directorios, indicando el contexto de cada objeto y las relaciones entre ellos. Se divide en dos interfaces, la API de programación y una SPI que permite conectar con proveedores de servicios de nombres y directorios siendo los principales LDAP, CORBA y RMI.
  - 7) **JAXP** (en inglés *Java API for XML Processing*). Soporta el procesamiento de documentos XML que cumplan con los esquemas del W3C a través de DOM, SAX y XSLT.
  - 8) **JMS** (en inglés *Java Message Service*). Proporciona la API de envío de mensajes para comunicarse con un MOM (en inglés *Message-Oriented Middleware*), una abstracción independiente del proveedor para comunicaciones entre sistemas.
  - 9) **JavaMail**. Proporciona la interfaz para controlar el envío y recepción de correos electrónicos. Puede soportar el formato MIME gracias a su integración con marco de trabajo JAF.
  - 10) **JAF** (en inglés *Java Beans Activation Framework*). API que proporciona el marco de trabajo para activación que soporta las peticiones de otros paquetes.
  - 11) **JTA** (en inglés *Java Transaction API*). Orientada hacia el manejo de

transacciones y a permitir la comunicación entre contenedor y componentes del servidor de aplicaciones como los monitores transaccionales y los administradores de recursos.

- 12) **JAX-RPC** (en inglés *Java API for XML-based RPC*). Proporciona soporte para comunicaciones remotas de tipo RPC entre clientes y servicios web con los estándares HTTP y SOAP. Soporta otros estándares como WSDL, así como SSL y TTL para autenticación. El SAAJ (en inglés *SOAP with attachments API for Java*) añade la posibilidad de archivos o notas aportados con los mensajes.

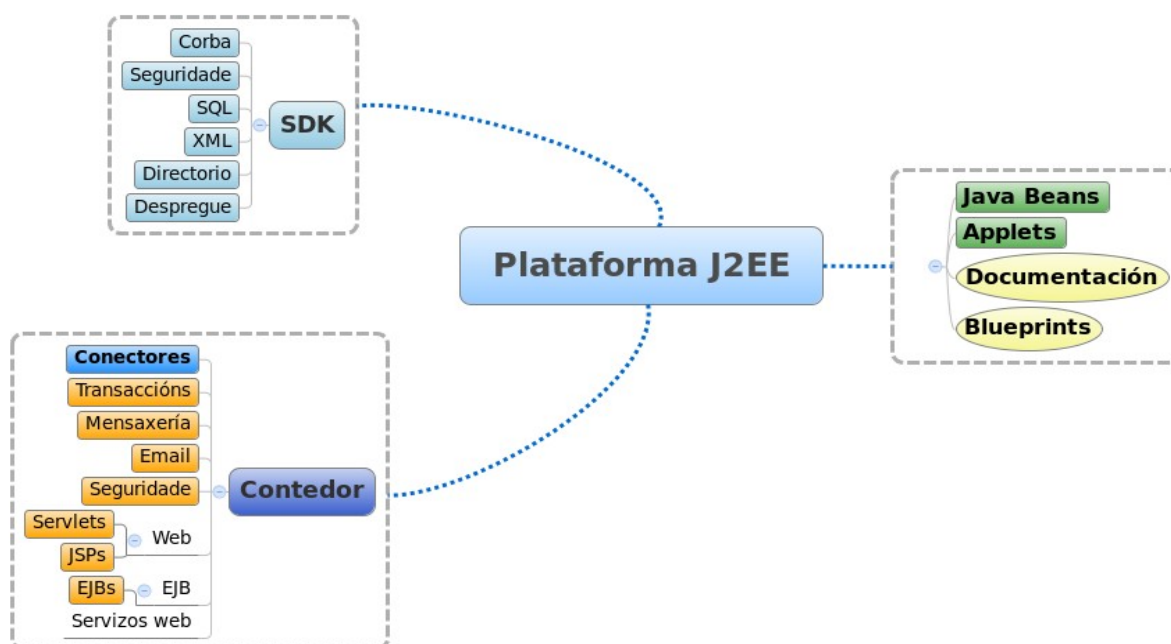
Cada componente se denomina **Módulo** JEE de modo que una aplicación estará formada por un conjunto de módulos siendo cada uno un componente para un contenedor. Existen tres tipos de módulos:

- 1) **Archivos JAR** (en inglés *Java Archive*). Agrupación de archivos Java y recursos según el formato ZIP. Empaquetan componentes EJB según la estructura de directorios del código, añadiendo una carpeta especial, META-INF, con metadatos.
- 2) **Archivos WAR** (en inglés *Web Application Archive*). Agrupan en un único archivo una aplicación web, incluyendo Servlets, archivos JSP, contenido estático y otros recursos web.
- 3) **Archivos EAR** (en inglés *Enterprise Application Archive*). Agrupa en un único archivo varios módulos de una aplicación como archivos WAR o componentes EJB y otras librerías en archivos JAR empaquetados con sus respectivos recursos. Asimismo se incluye el descriptor de despliegue de la aplicación en la carpeta META-INF.
- 4) **Archivos RAR** (en inglés *Resource Adapter Archive*). Contiene un adaptador de recursos de manera análoga a un controlador JDBC y similar a los EAR, pudiendo ir contenido en un archivo de este tipo. El formato viene definido en la especificación JCA (en inglés *Java EE Connector Architecture*).

De manera general conviene considerar a la plataforma como JEE, si bien, existen diferentes **ediciones**, siendo las principales:

- 1) **J2ME**. (en inglés *Java 2 Platform Micro Edition*). Para desarrollo de aplicaciones para dispositivos móviles, electrodomésticos y equipos PDA. Se desarrolló mediante el JPC bajo a especificación JSR 68.
- 2) **J2SE**. (en inglés *Java 2 Platform Standard Edition*). Para desarrollo de aplicaciones de uso general en estaciones de trabajo. Se desarrolló mediante el JPC bajo diferentes especificaciones según las versiones existentes: 1.4, 5.0 y 6.
- 3) **J2EE**. (en inglés *Java 2 Platform Enterprise Edition*). Para desarrollo de aplicaciones destinadas a servidores de aplicaciones para dar soporte a sistemas distribuidos en N capas. Estandarizada por el JPC a partir de la versión 1.4 suele denominarse JEE.

Para cada edición puede distinguirse entre la SDK (en inglés *Software Development Kit*), con el software y recursos destinados al desarrollo de aplicaciones y el JRE (en inglés *Java Runtime Environment*) con el entorno y librerías principales para permitir la ejecución de las aplicaciones.



**Figura 5: Plataforma J2EE.**

**TRADUCCIÓN TEXTO FIGURA 5:** Seguridad. Despliegue. Transacciones. Mensajería. Seguridad. Contenedor. Servicios Web.

### **27.3.1 Modelo de desarrollo**

El modelo de desarrollo más habitual en la arquitectura JEE es un modelo separado en múltiples capas siendo el habitual un mínimo de tres, pero pudiendo llegar a 5 ó 7 según la complejidad del sistema. El objetivo es minimizar el solapamiento entre ellas para que los cambios y modificaciones se limiten al mínimo necesario con el ideal de que se pueda cambiar una de las múltiples capas sin tener que modificar el resto. Se mejora la sostenibilidad, crecimiento del sistema y la reutilización de componentes en el mismo y entre sistemas. Asimismo se permite una mayor heterogeneidad de clientes o elementos de cliente y presentación al modificar sólo la capa más próxima al usuario. El diseño del modelo es análogo a soluciones en .NET pero la implantación diferente. Las 5 capas más habituales se describirán a continuación.

#### **27.3.1.1 Capa de cliente.**

Agrupar los elementos de la interfaz de usuario más próximos al cliente. Ejemplos de estos elementos serían el código (X)HTML/XML y Javascript, los Applets, archivos de recursos y tecnologías RIA. Los tipos de aplicaciones cliente más habituales serían los navegadores web, las aplicaciones de escritorio y actualmente cobran fuerza las aplicaciones para dispositivos móviles. Un aspecto importante en este modelo es garantizar que los EJB de la lógica de negocio sean accesibles tan sólo desde interfaces remotas a través del patrón *SessionFacade*. La variedad de interfaces actual hace que aparezcan *frameworks* de generación dinámica de los mismos basados en el lenguaje **XUL** (en inglés *XML User-Interface Language*), basado en XML. Permite incrustar XHTML y otros lenguajes como MathML o SVG además de CSS. Existen varias alternativas de librerías XUL como Luxor, XWT, Thinlets

o SwingML.

#### **27.3.1.2 Capa de presentación.**

Contiene toda la lógica de interacción directa entre el usuario y la aplicación. Se encarga de generar las vistas más acomodadas para mostrar la información a través de formatos y estilos adecuados. Se componen de una serie de Servlets y páginas JSP que se encargan de devolver el código que irá a la capa de cliente después de comunicarse con la capa de lógica de negocio para obtener los resultados. Puede localizarse en una aplicación de escritorio o en un contenedor web. Además de ensamblar las diferentes vistas, controla el flujo de navegación y hace funciones de autenticación, permisos de acceso y autorización de usuarios, etc. El patrón más habitual en esta capa será el MVC (en inglés *Model View Controller*). Entre las tendencias actuales se encuentran implementaciones de este modelo como Swing el JFace para aplicaciones de escritorio, mientras que dentro de los *frameworks* web se encontrarían Struts, JSF, Tapestry, Expresso y muchos otros, siendo Struts una especie de estándar de hecho. Estos *frameworks* además incorporan otros servicios como etiquetas personalizadas JSTL para interfaces de usuario, manejo de XML, acceso a datos, modelo, filtros, etc.

#### **27.3.1.3 Capa de lógica de negocio.**

Contiene los componentes de negocio reutilizables EJB o POJO, que representan el conjunto de entidades, objetos, relaciones, reglas y algoritmos del dominio o negocio en el que opere el sistema. En esta capa la solución POJO es una opción sencilla que puede mezclar elementos de las capas de integración y datos, mientras que los EJB distinguen los objetos de sesión y las entidades, recomendado en los Blueprints de Sun, emplazando cada uno en su correspondiente capa. El patrón básico en esta capa será el *SessionFacade* donde un único Bean de sesión se encarga de recibir las llamadas de cliente-presentación y dirigir las dentro del contenedor de EJB aislando esta capa.

De manera análoga se establece el modelo de los Bean de mensajería, que realizan comunicación asíncrona mediante JMS en un servidor MOM (en inglés *Messaging Oriented Middleware*), que puede ser un servidor externo al servidor de aplicaciones. También funciona con un patrón Fachada centralizando las llamadas remotas.

#### **27.3.1.4 Capa de Integración.**

Agrupar los componentes encargados del acceso a datos, sistemas *legacy*, motores de reglas de *workflow*, acceso a LDAP, etc. Pueden realizar cambios de formato en la información, pero transformaciones más complejas deberían realizarse en la capa de lógica de negocio, restringido esta a la lógica de acceso a datos o DAO (en inglés *Data Access Objects*) y los encapsuladores de datos y entidades VO (en inglés *Value Object*). Los VO pueden implementarse como POJO o EJB de entidad. Como ocurría anteriormente en el caso de los POJO, se gana en facilidad pero se pierden servicios y funcionalidades como los de persistencia. Los EJB suman complejidad, pero mejoran el rendimiento en memoria. En lo relativo al acceso a datos aparecen las siguientes alternativas:

- 1) **JDBC.** Para POJO o Beans de entidad con control de persistencia. Es una solución sencilla, con pocas funcionalidades pero que hace uso de una API de uso extendido.
- 2) **DAO.** Va un paso más allá que el JDBC incorporando interfaces para abstraer el acceso a datos y hacerlo independiente del lenguaje del gestor. Cada interfaz tendrá una implementación diferente para cada gestor de bases de datos.
- 3) **Frameworks de persistencia.** Hacen las funciones de motores de correspondencia de objetos a bases de datos relacionales definiendo entidades y relaciones vía XML. Realizan gran parte de las funciones de acceso a datos automáticamente. Las soluciones de uso más extendido son los *frameworks* Hibernate, iBatis, TopLink, JPA o a través de EJB.

- 4) **JDO** (en inglés *Java Data Objects*). Sistema de persistencia estándar a partir de una especificación JEE, añadiendo además de la correspondencia entre el modelo relacional y los objetos, la posibilidad de permitir definir los objetos sobre la base de datos. Las implementaciones de uso más extendido son OJB, XORM, Kodo JDO o LIDO.

#### **27.3.1.5 Capa de sistemas de información.**

Se encuentra integrada por los sistemas de bases de datos, ficheros, sistemas 4GL, ERP, Data Warehouse, Servicios web y cualesquiera otros sistemas de información de la organización. En esta capa irían los conectadores para diferentes sistemas de información heterogéneos y los propios recursos que integran los sistemas de información. Las soluciones más habituales se enumeran a continuación.

- 1) **JCA** (en inglés *J2EE Connector Architecture*). Define una interfaz de acceso común independiente del sistema, con la misma API para todos. Se basa en el concepto de adaptador de recursos, siendo cada adaptador un controlador específico para un sistema de información. Las operaciones básicas que define la especificación son: gestión de las conexiones de acceso a JCA, seguridad, transacciones, multiproceso, paso de mensajes y portabilidad dentro de los servidores de aplicaciones.
- 2) **JMS** (en inglés *Java Message Service*). El servicio de mensajes Java emplea colas de mensajes para el traspaso de información entre componentes software estableciendo una infraestructura MOM con dos modelos de API, Punto a punto, entre dos únicos clientes o Publicador/subscriptor donde varios clientes según su papel envían o leen mensajes.
- 3) **Servicios web**. Permiten la comunicación entre sistemas heterogéneos a través del acceso a la URL de aplicaciones



empleando protocolos basados en XML como SOAP o SAAJ. Los clientes acceden al servicio a partir de su interfaz definida ante WSDL (en inglés *Web Service Definition Language*) o insertada en algún registro de servicios web.



**Figura 6: Modelo de desarrollo en capas.**

**TRADUCCIÓN TEXTO FIGURA 6:** LÓGICA DE NEGOCIO. Servidor de aplicaciones. Transacciones. Ficheros. Servicios web.

### 27.3.2 Servidores de aplicaciones.

El servidor de aplicaciones será el encargado de soportar la mayoría de las funcionalidades y servicios de la tecnología JEE, siendo el núcleo de esta arquitectura. Cuando un servidor de aplicación implementa la tecnología JEE tiene que proporcionar todos los componentes definidos en la especificación y por tanto cualquier aplicación JEE podrá desplegarse y ejecutarse en el mencionado servidor.

El servidor de aplicaciones dispondrá de diferentes contenedores para Applets y aplicaciones clientes, web y EJB, siendo estos últimos los que se encargarán de operar con la lógica del dominio, gestión de transacciones,

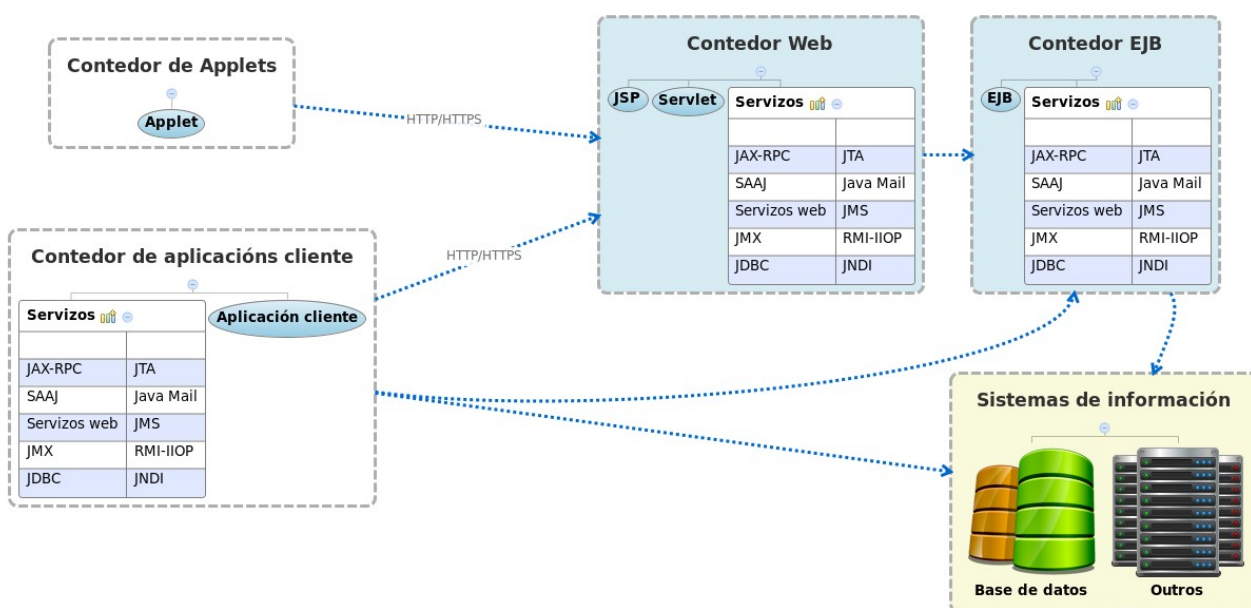
persistencia, control del flujo, etc.

- a) **Tomcat.** Sin presentar todas las funcionalidades de un servidor de aplicaciones, este servidor libre de Apache incorpora el servidor web Apache y soporte para JSP y Servlets con el contenedor Catalina. Presenta diferentes módulos de soporte de aplicación como seguridad SSL, SSO, JMX, AJP, JSF, conector Coyote para peticiones HTTP, soporte para Comet, Recolector de basura reducida así como herramientas web para despliegue y administración.
- b) **JBoss.** Uno de los servidores de aplicaciones libres de uso más extendido compuesto por un Contenedor de Servlets para JSP y Servlets y un Contenedor de Beans. A diferencia de Tomcat implementa todo el conjunto de servicios especificados por JEE. Como contenedor de Servlets emplea una adaptación de Tomcat o el contenedor Jetty. Entre los módulos y funcionalidades que soporta destaca que permite la creación de *cluster*, soporte EJB, JMX, Hibernate, JBoss AOP para dotar a clases Java de persistencia y funcionalidad transaccional, sistema caché, JSF, Portlets, JMS, Servidor de correo, gestión de contenidos foros y portales, entre otras muchas.
- c) **Geronimo.** Otro producto libre de Apache, compatible con JEE que incluye JDBC, RMI, JM, Servicios web, EJB, JSP, Servlets y otras tecnologías. La principal característica de este servidor es que integra un gran número de otras soluciones ya existentes: Tomcat y Embarcadero como contenedores web, OpenEJB como contenedor de Servlets, OpenJPA, Apache Axis, Apache CXF y Scout Apache para servicios web, Derby para el acceso a datos, y WADI para establecer clusters y balanceo de carga, entre otros.
- d) **JonAS.** Otra alternativa libre a JBoss, aunque no soporta por completo JEE. Permite integración con Tomcat o Jetty como contenedores web y tiene contenedor de EJB. Entre módulos y

servicios incorpora: Xplus, Hibernate, TopLink, OpenJPA, JORAM como implementación de JMS, varios protocolos RMI (IIOP, JRMP, IRMI), soporte LDAP, servicios web Axis y otros muchos.

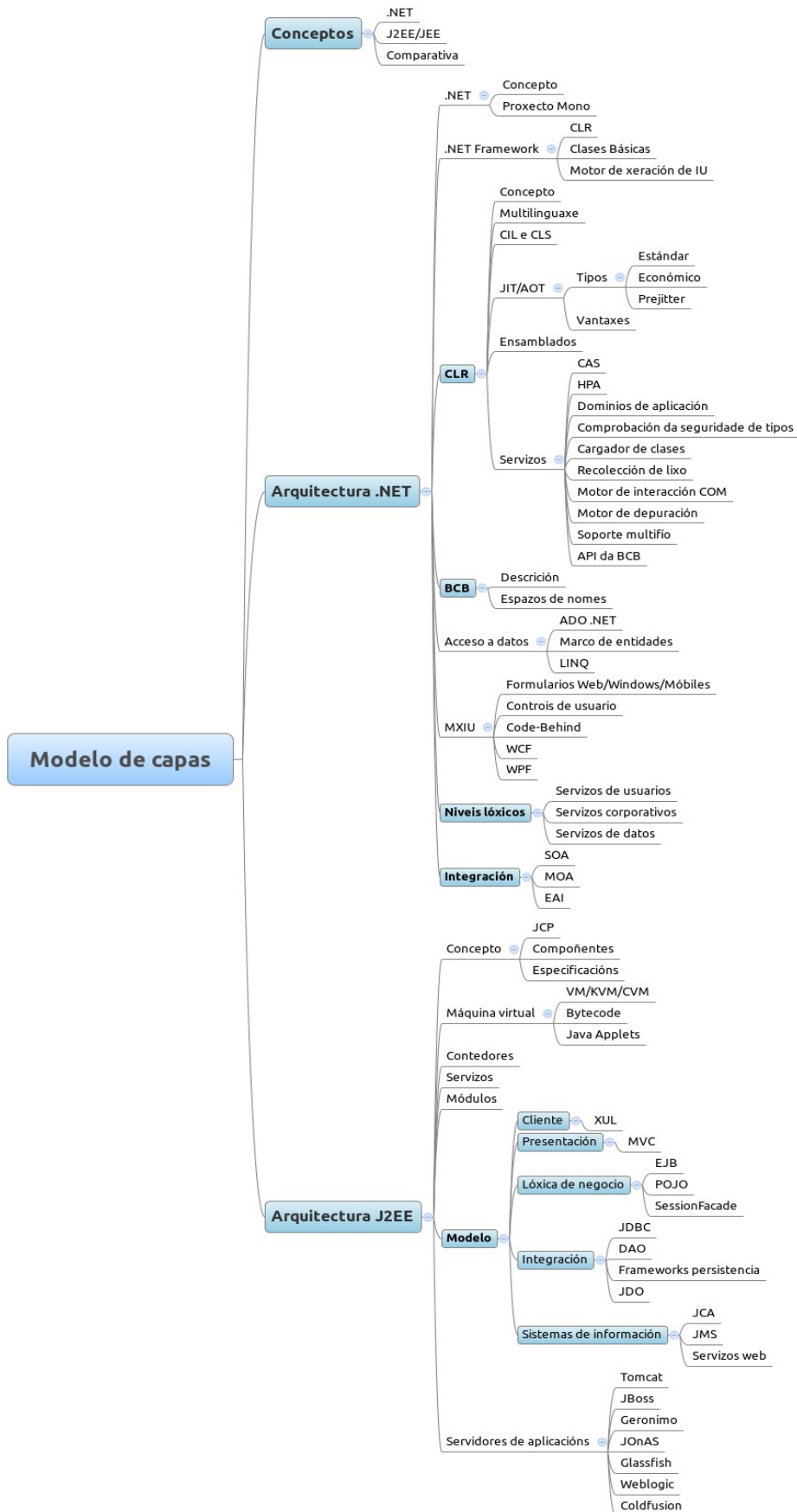
- e) **Glassfish.** Alternativa libre de Sun, ahora Oracle, que tiene como base el *framework* para persistencia Toplink. Incorpora además módulos para soporte EJB, JAX-RS, JSF, RMI, JMS, servicios web, en la línea de los anteriores, y novedades como Apache Félix, una implementación de OSGi (en inglés *Open Services Gateway*) y Grizzly que hace uso de la nueva API de Java de E/S (NIO) para mejorar la escalabilidad.
- f) **WebSphere.** Alternativa comercial de IBM con una versión de libre distribución. La versión libre, más ligera, se basa en el servidor Geronimo diferenciándose de este en que incluye soporte para DB2, Informix, soporte RAC de Oracle y otras bases de datos así como mejores librerías para XML. Otras tecnologías serían: los Servlets SIP (en inglés *Session Initiation Protocol*) que utilizan elementos multimedia en tiempo real, mensajería instantánea y juegos on line; el *framework* Spring; protocolos de seguridad Kerberos y SAML. La diferencia con otros servidores es que posee herramientas de administración más avanzadas, sobre todo para sistemas en cluster y soporte para *mainframes*.
- g) **Weblogic.** Alternativa comercial de Oracle basada en Glassfish, incorporando servicios del Weblogic server sobre la JVM JRockit. En concreto Weblogic Server proporciona los Servicios web Oracle WebLogic Server Services Web, la Application Grid como solución de grid de datos, soporte de conectividad con Tuxedo (WTC), soporte de RAC para Oracle, SAML, una API de integración con .NET a JMS.NET, Spring y el *framework* de diagnóstico WLDF.
- h) **Coldfusion.** Alternativa comercial de Adobe de las más valoradas actualmente, diferenciándose por el soporte a tecnologías RIA

principalmente Flash. Implementa parte de los servicios JEE pero puede integrarse con otros servidores de aplicaciones como WebSphere o Jboss, pudiendo desplegarse cómo aplicación Java. Además lleva incorporado el servidor de aplicaciones Adobe JRun. Destaca por el soporte en tecnologías AJAX, Flex, PDF, RSS, Flash Remoting, integración .NET y herramientas de administración avanzadas.



**Figura7: Arquitectura J2EE**

## 27.4. ESQUEMA



**TRADUCCIÓN TEXTO ESQUEMA:** Proyecto Mono. Motor de generación de IU. Multilenguaje. Ventajas. Comprobación de la seguridad de tipos. Recolección de basura. Soporte multihilo. Descripción. Espacios de nombres. Controles de usuario. NIVELES LÓGICOS. Servicios de usuario. Servicios Corporativos. Servicios de datos. Componentes. Especificaciones. Contenedores. Servicios. Lógica de negocio. Servidores de aplicaciones.

## **27.5. REFERENCIAS**

Varios autores.

Biblioteca MSDN de Microsoft. (2003).

Jef Ferguson y otros.

La biblia de C#. (2003).

Benjamín Aumaille.

J2EE. Desarrollo de aplicaciones Web. (2002).

I. Singh, B. Stearns y otros.

Desingning Enterprise Applications with the J2EE Platform. (2002).

**Autor: Juan Marcos Filgueira Gomis**

**Asesor Técnico Consellería de Educación e O. U.**

**Colegiado del CPEIG**

# **28. ARQUITECTURA SOA. SERVIZOS WEB. TECNOLOGÍAS XML.**

## TEMA 28. ARQUITECTURA SOA. SERVICIOS WEB. TECNOLOGÍAS XML.

### 28.1 INTRODUCCIÓN Y CONCEPTOS

### 28.2 ARQUITECTURA SOA

### 28.3 SERVICIOS WEB

### 28.4 TECNOLOGÍAS XML

### 28.5 ESQUEMA

### 28.6 REFERENCIAS

#### 28.1. INTRODUCCIÓN Y CONCEPTOS

Los sistemas actuales tienen una gran complejidad debido a la integración de múltiples componentes heterogéneos. La comunicación y relación entre estos componentes es uno de los grandes problemas actuales siendo **SOA** (en inglés *Service Oriented Architecture*) uno de los actuales modelos de solución. Esta arquitectura entiende la comunicación entre aplicaciones y componentes como servicios, no necesariamente servicios web, demandados por clientes o subscriptores y proporcionados y publicados por proveedores. Las arquitecturas para servidores de aplicaciones de uso más extendido como .NET y JEE acostumbran a definir una **capa de integración** que agrupa los componentes encargados del acceso a datos, sistemas *legacy*, motores de reglas *de workflow*, acceso a LDAP, etc. Para la comunicación de los componentes de esta capa existen varias soluciones como JCA, JMS y servicios web, está última una de las más aceptadas actualmente.

Los **Servicios web** permiten la comunicación entre sistemas heterogéneos a través del acceso a la URL de aplicaciones empleando protocolos basados en XML como SOAP o SAAJ. Los clientes acceden al servicio a partir de su interfaz definida ante WSDL (en inglés *Web Service Definition Language*) o insertada en algún registro de servicios web.



El uso de XML se convierte en un estándar de integración, siendo la base de las comunicaciones en esta capa tanto para estructurar como para almacenar e intercambiar información, extendiendo su uso a otros ámbitos. Las **tecnologías XML** son un conjunto de módulos que ofrecen servicios como: XSL/XSLT para diseño de documentos, Xpath como lenguaje de rutas para acceso a documentos, el lenguaje de consulta XQL, y otros como XLink o XPointer.

## **28.2 ARQUITECTURA SOA**

SOA define una arquitectura orientada a servicios que busca simplificar el modelo de integración de sistemas distribuidos heterogéneos. En esta arquitectura los componentes publican e invocan servicios en la red a través de mecanismos de comunicación como JCA, JMS, SOAP, RPC o Servicios web. Los servicios son funcionalidades de la lógica de negocio que pueden invocarse de manera remota para obtener un resultado. Se definen ante una interfaz explícita, por ejemplo a través de WSDL, independiente de su implementación empleando estándares de comunicación basados en XML. SOA define tres **bases** fundamentales:

- 1) **Orientación al intercambio de mensajes.** La base del sistema es la comunicación entre los nodos del sistema.
- 2) **Abstracción de componentes.** Cada sistema se reduce a su interfaz y el conjunto de servicios que define, con lo cual permite la integración entre cualquier tipo de sistema.
- 3) **Metadatos.** Descripciones e información asociada a servicios y mensajes, mejorando la capacidad semántica del sistema.

A nivel lógico los principales componentes en una arquitectura SOA son:

- a) **Servicios.** Entidades o funcionalidades lógicas definidos en interfaces públicas, que pueden o no requerir autenticación.
- b) **Proveedor de servicios.** Componente software que implementa un

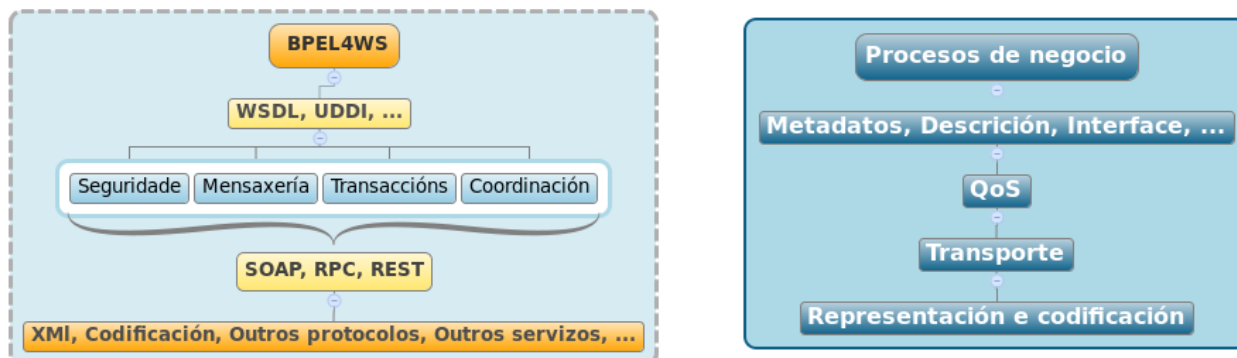
servicio y publica su interfaz.

- c) **Ciente de servicios.** Componente software que invoca un servicio de un proveedor.
- d) **Localizador de servicios.** Proveedor de servicios que registra las interfaces y permite a los clientes buscar en el registro y acceder a su localización.
- e) **Servicio de interconexión.** Proveedor que comunica solicitudes de servicio a otros proveedores.

El concepto de BPM o Gestión de procesos de negocio (en inglés *Business Process Management*), está muy relacionado con SOA. BPM es un modelo de gestión centrado en procesos de negocio y de cómo integrar sus funcionalidades en sistemas heterogéneos. A partir de la identificación y gestión de los procesos de la organización puede implantarse una solución BPM a través de una arquitectura SOA. Fruto de esta idea aparecen soluciones como:

- ✓ **BPMN.** Nota para el modelado de procesos de negocio.
- ✓ **BPEL.** Lenguaje de ejecución de procesos de negocio con servicios web para la orquestación de servicios. Generalmente se realiza una conversión de BPMN a BPEL.
- ✓ **BPEL4WS.** Lenguaje de definición y ejecución de procesos de negocios empleando servicios web (en inglés *Business Process Execution Language for Web Services*). BPEL4WS es resultado de la convergencia de WSFL (en inglés *Web Services Flow Language*) y XLANG, permitiendo componer Servicios web como servicios compuestos denominados Servicios de negocio.

## Arquitectura SOA



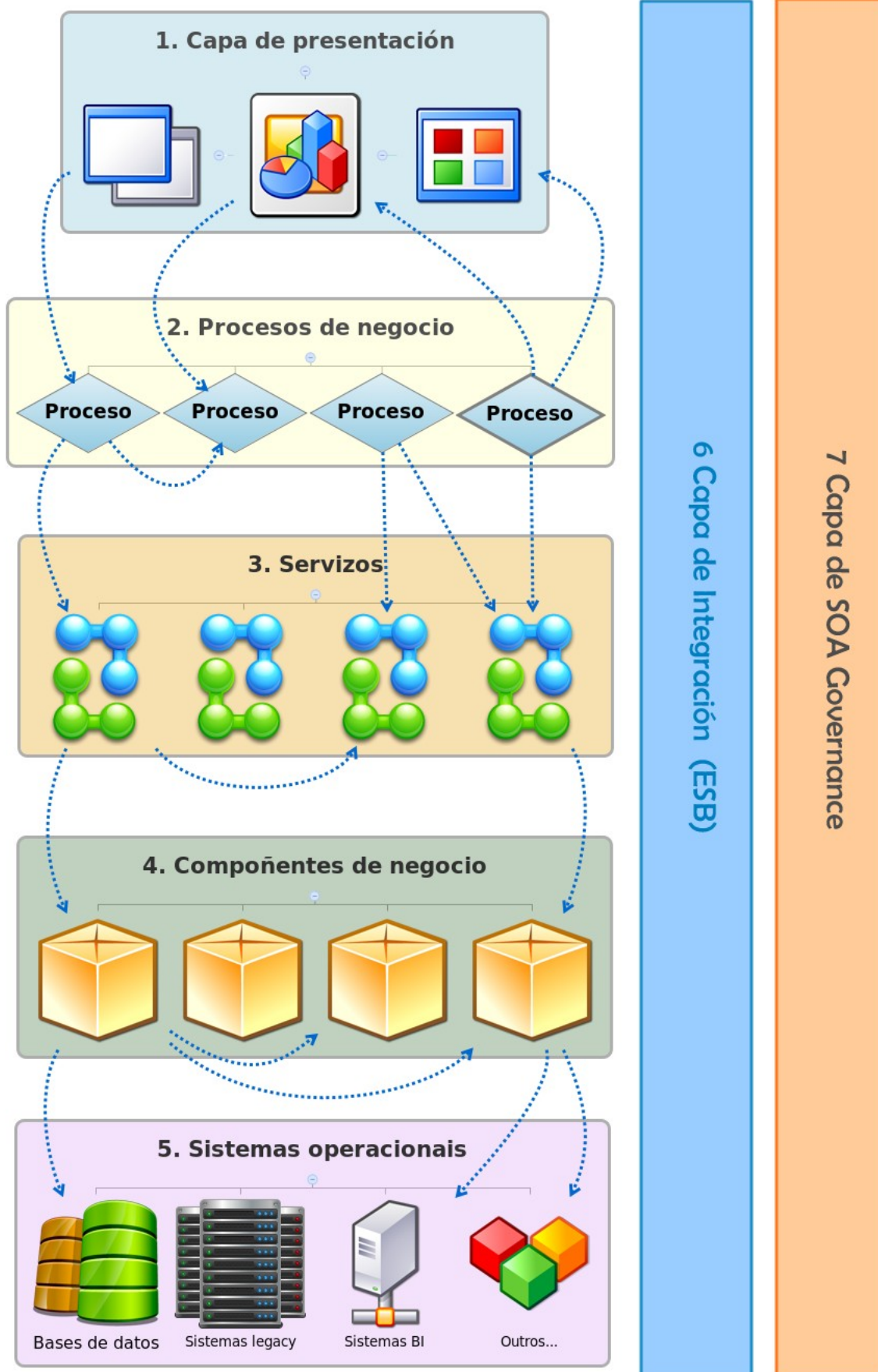
**Figura 1: Arquitectura SOA.**

**TRADUCCIÓN TEXTO FIGURA 1:** Seguridad. Mensajería. Transacciones. Otros protocolos. Otros servicios. Descripción. Interfaz.

La integración en la arquitectura SOA de los BPM permite establecer lo que se da en llamar **SOA Governance**, una estructura para la toma de decisiones y establecimiento de responsabilidades en la organización a través de la implantación de políticas, monitorización de servicios, incorporación de buenas prácticas, principios arquitectónicos, avance continuo de los procesos de negocio, etc. En definitiva, análisis y diseño de soluciones que permitan cumplir con éxito a implantación de SOA en una organización.

A nivel **conceptual SOA** describe una serie de guías o patrones para servicios alineados con un modelo de negocio. Un modelo conceptual de SOA permite definir un diseño en múltiples capas donde los servicios se relacionan con procesos de negocio y sistemas de información. La división más habitual considera 7 capas diferenciadas, que se pueden ver gráficamente en la Figura 2:

- 1) **Capa de Presentación.** Interfaces de usuario en sitios web, aplicaciones y portales que invocan funcionalidades de los procesos de negocio.



**Figura 2: Modelo conceptual SOA.**

- 2) **Capa de Procesos de negocio.** Representa los procesos y flujos operativos o *workflows* que invocan los clientes desde la capa de presentación u orquestación de servicios. Los procesos por norma general se representarán con BPEL y se implementarán con alguna herramienta de transformación.
- 3) **Capa de Servicios.** Funcionalidades de los componentes de la lógica de negocio que se publican para uso de los clientes. Se referencian a partir de la interfaz siendo transparentes a su implementación.
- 4) **Capa de Componentes de negocio.** Son los encargados de proporcionar las funcionalidades que se publicarán en los servicios así como cualquier otra intermedia o común al sistema. A este nivel irían los servidores de aplicaciones, otros servicios web, aplicaciones, paquetes y librerías.
- 5) **Capa de Sistemas operacionales.** En este nivel irían los sistemas de información de la organización, sistemas *legacy*, sistemas CRM o ERP, aplicaciones de BI, etc.
- 6) **Capa de Integración.** Agiliza la integración de servicios a través de sistemas tipo Buses de Servicios Empresariales o ESB, que realizan funciones de enrutamiento, monitorización y administración, transformaciones de los mensajes, etc. dentro del área de comunicaciones.
- 7) **Capa de SOA Governance.** Realiza funciones de administración, monitorización y control de la calidad del servicio en áreas como seguridad, disponibilidad y otros factores generales no recogidos en la capa de integración.

Existen **patrones de diseño** tomando como punto de partida esta arquitectura, se trata de patrones para Servicios web y patrones **POSA** (en inglés *Service-Oriented Architecture Patterns*). Algunos de los principales son:

- **Service Oriented Architecture.** Patrón que define la arquitectura SOA estableciendo reglas, relaciones y dependencias entre los componentes del sistema. Permite buscar servicios dinámicamente con independencia de la plataforma y sin requerir implementación, con transparencia. Este patrón es una variante ampliada del **Broker Pattern** de POSA. En este patrón un Servicio o nodo intermedio ayuda a localizar el servicio y puede obligar a realizar todas las comunicaciones a través de él o bien una vez establecida dejar que esta sea directa entre el cliente y el servicio.
- **Architecture Adapter.** Patrón genérico que facilita la comunicación entre diferentes arquitecturas gracias a la independencia de usar XML/SOAP y la generación de clases proxy. Este patrón es implementado por *frameworks* para Servicios web como Apache Axis (Java).
- **Service Directory.** Facilita la localización de Servicios web a partir de una especificación fuerte de las interfaces a través del catálogo UDDI de interfaces WSDL.
- **Service Factory.** Permite la selección de servicios del proveedor aislando el código de comunicación UDDI. Del mismo modo el patrón de extendido Service Factory Cache hace funciones de caché en el servicio. Simplifica en parte a API del patrón Service Directory.
- **Service Facade.** Proporciona un servicio web controlador que actúe como punto de entrada de la lógica de negocio u objeto de fachada. Puede emplear simultáneamente otros mecanismos de comunicación como CORBA.
- **Event Monitor.** Se emplea para notificar que un Servicio web de larga duración invocado remotamente completa la solicitud. Cuando el Servicio no dispone de mecanismos de notificación hace falta establecer un intermediario.
- **Business Object.** Un BO engloba un concepto del dominio, equiparable a un VO para entornos distribuidos.

- **Business Process.** Un BP engloba un proceso de la lógica de negocio, representando la jerarquía formada por las diferentes implementaciones de sus funcionalidades y la interfaz del servicio.
- **Asynchronous Business Process.** Este patrón se encarga de gestionar la llamada y notificación de respuesta al cliente cuando estas pueden ser de larga duración.
- **Business Object Collection.** Agrupa diferentes procesos de negocio en un mismo BOC.
- **Observer Services.** Se basa en un registro de servicios donde el observador notifica al cliente sobre eventos que derivados de los servicios en los que está registrado.
- **Publish/Suscribe Services.** Evolución del patrón Observer Services incorporando un sistema de notificaciones para sustituir al registro. Se emplea cuando los servicios web no incorporan un sistema de notificación y precisan un intermediario.
- **Data Transfer Object.** Permite enviar múltiples objetos en una misma llamada reduciendo el número de conexiones.
- **Partial Population.** Permite que los clientes seleccionen parte de la información del mensaje de respuesta a la solicitud de servicio buscando un mejor aprovechamiento del ancho de banda.
- **Microkernel.** Separa un núcleo de funcionalidad mínimo de partes especificadas por el cliente.
- **Web Service Interface.** Proporciona una interfaz que puede emplearse desde los clientes para invocar los métodos de un proxy de Servicio web genérico en lugar de depender de la clase proxy generada a partir de la WDSL.

REST (en inglés *Representation State Transfer*) representa un modelo de comunicación donde cada petición HTTP contiene la información necesaria para responder a la petición sin tener que almacenar el estado de la sesión. En REST todos los servicios son recursos, identificados por URIs y se diseñan sus representaciones mediante XML, JSON o microformatos. REST



representa una arquitectura SOA que no hace uso de Servicios web, SOAP ni RPC.

Actualmente dentro del marco de la Web 2.0 surge una nueva variante en las arquitecturas SOA, el concepto de **Mashup**, un sitio o aplicación web que hace uso de contenido de otras aplicaciones o servicios vía HTTP. Este contenido es recuperado en un modelo de Servicios web a través de su API pública evitando caer en el Web Scraping. Para emplear los Mashups como XML se emplean lenguajes específicos como EMMML (en inglés *Enterprise Mashups Markup Language*). Las arquitecturas Mashup constan de tres componentes:

- ✓ **Los proveedores de servicios.** Orígenes de datos que publican a través de una interface los métodos de acceso a sí mismos y permiten su consulta vía Atom, RSS, REST, JSON, Bases de datos o interfaces WSDL de Servicios web.
- ✓ **Aplicación o Servicio web Mashup.** Proporciona un nuevo servicio a partir de la información obtenida de los proveedores.
- ✓ **Clientes.** Usuarios finales, u otras aplicaciones o servicios que hacen peticiones al Mashup. En los clientes acostumbran a emplearse tecnologías RIA del tipo de AJAX o Comet.

Otro concepto que se puede relacionar con SOA es el de la **Nube** (en inglés *Cloud Computing*). La nube se fundamenta en emplear la red Internet para publicar servicios, que pueden o no requerir identificación. En la nube todo son servicios, aplicaciones, bases de datos, redes, y se gestionan y se acceden como tal. La arquitectura de la nube se estructura habitualmente en tres capas:

- 1) **Software como servicio** o SaaS (en inglés *Software as a Service*). Sería el nivel más alto, orientado a los usuarios y clientes finales. Se incluirían las aplicaciones propias y aplicaciones de terceros del estilo de Google Aps. La misma infraestructura del proveedor sirve a

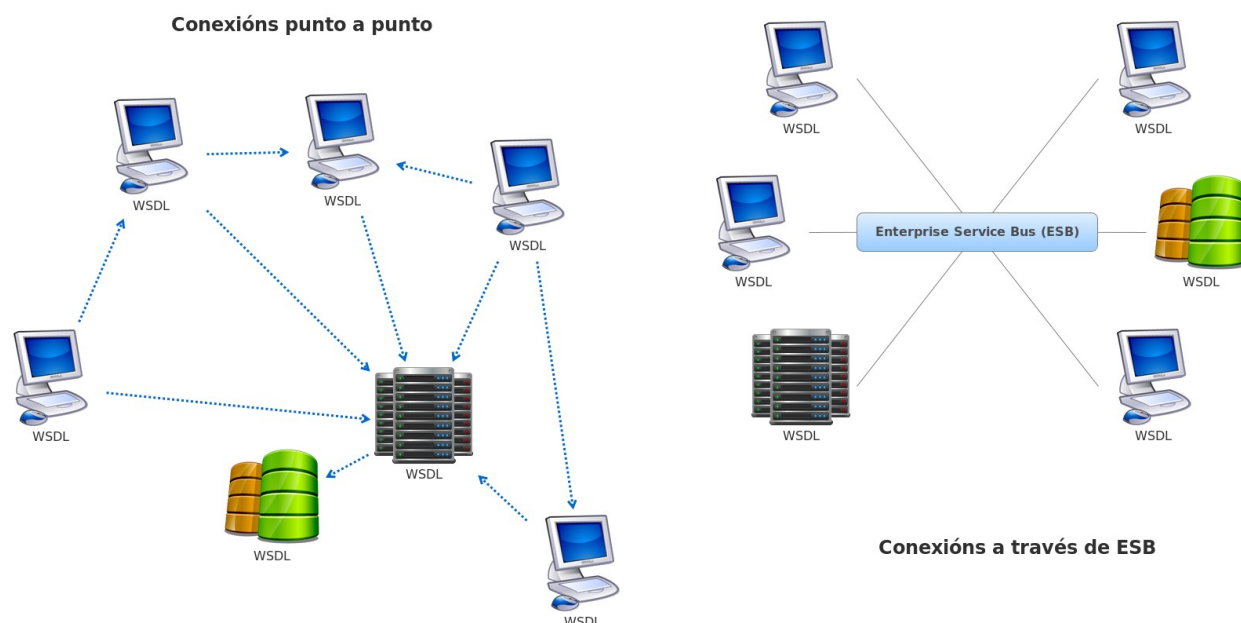


múltiples organizaciones finales.

- 2) **Plataforma como servicio** o PaaS (en inglés *Platform as a Service*). Serían la capa intermedia encargada de encapsular sistemas o *middleware* permitiendo correr aplicaciones sobre ellas. Ejemplos de este servicio serían Google App Engine o Windows Azure. De este modo una organización externa proporciona un servicio de infraestructura y soporte para otras organizaciones.
- 3) **Infraestructura como servicio** o IaaS (en inglés *Infrastructure as a Service*). En este caso se ofrecen servidores para computación, red, almacenamiento o bases de datos a través de diferentes técnicas, por ejemplo a través de máquinas virtuales.

El **Bus de Servicios Empresariales** o ESB (en inglés *Enterprise Service Bus*) representa otras de las características de SOA, aunque no es imprescindible en una arquitectura de este tipo. Se trata de un componente que hace abstracción del sistema de mensajes de la organización a través de un sistema único para todos los elementos de un sistema SOA. El ESB proporciona funciones de transformación, adaptación, conexión y enrutamiento que pueden ser implementadas en SOA. En una arquitectura SOA con un ESB todas las aplicaciones y servicios se conectan a un punto único y central que administra las comunicaciones, realizando funciones de *middleware*. El ESB se construye sobre tecnologías XML, XSLT, XPath, JMS o propias de Servicios web. Hace uso de elementos denominados Contenedores de Servicios o Brokers que hacen la función de servidores de comunicaciones. Entre los servicios que proporciona se encuentran:

- ✓ Funcionalidades de enrutamiento, fraccionamiento y combinación de mensajes bajo la base de los patrones EIP (en inglés *Enterprise Integration Pattern*).
- ✓ Funciones de supervisión y control de la calidad del servicio a través de Acuerdo de Nivel de Servicios o SLA de los servicios.
- ✓ Funciones de monitorización, seguridad y mediación de protocolos.



**Figura 3: Bus de Servicios Empresariales (ESB).**

**Traducción texto Figura 3:** Conexiones punto a punto. Conexiones a través de ESB

El Bus sustituye la comunicación directa entre dos aplicaciones o servicios, de manera que la comunicación se hace de modo transparente a través del ESB. Emplea un sistema de mensajería, como por ejemplo Tibco, soportando varios MEP o patrones de intercambio de mensajes, así como colas para reenviar las peticiones a los proveedores de servicios y las respuestas a las solicitudes a los clientes. Existen *frameworks* que recogen los componentes precisos para implantar un ESB como Mule ESB, este es un *framework* ligero destinado a mensajería y control de eventos. Permite integraciones con otros *frameworks* como Struts o Spring y soporta muchos componentes de servicio como JMS, SOAP, BPEL, JBI (en inglés Java Business Integration), y otros.

Por último hay que señalar que las arquitecturas SOA pueden completarse con módulos específicos según las necesidades de la organización, como:

- ✓ **Seguridad.** Con la adopción de diferentes tecnologías, SSL, Kerberos, X.509, Firmas XML, Encriptación XML, XML Canonicalization, SAML (en inglés *Security Assertion Markup Language*) o XKMS (en inglés *XML Key Direction Specification*), que administra la llave pública o PKI de las infraestructuras.
- ✓ **Orquestación y coreografía de servicios.** En este modelo la interacción entre servicios no se produce directamente sino que se define una entidad que define la lógica de interacción, facilitando la colaboración que será un servicio de control primario. En BPEL el servicio primario será un proceso BPEL, pero también se puede definir con BPEL4WS, WSFL o XLANG. Mientras la orquestación precisa de un director de orquesta o servicio central, el modelo de coreografía establece interacciones punto a punto a partir de reglas de colaboración generales. Para la coreografía existen lenguajes específicos como WS-CDL (en inglés *Web Services Choreography Description Language*) que tienen definida la forma de representar las interacciones.
- ✓ **Gestión transaccional.** Existen varias tecnologías que coordinan las transacciones entre servicios autónomos. El BTP (en inglés *Business Transaction Protocol*) donde ninguno de los servicios gestiona una transacción, sino que esta se comunica a todos y deciden si se unen o no, con comunicaciones basadas en XML en un formato propio. Otros mecanismos como WS-Transaction y WS-Coordination se encargan de gestionar transacciones llevadas a cabo por varios servicios a la vez, con protocolos SOAP y WSDL. Por su parte JEE dispone de la especificación JAXTX para transacciones complejas con el objetivo de aislar éstas de los contenedores.

## **28.3 SERVICIOS WEB**

Los Servicios web son uno de los modelos de implementación de SOA. Un Servicio web que proporciona un servicio vía web en una red a través de

una interfaz que le permite recibir peticiones y transmitir respuestas. Para soportar este sistema se desarrollaron una gran variedad de protocolos y tecnologías. Los principales son el HTTP/HTTPS para peticiones y respuestas y el XML como formato de intercambio. Los principales **componentes** comunes a los servicios web serían:

- a) **SOAP** (en inglés *Simple Object Access Protocol*). El protocolo de comunicación, sobre la capa de transporte basado en XML, que sirve para invocar los servicios a través de un protocolo, siendo los más habituales HTTP o SMTP, pero realmente es independiente y permite otros como POP3 o JMS. Permite tanto describir el contenido del mensaje y reglas de codificación de los tipos de datos, como aspectos de seguridad y transaccionalidad. Se encuentra estandarizado por el W3C, lo que garantiza la comunicación entre sistemas heterogéneos que lo implementen.
- b) **UDDI** (en inglés *Universal Description, Discovery and Integration*). Directorio donde se publican los servicios proporcionando la información necesaria para permitir su invocación. Presenta dos API que permiten a los servicios publicar sus funcionalidades y a los clientes enviar las peticiones y obtener los resultados. Cada servicio se publica en el UDDI proporcionando la URL de su WSDL y meta-información. De manera general se entiende que UDDI proporciona tres tipos de servicios: información general sobre los proveedores de los servicios (páginas blancas), categorías y clasificaciones de servicios (páginas amarillas) y las reglas de negocio o información técnica sobre los servicios (páginas verdes).
- c) **WSDL** (en inglés *Web Services Description Language*). Lenguaje basado en XML y XML Schema que permiten la descripción de la interfaz de los Servicios web y que está estandarizado por el W3C. En un documento WSDL se definen los tipos de datos, los mensajes, los *endpoints*, los *bindings* y los servicios.
- d) **Serialización de datos**. Se emplean definiciones de XML Schema

para especificar cómo codificar los datos en conjunción con las reglas de codificación de SOAP. Aunque el mecanismo más habitual sea el SOAP Document/Literal existen otros mecanismos como: RPC/Encoding, Document/Encoding o RPC/Literal.

Según lo visto anteriormente para las arquitecturas SOA en general, se pueden definir varios tipos de servicios según su complejidad, comenzando por los de nivel básico a los de niveles más complejos.

	<b>Servicios de nivel básico</b>	<b>Servicios de alta complejidad</b>
<b>Función</b>	Integración de la funcionalidad de una aplicación	Elemento llave de una arquitectura SOA
<b>Protocolos y tecnologías</b>	SOPA, UDDI, WSDL	ebXML, BPEL, BTP, RossetaNet, Apache Axis, ...
<b>Tipo de contenido</b>	Plano	MIME, PDF, ...
<b>Comunicaciones</b>	Punto a punto	Multiparty, ESB, ...
<b>Mensajería</b>	JMS, RPC, ...	Colaboración y <i>workflows</i>
<b>Transaccionalidad</b>	No transaccional	Transaccional
<b>Seguridad</b>	SSL, autenticación, ...	Firma digital, XML-encryption, Kerberos, ...

***Tabla 1: Complejidad de los servicios web.***

Según el tipo de comunicaciones las APIs más habituales son:

- API de mensajería.** Clientes y servicios disponen de sistemas de mensajería que les permiten comunicarse en formato XML. Al estar orientadas a los sistemas de mensajería presentan una alta QoS.
- API de RPC.** La solución más habitual, que emplea un compilador

intermedio de WSDL para generar el *stub* y el *skeleton* para cliente y servidor respectivamente, tal y como sucede con CORBA. Este sistema es el que acostumbran a emplear los *frameworks* actuales como Apache Axis.

- c) **API para servidores de aplicaciones (JEE/.NET).** Estas API vienen disponibles en las bibliotecas de clases de cada arquitectura, por ejemplo en JEE se disponen de: JAXM (en inglés *Java API for XML Messaging*) para intercambio de mensajes; JAX-RPC (en inglés *Java API for XML-based RPC*), que permite enviar peticiones remotas a terceros y recibir resultados; y JAXR (en inglés *Java API for XML Registries*), que proporciona acceso a registros de negocio y mecanismos para compartir información.

El **proceso de implementación de un servicio web** consiste en implementar las funcionalidades del servicio reutilizando las clases generadas a partir de un WSDL o de una API (JAX-RPC, Axis,...). Se pueden aprovechar las herramientas existentes en los ID, u otras más específicas con Ant o Wsdl2java. Una vez implementadas las clases con la lógica del servicio se generan las clases en un war y se despliegan en un contenedor de Servlets o en un IIS. Sobre el modelo de programación se emplean diferentes variantes:

- a) **Estilo CORBA.** Se generan todas las clases al compilar empleando clases de las API (Axis, JAX-RPC,...)
- b) **Dynamic Proxy.** La interface WSDL se crea al compilar, pero el proxy en el cliente sólo se compila en tiempo de ejecución.
- c) **Dynamic Invocation Interface.** Tanto WSDL como cliente se generan en tiempo de ejecución. El cliente busca e invoca el servicio vía *broker*.

Otro de los factores a considerar es el tema de la **seguridad** en los

Servicios web, que por la propia naturaleza de las arquitecturas SOA resulta un tema complejo. Los principales elementos de seguridad en lo que respecta la JEE, aunque muchas serían extensibles la .NET, serían:

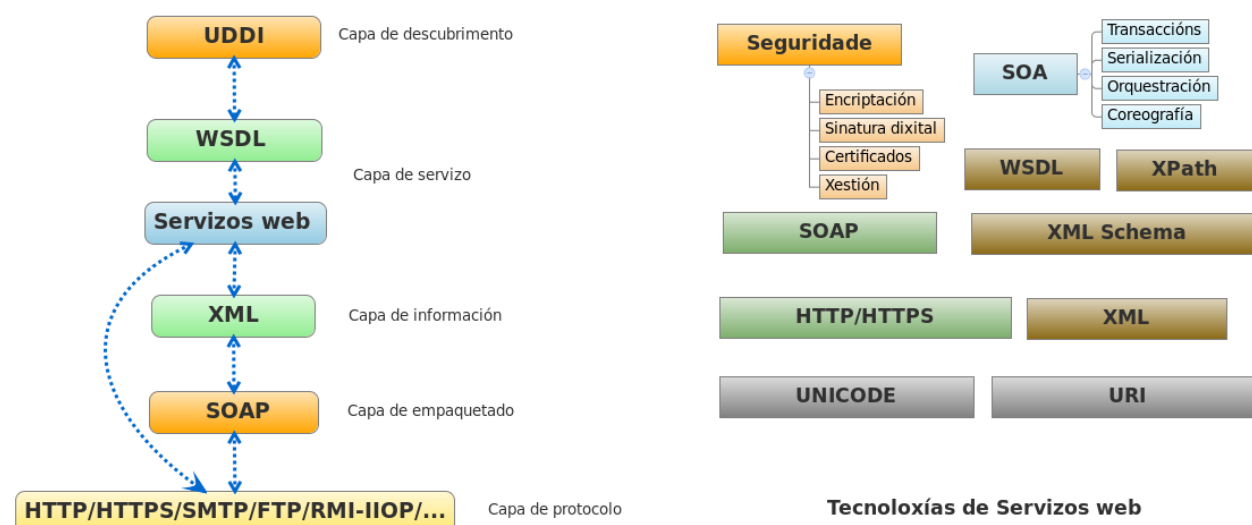
- ✓ Para JAX-RPC a API **XWS-Security** que facilita la integración de aspectos de seguridad.
- ✓ El estándar **XML-DigitalSignature** para firma digital.
- ✓ El estándar **XML-Encrytion** para encriptación de mensajería.
- ✓ **Certificados X.509** para autenticación.
- ✓ Bases de datos de certificados basadas en **JKS** (en inglés *Java Key Store*).

Dentro de la arquitectura los diferentes mecanismos se integrarían nivel a nivel de la siguiente manera:

- 1) **Nivel de transporte.** Autenticación básica, autenticación por certificado vía SSL/TLS. Codificación de usuario/contraseña en los *stubs* y reglas de seguridad para *endpoints*.
- 2) **Nivel de mensaje.** Firma de contenidos con certificados XML-DigitalSignature, certificados X.509 y encriptación.

Entre las **posibilidades** existentes para implementar Servicios web se encuentran:

- ✓ APIs Java: JAX-RPC, JAXM, SAAJ (mensajes SOAP como objetos), JWSDL (Acceso a descripciones WSDL), JAXR (Acceso al UDDI), *framework* Apache Axis,...
- ✓ .NET: ASP .NET, MS SOAP Toolkit,...
- ✓ Otras tecnologías: NuSOAP para PHP, Axis para C++,...



**Figura 4: Tecnologías de Servicios web.**

**Traducción Texto Figura 4:** Capa de descubrimiento. Capa de servicio. Seguridad. Firma digital. Gestión. Transacciones. Orquestación. Tecnologías de servicios web.

## 28.4 TECNOLOGÍAS XML

El lenguaje XML (en inglés *extensible Markup Language*) es un metalenguaje para etiquetado desarrollado por el W3C. En SOA el XML representa el estándar para intercambio de información estructurada entre sistemas heterogéneos. En esencia XML es un lenguaje de marcas que permite la creación de otros lenguajes de marcas, con diferentes usos en SOA, cada uno de estos lenguajes se denomina Aplicación XML y representa un modelo de datos de acuerdo a un esquema semántico.

El XML resulta más estricto que otros lenguajes como HTML, admitiendo varios mecanismos de validación o corrección:

- ✓ **Formación.** Un documento XML se denomina “bien formado” cuando



sigue las reglas léxicas (tipo de caracteres, codificación,...) y sintácticas (nidación correcta, marcas de apertura y cierre de estructura,...) debidas.

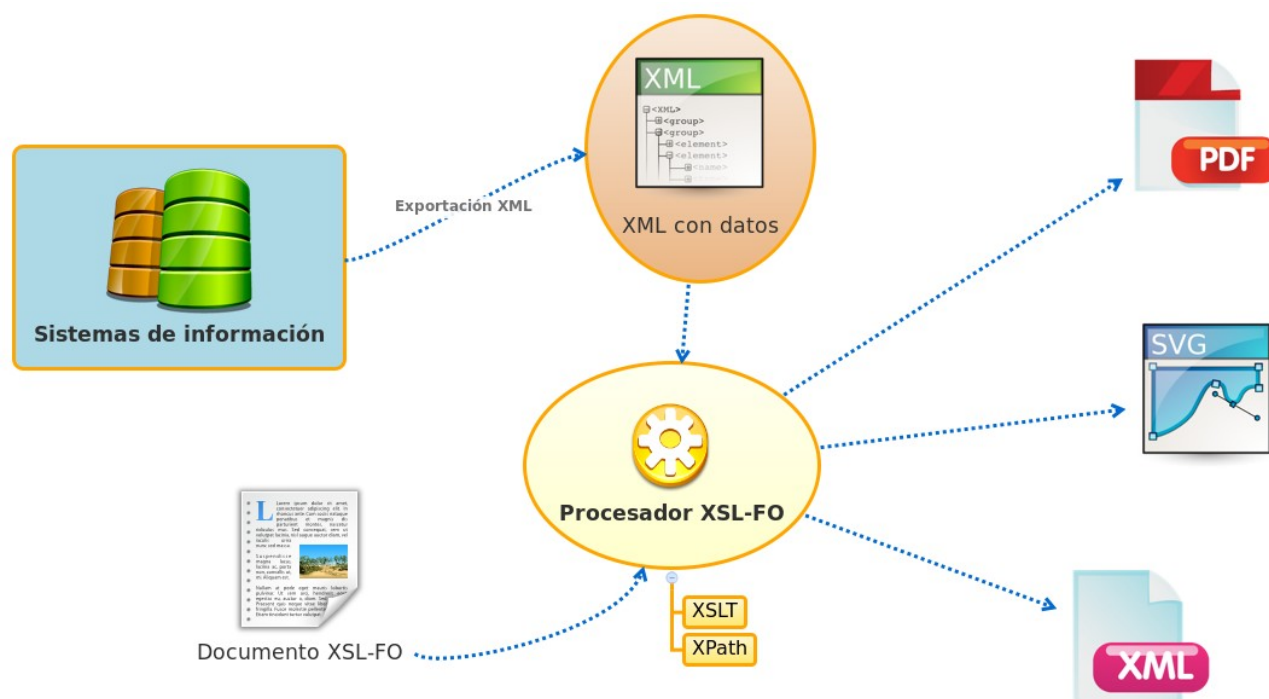
- ✓ **Validación.** Un documento XML se considera “validado” cuando cumple un conjunto de reglas y limitaciones denominadas en conjunto gramática o Esquemas XML (en inglés *XML Schema*). Existen varios formatos para gramáticas: los DTD heredados del SGML, los XML-Schema, que son recomendación del estándar por el W3C y otros más específicos como el XML Data.

Las **tecnologías principales más empleadas** del modelo XML en arquitecturas SOA vienen dadas por:

- ✓ **XML Schema.** Lenguaje de esquema para describir la estructura y reglas de validación de un documento XML. Se diferencia del DTD en que permite un gran número de tipos de datos. Los documentos de esquema son de extensión XSD (en inglés *XML Schema Definition*). La programación de esquemas se basa en los espacios de nombres y los elementos y atributos que contienen. Después de validar un documento contra un XSD se puede expresar su estructura y contenido en términos del modelo de datos del esquema. Esta funcionalidad se denomina PSVI (en inglés *Post Schema Validation Infoset*), y permite transformar el documento en una jerarquía orientada a objetos.
- ✓ **XSL.** XSL funciona como un lenguaje avanzado para crear hojas CSS transformando y realizando otras operaciones sobre documentos XML, dándoles formato. A su vez puede descomponerse en tres lenguajes o dialectos XML, todos recomendaciones del W3C, que integran la familia XSL:
  - ✓ **XSLT** (en inglés *Extensible Stylesheet Language Transformations*). Estándar para documentos XML que permiten transformar documentos XML en base a modelos de una sintaxis a otra

permitiendo estructuras de programación, funcionando a manera de intérprete. Las reglas de los modelos se definen programáticamente y la principal capacidad de este lenguaje es que permite separar el contenido de la presentación, o diferentes presentaciones en documentos XML lo que se adapta perfectamente a los modelos de separación en capas vistos.

- ✓ **XSL-FO** (en inglés *Extensible Stylesheet Language Formatting Objects*). Documentos XML que especifican formatos de datos u objetos para su presentación. La utilidad básica de estos documentos es la presentación, con lo cual se complementan con XSLT en la salida de datos de las aplicaciones. Permite la generación de documentos multiformato: XML, (X)HTML e incluso PDF. Existen procesadores específicos para este tipo de operaciones como Apache FOP.
- ✓ **XPath o XML Path Language**. Permite identificar partes de un documento XML, accediendo a sus atributos y elementos como si fuesen nodos, a través de la construcción de expresiones que recorren y procesan un documento XML. En XSL permite seleccionar y recorrer el documento XML de entrada de la transformación, pero por extensión tiene otros muchos usos actualmente, sirviendo de base para otros lenguajes XML.



**Figura 5: Familia XSL.**

- ✓ **XPointer.** Recomendación del W3C que permite localizar puntos concretos o fragmentos en un documento que expande las funcionalidades de XPath a través de rangos.
- ✓ **XLink.** Recomendación del W3C que define un mecanismo para añadir hipervínculos en archivos XML u otros recursos, con la opción de navegar en los dos sentidos, con vínculos bidireccionales o varios archivos enlazados, multivínculos. En XLink todo en la red es un recurso, y se puede enlazar desde un localizador, definiendo las relaciones entre recursos con arcos. Asimismo permite agregar a un vínculo información sobre sí mismo como metadatos.
- ✓ **XQuery.** Lenguaje de consulta desarrollado por el W3C para recuperar colecciones de datos XML, con muchas similitudes con SQL. Permite extraer y manipular información de documentos XML o cualquier otro sistema de información que permita representación vía XML como Bases de datos o documentos ofimáticos. Emplea XPath para acceder a los documentos añadiendo unas expresiones propias

denominadas FLWOR. Asimismo realiza transformaciones de documentos XML y búsquedas de elementos textuales en la web o en recursos XML/(X)HTML. En las arquitecturas SOA resultan especialmente útiles para recuperar información de Bases de datos y presentarla a través de Servicios web.

- ✓ **XForms.** Lenguaje de definición de Interfaces de usuario desarrollada por el W3C, centrada especialmente en la parte de formularios web y su integración en documentos (X)HTML, ODF o SVG. Se aplica el paradigma de separar el contenido, propósito y estructura. En aplicación del MVC incorpora un modelo declarativo de compuesto de reglas y validación para datos y tipos de datos de los formularios, así como envío de parámetros; una capa de vista compuesta de los controles de la interfaz de usuario; un controlador para orquestar las manipulaciones de datos, interacciones entre el modelo y la vista y envíos de datos. Otras evoluciones de esta tecnología serían AJAXForms y XSLTForms, incorporando AJAX y XSLT a esta tecnología. Por otra parte, existen otros lenguajes relacionados con las interfaces de usuario que siguen dialectos de XML, muchos de ellos con capacidad para interactuar con XForms como: XAML, XUL, UIML, UsiXML, AUIML,...

El grupo de tecnologías anteriores podrían definirse como lenguajes XML de propósito general. En muchas ocasiones el XML se emplea de manera concreta para representación de datos complejos o con necesidades específicas para nuestro dominio. En la tabla 2 se recogen algunos ejemplos de lenguajes XML empleados para representación o adaptación de información a necesidades concretas, bien para entornos de trabajo como XHTML y WML o bien en dominios específicos como aplicaciones de información geográfica o diseño gráfico.

	<b>Función</b>
<b>XHTML</b>	HTML con especificaciones más estrictas para presentar una mayor compatibilidad con la web semántica y los otros estándares XML
<b>MathML</b>	Expresar planteamientos matemáticos
<b>SVG</b>	Especificación para describir gráficos vectoriales y animaciones
<b>SMIL</b>	Permitir la integración multimedia en XHTML y SVG
<b>WML</b>	Adaptación del HTML para móviles y PDA
<b>VoiceXML</b>	Convertir habla en XML a partir de gramáticas de reconocimiento de voz
<b>SSML</b>	Para habla sintética
<b>GML/KML</b>	Para sistemas de modelado e información geográfica
<b>X3D</b>	Representación de gráficos en 3D
<b>EBML</b>	Para almacenar jerarquías de datos en formato binario de longitud variable

***Tablas 2: Lenguajes XML complementarios.***

El uso tan extendido del XML obliga a disponer de herramientas que permitan el tratamiento sencillo de los documentos, recorrer, manipular, procesar, etc. Muchas tecnologías disponen de *frameworks* específicos para **tratamiento de XML**:

✓ **DOM** (en inglés *Document Object Model*). Especificación del W3C de una API ([org.w3c.dom](http://org.w3c.dom)) para manipular documentos XML/HTML, acceder a su contenido, estructura y estilos, a través de un analizador sintáctico. DOM genera un árbol jerárquico en memoria donde almacena todo el documento. A través de un procesador se permite acceder a cualquier nodo del árbol, o insertar/eliminar nuevos nodos. El principal inconveniente de este modelo es que precisa gran

cantidad de memoria por la necesidad de cargar todo el documento, pero tiene las ventajas de ser muy sencillo de implementar y de permitir la generación de XML. Se apoya en tecnologías XSLT y Xpath. Frameworks como Xerces se basan en DOM para tratamiento de XML así como otros basados en AJAX del tipo de JQuery, Prototype, Dojo, etc. Asimismo existen alternativas recientes similares a DOM, diseñadas explícitamente para JEE que resultan más fáciles de emplear, JDOM (org.jdom) y DOM4J (org.dom4j).

✓ **SAX** (en inglés *Simple API for XML*). API inicialmente para Java (org.xml.sax), pero que después evolucionó a otros lenguajes como C++, Perl, Python..., que disponen de un analizador que genera eventos al conseguir puntos llave del documento analizado. Recorre el documento de manera secuencial a través de un administrador de eventos, el `DocumentHandler`, evento a evento, con lo cual no precisa cargar el documento en memoria pero no permite vuelta atrás sin ir de nuevo al inicio. Esto lo hace muy adecuado para documentos de gran tamaño. Otros *frameworks* más completos se basan a su vez en SAX, como: Xerces, Crimson, Piccolo u Oracle XML Parser.

✓ **StAX** (en inglés *Streaming API for XML*). Define un analizador sintáctico de flujo de datos integrado en JEE, con soporte para generación de XML. Se emplean dos estilos de análisis Cursor API e Iterador Event Iterator API, ambos basados en iteraciones para solventar las limitaciones de SAX y DOM. En este modelo el documento XML se transmite en un flujo de datos donde se va solicitando el siguiente evento (*Pull*) con el fin de optimizar recursos de memoria. Se distingue entre Streaming Pull Parsing donde el cliente sólo obtiene los datos solicitados previamente (SAX) y el Streaming Push cuando el analizador envía al cliente datos del XML al localizar un elemento.

✓ **JAXP** (en inglés *Java API for XML Processing*). API de Java (javax.xml.parsers y javax.xml.transform) que proporciona acceso a través de dos factorías abstractas para trabajar con instancias de analizadores DOM y SAX a través de diferentes implementaciones, así como soporte para StAX, espacios de nombres y XSLT (Xalan). También lleva incorporado el analizador Crimson. Se acostumbra a integrar en entornos con Servicios web para agrupar en un mismo *framework* todas las posibilidades de tratamiento de XML.

✓ **JAXB** (en inglés *Java Architecture for XML Binding*) . API JEE (javax.xml.bind) que proporciona un conjunto de interfaces para analizar y generar XML de manera automática. A partir del modelo definido en XML realiza la generación de clases Java equivalentes. El esquema se acostumbra a definir vía DTD, a partir del cual un desarrollador puede construir un árbol de objetos Java que se corresponden con el XML. De este modo se evitan las limitaciones de memoria de DOM.

Paralelamente se dispondrá de componentes específicos para entornos basados en **Servicios web** como WSDL, los más habituales serían:

✓ **SAAJ**. API (javax.xml.soap) de SOAP y SOAP con adjuntos (en inglés *SOAP with Attachments*) que permite enviar documentos XML y adjuntos en formato MIME que pueden ser o no XML. Suele emplearse a bajo nivel por otras API para operaciones de mensajería.

✓ **JAX-RPC**. API de JEE para facilitar el desarrollo de componentes software que hagan uso de XML para comunicaciones a través de llamadas a procedimientos remotos (RPC), en la línea de IDL-CORBA y RMI. A diferencia de estas alternativas JAX-RPC emplea XML como soporte a Servicios web. Permite correspondencia entre objetos y estructuras XML. En arquitecturas SOA el JAX-RPC sería la tecnología a

través de la que el cliente envía la petición de servicio. Por debajo emplea SOAP, pero este nivel permanece transparente a la API. Sus funciones abarcan: Mensajería asíncrona, Enrutamiento de mensajes, Mensajería con entrega garantizada.

- ✓ **JAXR** (en inglés *Java API for XML Registries*). API de JEE para acceso a registros de servicios en estándares abiertos como ebXML o UDDI. Permite a los servicios la posibilidad de auto-registrarse. Asimismo soporta el uso de consultas SQL para la búsqueda de registro a través del objeto SQLQueryManager. Hace uso de JAXM para mensajería.
- ✓ **JAX-WS** (en inglés *Java API for XML Web Services*). Componente del servicio web base Metro, que sería evolución y ampliación de JAX-RPC y se encontraría integrado con JEE (javax.xml.ws). Hace uso de anotaciones Java para describir elementos de las clases, como metadatos y permiten automatización de tareas.

Por último dentro del ámbito de la **seguridad**, existen una serie de soluciones derivadas de la capacidad insuficiente ante arquitecturas SOA de TLS o SSL. Así es necesario destacar las siguientes tecnologías:

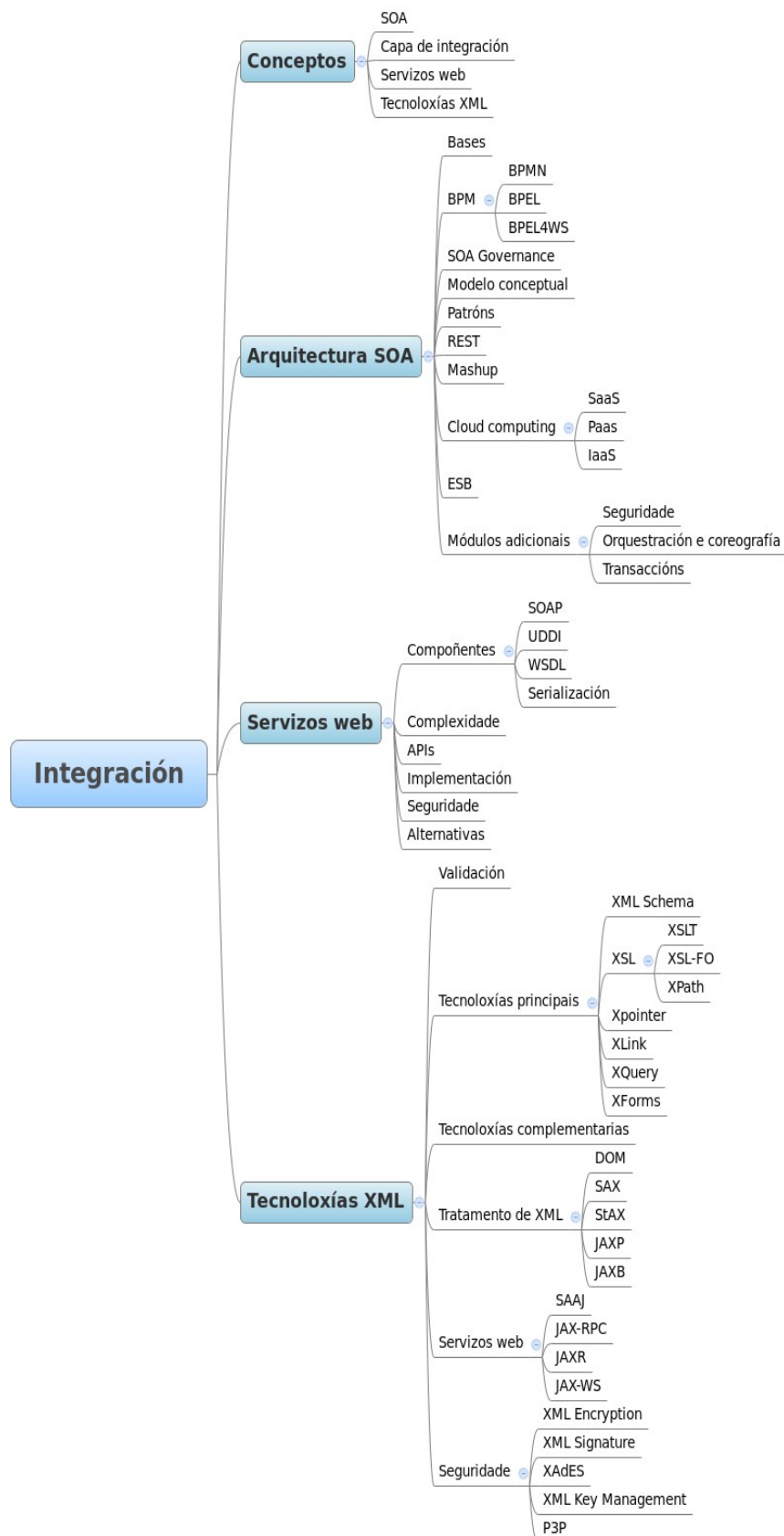
- ✓ **XML Encryption**. La encriptación XML es una recomendación del W3C que especifica el proceso para cifrar datos o documentos completos y representar esa información encriptada en un documento XML. Permite supercifrado y soporta los algoritmos TripleDES, AES y RSA.
- ✓ **XML Signature**. Firma digital que garantiza la integridad de las partes en una comunicación. Asimismo proporciona autenticación de mensajes, integridad de datos, soporte de transacciones sin rechazo y firmas para cualquier contenido digital o XML. En el documento se añade un elemento Signature que encapsula el contenido de la firma



digital incluyendo una referencia al objeto firmado, la indicación del algoritmo de canonización, y el valor resultante de la firma.

- ✓ **XAdES** (en inglés *XML Advanced Electronic Signatures*). Firma digital avanzada XML, que añade un conjunto de extensiones a XML Signature, permitiendo por ejemplo que las firmas sean válidas durante largos períodos de tiempo
- ✓ **XML Key Management**. Protocolo para distribuir y registrar llaves públicas y certificados evitando la complejidad de PKI. Está compuesto de dos partes: X-KRSS o registro de llave pública, un conjunto de protocolos que soportan el registro de pares de llaves; y X-KISS información de llave pública, que define un conjunto de protocolos para procesamiento y envío de información asociada e identificada con XML Signature y cifrada con XML Encryption.
- ✓ **P3P** (en inglés *Platform for Privacy Preferences*). Especificación del W3C que define un estándar de gestión de datos y de privacidad, así como un formato XML para expresar políticas de privacidad, con el objetivo de permitir a los usuarios si y cómo quieren revelar información personal.

#### **28.4. ESQUEMA**



**Traducción texto Esquema:** Servicios web. Tecnologías XML. Patrones.

Módulos adicionales. Orquestación. Transacciones. Componentes. Complejidad. Seguridad. Tecnologías principales. Tecnologías complementarias. Tratamiento de XML. Seguridad.

## **28.5. REFERENCIAS**

Varios autores.

Web Services Architecture. W3C Working Group. (2004).

César de la Torre y Roberto González.

Arquitectura SOA con tecnología Microsoft. Buenas prácticas y diseño de aplicaciones empresariales. (2008).

Joan Ribas Lequerica.

Web Services. (2003).

Patrick Cauldwell y otros.

Servicios Web XML. (2002).

**Autor: Juan Marcos Filgueira Gomis**

**Asesor Técnico Consellería de Educación e O. U.**

**Colegiado del CPEIG**



## **29. PATRONES DE DISEÑO Y FRAMEWORKS. MVC, JSF. ANTIPATRONES.**

## **TEMA 29. PATRONES DE DISEÑO Y FRAMEWORKS. MVC. JSF. ANTIPATRONES.**

### **29.1 INTRODUCCIÓN Y CONCEPTOS**

### **29.2 PATRONES DE DISEÑO Y FRAMEWORKS**

### **29.3 MVC**

### **29.4 JSF**

### **29.5 ANTIPATRONES**

### **29.6 ESQUEMA**

### **29.7 REFERENCIAS**

### **29.1 INTRODUCCIÓN Y CONCEPTOS**

Para muchos problemas de diseño que se repiten tanto en desarrollos software como en implantaciones hardware existen soluciones comunes de aplicación dentro del mismo contexto. Estas soluciones recurrentes se denominan **patrones de diseño** y se basan en el concepto de reutilización y aprovechamiento de soluciones ya existentes en problemas nuevos. Los patrones según el autor suelen dividirse en diferentes familias, siendo la clasificación más habitual en patrones de diseño, de arquitectura e interacción. A su vez dentro de los patrones de diseño se clasifican según creacionales, estructurales y de comportamiento. Asimismo se definen los patrones de programación como patrones específicos para lenguajes de programación o sistemas concretos. Hace falta señalar que en una misma solución o diseño puede convivir cualquier número de patrones que sea necesario, ya que en muchos casos se trata de soluciones parciales a problemas concretos no de soluciones generales. Frente al concepto de patrón surge el de antipatrón que definen errores de diseño comunes o problemas que se repiten a menudo para ayudar a identificarlos. Los patrones y antipatrones pueden definirse ante lenguajes de definición del estilo del Lenguaje Unificado de Modelado o UML (en inglés *Unified*

*Modeling Language*), que soportado por el OMG es uno de los más empleados actualmente.

En las aplicaciones JEE y .NET uno de los patrones de uso más extendido es el Modelo Vista Controlador o MVC (en inglés *Model View Controller*) que se basa en la separación de una aplicación en tres capas o componentes diferenciados, interfaz de usuario, lógica de negocio y sistemas de información.

Este patrón se integra en **frameworks**, o componentes software que implementan funcionalidades comunes a conjuntos de aplicaciones, y que pueden seguir los modelos de patrones de diseño. El patrón sería la solución de diseño abstracta y el *framework* una implementación del mismo concreta. Algunos *frameworks* como Struts representan el esqueleto de una aplicación, con implementación del patrón MVC entre otros, aportando así directamente todas las funcionalidades precisas para su funcionamiento interno. Si en una misma aplicación añadimos nuevos *frameworks* dispondremos de funcionalidades añadidas. De este modo el *framework* **JSF** (en inglés *Java Server Faces*) proporciona, complementariamente a Struts herramientas que facilitan el desarrollo de interfaces de usuario.

## **29.2 PATRONES DE DISEÑO Y FRAMEWORKS**

Las principales ventajas del empleo de patrones en soluciones software pasan por facilitar la comunicación interna entre componentes, ahorrar tiempo y otros recursos, mejorar la calidad de las operaciones de todo el ciclo de vida de desarrollo y facilitar el aprendizaje. A día de hoy es un hecho consumado que su aplicación correcta reporta un beneficio directo en cualquier desarrollo o implantación.

### **29.2.1 Clasificación general**

Existen muchas clasificaciones de los patrones, según el autor (res), pero la más habitual hace referencia al ámbito de aplicación del patrón tomando como referencia la ingeniería del software:

- 1) **Patrones de diseño.** Proporcionan un esquema de aplicación en partes de un sistema software. Definen estructuras que resuelven un problema de diseño de utilidad en diferentes aplicaciones.
- 2) **Patrones de arquitectura.** Proporcionan un esquema u organización estructural para definir sistemas completos o subsistemas incluyendo responsabilidades y relaciones entre sistemas.
- 3) **Patrones de interacción.** Proporcionan un diseño de interfaz para aplicaciones o aplicaciones web.
- 4) **Patrones de programación** (en inglés *Idioms patterns*). Patrones a bajo nivel para lenguajes de programación o tecnologías específicas. Definen representaciones de implementaciones de componentes y relaciones considerando funcionalidades propias de cada lenguaje.

### **29.2.2 Clasificación de patrones para tecnologías de servidores de aplicaciones**

A mayores fueron surgiendo patrones para otros ámbitos de aplicación, como programación multihilo, flujos de trabajo para procesos de sistemas empresariales, arquitecturas SOA o integración de sistemas. En definitiva, puede concluirse que el concepto de patrón puede extenderse a cualquier problema que nos encontremos y el nivel de abstracción que precisemos en la solución. Existen diferentes catálogos de patrones, siendo los más conocidos:

- ✓ **GoF**, (en inglés *Gang of Four*) para problemas de diseño. (1995).
- ✓ **POSA**, (en inglés *Pattern Oriented Software Architecture*) para soluciones en arquitecturas SOA. (1996).

- ✓ **J2EE**, para soluciones específicas de esta tecnología. (2003).
- ✓ **PoEAA** (en inglés *Patterns of Enterprise Application Architecture*). Para sistemas complejos en arquitecturas empresariales distribuidas en capas. (2003).
- ✓ **GRASP** (en inglés *General Responsibility Assignment Software Patterns*). Patrones generales para asignación de responsabilidades y transiciones. (2005).

En concreto para el ámbito de los servidores de aplicaciones como .NET y JEE en cuanto a arquitectura y análisis pueden destacarse los siguientes atendiendo a su nivel de utilización:

- a) **Patrón de análisis Party (Grupo)**. Agrupa las responsabilidades similares de los tipos de colectivos de una organización en un supertipo. Se emplea para facilitar el modelado de estructuras en organización, siendo cada tipo una organización, empresa, rol o papel y almacenar los datos personales de cada miembro. Situaciones especiales obligan a adaptaciones de este patrón como ocurre en el Party Type Generalizations que permite la generalización de tipos de grupo que heredan de un subtipo, por ejemplo para una persona tiene varios papeles a un tiempo.
- b) **Patrón de análisis Accountability**. Establece una relación de responsabilidad entre dos partes o perfiles. Con los tipos Accountability y Accountability Type permite expresar la clase de relación entre ambos. Puede hacer uso del patrón Party para obtener una mayor flexibilidad. Según sean las relaciones puede dar lugar a patrones más complejos como Hierarchic Accountability o Jerarquía de responsabilidad que añade limitaciones a los elementos de responsabilidad; o que tiene aplicación a la hora de delegar tipos de responsabilidad a un subpatrón Party.
- c) **Patrón arquitectónico MVC** (en inglés *Model View Controller*).



Estructura un componente software en 3 capas, el modelo con la lógica de negocio, funcionalidades y sistemas de información, la vista con la interfaz de usuario y el Controlador que recibe los eventos de la entrada y coordina las actividades de la vista.

- d) **Patrón arquitectónico PAC** (en inglés *Presentation Abstraction Control*). Similar al MVC este patrón define un sistema interactivo basado en una jerarquía de agentes cooperantes que realizan funcionalidades concretas. Se divide en tres capas: Presentación con interacción persona-máquina, Abstracción con la lógica y sistemas de información y el Control que centraliza las comunicaciones entre agentes, procesa eventos externos y actualiza el modelo. La principal diferencia con el MVC radica en que se pueden hacer diferentes agentes o subsistemas de aplicación, operando de forma independiente o jerarquizada.
- e) **Patrón arquitectónico Capas** (en inglés *Layers*). Representaría la abstracción genérica de los patrones anteriores a un sistema multicapa, orientado a distribución jerárquica de papeles y responsabilidades. Permite aumentar o disminuir el nivel de abstracción, más o menos capas, aislando el mantenimiento y actualización de cada capa. Cada nivel o capa ofrece servicios a la capa superior y usa los de la inferior.
- f) **Patrón arquitectónico Pipes and Filters**. Orientado también a arquitecturas SOA, en este modelo cada componente posee un conjunto de entradas y salidas. Representa la lectura de flujos de datos, transformándolos en un flujo de salida sin tener que procesar toda la entrada, como ocurre en los modelos Streaming y de ahí que se denominen Filtros a los componentes que reciben las entradas y tuberías o conductos a los que encaminan el flujo hacia salida. Permite representar procesamientos en paralelo así como ejecución concurrente.
- g) **Patrón arquitectónico Blackboard**. Proporciona un modelo de soluciones aproximadas, cuando no se puede aplicar una solución

concreta. Permite reutilizar las fuentes de conocimiento y un mejor soporte de cambios y mantenimiento de la solución aproximada.

- h) **Patrón Microkernel.** Dentro de los patrones para sistemas adaptables, este modelo separa un kernel funcional mínimo del extendido para soportar sistemas software con requerimientos que cambian a lo largo del tiempo. Ideado para sistemas operativos, cada uno de ellos sería una vista del Microkernel central, permitiendo que se pueda extender el sistema de manera fácil.
- i) **Patrón Reflection.** Otro patrón sistemas adaptables que modela un mecanismo para cambiar la estructura y comportamiento de un sistema dinámicamente. Establece dos niveles: Metadatos para que los software lleve una descripción de sí mismo y Lógica de aplicación. Los cambios de comportamiento pueden reflejarse en los metadatos, pero esto puede pasar inadvertido.
- j) **Patrón arquitectónico Broker.** Orientado a arquitecturas SOA y sistemas distribuidos donde varios clientes hacen peticiones a un servidor o servicio remoto. El agente Broker se encarga de coordinar la comunicación entre el cliente y el proveedor del servicio. Las principales ventajas de este patrón son permitir la transparencia de localización del servicio, permitir cambios y ampliación de nuevos componentes sin que el sistema se vea afectado, avance de la portabilidad e interoperabilidad con otros agentes Broker.
- k) **Patrón Publisher Subscriber.** Orientado a arquitecturas SOA y sistemas distribuidos inserta una capa entre clientes y servidores que se encarga de llevar cuenta de la comunicación de manera transparente. Representa una arquitectura de mensajería sin acoplamiento.

En lo tocante al diseño, los principales patrones suelen agruparse en tres grandes categorías: Patrones creacionales, estructurales y de comportamiento. Los **creacionales** incluirían:

- a) **Abstract Factory.** Proporciona una interfaz que permite la creación de familias de objetos dependientes o relacionales sin tener que especificar las clases completas. Ejemplos de este patrón serían los Widgets y componentes de interfaces gráficas.
- b) **Builder.** Constructor virtual que separa la construcción de un objeto complejo de su representación, de tal manera que se obtienen diferentes representaciones en un mismo proceso.
- c) **Factory Method.** Patrón que define una interfaz para la creación de objetos dejando que las subclases decidan qué clase instanciar, haciendo que el proceso de generación del subtipo sea transparente al usuario.
- d) **Prototype.** Permite la creación de nuevos objetos clonándolos de una instancia de un objeto ya existente.
- e) **Singleton.** Patrón de instancia única que asegura que de una clase sólo existirá una única instancia definiendo un punto de acceso común a la misma.
- f) **Object Pool.** Patrón para la obtención de objetos por clonación. Se crea una instancia de un tipo de objeto de la clase a clonar. Está pensado para casos donde la creación tenga un coste muy alto y se permita la utilización de objetos genéricos del Pool.

Por otra parte, dentro de los **estructurales**:

- a) **Adapter.** Patrón que convierte la interfaz de una clase en otra interfaz adaptada a necesidades específicas como determinados clientes o interfaces requeridas por compatibilidad.
- b) **Bridge.** O patrón Handle/Body, separa una abstracción de su implementación de manera que ambas puedan cambiar de forma independiente, sin que cambios en una afecten a la otra.
- c) **Composite.** Patrón que permite manipular objetos compuestos como si de un simple se tratara. Hace uso de la composición recursiva y la

estructuras en forma de árbol para poder presentar una interfaz común.

- d) **Decorator**. Responde a la necesidad de añadir funcionalidades a objetos dinámicamente. Crea una jerarquía de clases donde las hijas heredan de las madres las funcionalidades e incorporan las suyas propias.
- e) **Facade**. Proporciona una interfaz común de acceso a un conjunto de interfaces de un sistema. Facilita el empleo del sistema interno con otras interfaces de alto nivel. Los clientes sólo pueden comunicarse a través de la interfaz única que hace de fachada.
- f) **Flyweight**. Permite eliminar la redundancia entre objetos que presentan la misma información. Factoriza los atributos comunes a estos objetos en una clase ligera.
- g) **Proxy**. Proporciona un punto de control de acceso o intermediario para el control de otro(s) objeto(s). Presenta diferentes niveles de aplicabilidad:
  - ✓ *Proxy remoto*. Representa a un objeto remoto de manera local, codificando la petición y argumentos antes de enviarla al objeto remoto.
  - ✓ *Proxy virtual*. Crea objetos de alto coste bajo demanda, con posibilidad de caché de la información de los mismos limitando los costes de acceso.
  - ✓ *Proxy de protección*. Controla el acceso a objetos remotos comprobando que los clientes disponen de los permisos necesarios.
  - ✓ *Proxy de referencia inteligente*. Análogo a un puntero con operaciones adicionales sobre un objeto para temas de concurrencia, acceso a memoria, etc.

El último bloque serían los patrones de comportamiento:

- a) **Chain of responsibility.** El patrón cadena de responsabilidad permite establecer la línea que deben llevar los mensajes, denominada cadena de objetos receptores, permitiendo que varios objetos puedan capturar un mensaje, como puede ser una excepción Java. Cualquiera de los receptores podría responder a la petición según el criterio establecido.
- b) **Comando u Orden.** Patrón que encapsula una operación en un objeto, de suerte que se puedan hacer operaciones extendidas como almacenamiento y colas de peticiones y soporte de acciones de hacer y deshacer.
- c) **Intérprete.** Define una representación para la gramática de un lenguaje junto con su intérprete.
- d) **Iterator.** Patrón con el objetivo de permitir recorrer objetos compuestos como pueden ser las colecciones sin necesidad de contemplar aspectos de implementación o representación interna de los mismos. Define una interfaz donde se ofrecen diferentes métodos para recorrer el objeto complejo.
- e) **Mediador.** Define un objeto que facilita la interacción entre otros de distinto tipo, coordinando la comunicación entre ellos. El objetivo es encapsular la interacción de esos objetos para evitar el acoplamiento entre ellos.
- f) **Memento.** Representa el estado de un objeto o sistema complejo para permitir su almacenamiento y modificación, de suerte que se pueda restaurar volviendo a estados anteriores en el tiempo.
- g) **Observador.** Permite definir una dependencia de uno a muchos, de modo que eventos o modificaciones de estado disparen la notificación de los cambios a todos los objetos o sistemas dependientes.
- h) **Estado.** Se emplea para permitir que un objeto cambie su comportamiento en el caso de modificarse su estado. De este modo

diferentes clases pueden representar a un mismo objeto a lo largo del tiempo.

- i) **Estrategia.** Permite definir una familia de algoritmos o métodos de resolución, permitiendo seleccionar dinámicamente cuáles aplicar y que de este modo sean intercambiables.
- j) **Template Method.** Define el esqueleto de un algoritmo para una operación, delegando partes del mismo a las clases concretas. De este modo las subclases pueden redefinir pasos concretos del método de resolución.
- k) **Visitor.** Representa un algoritmo u operación realizada sobre la estructura de un objeto, permitiendo la definición de nuevas operaciones sin alterar el tipo de los elementos sobre los que se realiza la operación.

En una última categoría podrían incluirse los patrones propios de lenguajes de programación o tecnologías concretas, siendo los **patrones JEE**, la mayoría Core J2EE Patterns, aquellos con un uso más extendido dentro del mundo de los servidores de aplicaciones y los servicios web:

- a) **Intercepting Filter.** Intercepta las peticiones de la capa de presentación antes o después de su procesamiento permitiendo realizar operaciones sobre los datos como auditorías, comprobaciones de seguridad, conversiones o validaciones. Permiten conectarse en cascada y activar o desactivar sin que afecte al funcionamiento general de una aplicación. Permite diferentes estrategias como: Custom Filter, Estándar, Base Filter y Template Filter.
- b) **Front Controller.** Centraliza el control de las peticiones de la capa de presentación, dirigiéndolas hacia el componente acomodado para validación de parámetros, invocación de elementos de la lógica de negocio, etc. Un controlador se encarga de recoger las peticiones y factorizar el código repetitivo.

- c) **View Helper.** Proporciona una clase que engloba código común, con aplicación tanto para la capa de negocio como para la de presentación. Cada vista contiene código para formato, delegando las responsabilidades de procesamiento en las clases de ayuda implementadas como Java Beans o Custom Tags. Asimismo pueden almacenar modelo de datos intermedios haciendo adaptaciones previas del negocio, como conversiones o validaciones, lo lógico por la separación en capas es que estas operaciones no sean muy complejas.
- d) **Composite View.** Define una jerarquía de vistas compuestas de diferentes vistas particulares permitiendo modificar las partes en tiempo de ejecución y a partir de modelo. De este modo se incluyen dinámicamente las vistas concretas en vistas compuestas de la aplicación a través de los mecanismos que disponen para tal efecto JSP y Servlets.
- e) **Service to worker.** Agrupa varios patrones a modo de framework para permitir combinar un controlador (Front Controller), y un Dispatcher o controlador de vistas (View Helper), para manejar las peticiones de los clientes y generar la presentación dinámicamente cómo respuesta. Los controladores solicitan el contenido a los Helpers que llenan el modelo de negocio intermedio.
- f) **Dispatcher View.** Con una estructura similar a la del Service to worker, en este modelo tanto Controlador como Dispatcher tienen responsabilidades más limitadas ya que lógica de procesamiento y control de la vista son básicas.
- g) **Business Delegate.** Permite la abstracción a implementación de componentes complejos como EJB o JMS de la capa de presentación. De este modo pueden crearse clases Proxy que almacenen y encolen las peticiones pudiendo proporcionar control de prioridades, gestión de excepciones o caché. El patrón emplea un componente denominado Lookup Service, responsable de ocultar los detalles de implementación del código de búsqueda dentro de la lógica de

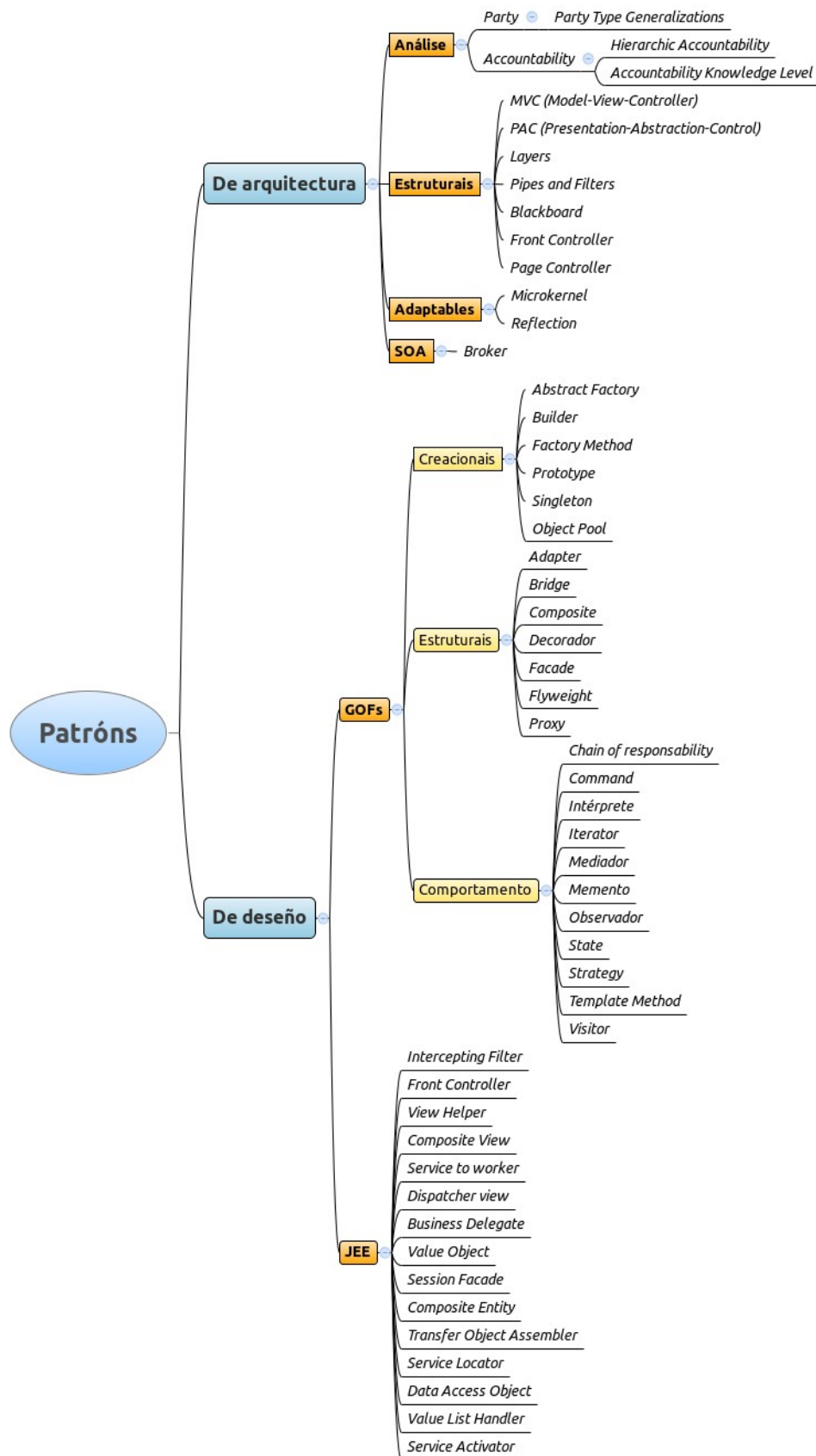
negocio.

- h) **Value Object (VO).** Encapsula un conjunto de datos que representan un objeto o entidad del negocio. Cuando se solicita a un Bean un conjunto de información este puede crear el objeto Value Object y llenar sus atributos para devolverlo al cliente.
- i) **Session Facade.** Emplea un Bean de sesión como fachada para encapsular las interacciones de los componentes de negocio y ofrecer un servicio de acceso uniforme, a través de los interfaces requeridos sólo a través de los casos de uso. Proporciona una abstracción de alto nivel implementada a modo de Bean.
- j) **Composite Entity.** Permite ampliar los Beans de entidad cuando estos son de pequeño tamaño, de este modo pueden aumentarse manteniendo la compatibilidad. El abuso de este patrón se considera un antipatrón ya que puede dar lugar a estructuras muy complejas. Un Bean Composite Entity representa un grafo de objetos, por tanto debe emplearse con cuidado.
- k) **Transfer Object Assembler.** Simplifica el acceso a los sistemas de información a través de un conector común. Cada objeto de negocio tendrá un Transfer Object (TO) con los detalles de acceso a datos (Beans, JDO, JDBC,...) y un Bean de sesión funcionará como interface común.
- l) **Service Locator.** Se emplea para abstraer la utilización de JNDI a través de un objeto Service Locator y para ocultar las complejidades de la creación del contexto inicial, así como de la búsqueda e instanciación de EJBs a través de un punto de acceso común.
- m) **Data Access Object (DAO).** Se emplea un objeto como medio de acceso a sistemas de información, en especial Bases de datos. Abstrae y encapsula las operaciones relacionadas con la tecnología de persistencia empleada (JDBC, JDO, LDAP, Beans, TopLink,



Hibernate, iBATIS, etc.). Controla los parámetros de conexión, obtención de datos y almacenamiento proporcionando una interface de acceso común.

- n) **Value List Handler.** Implementado como Beans de sesión, se encarga de manejar la ejecución de consultas SQL, cachearlas y procesar los resultados. Accede directamente a un DAO que se encarga a su vez de hacer la conexión con el sistema de información y recuperación de los datos. Una vez obtenidos los almacena como TO o VO permitiendo al cliente recorrerlos gracias a la implementación del patrón Iterador.
- o) **Service Activator.** Proporciona un modelo para mensajería asíncrona como JMS. El Service Activator recibe los mensajes y localiza y llama a los métodos de los componentes de negocio que se van a encargar de resolver la petición.



***Figura 1: Resumen de los principales patrones en arquitecturas de***

### ***servidores de aplicaciones.***

**Traducción texto Figura 1:** Patrones. Análisis. Estructurales. De diseño.  
Creacionales. Estructurales. Comportamiento

La implementación de estos patrones no acostumbra a hacerse a medida sino que se recurre a los **frameworks**. Muy relacionados entre sí, los *frameworks* representan una arquitectura de pequeño tamaño que proporciona una estructura genérica integrando diferentes patrones de modo que puedan ser reutilizados o integrados de manera sencilla en las aplicaciones. En un *framework* los patrones tienen una implementación concreta sobre la definición abstracta del patrón. En última instancia son un conjunto de clases e interfaces que cooperan para ofrecer un software reutilizable.

## **29.3 MVC**

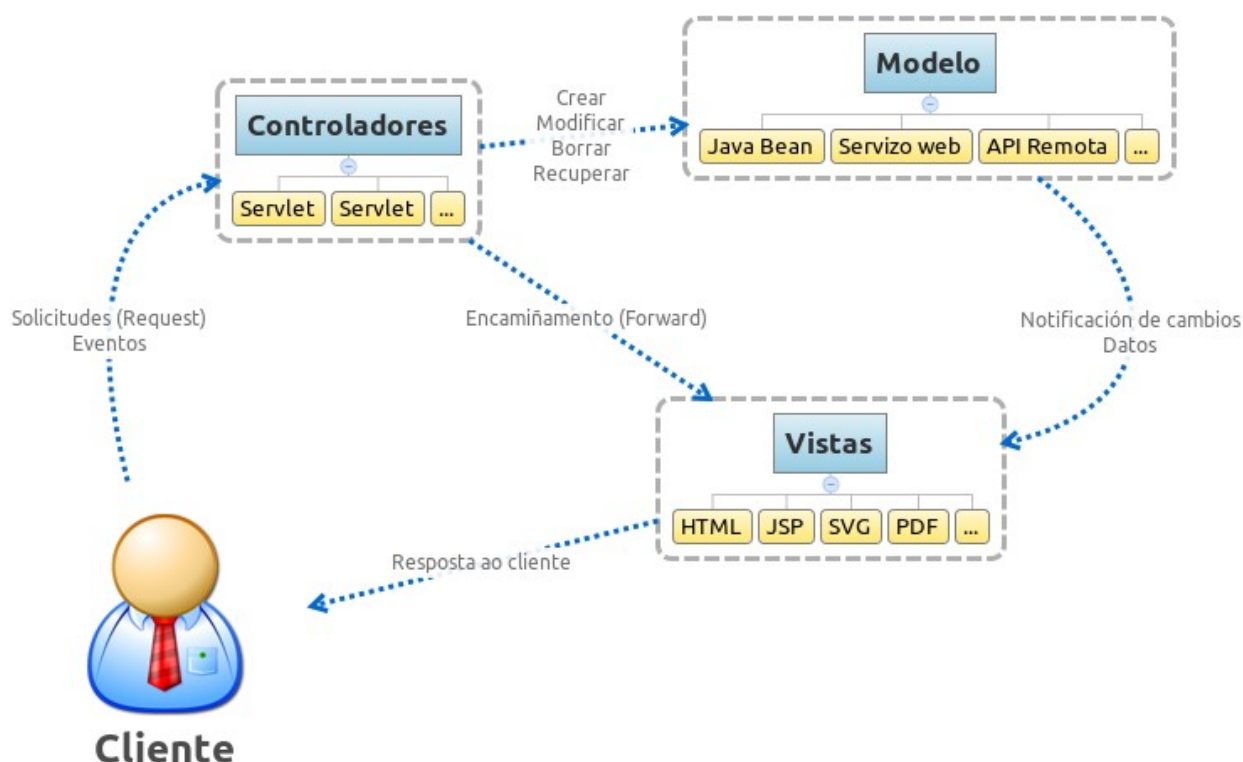
El patrón Modelo-Vista-Controlador es el más empleado para estructurar una aplicación atendiendo a una correcta separación en capas: entrada, procesamiento y salida. Sus principales **ventajas** son una reducción del acoplamiento, facilidad de desarrollo, claridad en el diseño, avance en el mantenimiento, mayor escalabilidad, una mayor cohesión con cada capa fuertemente especializada, y una mayor flexibilidad y agilidad en las vistas, permitiendo su modificación dinámica, sincronización, anidamiento y la existencia de múltiples vistas.

Las **capas** del modelo se concretan en:

- 1) **Modelo** (en inglés *Model*). Encapsula tanto datos como las funcionalidades o casos de uso. Tiene que funcionar independientemente de cualquier representación que tomará los datos en la salida y cualquier comportamiento que se especifique en

la entrada del sistema. A todos los efectos será una caja negra que recibe peticiones y devuelve resultados, encargándose de manejar los datos y controlar sus transformaciones. Normalmente implementa los patrones DAO, VO y Fachada.

- 2) **Vista** (en inglés *View*). Capa en la que se integran todos los componentes que afecten a la interfaz de usuario. Recibe las peticiones del usuario y las envía al controlador, obteniendo de este las respuestas. Se permiten múltiples vistas del mismo modelo, pero toda la lógica de presentación debe ir en esta capa.
- 3) **Controlador** (en inglés *Controller*). Recibe peticiones de la vista, tales como eventos, refrescos, etc. que recoge con un gestor de eventos o Handler y son traducidos a solicitudes de servicios o casos de uso, enviando las peticiones al modelo. A menudo implementan patrones como Comando o Front-Controller para encapsular las acciones.



**Figura 2: Modelo-Vista-Controlador con tecnologías JEE.**

**Traducción texto Figura 2:** Encaminamiento. Respuesta al cliente.

Como patrón de arquitectura el MVC puede contener a su vez los siguientes **patrones** de diseño:

- ✓ **Observador.** Para proveer el mecanismo de publicación y suscripción que permita notificar cambios del modelo en las vistas.
- ✓ **Composite View.** Para permitir la creación de vistas compuestas en una jerarquía.
- ✓ **Estrategia.** Para llevar cuenta de la relación entre las vistas y los controladores, ya que permite modificar dinámicamente aspectos del control.
- ✓ **Factory Method.** Para especificar al controlador una vista como predeterminada.
- ✓ **Decorador.** Para añadir funcionalidades adicionales a las vistas.
- ✓ **Proxy.** Para distribuir la arquitectura en diferentes emplazamientos y mejorar características de rendimiento.

El modelo MVC se implementa tanto en *frameworks* .NET (Windows Forms, ASP .NET, Spring .NET, Maverick .NET, MonoRail,...) como en JEE (Struts, Spring, Tapestry, Aurora, JSF, etc.). Asimismo es un modelo que se encuentra extendido a muchas otras tecnologías como PHP, Ruby, Perl, Phyton, etc.

Los *frameworks* que implementan el MVC suelen presentar una serie de características **generales**, comunes a todos ellos y que incluyen:

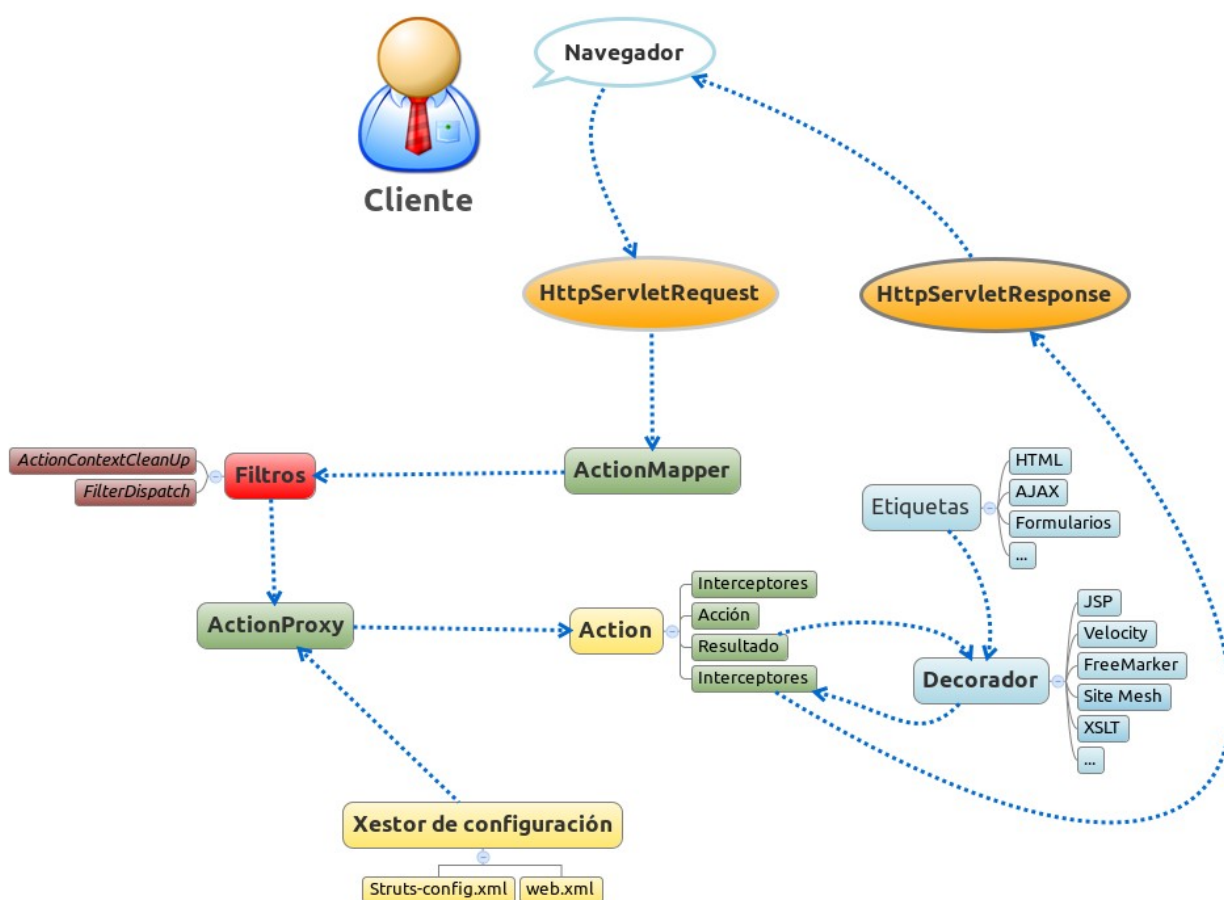
- ✓ Implementación de diferentes patrones de diseño orientados a la reutilización de diseño y código.
- ✓ Controles de validación de campos de formularios.
- ✓ Control de errores y excepciones.
- ✓ Mensajería y localización de cadenas de textos.
- ✓ Librerías de etiquetas o componentes (TagLibs, Widgets, etc.)
- ✓ Componentes de la Interfaz de Usuario como etiquetado de componentes de formularios, pestañas, controles AJAX, etc.
- ✓ Presentación de información a través de listados y tablas con paginación.
- ✓ Integración con *frameworks* con el patrón Decorador o basados en modelos como Tiles, FreeMarker, Velocity, etc.
- ✓ Acceso datos en diferentes Sistemas de Información: Bases de datos, XML, etc.
- ✓ Abstracción de direcciones URL, Request y sesiones.
- ✓ Autenticación y control de usuarios, papeles y filtros.

Entre los *frameworks* que implementan el MVC destaca Apache Jakarta **Struts** (que tiene una evolución en Struts 2.0 al fusionarlo con WebWork), uno de los más empleados en tecnologías JEE y que resulta casi un estándar de facto debido a su integración en otros *frameworks* con más funcionalidades. Se emplea para la implementación de aplicaciones web basadas en Servlets y JSP. Proporciona un conjunto de etiquetas JSP personalizadas (en inglés *Custom Tags*) que permiten encapsular

funcionalidades en la vista. Con el modelo de Struts tiene implantación directa el modelo MVC y otros patrones de diseño pre-construidos, permitiendo la configuración directa de objetos reutilizables ante la configuración de XML. Además proporciona las características anteriormente especificadas: validación, localización, modelo, etc.

Transporta automáticamente los datos insertados por el cliente hasta el controlador a través de Acciones (en inglés *Actions*) mediante formularios ActionForms integrados en el *framework* y viceversa para su presentación. Distingue entre una parte común a cualquier aplicación que haga uso del *framework* que hace de Controlador (ActionServlet) y otra parte configurable a través de archivos de configuración en XML (*struts-config.xml*, *web.xml*,...). Su principal desventaja es no abarcar hasta el nivel de acceso a datos, haciendo que sea necesario el empleo de otros *frameworks* especializados en esta capa para la elaboración de DAO, VO y otras operaciones complementarias.





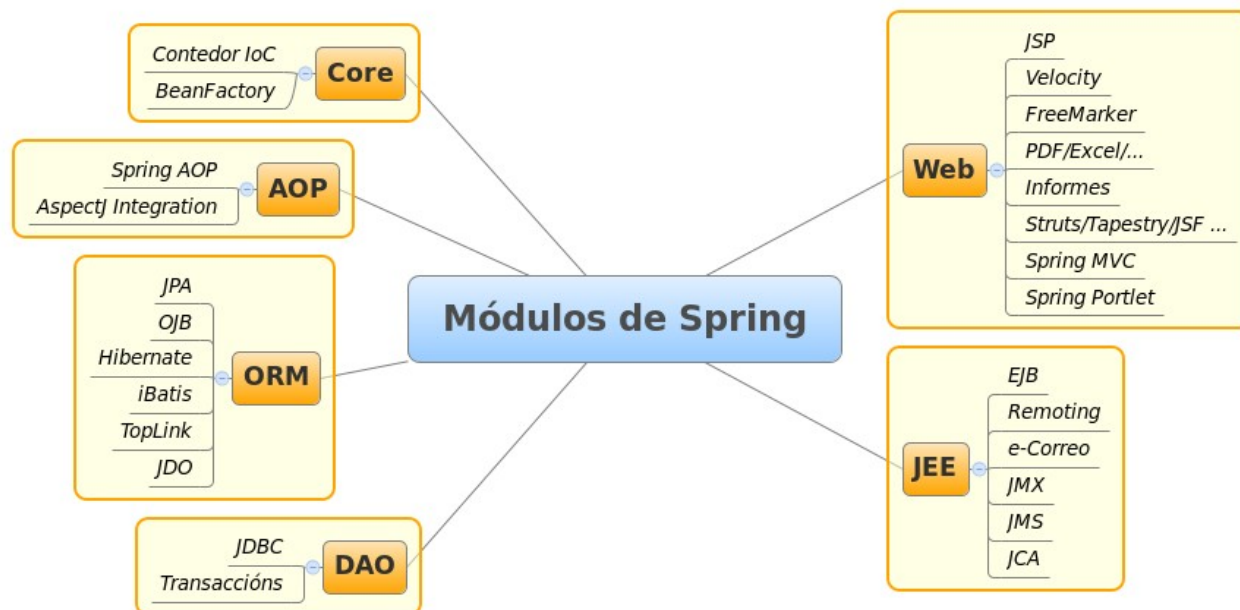
**Figura 3: Funcionamiento interno Struts/Struts 2.0.**

La principal alternativa a Struts sería **Spring** Framework, aunque también permiten integración conjunta y con otros *frameworks* como JSF, Tapestry o WebWork. Aunque su orientación principal sea la plataforma JEE, está disponible en .NET a través del *framework* Spring .NET. Tiene soporte para JTA, JDO, JDBC y ODBC, y permite integración con terceros como Acegi, Hibernate, iBatis y OJB. Como novedad permite programación orientada a aspectos o AOP (en inglés *Aspect-Oriented Programming*) que busca emplear los servicios secundarios cómo seguridad, registro de log, manejo de transacciones, etc. de las funcionalidades del modelo. Con AOP pueden emplearse los servicios de la aplicación de forma declarativa, o ante archivos XML de configuración o mediante estándares JSR. Asimismo realiza Inversión de Control o IoC, que promueve el bajo acoplamiento a partir de

la inyección de dependencias entre objetos. Las principales desventajas de Spring son que implica una configuración compleja, ya que cada servicio lleva su XML propio, aunque existe la alternativa del JSR. Su contenedor no resulta ligero lo que impide que tenga aplicación práctica en algunos entornos como pueden ser los dispositivos móviles.

La arquitectura **de Spring** está compuesta por los siguientes componentes:

- ✓ **Core.** El núcleo que aloja el contenedor principal o BeanFactory.
- ✓ **Módulo AOP.** Proporciona la implementación de AOP, permitiendo desarrollar interceptores de método y puntos de ruptura para desligar el código del modelo de las funcionalidades transversales.
- ✓ **Módulo DAO.** Proporciona la capa de abstracción de acceso a datos y sistemas de información sobre los diferentes conectadores disponibles. Además provee de manejo de transacciones vía AOP y otros servicios.
- ✓ **Módulo ORM.** Proporciona integración para las distintas API de correspondencia entre objetos y entidades de bases de datos con soporte de diferentes tecnologías e integración con *frameworks* de terceros.
- ✓ **Módulo JEE.** Integración con aplicaciones y servicios JEE.
- ✓ **Módulo Web.** Aporta componentes especiales orientados a desarrollo web e integración con *frameworks* alternativos como Struts o JSF, además de una implementación del paquete Spring MVC.

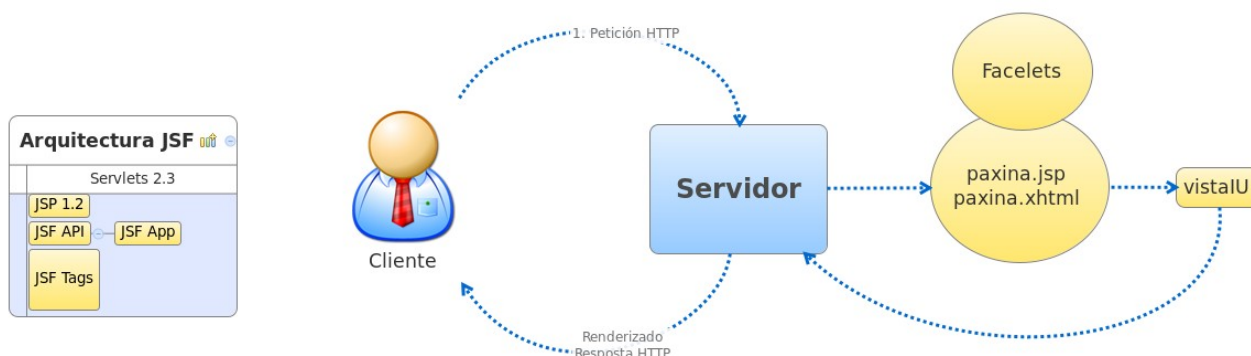


**Figura 4: Arquitectura de Spring.**

## 29.4 JSF

La tecnología Java Server Faces proporciona un *framework* de interfaz de componentes de usuarios para el lado del servidor de aplicaciones. En su base emplea JSP pero permite otras tecnologías para interfaces de usuario como XUL. Entre los **componentes** de JSF se encuentran:

- 1) Un conjunto de APIs para representar y manejar componentes de la interface de usuario. Entre las opciones que gestionaría se encontrarían: control de estado y eventos, validaciones de formularios, conversión de datos, control de navegaciones y soporte de localización y accesibilidad.
- 2) Un conjunto de componentes de la interfaz de usuario reutilizables.
- 3) Dos librerías de etiquetas personalizadas (en inglés *Custom Tags*) para JSP.
- 4) Modelo de eventos para el lado del servidor.
- 5) Soporte para Managed Beans de control de eventos.



**Figura 5: Arquitectura JSF y funcionamiento básico.**

Uno de los componentes de JSF es el *framework* JavaServer **Facelets**, destinado a la gestión de modelo (en inglés *templates*). Las principales características de este *framework* son:

- ✓ Coste de tiempo cero para el desarrollo de etiquetas de componentes de la Interfaz de Usuario.
- ✓ Facilidad de creación de modelo de páginas y componentes reutilizables.
- ✓ Soporte para UEL (en inglés *Unified Expression Language*) y validaciones EL.
- ✓ Compatibilidad con cualquier RenderKit.
- ✓ Se integra plenamente con JSTL cosa que en JSF puede ocasionar problemas.
- ✓ Compilación más rápida que con JSP.

Actualmente existen numerosas **implementaciones** de JSF que pueden complementar a la especificación oficial JEE. Existe la posibilidad de combinar diferentes implementaciones en una misma aplicación, siendo las más habituales:

- a) **MyFaces Tomahawk/Sandbox.** Desarrollado por Apache proporciona un conjunto de componentes reutilizables compatibles con las especificaciones JSF 1.1, JSF 1.2 y JSF 2.0.
- b) **Trinidad.** Subproyecto de MyFaces, a partir de la inclusión de los componentes ADF Faces y otros avances. Proporciona los siguientes elementos: Una implementación de JSF, varias librerías de componentes Widgets, la extensión MyFaces Orchestra y módulos de integración para otras tecnologías y estándares como MyFaces Portlet Bridge.
- c) **Tobago.** Otro proyecto basado en MyFaces en una aproximación del diseño de páginas web al de aplicaciones de escritorio. Proporciona una serie de componentes de la Interface de Usuario como abstracciones del HTML. Presenta un conjunto de temas para clientes HTML con vistas independientes de HTML/CSS/Javascript.
- d) **ICEfaces.** Contiene diversos componentes de interfaces de usuario enriquecidos, basados en AJAX y compatibles con SSL, como editores de texto, reproductores multimedia, etc. Soporta Facelets y Seam, además de ser compatible con Spring, WebWork y Tomahawk.
- e) **RichFaces.** Otro *framework* AJAX que incluye ciclo de vida, validaciones, conversiones y gestión de recursos en las aplicaciones. Soporta Facelets y Seam, además de ser compatible con Spring y Tomahawk.
- f) **Ajax4JSF.** Otra alternativa más que proporciona un *framework* AJAX que incluye ciclo de vida, validaciones, conversiones y gestión de recursos en las aplicaciones. Soporta Facelets y Seam, además de ser compatible con Spring y Tomahawk. Incluye los siguientes componentes:
  - ✓ *Ajax Filter.* Filtro de peticiones para AJAX.
  - ✓ *Ajax Action Components.* Envían las peticiones desde el cliente.
  - ✓ *Ajax Containers.* Interfaz que describe zonas dentro de las JSP.

- ✓ *Javascript Engine*. Motor en el lado del cliente que actualiza diferentes zonas de las JSP en función de la respuesta AJAX.

## 29.5 ANTIPATRONES

Contrarios al **concepto** de patrones, los antipatrones representan malos usos habituales, o soluciones que, sobre todo a lo largo del tiempo, presentan más problemas de los que resuelven, se trata en definitiva de malas prácticas. Existen dos variantes principales, los que describen una mala solución para un problema habitual y que produce consecuencias difíciles de arreglar a lo largo del tiempo; y aquellos que describen cómo poner remedio a un problema y convertirlo en una buena solución. Por norma general los antipatrones se ven como una buena idea al inicio, que falla de mala manera a la hora de su implementación.

Las **motivaciones** o razones para tener en cuenta los antipatrones como caso de estudio atienden a los siguientes puntos:

- ✓ Permiten identificar soluciones de riesgo para problemas habituales.
- ✓ Proporcionan experiencia del mundo real para detectar problemas que se repiten a lo largo del tiempo, ofreciendo posibles soluciones o alternativas para sus implicaciones más habituales.
- ✓ Proporcionan un marco común para la identificación y documentación de los problemas y diseño de las soluciones.

Como sucedía con los patrones, los antipatrones acostumbran a agruparse en diferentes **categorías**, siendo las principales:

- 1) **Antipatrones de desarrollo software**. Definen problemas asociados al desarrollo software a nivel de aplicación, al nivel de los

patrones de diseño.

- 2) **Antipatrones de arquitectura de software.** Se centran en la distribución y relaciones de las aplicaciones, servicios y otros componentes software a nivel de organización.
- 3) **Antipatrones de gestión de proyectos software.** Identifican escenarios críticos sobre la comunicación entre personas y resolución de problemas en equipos, viendo cómo afectan a un proyecto o proceso software.

Asimismo los antipatrones tienen aplicación en muchas otras áreas como metodología, gestión de la configuración, TDD, diseño web, accesibilidad, usabilidad, etc.

Dentro de los **antipatrones de desarrollo** software nos encontramos entre los más comunes:

- a) **Blob u objeto todopoderoso** (en inglés *God Object*). Se emplea un único objeto, clase o módulo para aglutinar un amplio conjunto de funcionalidades que deberían encontrarse divididas. Con este patrón se cae en un código ampliamente desorganizado y muy acoplado.
- b) **Flujo de lava o lava seca** (en inglés *Lava Flow*). Representa aquellos tipos de programación por impulsos o erupciones de código, de manera desestructurada, desorganizada y con poca documentación. El sistema crece de manera desproporcionada y pasado un tiempo los bloques de código más antiguos se consideran metafóricamente solidificados en lo relativo a la dificultad de solucionar cualquier tipo de problema en el que se encuentren involucrados.
- c) **Descomposición funcional.** Diseño no orientado a objetos, fruto de la migración desde lenguajes estructuradas a POO.

- d) **Poltergeists.** O clases fantasma debido al desconocimiento dentro de la aplicación de cuál es el objetivo de algunas clases, siendo en muchos casos su única función transmitir información entre clases.
- e) **Martillo dorado.** Emplear la misma solución para cualquier problema que surja, sin contemplar otras posibles alternativas.
- f) **Código spaghetti.** Hace referencia a código de aplicación con una estructura compleja e incomprensible con multitud de tecnologías mezcladas. La analogía se hace a partir de las relaciones entre el código que parecen un gran número de hilos mezclados y enrollados.
- g) **Programación copiar y pegar.** Solución en la que en lugar de crear soluciones genéricas se copian y se adaptan soluciones ya existentes.

En el tocante a los **antipatrones de arquitecturas** software destacan por ser los más habituales:

- a) **Reinventar la rueda.** Implementar componentes ya disponibles o que pueden aprovecharse con ligeras modificaciones. Se da por la tendencia a hacer todo uno mismo o el desconocimiento de la arquitectura y soluciones disponibles en el mercado o alternativas de código abierto.
- b) **Vendor Lock-In.** Construir una arquitectura dependiente de un producto de terceros, en especial cuando se trata de software privativo. Se ponen en peligro la escalabilidad del sistema y aumentan los costes de mantenimiento.
- c) **Aislamiento en la organización.** En una misma organización o conjunto de sistemas se crean diferentes unidades aisladas entre sí que crecen en paralelo solucionando problemas comunes de manera independiente. En este modelo puede crecer sobre manera el coste



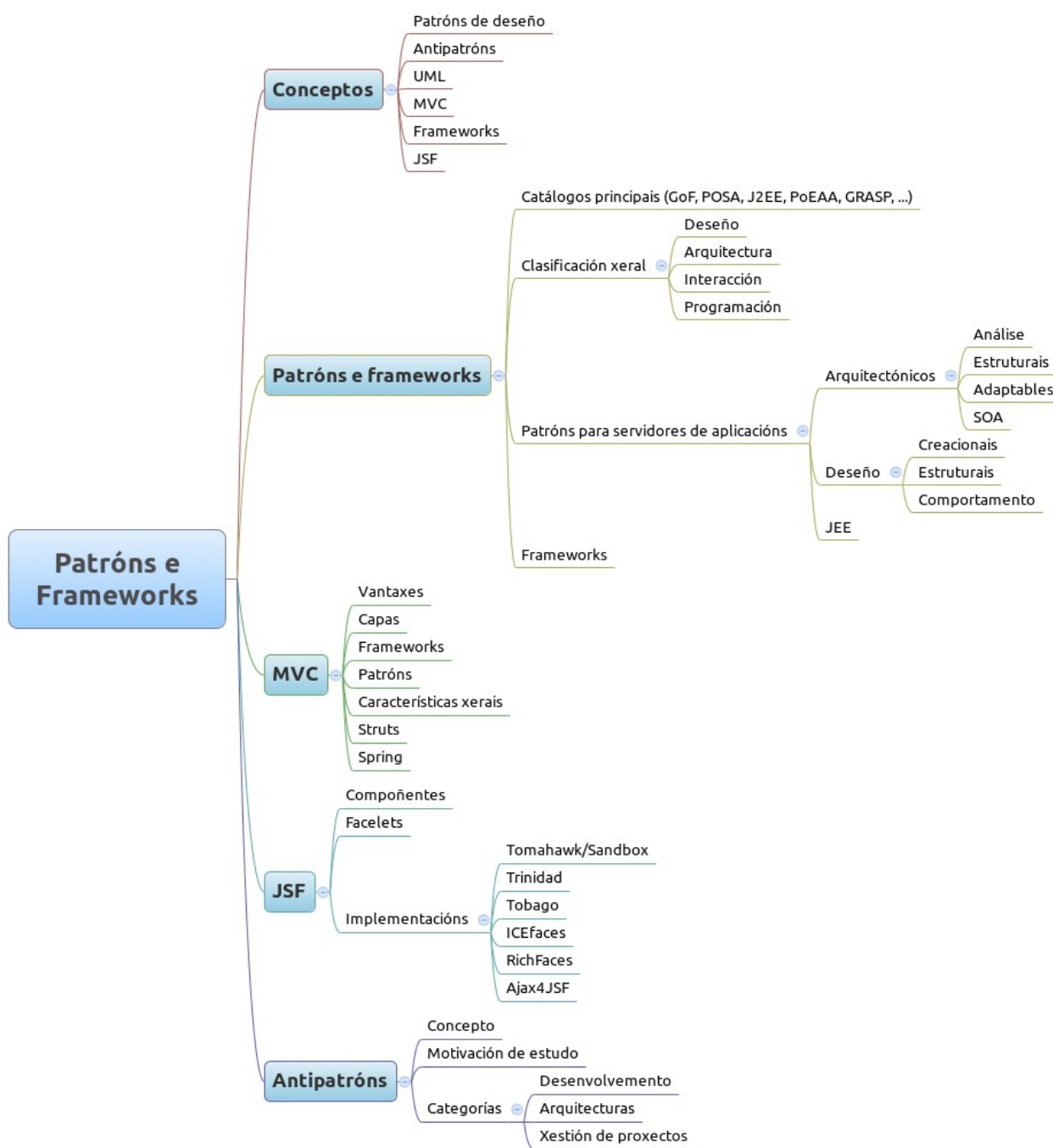
de integración llegada la necesidad del mismo.

- d) **Diseño por comité.** Demasiadas personas participan de los requerimientos del proyecto dando lugar a un diseño demasiado abstracto y excesivamente complejo a causa de contemplar demasiados puntos de vista particulares. Se complica la toma de requisitos dando lugar a demasiadas reuniones de larga duración, que dificultan y provocan errores al largo de todo el ciclo de vida de desarrollo.
- e) **Arquitectura por implicación.** No existe documentación de la arquitectura del sistema, ni de los procesos, ni de las tareas automatizadas más habituales.

En lo que respecta a los **antipatrones de gestión de proyectos** software destacan por ser los más habituales:

- a) **Parálisis de análisis.** Los procesos de análisis y diseño se prolongan tanto que el proyecto termina muriendo en él sin llegar a implementarse. Son desarrollos opuestos a los modelos basados en prototipos e iterativos.
- b) **Muerte por planificación.** Demasiada planificación y reuniones sin llegar a concretar puntos de partida para el desarrollo. De nuevo son desarrollos opuestos a los modelos basados en prototipos e iterativos.
- c) **Personas problemáticas** (en inglés *corncob*). Personas difíciles para participar en equipos, o con poca capacitación o aptitud, obstruyen, desvían e incluso sabotean el desarrollo.
- d) **Gestión irracional.** La falta de decisión y capacitación sumado a la nula planificación pueden dar lugar a la toma de decisiones con posterioridad y desarrollos de urgencia.
- e) **Proyectos sin gestión.** No se atiende al análisis y el diseño, sólo a implementación. Se van arreglando incidentes según se producen en modo pila, las últimas primero.

## **29.6 ESQUEMA**



**Traducción texto Esquema:** Patrones de diseño. Antipatrones. Catálogos principales. Clasificación general. Diseño. Patrones para servidores de aplicaciones. Análisis. Estructurales. Creacionales. Comportamiento. Ventajas. Características generales. Componentes. Implementaciones.

Motivación de estudio. Gestión de proyectos.

## **29.7 REFERENCIAS**

Deepak Alur y otros.

Core J2EE Patterns. Best Practices and Design Strategies. (2003).

William Crawfor y Jonathan Kaplan.

J2EE Design Patterns. (2003).

Steven Metsker y William Wake.

Design Patterns in Java. (2006).

**Autor: Juan Marcos Filgueira Gomis**

**Asesor Técnico Consellería de Educación e O. U.**

**Colegiado del CPEIG**



## **30. HTML. APLICACIÓNS DA INTERNET ENRIQUECIDAS (RIA).**

## **TEMA 30. HTML. APLICACIONES DE INTERNET ENRIQUECIDAS (RIA).**

### **30.1 INTRODUCCIÓN Y CONCEPTOS**

### **30.2 HTML**

### **30.3 AJAX**

### **30.4 RIA PARA MULTIMEDIA Y ANIMACIONES**

### **30.5 OTRAS TECNOLOGÍAS RIA**

### **30.6 ESQUEMA**

### **30.7 REFERENCIAS**

### **30.1 INTRODUCCIÓN Y CONCEPTOS**

Los documentos web y MIME pueden definirse a través del lenguaje de marcado HTML, (en inglés *HyperText Markup Language*) que permite describir la estructura y contenido de un documento a través de etiquetas, además de permitir otros elementos complementarios como estilos o *scripts*. Variantes de este lenguaje permiten un dominio más estricto adaptado a ámbitos concretos como el XHTML o MathML y más abiertos como SGML (en inglés *Standard Generalized Markup Language*). La última versión del lenguaje HTML5 incorpora etiquetas específicas para contenidos multimedia, grandes conjuntos de datos, mejoras en formularios, web semántica y otras opciones, que responden a la evolución de las necesidades de la web hoy en día.

En extensión, las aplicaciones de internet enriquecidas o **RIA** (en inglés *Rich Internet Applications*), son un conjunto de tecnologías que buscan acercar las interfaces de las aplicaciones web a las de las aplicaciones de escritorio dotándolas de nuevas funcionalidades, de ahí la riqueza, y agilizando aspectos como las recargas de datos. Por norma general precisan de un *framework*, componente adicional o *plug-in* en el navegador que permitan su interpretación. En las aplicaciones RIA la mayor parte de la

comunicación se hace de manera asíncrona en comunicaciones transparentes al usuario que evitan gran parte de las recargas de páginas para realizar actualizaciones de datos. Frente a estas ventajas en lo tocante a la usabilidad en cuanto a la mejora de las funcionalidades y actualizaciones de datos, la principal desventaja será la accesibilidad de la página para usuarios que presenten dificultades de acceso a la información en la web.

Muchas de estas **tecnologías** pertenecen al mundo del software propietario encontrando gran dependencia respecto de las compañías que las desarrollan. Las principales tecnologías se encuentran en los andenes Flash, Flex y AIR de Adobe, Silverlight de Microsoft, OpenLaszlo, incontables *frameworks* AJAX y Javascript, y otras tecnologías como las ya maduras, Applets y Java WebStart y las emergentes como XUL, JavaFX, GWT o Bindows. Cualquiera de estas tecnologías se localiza en la capa de vista, usuario o cliente como complemento a la (X)HTML/CSS y se integran con las tecnologías de servidores de aplicaciones como .NET/JEE.

Dentro de los **casos de éxito** de esta tecnología, actualmente la mayor parte de las aplicaciones y servicios web globales de mayor uso dentro de la web 2.0 hacen uso de tecnologías RIA en sus interfaces, siendo Google Maps, Gmail, Flickr, Meebo, Orkut, y un largo etcétera.

## **30.2 HTML**

El conjunto de estándares HTML presentan un conjunto de normas definido por el W3C, siendo la versión estándar más habitual la HTML 4.01, la reciente HTML 5 y la variante XHTML 1.0. Un documento HTML describe la

estructura y contenido de una página o sitio web, permite acceso al DOM y hace uso de archivos multimedia y *scripts* externos que a su vez pueden adoptar la forma de tecnologías RIA o DHTML. En este caso en el documento HTML se especifica un objeto incrustado en la tecnología correspondiente y cualquier navegador lo identifica y lo reproduce según el correspondiente plug-in. Gracias al HTML puede definirse un documento con los siguientes elementos:

- Publicación de contenidos en línea, como textos, encabezados, listas, tablas, párrafos, formatos y estilos, etc.
- Asociación de documentos vía enlaces de hipertexto.
- Diseño de formularios para permitir la entrada de información y comunicación con servidores remotos.
- Incrustación en los documentos de objetos externos como elementos multimedia u objetos de aplicaciones externas como hojas de cálculo, documentos PDF y otros.

Los documentos HTML especifican también la **codificación de caracteres** internacional del documento siendo UTF-8 o Latin-1. Esta identificación de la codificación permite que otras aplicaciones, como navegadores o procesadores de texto, puedan reconocer y presentar adecuadamente la información del documento. Asimismo dentro de esta codificación, los caracteres especiales se representan a través de **entidades HTML**, que definen de manera unívoca el carácter especial a través de un código HTML específico.

Carácter especial	Entidad HTML
Á	&Aacute;
á	&aacute;



<b>Espacio en blanco</b>	&nbsp;
<b>Ñ</b>	&Ntilde;
<b>ñ</b>	&ntilde;
<b>Ü</b>	&Uuml;
<b>&gt;</b>	&gt;
<b>&lt;</b>	&lt;

***Tabla 1: Ejemplos de entidades HTML.***

### **30.2.1 ACCESIBILIDAD WEB**

Otro aspecto asociado a la definición de documentos web es el de **accesibilidad web**, donde el W3C elaboró las Pautas de Accesibilidad al Contenido Web 1.0 o WCAG adaptadas en el caso español con la Norma UNE 139803. Estas guías definen una serie de reglas y normas que deben cumplir los documentos web para permitir el acceso a su información a personas con discapacidad o que presentan dificultades específicas.

Las WCAG 1.0 son catorce pautas o patrones de diseño web con soluciones comunes para resolver problemas de acceso a contenidos web. Las pautas contienen una serie de puntos de verificación que facilitan la detección de posibles problemas de acceso. Cada uno de los puntos ha asignado un nivel de prioridad o peso según a la importancia o dificultad de acceso a la información que lleve asociado. Estos niveles son:

- **Prioridad 1.** Puntos que debe cumplir un sitio web o de otra manera no se permitirá que ciertos grupos de usuarios no podrán acceder a los contenidos del sitio.



- **Prioridad 2.** Puntos que debe cumplir un sitio web o de otra manera a ciertos grupos de usuarios les será muy difícil acceder a los contenidos del sitio.
- **Prioridad 3.** Puntos que debe cumplir un sitio web o de otra manera algunos grupos de usuarios podrían tener dificultades a la hora de acceder a los contenidos del sitio.

En función al cumplimiento de estos puntos de verificación se establecen tres niveles de conformidad de un sitio web con las pautas de accesibilidad:

- Nivel de Conformidad **A**. Se cumplen todos los puntos de verificación de prioridad 1.
- Nivel de Conformidad **Doble A**. Se cumplen todos los puntos de verificación de prioridad 1 y 2.
- Nivel de Conformidad **Triple A**. Se cumplen todos los puntos de verificación de prioridad 1,2 y 3.

Prioridad	Punto de verificación	Cumplimiento
1	Proporcionar texto equivalente a elementos no textuales	S/N/N.A.
1	Asegurar que la información transmitida con color tiene alternativa sin color	S/N/N.A.
1	Identificación de cambios de idioma	S/N/N.A.
1	Organizar el documento independientemente de las hojas de estilo	S/N/N.A.
1	Garantizar que las alternativas al contenido dinámico se actualizan con él	S/N/N.A.
1	No provocar destellos de pantalla	S/N/N.A.
1	Emplear un lenguaje claro y simple en el sitio	S/N/N.A.
1	Proporcionar enlaces de texto redundantes en mapas de imágenes	S/N/N.A.

<b>1</b>	En las tablas identificar las cabeceras de fila y columna	S/N/N.A.
<b>1</b>	Titular cada marco o <i>frame</i> de la página	S/N/N.A.
<b>1</b>	Permitir el funcionamiento sin lenguajes de scripts de cliente activados	S/N/N.A.
<b>1</b>	Proveer alternativas auditivas sincronizadas en contenidos multimedia	S/N/N.A.
<b>1</b>	En el caso de que sea imposible cumplir con los puntos anteriores hay que crear un documento completo alternativo	S/N/N.A.
<b>2</b>	Garantizar contraste de color de texto y fondo legible	S/N/N.A.
<b>2</b>	Emplear hojas de estilo para maquetación y presentación	S/N/N.A.
<b>2</b>	Emplear documentos validados formalmente	S/N/N.A.
<b>2</b>	Emplear elementos de encabezado	S/N/N.A.
<b>2</b>	Emplear unidades relativas frente a absolutas	S/N/N.A.
<b>2</b>	Marcar adecuadamente listas, ítems y citas	S/N/N.A.
<b>2</b>	Evitar navegación o refresco de páginas automáticos	S/N/N.A.
<b>2</b>	Evitar abrir nuevas ventanas sin informar al usuario	S/N/N.A.
<b>2</b>	Emplear índices o mapas de los sitios y metadatos para aportar información semántica	S/N/N.A.
<b>3</b>	Proporcionar un orden lógico de tabulación	S/N/N.A.
<b>3</b>	Proporcionar atajos de teclado	S/N/N.A.
<b>3</b>	Incluir caracteres por defecto en los formularios	S/N/N.A.
<b>3</b>	Proporcionar abreviaturas para los encabezados de las páginas	S/N/N.A.
<b>3</b>	Proporcionar resúmenes para las tablas	S/N/N.A.
<b>3</b>	Proporcionar barras de navegación	S/N/N.A.

**Tabla 2: Ejemplo de lista de puntos de verificación.**

Paralelamente se desarrollaron las **Pautas de accesibilidad para herramientas de autor** o ATAG (en inglés *Authoring Tool Accessibility Guidelines 1.0*) con el objetivo de ayudar a los desarrolladores a la hora de

crear herramientas de autor destinadas a la elaboración de contenidos web accesibles; las **Pautas de accesibilidad para XML** o XAG; y las **Pautas de accesibilidad para agentes de usuario** 1.0 o UAAG orientadas a definir un marco accesible de soporte a componentes software que acceden a contenidos web como navegadores, reproductores multimedia, procesadores de texto y otras tecnologías asistivas. Por último dentro del marco de las aplicaciones RIA la iniciativa **WAI-ARIA** comienza a dar soporte a requerimientos de accesibilidad para este tipo de tecnologías.

Una de las reglas de accesibilidad web es cumplir con la validez del documento. Para evaluarla una de las etiquetas, DOCTYPE, permite especificar la versión de HTML empleada para elaborar el documento. Además de permitir corregir problemas de compatibilidad con navegadores permite validar que el documento esté bien formado.

Existen tres **definiciones del tipo de documento** o DTD:

- **Strict.** Estricto, respetando escrupulosamente las normas y sintaxis del documento, y evitando parte del conjunto de etiquetas posible total.
- **Transitional.** Con más libertad, hace uso de todo el conjunto de etiquetas y permite licencias en la sintaxis.
- **Frameset.** Reservado para documentos que hacen uso de marcas para organizar la estructura del sitio web.

A partir de esta especificación existen herramientas de **validación** en línea y para descarga, que orientadas hacia la accesibilidad o hacia la corrección del código permiten evaluar la calidad del documento.

### **30.2.2 HTML PARA DISPOSITIVOS MÓVILES**

Asimismo al aumentar las posibilidades de conexión de los **dispositivos móviles**, creció la demanda de que den soporte web con esta orientación y por tanto de adaptar los documentos a las necesidades específicas de este tipo de equipos, muchas de las mismas muy ligadas al ámbito de la accesibilidad y el diseño líquido. Los protocolos más habituales **WAP**, **WAP 2.0** y **WCSS** recogen los DTD más empleados. El W3C proporciona un subconjunto específico para móviles el XHTML 1.1 Basic, y la OMA (en inglés *Open Mobile Alliance*) la especificación XHTML Mobile Profile, ambos garantizan el soporte de los documentos que cumplan las convalidaciones con los dispositivos más restringidos. En el caso de WAP no se encuentran soportados los protocolos de internet, por lo que hay que emplear nodos pasarela. Asimismo dispone de un lenguaje de *script* propio, **WMLScript** basado en ECMAScript y empleado en las páginas **WML** (en inglés *Wireless Markup Language*) que emplean a su vez XML. La siguiente versión WAP 2.0 sí adopta los protocolos de internet, lo que hace que no sea imprescindible la pasarela, presentando además mejoras de rendimiento para la comunicación en dispositivos móviles dentro de estos protocolos.

Otras tecnologías relacionadas con el diseño web para móviles serían **i-mode**, que presenta un lenguaje muy similar a HTML para la creación de micropáginas y situada a niveles de mercado cerca del WAP en países como Japón. Otra tecnología complementaria sería el **J2ME** (en inglés *Java2 MicroEdition*) versión ligera, o subconjunto, destinado a dispositivos móviles de la plataforma Java.

### **30.2.3 HTML 5**

El HTML busca dar respuesta a muchas de las necesidades actuales de los sitios web y que no fueron recogidas en las versiones anteriores de la especificación. En esta versión tiene lugar una mayor integración con el

DOM, a través del IDL (en inglés *Interfaz Definition Language*). Dentro de las nuevas características de HTML5 se encuentran:

- **Una nueva estructura de documento.** Con secciones como NAV, bloque de navegación del sitio web, HEADER, FOOTER, ARTICLE o SECTION que aportan información semántica más relevante que emplear únicamente DIVS o tablas.
- **Nuevas etiquetas semánticas.** Como TIME para fechas/tiempo o MARK para destacar elementos, METER, PROGRESS, así como incorporación en los documentos de archivos RDF o OWL con metadatos.
- **Nuevas etiquetas multimedia.** Como AUDIO y VIDEO que presenta además los códecs necesarios para reproducir los contenidos incrustados.
- **Etiquetas para gráficos.** Las mejoras en el componente CANVAS permiten la representación de gráficos 2D/3D, como formas, degradados, transformaciones y otras opciones gráficas.
- **Nuevos componentes para formularios.** Con nuevos tipos de datos para correo electrónico, teléfono, números, fechas, etc. que permiten la validación en el cliente sin emplear Javascript.
- **Componentes para operar con grandes conjuntos de datos.** Componentes como DATAGRID, MENU o COMMAND, proporcionan a estos elementos comunes de los sitios web características dinámicas que facilitan el control del comportamiento de la página sin librerías externas.
- **Nuevas APIS.** Librerías para operaciones tan variadas como arrastrar objetos, funcionamiento fuera de línea, geolocalización, almacenamiento persistente en local basadas en SQL Lite, WebSockets para comunicación entre sitios distribuidos, WebWorkers para trabajo con hilos de ejecución paralelos y otras posibilidades que se encuentran en estudio o desarrollo.

**Figura 1: Ejemplo de estructura de un documento en HTML5.**

### **30.3 AJAX**

**AJAX** (en inglés *Asynchronous JavaScript And XML*), Javascript Asíncrono y XML, se origina para aprovechar que la comunicación entre el usuario y la interfaz no es fluida permitiendo realizar comunicaciones asíncronas con el servidor y realizar así un mayor aprovechamiento del ancho de banda y obtener una mayor velocidad de respuesta. Los datos se cargan en un segundo plano sin afectar a la interfaz. AJAX no representa una tecnología en sí misma, sino que se trata de la combinación de un grupo de **tecnologías** ya existentes:

- (X)HTML y CSS para el diseño de las páginas web.
- DOM (en inglés *Document Object Model*), o Modelo de Objetos del Documento, API que representa un conjunto de objetos para manipular y modificar dinámicamente documentos (X)HTML y XML a través de lenguajes de script como Javascript, JScript o ECMAScript.
- Objetos de los tipos `Iframe` o `XMLHttpRequest` para intercambiar datos de manera asíncrona con el servidor.
- XML para los formatos de intercambio de datos y comunicaciones a través de JSON o EBML.
  
- Otras tecnologías para facilitar la implantación de soluciones específicas como XSLT, RSS, PDF u otras tecnologías RIA.
- Javascript para proporcionar el nexo común a todo el conjunto.

Como ocurre con las tecnologías RIA en general, es requisito que el navegador tenga **soporte** para el conjunto de tecnologías AJAX, de otra

manera habría que proporcionar una alternativa HTML básica. Aun así cada vez se encuentra soportado por un mayor número de navegadores y no requiere la instalación de complementos.

El **funcionamiento básico** de AJAX se basa en el objeto de comunicación asíncrona, por ejemplo el XMLHttpRequest, que se instala con una librería, *framework* o motor AJAX en el lado del cliente. El motor AJAX proveerá los métodos para permitir la comunicación asíncrona de datos además de definir a los componentes reutilizables AJAX con la definición de su comportamiento, y el contenido y estructura general del documento. El hecho de realizar todas estas funciones desde el cliente de manera asíncrona supone la gran mejora de rendimiento de AJAX sobre el modelo tradicional de desarrollo web.

**Figura 2: Funcionamiento básico de AJAX.**

La **refactorización** de aplicaciones tradicionales al modelo AJAX implica cambios en la estructura interna con la preservación de las funcionalidades. La metodología de refactorización habitual implica una serie de cambios a pequeña escala, para lo cual es ideal la programación orientada a objetos. Estos cambios van desde:

- **Refactorización a nivel de método/clase.** En sistemas con poco acoplamiento, evitando por ejemplo que se modifiquen los atributos de las clases entre objetos. Estos accesos múltiples pueden integrarse en un nivel superior una vez resueltas las dependencias del método/clase.

Tecnologías RIA

Tecnología  
s

Contenido y

- **Creación de nuevas clases.** Se definen nuevas clases especializadas en la arquitectura AJAX con responsabilidades e interfaces bien definidos.
- **Eliminación de clases intermediarias.** Se eliminan las delegaciones en exceso entre clases del modelo tradicional en



aplicación de las consideraciones sobre antipatrones de exceso de capas.

- **Componentes y etiquetado.** Diferentes componentes sobre todo de la vista presentan funcionalidades comunes que se pueden factorizar a través de componentes AJAX y etiquetados, para facilitar el mantenimiento y la reutilización.

En la actualidad existen infinidad de **frameworks y librerías AJAX**, incorporando componentes de la vista, diferentes funcionalidades Javascript, elementos para comunicación asíncrona tanto en el cliente como en el servidor y otros elementos para integración con otras tecnologías RIA. El objetivo de estos *frameworks* se resume en facilitar el desarrollo de aplicaciones web basadas en AJAX, haciendo hincapié en aspectos de la capa de vista o cliente. Los principales *frameworks* en cuanto a su uso más extendido, son:

- **Prototype.** Puede ser el *framework* de uso más extendido, también de código abierto dispone de la extensión **Scriptaculous** para añadir animaciones y efectos en los documentos, y JSON para intercambio de datos. Sirve a su vez a base de otros *frameworks* AJAX. Permite una gran integración en aplicaciones desarrolladas con *Ruby on Rails* pero también puede operar de manera independiente. Sus principales características son:
  - a) **DOM Extendido.** Referencia ágil a objetos DOM, como por ejemplo empleando la función `$()` en lugar de `document.getElementById()`.
  - b) **Scriptaculous.** Aporta al *framework* un constructor de objetos DOM (*builder.js*), un repositorio de efectos visuales (*effects.js*, *slider.js*), funcionalidades de control de elementos (X)HTML (*dragdrop.js*, *controls.js*) y métodos para realizar test de verificación unitarios (*unittest.js*).



- **Dogo Toolkit.** Proyecto de software libre, actualmente con el soporte de IBM y Sun entre otros, que contiene varias APIs Javascript modulares y una amplia colección de widgets para uso bajo demanda, agrupados en un sistema de paquetes al estilo de JEE. Entre sus principales características están la aportación de:
  - a) **Componentes empaquetados en Dijit.** Widgets para la estructura de las páginas como menús, pestañas; específicos para calendarios, relojes, gráficos, vectores 2D/3D, ordenación de tablas y paginación; además de elementos para formularios y su convalidación, elementos HTML5 y componentes para mejorar la accesibilidad. Asimismo presenta un editor de texto enriquecido y soporte drag & drop entre sus componentes.
  - b) **Comunicación asíncrona.** Provee de una capa de abstracción para la comunicación transparente entre el navegador y el servidor web que hace uso de elementos Iframes ocultos para el refresco de datos.
  - c) **Almacenamiento en el servidor.** Implanta diferentes mecanismos de almacenamiento de datos vía CVS, OPML, RDF o el servicio web Del.icio.us.
  - d) **Soporte para otras tecnologías.** Permite integración con tecnologías RIA como las aplicaciones AIR basadas en Javascript a través de API y soporte para móviles.
  
- **jQuery.** Otro proyecto de software libre que en este caso, busca simplificar el acceso a documento (X)HTML, el modelo DOM, la manipulación de eventos, utilidades, animaciones y efectos. Permite la instalación a través de un paquete básico muy ligero (*jquery.js*) que puede ser ampliado a través de plug-ins como JExpand, para tablas, y JQueryUI para widgets con efectos visuales, entre otros muchos. Entre sus principales características se encuentran:

- a) Integración en la plataforma .NET y con los *frameworks* ASP .NET MVC y ASP .NET AJAX.
- b) Soporte para CSS 3 y XPath.
- c) Soporte para manipulación de (X)HTML y JSON.
- d) Hace uso de programación no intrusiva.
- e) Ligero y extensible.

□ **Qooxdoo.** Colección de librerías Javascript multipropósito de código abierto, que como las vistas anteriormente permite control ágil a alto nivel de (X)HTML, CSS y DOM, además de proporcionar funcionalidades extendidas. La principal diferencia respecto a las anteriores es que proporciona widgets de última generación muy similares a los de las aplicaciones de escritorio. Entre sus principales características se encuentran:

- a) **Abstracción del navegador.** Establece una capa intermedia de abstracción del navegador con las especificaciones necesarias para los principales tipos de navegadores definiendo así una interfaz estándar que mejora la compatibilidad sin necesidad de instalar *plug-ins* adicionales.
- b) **Administración de eventos.** Provee una interfaz propia con métodos para registrar y eliminar eventos.
- c) Herramienta de desarrollo de interfaces de usuario.
- d) Soporte de internacionalización i18n y localización l10n que permite el formato de traducción basado en archivos .po.
- e) Provee *frameworks* para depuración de test unitarios y simulaciones.

□ **Mootools.** *Framework* de código abierto modular y extensible que permite al desarrollador seleccionar que componentes a emplear



para minimizar el peso final de la librería en el cliente. Presenta un componente para la incorporación de efectos avanzados y transiciones, estrechamente relacionados con Flash siendo un punto fuerte su integración con esta otra tecnología RIA. Provee de los siguientes componentes:

- a) **Core.** Núcleo de funciones básicas que emplean todos los demás componentes del *framework*.
- b) **Class.** Librería para instanciación y manipulación de objetos.
- c) **Natives.** Extensiones de funciones básicas Javascript.
- d) **Element y Effects/FX.** APIs para manejo de documentos HTML y aplicación de efectos sobre sus elementos.
- e) **Remote.** Para intercambio de datos con el servidor a través de peticiones XmlHttpRequest, JSON o Cookies.

- **ExtJS.** Conjunto de librerías derivadas de la Yahoo! UI, actualmente se emplea como extensión de JQuery y Prototype incorporando *widgets* especializados, en especial en la representación de gráficas y *grids*. Existe además una adaptación específica para GWT denominada ExtGWT, con muchas optimizaciones para este contorno. Incorpora una capa propia dentro de una arquitectura MVC lo que le permite proveer de flexibilidad en lo tocante a los estilos, haciendo uso de la extensión SASS (en inglés *Syntactically Awesome Style Sheets*), una extensión de CSS3. Asimismo provee de librerías que facilitan la integración con AIR y Spring como backend. Dentro de los componentes de datos dispone de varios lectores tanto para XML como JSON. La arquitectura general incluye los paquetes Base y Core con las funcionalidades comunes; los componentes de la interfaz de usuario con los widgets y gadgets; Remoting para la ejecución de métodos en el servidor vía RPC; los servicios de datos para lectura de

vectores, XML y JSON; y el *miniframeframework Drag and drop* para permitir soporte de arrastre entre los componentes del *framework*.

□ **Rico.** Basado en Prototype y orientado hacia la Web 2.0 las principales aportaciones de este *framework* incluyen efectos de animaciones que permiten realizar transiciones que pueden ser interrumpidas, pausadas o reiniciadas, permitiendo el solapamiento de animaciones. Permite la creación de efectos cinematográficos y otros efectos visuales.

Asimismo proporciona las funcionalidades básicas para soporte AJAX y extiende parte del repertorio de Prototype con mejoras.

□ **DWR.** (En inglés *Direct Web Remoting*). Framework de código abierto orientado a la integración de AJAX con aplicaciones JEE a través de mecanismos de RPC como RMI o SOAP. Permite ejecutar código Java en un servidor de aplicaciones como si estuviese en el navegador del cliente, invocando los objetos como si fuesen locales. Consta de dos elementos principales: un *framework* Javascript en el cliente y un Servlet en el servidor para procesar las peticiones y generar las respuestas. El Javascript actuará como proxy de las clases Java permitiendo que en ese código se incluyan las clases Java bajo servidor. En una llamada a un método de una clase, DWR genera dinámicamente una versión Javascript de la clase *AjaxService*, invocada a través de un manejador de eventos que gestiona la interacción con el servidor. Cuando llega la respuesta al cliente se invoca una función *callback* para actualizar el contenido del documento. Este método se denomina Reverse AJAX, soportando tres métodos básicos de envío de datos:

- 1) **Polling.** El navegador pregunta al servidor en intervalos regulares si completó la petición.



- 2) **Piggyback.** El servidor espera a la siguiente petición del navegador para darle la respuesta.
- 3) **Comet.** El servidor responde al navegador de manera planificada tipo Streaming en una respuesta Http larga.

Asimismo, provee de dos opciones de comunicación remota:

- 1) **DWR nativo.** Empleando un superconjunto de JSON donde el motor DW (*engine.js*) maneja las peticiones y prepara la ejecución de las llamadas al servidor.
- 2) **JSON/JSONP.** API para JSON que facilita la integración con otros *frameworks* como Dogo, ExtJS o JQuery.

En lo tocante al tema de la seguridad DWR contempla protecciones específicas contra ataques XSS (en inglés *Cross Site Scripting*) y CSRF (en inglés *Cross Site Request Forgery*).

- **SAJAX.** (En inglés *Simple AJAX Toolkit*). Herramienta de código abierto que de manera análoga a DRW permite realizar llamadas a métodos del servidor en PHP, ASP, Coldfusion, Ruby, Perl, Phyton y otros lenguajes desde Javascript en el navegador, sin tener que recargar la página.

Provee de una API para cada lenguaje de servidor, por ejemplo *Sajax.php*, que se incluye en el código de este para permitir la integración con el Javascript del navegador.

- **GWT.** (En inglés *Google Web Toolkit*). *Framework* de desarrollo AJAX dentro de aplicaciones Java. Este contorno permite que al definir una interfaz Java se traduzca con el compilador GWT de manera transparente a Javascript y HTML. El principal objetivo de este *framework* es integrar en un mismo ID el desarrollo de la aplicación y de la parte de interfaces de usuario con AJAX pero además provee de otras funcionalidades como componentes HTML dinámicos y reutilizables, protocolos de transferencia XML y JSON,

internacionalización i18n, integración con JUnit incluso en las llamadas RPC y con Javascript a través de JSNI. La arquitectura de GWT se estructura en los siguientes elementos:

1. **Compilador Java a Javascript GWT.** Para aplicaciones web genera automáticamente el código Javascript necesario para la interfaz Java definida.
2. **Hosted Web Browser.** Motor de ejecución de aplicaciones Java sin traducirlas a Javascript a modo de maquinilla virtual Java.
3. **Librería de Emulación JRE.** Contiene los principales paquetes Java de uso común soportadas por GWT.
4. **Librería de clases de Interfaces de Usuario GWT.** Provee de un conjunto de componentes para interfaces de usuario.

A mayores de estos *frameworks* existen infinidad de alternativas y librerías para temas específicos o versiones más o menos simples de propósito general, como:

- 1) **AjaxAC.** *Framework* PHP que emplea AJAX en el cliente y se orienta a la reutilización por disponer de clases muy simples.
- 2) **AJAX .NET Professional.** Librería AJAX para ASP .NET con funcionalidades básicas para controles de usuario y utilidades de uso general.
- 3) **ATLAS.** También denominado ASP .NET AJAX, integra en un mismo *framework* un conjunto de extensiones para integrar AJAX en .NET, que incluye a Microsoft Ajax Library.
- 4) **BAJAX.** Librería Javascript muy ligera (<6 K), para integrar AJAX de la forma más simple posible.
- 5) **Taconite.** *Framework* para desarrollo AJAX, que automatiza tareas para gestionar el objeto XMLHttpRequest o la creación de contenido dinámicamente.

- 6) **Spry Framework fuera Ajax.** Librería a Javascript de Adobe para la integración de AJAX con orígenes de datos XML, JSON y HTML para lenguajes de servidor como Coldfusion, PHP o ASP .NET. Spry ofrece tres componentes principales: Datos, Widgets y Efectos.
- 7) **Tacos.** Librería que proporciona componentes, efectos, validaciones y funcionamiento AJAX para el *framework* Tapestry.
- 8) **XAJAX.** *Framework* AJAX para desarrollo en PHP, que permite desde el navegador llamar a funciones del servidor.
- 9) **Zephyr.** *Framework* AJAX para desarrollo en PHP5 bajo el modelo MVC. Provee de un motor de modelos, soporte para datos adoDB y otras opciones.
- 10) **ZK.** *Framework* para desarrollo de aplicaciones Java que busca hacer transparente la tecnología Javascript. Los componentes de la interfaz de usuario se relacionan con componentes POJO en el servidor. Se recomienda la integración con Spring, Toplink o Hibernate y aporta protecciones contra ataques XSS, CSRF y DoS. Se verá detalladamente más adelante.

## **30.4 RIA PARA MULTIMEDIA Y ANIMACIONES**

Al ir mejorando el ancho de banda de las conexiones fueron en aumento las tecnologías RIA destinadas a mejorar las funciones multimedia, gráficos vectoriales, animaciones e interactividad. La pionera de estas tecnologías fue Flash para posteriormente aparecer Flex, AIR, JavaFX, OpenLaszlo y Silverlight como principales alternativos.

### **30.4.1 FLASH RIA**





La plataforma Flash evolucionó de plug-in en el cliente (Flash Player) para la visualización de imágenes vectoriales y animaciones, hacia una arquitectura RIA para dotar de nuevas funcionalidades a las interfaces web. Representa la primera tecnología RIA, siendo el marco que engloba diferentes tecnologías dentro del que se denomina Flash RIA. Dentro de las Flash RIA se encuentran las tres tecnologías principales soportadas por Adobe, y anteriormente Macromedia, las cuales tienen el uso más extendido:

- 1.1. **Flash.** Empaqueta de aplicación en archivos SWF a modo de componentes.
- 1.2. **Flex.** A partir de un servidor de aplicaciones JEE realiza la comunicación con el SWF permitiendo llamar a objetos del servidor.
- 1.3. **AIR.** Para ejecución de aplicaciones Flash en el equipo del cliente sin necesidad de navegadores como intermediarios.

Alternativamente, existen otros proveedores de tecnologías Flash RIA, entre los que se encuentran:

1. **OpenLaszlo.** Muy similar a Flash pero con un lenguaje de programación propio LZX.
2. **SnappMX.** Orientada a servicios web.
3. **Zulu.** Permite desarrollar aplicaciones conjuntamente con el estándar XUL y Flash.
4. **XAMLON.** Permite desarrollar aplicaciones Flash con el lenguaje de marcado XAML dentro de la plataforma .NET.

La orientación principal de Flash a diferencia de AJAX es la de ser un complemento del interfaz de usuario extendiendo determinadas funcionalidades, en especial en el campo de las animaciones y de la interacción con el usuario. Los entornos de desarrollo de Flash están más

orientados hacia la edición de animaciones que hacia el desarrollo web, pero se fue abriendo camino en el campo de la elaboración de widgets, soporte multilenguaje efectos 3D control y validación de formularios. Permite integración con AJAX y Javascript pero dispone de un lenguaje de *script* propio denominado **ActionScript**. Este *script* de programación orientada a objetos sigue el estándar ECMA-262, implementando E4X (en inglés *ECMAScript fuera XML*). Opera con un modelo de eventos basado en la especificación DOM, aunque no lo sigue completamente. Se lanza en una maquinilla virtual específica AMV2 (en inglés *ActionScript Virtual Machine*) alojada en el contorno de ejecución Flash Player. Por último destacar que permite conectividad con servicios web y bases de datos de manera remota a través de la clase *DataProvider*.

Como sucede en la mayoría de *frameworks* de desarrollo la plataforma Flash puede ampliarse con paquetes de librerías de terceros, entre estas destacan:

- a) **SPL** (en inglés *Spelling Plus Library*). Para diseño de editores de texto enriquecido con corrección ortográfica.
- b) **Red5**. Servidor Flash de software libre.
- c) **Papervision3D**. Motor de generación 3D de software libre.
- d) **As3corelib**. Paquete de librerías ActionScript 3 que contiene clases y utilidades de uso común. Incluye codificadores de imagen, serialización JSON, APIs para fechas, Strings y otros tipos de datos y codificación de llaves MD5 y SHA 1.
- e) **SWFObject**. Javascript para incrustar contenido Flash en documentos (X)HTML
- f) **Tweener**. Para crear animaciones y transiciones directamente trabajando con ActionScript.
- g) **Gaia**. *Framework* orientado hacia la agilización del desarrollo Flash.

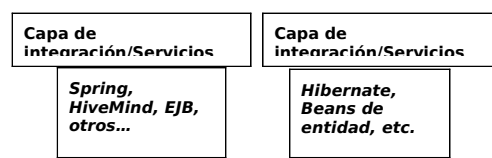
### 30.4.2 FLEX

Flex es la evolución de Flash ampliando el ámbito de desarrollo RIA con nuevas tecnologías y formatos. Se diferencia de Flash en su facilidad de integración con tecnologías de lenguajes de servidor lo que facilita el uso de patronos de diseño y que emplea MXML (en inglés *Macromedia eXtensible Markup Language*) para definir el aspecto y comportamiento de los interfaces de usuario. Como Flash, soporta el lenguaje ActionScript y su plataforma incorpora librerías de componentes para interfaces de usuario específicas. Permite integración con otras tecnologías del lado del servidor como servicios web, REST o AMF. Las aplicaciones Flex pueden integrarse en un documento HTML de manera que este puede actualizar dinámicamente la vista y enviar y recibir datos asíncronamente con el servidor de fondo, de manera similar a AJAX.

En las **comunicaciones cliente servidor**, Flex en el cliente se comunica con el servidor vía HTTP, disponiendo de tres API RPC: HTTPService, WebService y RemoteObject. No acceden directamente a bases de datos remotas sino que lo hacen a través de capas intermedias. A través de HTTPService solicita archivos JSP o XML con los datos en formato de variables String, formatos de intercambio XML, E4X u objetos ActionScript. En el caso de devolver JSON, Flex dispone de librerías especializadas para serialización así como para SOAP. A través de la API de RemoteObject permite realizar peticiones **Flash Remoting** que devuelven mensajes binarios **AMF** (en inglés *Action Message Format*) sobre HTTP. Cuando este formato tiene aplicación se obtiene un mayor rendimiento que en otras tecnologías, como JSON o SOAP.

#### **Figura 3: Integración de Flex en JEE.**

Asimismo Flex permite intercambio de datos en tiempo real ante dos vías: **XML Socket** y **Socket Binario**. Con el **XML Socket** se crea una conexión



que permanece abierta mientras dure la comunicación, o es cerrada explícitamente. En el Socket Binario el funcionamiento es similar pero cliente y servidor no precisan intercambiar paquetes XML específicamente sino que envían los datos como información binaria, lo que permite conectar con servidores de correo como POP3, SMTP y IMAP o servidores de nuevas como NNTP.

En lo tocante a la **seguridad** a la hora de integrar a Flex en una aplicación JEE será la arquitectura de esta la que imponga el modelo de seguridad, variando desde un *framework* de autenticación/autorización específico a un directorio LDAP, o archivos de configuración XML. La información de seguridad debe añadirse a los servicios BlazeDS y LiveCycle Data de manera que soliciten credenciales en las comunicaciones con el servidor.

Como en otras tecnologías, en Flex está disponible una serie de **frameworks** multipropósito. Muchos de ellos también son válidos para AIR. Los más empleados son:

- a) **Cairngorm**. Microarquitectura que aplica un pequeño conjunto de patrones de diseño (Service Locator, Front-Controller...) probados en conjunto. Se centra en tres áreas clave: manejar acciones de usuario, encapsular las interacciones con servidor y la lógica de negocio y gestionar el estado del cliente representándolo en la Interfaz de usuario.
- b) **Mate**. *Framework* orientado a eventos basado en etiquetas, implementadas completamente en MXML. Implementa la idea de inyección de dependencia, construyendo los objetos para a continuación inyectar en las clases los datos. Los objetos no solicitan la información pero esta le es entregada por el sistema.
- c) **PureMVCFramework**. Como Cairngorm representa también una microarquitectura con un pequeño conjunto de patrones de diseño,

con MVC y Fachada como núcleos centrales, cada una a través de un patron instancia única.

- d) **Swiz.** *Framework* de control de inversión (Ioc), que provee metodologías para simplificar el manejo de eventos y las llamadas asíncronas a procedimientos remotos. Emplea MVC pero a diferencia de los anteriores sólo en lo que respecta a la estructura de clases y no de directorios.
- e) **Parsley.** Conjunto de librerías ActionScript para correspondencia de objetos y entidades, registro de depuración, inyección de dependencia, mensajería y otras funcionalidades extendidas.

Para la integración con sistemas de información Flex provee del servicio de gestión de datos dentro del servicio LiveCycle Data. En este servicio se incluye sincronización de datos en tiempo real entre cliente, servidor y otros clientes, replicación de datos, paginación bajo demanda, y para aplicaciones AIR sincronización de datos locales para conexiones ocasionales de las aplicaciones.

### **30.4.3 SILVERLIGHT/MOONLIGHT**

Complemento para navegadores que permite integrar en la misma extensión elementos multimedia, animaciones e interactividad, de manera similar al WPF. Se encuentra basado en XAML para la definición de las interfaces de usuario, a partir de las que invoca a métodos del servidor de aplicaciones .NET. Permite la carga dinámica de XML con el que se puede operar a través de DOM a la manera de AJAX. Proporciona extensiones Javascript e. Las principales **característicos** del *framework* son:



- a) **WPF**. Incluye un subconjunto de WPF que extiende en gran medida elementos de la Interfaz de Usuario.
- b) **XAML**. Definición de la Interfaz de Usuario a través de un lenguaje de marcado declarativo.
- c) **Integración** con Javascript y ASP .NET AJAX.
- d) Acceso a **objetos del lado del servidor** .NET.
- e) Conexión con **servicios de red** WCF, SOAP y ASP .NET AJAX, permitiendo orígenes de datos JSON, XML y RSS.
- f) Soporta **LINQ** para implementar el acceso a datos

La **arquitectura** de Silverlight se compone de 3 partes fundamentales:

- 1) **Framework de presentación básico**. Componentes y servicios orientados a la Interfaz de usuario y la interacción con el usuario, elementos multimedia y soporte XAML.
- 2) **.NET Framework para Silverlight**. Subconjunto de .NET Framework para Silverlight que contiene componentes y librerías, recolector de basura, WCF y CLR. Asimismo incluye los controles de la Interfaz de Usuario, XLINQ, RSS/Atom, serialización XML y DLR (en inglés *Dynamic Language Runtime*)
- 3) **Componente de instalación y actualización**. Control de instalación y actualización de la extensión.

Mención especial merece el apartado de la **seguridad**, como ocurría con otras tecnologías RIA, Silverlight incorpora políticas de seguridad específicas para:

- a) Seguimiento del ciclo de vida de seguridad de Microsoft SDL (En inglés *Security Development Lifecycle*)
- b) Evitación de ataques XSS.
- c) Aislamiento de código de archivos de configuración XAP.



- d) Proveer acceso seguro a recursos de red.
- e) Servicios criptográficos para protección de datos de usuario.
- f) Firma digital de las aplicaciones.

**Figura 4: Arquitectura Silverlight.**

#### 30.4.4 JAVA FX

Librerías AJAX

Java web, JEE

Controles  
Diseño  
Enlace de  
datos  
Edición

FX es una plataforma que se compone de elementos multimedia y scripting junto con tecnologías de servidor para el desarrollo de aplicaciones multiplataforma. Puede funcionar de manera independiente del navegador siempre que tenga en el equipo instalada una máquina virtual Java compatible. Promueve el concepto de "Perfil común" para intentar unificar todos los dispositivos que soporten JavaFX, de manera que el mismo modelo de desarrollo se adapte a cualquier entorno.

Motor de ejecución CLR

Los principales **componentes** de Java FX son:

- a) **JavaFX Script.** Lenguaje de programación declarativo, con tipos estáticos, que permite invocar cualquier API de Java de la plataforma. Métodos de la plataforma.
- b) **Entorno de ejecución JavaFX.** Especializado para el dispositivo, Escritorio/Web, Mobile, o TV.
- c) **Aplicaciones JavaFX.** Independientes o empaquetadas como archivos JAR.
- d) **Herramientas de desarrollo.** Incluye el compilador para JavaFX Script, Plug-ins para IDEs como Eclipse, y librerías especializadas para gráficos, multimedia o servicios web, entre otros.

Ratón

Vector  
Imágenes  
Animaciones  
Texto

La **arquitectura** de JavaFX presenta en un primer nivel:

1. Las **APIs públicas** de JavaFX. Las principales funcionalidades de estas APIs son permitir la integración con otros lenguajes como JRuby, Groovy y Javascript, funcionalidades genéricas y extensiones para las interfaces de usuario.
2. **Grafo de Escena**. En este grafo, definido en la API *javafx.scene* se representa en una estructura en árbol con nodos representando todos los elementos visuales de la interfaz de usuario. Cada nodo puede llevar sus estilos, además de efectos, manejadores de eventos y control de estado.

Bajo ellos se encuentra el motor de ejecución compuesto por los siguientes componentes:

**Prism**. Motor gráfico de alto rendimiento que soporta Java2D, OpenGL y DirectX.

**Quantum Toolkit**. Gestiona las reglas de procesos para representación gráfica frente al manejo de eventos.

**Glass Windowing Toolkit**. Sistema de control de ventanas en el nivel más bajo de la arquitectura gráfica de JavaFX. Provee de los servicios operativos nativos del sistema además de ser responsable de gestionar la cola de eventos.

**Motores web y multimedia**. Incluyen APIs para soporte de medios visuales y de sonido. Asimismo el motor web soporta los estándares HTML5, CSS, Javascript, DOM y SVG.

***Figura 5: Arquitectura JavaFX.***



Si bien las tecnologías con anterioridad representan las soluciones de uso más extendido actualmente, existen otras multipropósito o más específicas que buscan hacerse un hueco en el mundo de las RIA. Dentro de estas otras tecnologías, OpenLaszlo merece mención especial, si bien no representa una tecnología en sí sino un conjunto de tecnologías, con algunas adaptaciones específicas.

### **30.5.1 OPENLASZLO**

OpenLaszlo es un *framework* y plataforma de desarrollo para aplicaciones RIA con licencia GPL, que precisa de un servidor propio para el alojamiento de las aplicaciones desarrolladas. Una misma aplicación definida a través de un lenguaje de definición propio puede exportarse a diferentes formatos multinavegador y multiplataforma.

Las aplicaciones OpenLaszlo pueden desplegarse en el servidor de aplicaciones de la plataforma, denominado **Despliegue en modo proxy**, o bien en modo Despliegue "SOLO" con independencia del servidor, por norma general empaquetada en un archivo SWF. Otra característica peculiar de funcionamiento son las **Librerías dinámicas**, que permiten una descarga "parcial" de la aplicación, para obtener una carga inicial más rápida, siendo el resto de la carga de la aplicación bajo demanda. Asimismo la funcionalidad **Krank** permite cargar las aplicaciones OpenLaszlo más rápidamente, realizando un preprocesamiento de la vista y *scripts* de inicialización.

OpenLaszlo aporta un lenguaje declarativo propio, el LZX diseñado para describir las interfaces del usuario, al estilo de XUL y XForms. Este lenguaje incorpora un *framework* de etiquetas dividido en categorías como: elementos de la interfaz, orígenes de datos, efectos multimedia y acciones. Se emplea conjuntamente con Javascript/AJAX siendo este último el encargado de la interacción con el usuario.

La plataforma OpenLaszlo incorpora a los siguientes **componentes**:

- a) **Compilador**. Permite que una aplicación definida ante el lenguaje declarativo LZX pueda transformarse a Flash (SWF) o DHTML-AJAX. Asimismo presenta una serie de módulos específicos para:
  - 1) **Compilación XML IU**. El compilador transforma las definiciones de la interfaz de usuario al formato de salida especificado para la aplicación.
  - 2) **Compilación ECMAScript**. Clases e instancias LZX se traducen a ECMAScript y controladores de eventos, transformándolas a Bytecode.
  - 3) **Compilación multimedia**. Se codifican los archivos en formatos de imagen y sonido, así como las fuentes TrueType en ficheros OBJ para SWF o XML.
- b) **Servidor**. Hace las funciones de alojamiento de la aplicación y proxy, manteniendo comunicación bidireccional con los back-ends a través de JAVARPC u otros protocolos de servicios web. Asimismo proporciona servicios de transformación de formatos, mensajería, *streaming*, encriptación SSL y autenticación.
- c) **Entorno de ejecución o LCF** (en inglés *Laszlo Foundation Class*). Incluye componentes de la interfaz de usuario, acceso a datos y servicios de red. La LCF se divide en cuatro componentes principales:
  - 1) **Data Loader/Binder**. El cargador y encargado de asociar y relacionar los datos. Dirige el tráfico de datos incluyendo el flujo de datos hacia el cliente.
  - 2) **Sistema de eventos**. Permite una programación basada en eventos al recoger los eventos detectados en el cliente.



- 3) **Layout & Animation System.** Sistema de animación y escenario de la aplicación que ofrece todos los elementos para la parte gráfica de la aplicación así como animaciones y efectos.
- 4) **Servicios para las aplicaciones.** Con funcionalidades extra como contadores, sonidos, etc ...
- d) **Framework.** Provee de una extensa API para animaciones, estructura de aplicación, acceso a datos, comunicaciones con el servidor y definición de la interfaz de usuario. Estructuralmente sigue un modelo MVC, pero ampliable a las capas que sean precisas para cada solución.
- e) **Servlet.** Se trata de un componente opcional para la aplicación que permite atender y dirigir peticiones multimedia o para servicios web como SOAP, JavaRPC o XML-RPC. Hace funciones de caché y proxy para priorización y bloqueo de peticiones así como de registro de polillas y auditoría.

### 30.5.2 ZK FRAMEWORK

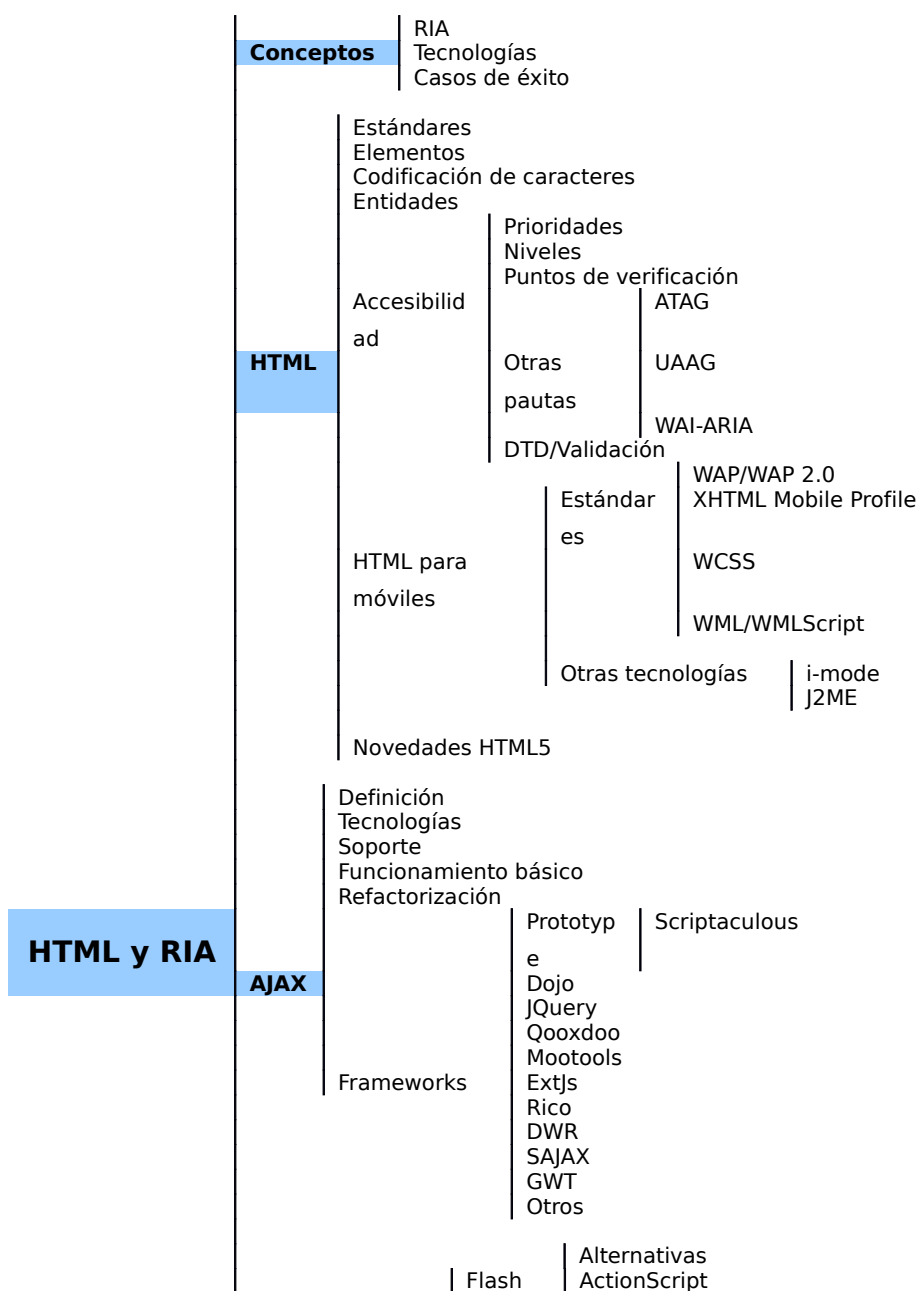
Se trata de otro *framework* orientado a eventos que en esencia podría incluirse con los *frameworks* AJAX aunque con algunas diferencias. En primer lugar emplea un lenguaje de marcado propio para las interfaces de usuario denominado ZUML que puede mezclarse con otros lenguajes de marcado como XUL y XHTML, además de lenguajes de *script* y expresiones EL para manipulación de componentes y datos. Se diseñó para integración con aplicaciones JEE incorporando las capacidades de AJAX en desarrollos ágiles y reutilizables.

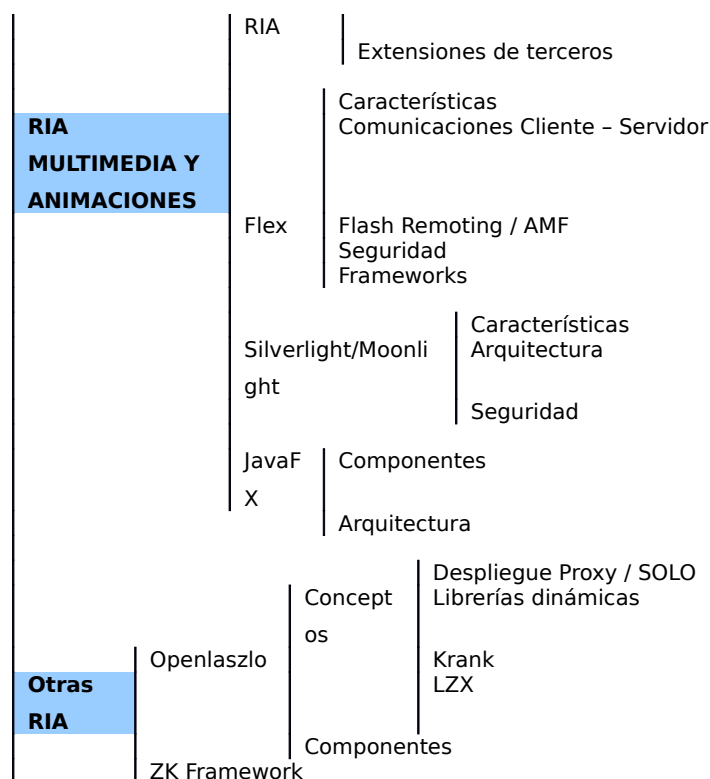
Provee de un *framework* con una implementación de ZK Spring adaptado del *framework* Spring, y librerías con componentes y etiquetas JSP con soporte para AJAX. En lo tocante a la seguridad añade protección para



ataques XSS, DoS y CSRF, además de reforzar la autenticación y los permisos con la incorporación de *frameworks* de terceros como Spring Security.

### 30.6 ESQUEMA





## 31.6 REFERENCIAS

Lee Babin

Beginning Ajax with PHP: from novice to professional. (2007).

Rebecca Riordan

Head First Ajax. (2008).

Michael Mahemoff.

Ajax Design Patterns. Creating Web 2.0 Sites with Programming and Usability Patterns. (2006)

**Autor: Juan Marcos Filgueira Gomis**

**Asesor Técnico Consellería de Educación y O. U.**

**Colegiado del CPEIG**



**XUNTA  
DE GALICIA**



*ESCOLA GALEGA  
DE ADMINISTRACIÓN  
PÚBLICA*

# **INGENIERÍA DEL SOFTWARE**

**31. INGENIERÍA DEL  
SOFTWARE. PROCESO  
SOFTWARE, MODELOS DE  
PROCESO SOFTWARE.  
PROCESO UNIFICADO. CICLOS  
DE VIDA. MODELOS DE CICLO  
DE VIDA. FASES DEL CICLO DE  
VIDA. MODELOS DE  
DESARROLLO. MODELOS  
ÁGILES. METODOLOGÍAS DE  
DESARROLLO DE SOFTWARE.  
MÉTRICA VERSIÓN 3.**

## **Tema 31. Ingeniería del software. Proceso software, modelos de proceso software. Ciclo de vida. Modelos de ciclo de vida. Fases del ciclo de vida. Modelos de desarrollo. Modelos ágiles. Metodologías de desarrollo de software. Métrica versión 3**

### **INDICE**

- 31.1 Ingeniería del software
  - 31.1.1 La crisis del software
  - 31.1.2 La Ingeniería del software
- 31.2 Proceso Software. Modelos de Proceso Software
- 31.3 Ciclo de vida
- 31.4 Modelos de ciclo de vida
  - 31.4.1 Modelo Codificar y Corregir
  - 31.4.2 Modelo por Etapas y Modelo en Cascada.
  - 31.4.3 Modelos basados en Prototipos
    - 31.4.3.1 Prototipo Rápido
    - 31.4.3.2 Prototipo Evolutivo
    - 31.4.3.3 Modelo Incremental
  - 31.4.4 Modelo en Espiral
  - 31.4.5 Modelos basados en Transformaciones
  - 31.4.6 Desarrollo Basado en Componentes
  - 31.4.7 Proceso Unificado de Desarrollo de software (PUDS)
  - 31.4.8 Modelo de métodos formales
  - 31.4.9 Programación Extrema (eXtreme Programming)
- 31.5 Metodologías Ágiles
  - 31.5.1 SCRUM
  - 31.5.2 DSDM
  - 31.5.3 Extreme Programing (XP)
  - 31.5.4 Agile Modeling (AM)
  - 31.5.5 FCC
  - 31.5.6 Familia Crystal



31.6 Metodologías de desarrollo de sistemas de información.

31.7 Métrica Versión 3

### **31.1 Ingeniería del software**

#### *31.1.1 La crisis del software.*

La rápida expansión de la Informática, sobre todo a partir de la segunda generación de ordenadores en los años 60, llevó a la escritura de millones de líneas de código antes de que se empezaran a plantear de manera seria metodologías para el diseño y la construcción de sistemas software y métodos para resolver los problemas de mantenimiento, fiabilidad, etc. Esta expansión sin control tuvo como consecuencia la denominada **crisis del software**, que es el nombre genérico que se ha acuñado para referirse a un conjunto de problemas que se han ido encontrando en el desarrollo del software. Esta problemática no sólo se limita al software que no funciona adecuadamente, sino que abarca otros aspectos como la forma de desarrollar el software, el mantenimiento de un volumen creciente de software existente y la forma de satisfacer la demanda creciente de software.

Los síntomas que hacen palpable la aparición de la crisis del software son, entre otros, los siguientes:

- **Expectativas:** los sistemas no responden a las expectativas que de ellos tienen los usuarios.
- **Fiabilidad:** los programas fallan demasiado a menudo.
- **Costo:** los costos del software son muy difíciles de prever y, frecuentemente, son muy superiores a lo esperado.
- **Plazos:** el software se suele entregar tarde y con menos prestaciones de las ofertadas.
- **Portabilidad:** es difícil cambiar un programa de su entorno hardware, aun cuando las tareas a realizar son las mismas.

- **Mantenimiento:** la modificación del software es una tarea costosa, compleja y propensa a errores.
- **Eficiencia:** los esfuerzos que se hacen para el desarrollo del software no hacen un aprovechamiento óptimo de los recursos disponibles (personas, tiempo, dinero, herramientas, etc.).

La solución a la crisis del software se centra, pues, en abordar y resolver los siguientes problemas principales:

- La planificación del proyecto software y la estimación de los costes de desarrollo, que son muy imprecisos.
- La productividad de las personas, que no se corresponde con la demanda de sus servicios.
- La calidad del producto software, que es, en muchos casos, inadecuada.

### *31.1.2 La Ingeniería del Software*

La Ingeniería del Software se puede definir como *el establecimiento y uso de principios de ingeniería orientados a obtener, de manera económica, software que sea fiable y funcione eficientemente sobre máquinas reales.*

La Ingeniería del Software abarca tres elementos clave: métodos, herramientas y procedimientos. Los **métodos** proporcionan la manera de construir técnicamente el software. Abarcan las tareas de planificación y estimación de proyectos, análisis de los requerimientos del sistema y del software, diseño de las estructuras de datos, de la arquitectura de programas y de los procedimientos algorítmicos, y la codificación, pruebas y mantenimiento. Las **herramientas** suministran el soporte automático o semiautomático para los métodos; esto es, dan soporte al desarrollo del software. Los **procedimientos** definen la secuencia en la que se aplican los métodos, los controles que ayudan a asegurar la calidad y a coordinar los cambios y las guías que facilitan a los gestores del software.

El trabajo que se asocia a la ingeniería del software se puede dividir según Pressman en **tres fases genéricas**, con independencia del área de aplicación, tamaño o complejidad del proyecto:

- **La fase de definición** se centra sobre el *qué*. Es decir, durante la definición, el que desarrolla el software intenta identificar qué información ha de ser procesada, qué función y rendimiento se desea, qué comportamiento del sistema, qué interfaces van a ser establecidas, qué restricciones de diseño existen, y qué criterios de validación se necesitan para definir un sistema correcto. Por tanto, han de identificarse los requisitos clave del sistema y del software. Aunque los métodos aplicados durante la fase de definición variarán dependiendo del paradigma de ingeniería del software (o combinación de paradigmas) que se aplique, de alguna manera tendrán lugar tres tareas principales: ingeniería de sistemas o de información, planificación del proyecto del software y análisis de los requisitos.
- **La fase de desarrollo** se centra en el *cómo*. Es decir, durante el desarrollo un ingeniero del software intenta definir cómo han de diseñarse las estructuras de datos, cómo ha de implementarse la función dentro de una arquitectura de software, cómo han de implementarse los detalles procedimentales, cómo han de caracterizarse interfaces, cómo ha de traducirse el diseño en un lenguaje de programación (o lenguaje no procedimental) y cómo ha de realizarse la prueba. Los métodos aplicados durante la fase de desarrollo variarán, aunque siempre tendremos: diseño del software, generación de código y prueba del software.
- **La fase de mantenimiento** se centra en el *cambio* que va asociado a la corrección de errores, a las adaptaciones requeridas a medida que evoluciona el entorno del software y a cambios debidos a las mejoras producidas por los requisitos cambiantes del cliente. Durante la fase de mantenimiento se encuentran cuatro tipos de cambios:

- o **Corrección.** Incluso llevando a cabo las mejores actividades de garantía de calidad, es muy probable que el cliente descubra los defectos en el software. El *mantenimiento correctivo* cambia el software para corregir los defectos.
- o **Adaptación.** Con el paso del tiempo, es probable que cambie el entorno original para el que se desarrolló el software. El *mantenimiento adaptativo* produce modificaciones en el software para acomodarlo a los cambios de su entorno externo (hardware, sistema operativo, reglas de negocio,...).
- o **Mejora.** Conforme se utilice el software, el cliente/usuario puede descubrir funciones adicionales que van a producir beneficios. El *mantenimiento perfectivo* lleva al software más allá de sus requisitos funcionales originales.
- o **Prevención.** El software de computadora se deteriora debido al cambio. En esencia, el *mantenimiento preventivo* hace cambios en programas de computadora a fin de que se puedan corregir, adaptar y mejorar más fácilmente.

Estas fases se complementan con un número de **actividades protectoras** se aplican a lo largo de todo el proceso del software. Entre las actividades típicas de esta categoría se incluyen: seguimiento y control del proyecto de software, revisiones técnicas formales, garantía de calidad del software, gestión de configuración del software, preparación y producción de documentos, gestión de reutilización, mediciones y gestión de riesgos.

### **31.2 Proceso software. Modelos de proceso software**

Según Sommerville, *un proceso del software es un conjunto de actividades que conducen a la creación de un producto software*. Estas actividades pueden consistir en el desarrollo de software desde cero en un lenguaje de programación estándar como Java o C. Sin embargo, cada vez más, se desarrolla nuevo software ampliando y modificando los sistemas existentes

y configurando e integrando software comercial o componentes del sistema.

Para Fugetta, un proceso software es *un conjunto coherente de políticas, estructuras organizacionales, tecnologías, procedimientos y artefactos que son necesarios para concebir, desarrollar, instalar y mantener un producto software*.

Los procesos del software son complejos y, como todos los procesos intelectuales y creativos, dependen de las personas que toman decisiones y juicios. Debido a la necesidad de juzgar y crear, los intentos para automatizar estos procesos han tenido un éxito limitado. Las herramientas de ingeniería del software asistida por computadora (CASE) pueden ayudar a algunas actividades del proceso, pero tienen limitaciones. Una razón por la cual la eficacia de las herramientas CASE está limitada se halla en la inmensa diversidad de procesos del software. No existe un proceso ideal, y muchas organizaciones han desarrollado su propio enfoque para el desarrollo del software. Los procesos han evolucionado para explotar las capacidades de las personas de una organización, así como las características específicas de los sistemas que se están desarrollando. Para algunos sistemas, como los sistemas críticos, se requiere un proceso de desarrollo muy estructurado. Para sistemas de negocio, con requerimientos rápidamente cambiantes, un proceso flexible y ágil probablemente sea más efectivo.

Aunque existen muchos procesos diferentes de software, algunas actividades fundamentales son comunes para todos ellos:

1. **Especificación del software** donde los clientes e ingenieros definen el software a producir y las restricciones sobre su operación.
2. **Desarrollo del software** donde el software se diseña y programa.
3. **Validación del software** donde el software se valida para asegurar que es lo que el cliente requiere.

4. **Evolución del software** donde el software debe evolucionar para cubrir las necesidades cambiantes del cliente.

Diferentes tipos de sistemas necesitan diferentes procesos de desarrollo. Por lo tanto, estas actividades genéricas pueden organizarse de diferentes formas y describirse en diferentes niveles de detalle para diferentes tipos de software. El uso de un proceso inadecuado del software puede reducir la calidad o la utilidad del producto de software que se va a desarrollar y/o incrementar los costes de desarrollo.

Los procesos del software se pueden mejorar con la estandarización. Esto conduce a mejorar la comunicación y a una reducción del tiempo de formación, y hace la ayuda al proceso automatizado más económica. La estandarización también es un primer paso importante para introducir nuevos métodos, técnicas y buenas prácticas de ingeniería del software.

De todos modos, la existencia de un proceso de software no es garantía de que éste será entregado a tiempo, de que satisfará las necesidades del cliente, o de que mostrará las características técnicas que conducirán a características de calidad a largo plazo. El proceso de software debe **evaluarse** para asegurarse de que cumpla una serie de criterios básicos que han demostrado ser esenciales para una ingeniería de software exitosa.

**CMMI** (Modelo de Capacidad de Madurez Integrado) es un modelo total del proceso, que describe las metas, prácticas y capacidades específicas con que debe contar un proceso de software maduro. El estándar **SPICE (ISO/IEC15504)** define un conjunto de requisitos para la evaluación del proceso de software. Lo que pretende es ayudar a las organizaciones en el desarrollo de una evaluación objetiva de la eficacia de cualquier proceso de software definido.

El **ISO 9001:2000 para software** es un estándar genérico que se aplica a cualquier organización que desee mejorar la calidad general de los productos, sistemas o servicios que provee. El ISO 9001:2000 subraya la

importancia que tiene para una organización identificar, implementar, gestionar y mejorar de manera continua la efectividad de los procesos necesarios para el sistema de administración de la calidad, y gestionar las interacciones de estos procesos para conseguir los objetivos de la organización.

El ISO 9001:2000 ha adoptado un ciclo de "planear-hacer-revisar-actuar" (PDCA Plan-Do-Check-Act en inglés, también conocido como Círculo de Deming) que se aplica a los elementos de gestión de calidad de un proyecto de software. Dentro de un contexto de software, "planear" establece los objetivos, las actividades y tareas del proceso necesarios para conseguir un software de alta calidad y una satisfacción del cliente; "hacer" implementa el proceso de software (incluidas las actividades del marco de trabajo y las actividades sombrija); "revisar" monitorea y mide el proceso para asegurarse de que todos los requisitos establecidos para la gestión de calidad hayan sido cumplidos; "actuar" inicia las actividades de mejoramiento del proceso de software, el cual tiene una continuidad de trabajo para mejorar el proceso.

Un **modelo de procesos del software** es una descripción abstracta y simplificada de un proceso del software que presenta una visión de ese proceso. Cada modelo de proceso representa un proceso desde una perspectiva particular y así proporciona sólo información parcial sobre ese proceso. Son abstracciones de los procesos que se pueden utilizar para explicar diferentes enfoques para el desarrollo de software. Puede pensarse en ellos como marcos de trabajo del proceso que pueden ser extendidos y adaptados para crear procesos más específicos de ingeniería del software. Cada modelo describe una sucesión de fases y un encadenamiento entre ellas. Según las fases y el modo en que se produzca este encadenamiento, tenemos diferentes modelos de proceso. Un modelo es más adecuado que otro para desarrollar un proyecto dependiendo de un conjunto de características del proyecto. Los modelos pueden incluir actividades que

son parte de los procesos y productos de software y el papel de las personas involucradas en la ingeniería del software. Alternativamente, a veces se usan los términos **ciclo de vida**, **modelo de ciclo de vida** y **modelo de desarrollo**.

La mayor parte de los modelos de procesos del software se basan en uno de los tres modelos generales o paradigmas de desarrollo de software:

1. **El enfoque en cascada.** Considera las actividades anteriores y las representa como fases de procesos separados, tales como la especificación de requerimientos, el diseño del software, la implementación, las pruebas, etcétera. Después de que cada etapa queda definida «se firma» y el desarrollo continúa con la siguiente etapa.
2. **Desarrollo iterativo.** Este enfoque entrelaza las actividades de especificación, desarrollo y validación. Un sistema inicial se desarrolla rápidamente a partir de especificaciones muy abstractas. Éste se refina basándose en las peticiones del cliente para producir un sistema que satisfaga las necesidades de dicho cliente. El sistema puede entonces ser entregado. De forma alternativa, se puede reimplementar utilizando un enfoque más estructurado para producir un sistema más sólido y mantenible.
3. **Ingeniería del software basada en componentes (CBSE).** Esta técnica supone que las partes del sistema ya existen previamente. El proceso de desarrollo del sistema se enfoca en la integración de estas partes más que desarrollarlas desde el principio.

Estos tres modelos de procesos genéricos se utilizan ampliamente en la práctica actual de la ingeniería del software. No se excluyen mutuamente y a menudo se utilizan juntos, especialmente para el desarrollo de sistemas grandes. Los subsistemas dentro de un sistema más grande pueden ser desarrollados utilizando enfoques diferentes. Por lo tanto, aunque es



conveniente estudiar estos modelos separadamente, debe entenderse que, en la práctica, a menudo se combinan.

Pressman separa entre el **modelo personal** (modelo utilizado por cada desarrollador) y **modelo en equipo** (cuando el proyecto es dirigido por varios profesionales) para el proceso de software. Ambos destacan la medición, la planeación y la autodirección como ingredientes clave para un proceso de software exitoso.

### **31.3 Ciclo de vida del software**

Según el estándar ISO-12207 **el ciclo de vida** de un sistema de información es *el marco de referencia que contiene los procesos, las actividades y las tareas involucradas en el desarrollo, la explotación y el mantenimiento de un producto de software, abarcando la vida del sistema desde la definición de los requisitos hasta la finalización de su uso*. También se podría definir el ciclo de vida de un sistema de información como *el conjunto de etapas por las que atraviesa el sistema desde su concepción, hasta su retirada de servicio pasando por su desarrollo y explotación*.

No existe un único **modelo de ciclo de vida** que defina los estados por los que pasa cualquier producto software. Dado que existe una gran variedad de aplicaciones y que dicha variedad supone situaciones totalmente distintas, es natural que existan diferentes modelos de ciclo de vida. No obstante, todo modelo de ciclo de vida debe cubrir los siguientes objetivos básicos:

- Definir las actividades a realizar y en qué orden, es decir, determinar el orden de las fases del proceso software.
- Establecer los criterios de transición para pasar de una fase a la siguiente.

- Proporcionar puntos de control para la gestión del proyecto, es decir, calendario y organización.
- Asegurar la consistencia con el resto de los sistemas de información de la organización.

Cada proyecto debe seleccionar el modelo de ciclo de vida que sea más apropiado para su caso, el cual se elige en base a considerar una serie de factores como: la cultura de la organización, el deseo de asumir riesgos, el área de aplicación, la volatilidad de los requisitos, la comprensión de dichos requisitos, etc. En cualquier proyecto software, el modelo de ciclo de vida permite responder a las cuestiones de ¿qué se hará a continuación? y ¿por cuánto tiempo se hará? Dado que cada modelo de ciclo de vida tiene sus ventajas y sus inconvenientes, no se suelen seguir en la práctica los modelos en su forma pura, sino que de acuerdo con las peculiaridades del sistema y la experiencia del personal, se pueden adoptar aspectos de otros modelos que sean más adecuados al caso concreto.

Es importante no confundir el concepto de ciclo de vida con el de metodología. Mientras que el ciclo de vida indica qué actividades hay que realizar y en qué orden, la **metodología** indica cómo avanzar en la construcción del sistema, esto es, con qué técnicas, y entre sus características está la de determinar los recursos a utilizar o las personas implicadas en cada actividad.

### **31.4 Modelos de ciclo de vida del software**

Una posible clasificación sería la que divide los modelos de ciclo de vida en:

- **Modelos tradicionales:** Son los de más amplia utilización. Dentro de este grupo estarían:
  - o Modelo en cascada.
  - o Modelos basados en prototipos:
    - Modelo de construcción de prototipos.

- Modelo de desarrollo incremental.
- Modelo de prototipado evolutivo.
- **Modelos alternativos**
  - o Modelo en espiral
  - o Modelos basados en transformaciones: La filosofía general es llegar a generar código a partir de unas especificaciones transformándolas por medio de herramientas. Según usemos unas u otras herramientas tendremos:
    - Las que usan técnicas de cuarta generación (Roger Pressman): lenguajes no procedimentales para consultas a BD; generadores de código, de pantallas, de informes; herramientas de manipulación de datos; facilidades gráficas de alto nivel.
    - Basados en modelos de transformación (Carma McClure)  
=> Basados en herramientas CASE que permiten, siguiendo el MCV clásico, pasar de una etapa a otra aplicando las transformaciones que dan las herramientas.
  - o Desarrollo de Software Basado en Componentes (DSBC o CBSB ).

Aparte de estos modelos de ciclo de vida en la actualidad existen nuevas alternativas:

- Proceso unificado de desarrollo del software. (PUDS)
- Programación Extrema.

#### *31.4.1 Modelo Codificar y Corregir (Code and Fix)*

Este es el modelo básico utilizado en los inicios del desarrollo de software. Contiene dos pasos:

- Escribir código.

- Corregir problemas en el código.

Se trata de primero implementar algo de código y luego pensar acerca de requisitos, diseño, validación, y mantenimiento. Este modelo tiene tres problemas principales:

- Después de un número de correcciones, el código puede tener una muy mala estructura, hace que los arreglos sean muy costosos. Esto hizo ver la necesidad de una fase previa de diseño antes de la de codificación.
- Frecuentemente, aún el software bien diseñado, no se ajusta a las necesidades del usuario, por lo que es rechazado o su reconstrucción es muy cara. Esto condujo a la necesidad de introducir una fase de análisis de requerimientos antes del diseño.
- El código es difícil de reparar por su pobre preparación para probar y modificar. Este problema hizo resaltar la necesidad de la planificación y preparación de las distintas tareas en cada fase.

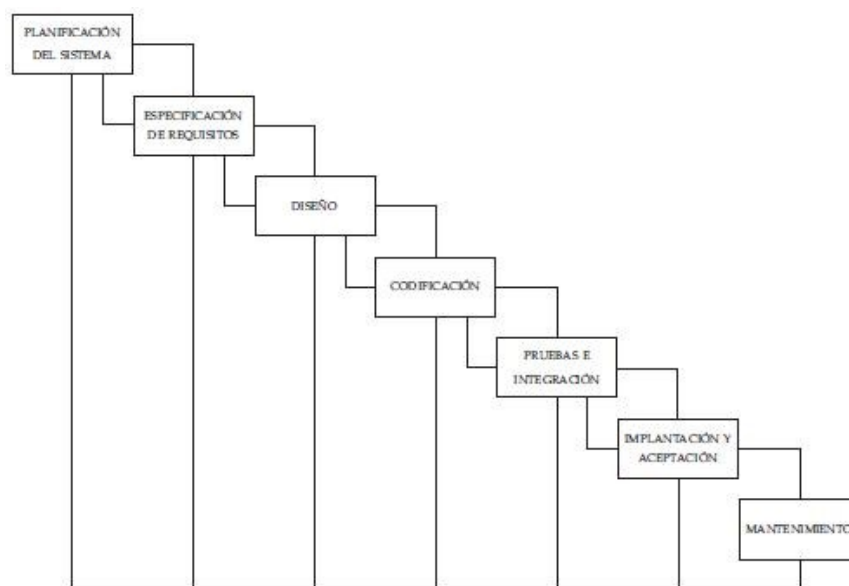
#### *31.4.2 Modelo por Etapas y Modelo en Cascada*

Los problemas apuntados del modelo Code and Fix llevaron a la necesidad de realizar el desarrollo del software siguiendo un modelo de etapas sucesivas, el **modelo por etapas** (Stage Wise) que considera las siguientes: Planificación, Especificaciones de operación, Especificaciones de codificación, Codificación, Prueba de cada unidad, Prueba de integración, Eliminación de problemas, y Evaluación del sistema.

El **modelo en cascada** introduce una serie de mejoras respecto al modelo por etapas tales como considerar la realización de bucles de realimentación entre etapas, permitiendo que se puedan resolver los problemas detectados en una etapa, en la etapa anterior y permitir la incorporación inicial del prototipado a fin de captar las especificaciones durante el análisis, o para probar distintas soluciones durante el diseño.

El modelo en cascada se compone de una serie de fases que se suceden secuencialmente, generándose en cada una de ellas unos resultados que serán necesarios para iniciar la fase siguiente. Es decir, la evolución del producto software se produce a través de una secuencia ordenada de transiciones de una fase a la siguiente, según un orden lineal. El número de fases en este modelo es irrelevante, ya que lo que le caracteriza es la *secuencialidad* de las mismas y la *necesidad de completar* cada una de ellas para pasar a la siguiente. El modelo del ciclo de vida en cascada está regido por la documentación, es decir, la decisión del paso de una fase a la siguiente se toma en función de si la documentación asociada a dicha fase está completa o no. Sin embargo, esta forma de proceder no es la más adecuada para algunos tipos de software como el que se usa en las aplicaciones interactivas y de usuario final.

Desde su presentación, el modelo en cascada ha tenido un papel fundamental en el desarrollo de proyectos software. Ha sido, y todavía sigue siendo, el más utilizado, tanto que este modelo se conoce con el nombre de ciclo de vida clásico, si bien incorporando infinidad de variaciones que eliminan el carácter simplista del mismo. Aun así, existen una serie de limitaciones que justifican la necesidad de definir otros modelos.



Como se ha indicado anteriormente, las fases que comprende el ciclo de vida clásico son irrelevantes, tanto en número, como en cuáles sean esas fases siempre que se produzcan secuencialmente. Posiblemente, el modelo clásico más utilizado sea el modelo de siete fases que son:

- **Planificación del sistema:** En esta fase es necesario fijar el ámbito del trabajo a realizar, los recursos necesarios, las tareas a realizar, las referencias a tener en cuenta, el coste estimado del proyecto, la composición del equipo de desarrollo y el orden de las actividades.
- **Especificación de requisitos:** En esta fase es preciso analizar, entender y documentar el problema que el usuario trata de resolver con el sistema y se han de especificar con detalle las funciones, objetivos y restricciones del mismo, a fin de que usuarios y desarrolladores puedan tomar éstas como punto de partida para acometer el resto del sistema. Es decir, en la fase de especificación de requisitos se trata de definir **qué** debe hacer el sistema, e identificar la información a procesar, las funciones a realizar, el rendimiento del sistema, las interfaces con otros sistemas y las ligaduras de diseño.
- **Diseño:** Arranca de las especificaciones de la fase anterior. En la fase de diseño, una vez elegida la mejor alternativa, se debe crear la

solución al problema descrito atendiendo a aspectos de interfaces de usuario, estructura del sistema y decisiones sobre la implantación posterior. La fase de diseño trata de definir el **cómo**.

- **Codificación:** Esta fase consiste en traducir las especificaciones y representaciones del Diseño a un lenguaje de programación capaz de ser interpretado y ejecutado por el ordenador.
- **Pruebas e integración:** Una vez que se tienen los programas en el formato adecuado al ordenador, hay que llevar a cabo las pruebas necesarias que aseguren la corrección de la lógica interna del programa y que éste cubre las funcionalidades previstas. La integración de las distintas partes que componen la aplicación o el sistema debe garantizar el buen funcionamiento del conjunto.
- **Implantación y aceptación del sistema:** El objetivo de esta fase es conseguir la aceptación del sistema por parte de los usuarios del mismo, y llevar a cabo las actividades necesarias para su puesta en producción.
- **Mantenimiento del sistema:** La fase de mantenimiento comienza una vez que el sistema ha sido entregado al usuario y continúa mientras permanece activa su vida útil. Puede deberse a errores no detectados previamente (correctivo), a modificaciones, mejoras o ampliaciones solicitadas por los usuarios (perfectivo, o aumentativo) o a adaptaciones requeridas por la evolución del entorno tecnológico o cambios normativos (mantenimiento adaptativo).

Las principales **críticas** al modelo se centran en sus características básicas, es decir secuencialidad y utilización de los resultados de una fase para acometer la siguiente de manera que el sistema sólo se puede validar cuando está terminado. En cuanto al flujo secuencial, los proyectos reales raramente siguen el flujo secuencias que propone el modelo. Siempre ocurren interacciones y en las últimas fases sobre todo se pueden realizar en paralelo algunas áreas como por ejemplo codificación y pruebas. Una

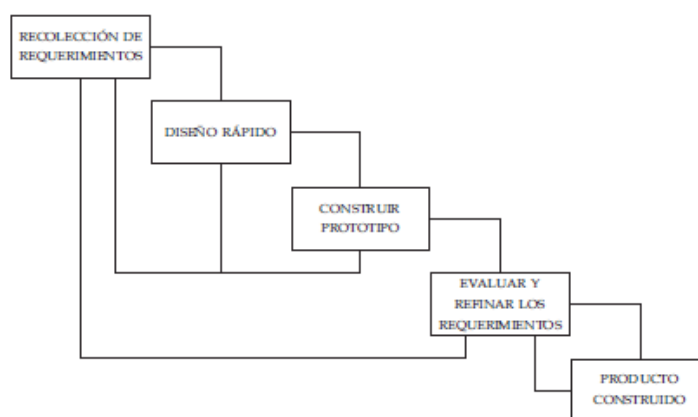
aplicación del modelo en sentido estricto obligaría a la “congelación” de los requisitos de los usuarios, supuesto este completamente alejado de la realidad. El modelo no contempla la posibilidad de realimentación entre fases. Por otro lado, el modelo no prevé revisiones o validaciones intermedias por parte del usuario, así los resultados de los trabajos sólo se ven al final de una serie de tareas y fases de tal forma que si se ha producido un error en las primeras fases este sólo se detectará al final y su corrección tendrá un costo muy elevado, puesto que será preciso rehacer todo el trabajo desde el principio.

#### *31.4.3 Modelos basados en prototipos*

Permiten a los desarrolladores construir rápidamente versiones tempranas de los sistemas software que pueden evaluar los usuarios. Existen varios modelos derivados del uso de prototipos

##### *31.4.3.1 Prototipado rápido*

Los prototipos deben poder construirse con facilidad para evaluarlos en una temprana fase del desarrollo y además han de ser baratos y desarrollados en poco tiempo. También se denominan de usar y tirar.





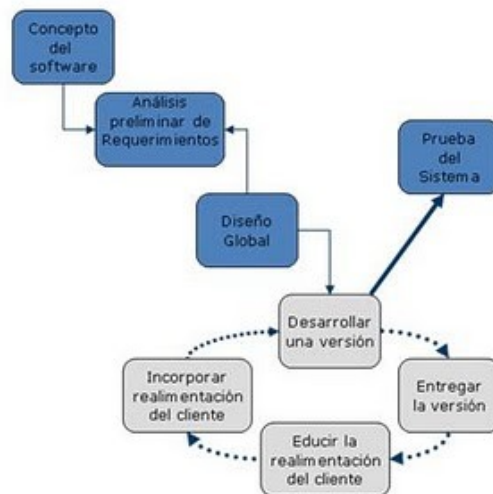
El prototipo sirve para crear y validar la especificación, y para que el usuario tenga una idea de cómo será el software antes de que comience el desarrollo. Es importante precisar que el prototipo se construye sólo para servir como mecanismo de definición de los requerimientos funcionales. Posteriormente ha de desecharse y debe construirse el sistema con los criterios normales de calidad y mantenimiento, siguiendo, por ejemplo, el ciclo de vida clásico, ya que generalmente el prototipo se ha construido tomando decisiones de implementación contrarias al buen criterio de desarrollo de software. Los objetivos del prototipo son:

- Reducir el riesgo de construir un producto que se aleje de las necesidades del usuario
- Reducir el coste de desarrollo al disminuir las correcciones en etapas avanzadas del mismo.
- Aumentar las posibilidades de éxito del producto.

El principal problema de este modelo es que el usuario ve en el prototipo lo que parece ser una versión de trabajo del software, sin saber que con la prisa de hacer que funcione no se ha tenido en cuenta la calidad del software global o la facilidad de mantenimiento a largo plazo. Cuando se informa de que el producto se debe construir otra vez para que se puedan mantener los niveles altos de calidad, el cliente no lo entiende y pide que se apliquen unos pequeños ajustes que puedan hacer del prototipo un producto final.

#### *31.4.3.2 Prototipado evolutivo*

En este tipo de ciclo de vida se construye una implementación parcial del sistema que satisface los requisitos conocidos, la cual es utilizada por el usuario para llegar a comprender mejor la totalidad de requisitos que desea.



Desde un punto de vista genérico, se puede decir que los modelos evolutivos se encaminan a conseguir un sistema flexible que se pueda expandir, de forma que se pueda realizar rápidamente un nuevo sistema cuando cambian los requisitos. Estos modelos consisten en implementar un producto software operativo y hacerle evolucionar de acuerdo con la propia experiencia operacional. Están especialmente indicados en situaciones en que se utilizan lenguajes de cuarta generación (L4G) y para aquellas otras en que el usuario no puede decir lo que requiere, pero lo reconocerá cuando lo vea. Los modelos evolutivos dan al usuario una rápida capacidad de operación inicial y una buena base para determinar mejoras del sistema. Está relacionado con el concepto de RAD (Rapid Application Development – Desarrollo Rápido de Aplicaciones), que identifica los asistentes, plantillas y entornos de fácil y rápida creación de software.

La *diferencia* fundamental entre el prototipado rápido y el evolutivo estriba en que mientras que en el primer caso se asume que existen una serie de requisitos reales, aunque para establecer lo que el usuario quiere realmente es necesario establecer una serie de iteraciones antes de que los requisitos se estabilicen al final, en el caso evolutivo se asume desde el principio que los requisitos cambian continuamente.

En el prototipo rápido lo lógico es implementar sólo aquellos aspectos del sistema que se entienden mal, mientras que en el prototipo evolutivo lo lógico es comenzar por los aspectos que mejor se comprenden y seguir construyendo apoyados en los puntos fuertes y no en los débiles. Como resultado de este modo de desarrollo, la solución software evoluciona acercándose cada vez más a las necesidades del usuario; ahora bien, pasado un tiempo el sistema software así construido deberá ser rehecho o sufrir una profunda reestructuración con el fin de seguir evolucionando.

El modelo de prototipado evolutivo (Evolutionary Development model) también tiene sus **dificultades**. Se le puede considerar como una nueva versión, utilizando lenguajes de programación de más alto nivel, del viejo modelo CODE-AND-FIX. Otro inconveniente que presenta es partir de la suposición, muchas veces no realista, de que el sistema operacional del usuario final será lo suficientemente flexible como para poder incorporar caminos de evolución futuros no planificados con anterioridad.

#### *31.4.3.3 Modelo de Desarrollo Incremental*

El modelo de desarrollo incremental consiste en desarrollar un sistema que satisfaga una parte de los requisitos especificados y posteriormente ir creando versiones que incorporen los requisitos que faltan, hasta llegar al sistema final. Actuando así, se pretende disponer pronto de un sistema que aunque sea incompleto, sea utilizable y satisfaga parte de los requisitos, evitando de paso el efecto big-bang, es decir, que durante un período largo de tiempo no se tenga nada y de repente haya una situación completamente nueva. Por otra parte, también se logra que el usuario se implique estrechamente en la planificación de los pasos siguientes.

El modelo de desarrollo incremental también se utiliza para evitar la demanda de funcionalidades excesivas al sistema por parte de los usuarios, ya que como a éstos les resulta difícil definir sus necesidades reales

tienden a pedir demasiado. Actuando con este modelo se atiende primero a las funcionalidades esenciales y las funcionalidades accesorias sólo se incluyen en las versiones sucesivas cuando realmente son necesarias.

La diferencia entre el modelo de desarrollo evolutivo y el modelo de desarrollo incremental radica en dos aspectos:

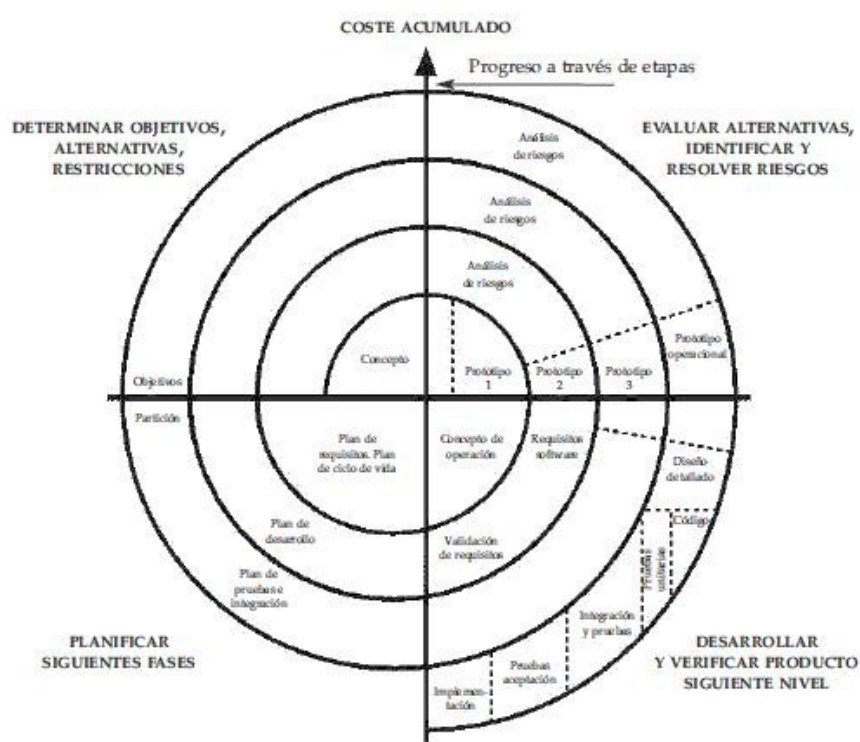
- El modelo incremental parte de la hipótesis de que se conocen todos los requisitos y éstos se van incorporando al sistema en versiones sucesivas. En el modelo evolutivo sólo se conocen unos pocos requisitos y los restantes se van descubriendo en sucesivas evoluciones del prototipo.
- Cada vez que se desarrolla una nueva versión, en el modelo evolutivo es una versión de todo el sistema, mientras que en el incremental es una versión anterior sin cambios, más un número de nuevas funciones.

#### *31.4.4 Modelo en Espiral*

El *modelo en espiral*, propuesto originalmente por Boehm, es un modelo de proceso de software evolutivo que conjuga la naturaleza iterativa de construcción de prototipos con los aspectos controlados y sistemáticos del modelo en cascada. Proporciona el potencial para el desarrollo rápido de versiones incrementales del software. En el modelo espiral, el software se desarrolla en una serie de versiones incrementales. Durante las primeras iteraciones, la versión incremental podría ser un modelo en papel o un prototipo. Durante las últimas iteraciones, se producen versiones cada vez más completas del sistema diseñado. Las principales diferencias entre el modelo en espiral y los modelos de ciclo de vida más tradicionales son:

- En el modelo en espiral hay un reconocimiento explícito de las diferentes **alternativas** para alcanzar los objetivos del proyecto.

- El modelo en espiral se centra en la identificación de los **riesgos** asociados a cada alternativa y en la manera de resolver dichos riesgos.
- En el modelo en espiral los proyectos se dividen en ciclos (ciclos de espiral), avanzándose en el desarrollo mediante consensos al final de cada ciclo.
- El modelo en espiral se adapta a cualquier tipo de actividad.



El modelo en espiral refleja la idea de que cada ciclo implica una progresión en el desarrollo del producto software que aplica la misma secuencia de pasos para cada parte del producto y para cada uno de sus niveles de elaboración, desde la concepción global hasta la codificación individual de cada programa. Esta secuencia de pasos, iterativa en cada fase del desarrollo, se compone de las cuatro actividades siguientes:

- **Planificación:** Este primer paso con el que comienza cada ciclo de espiral consiste en la identificación de los objetivos de la parte del producto que está siendo elaborada (funcionalidad, rendimiento,

adaptación a los cambios, etc.), identificación de las alternativas principales para realizar o implementar esta parte del producto, y la identificación de las restricciones impuestas (coste, plazo de realización, interfaces, etc.).

- **Análisis de riesgos:** Comienza con la evaluación de cada alternativa respecto a los objetivos y a las restricciones. Este proceso de evaluación identificará áreas de incertidumbre que son fuentes significativas de riesgo en el proyecto. Se decidirá como resolver los riesgos asociados a la alternativa elegida.
- **Ingeniería.** Este paso consiste en el desarrollo y verificación del producto objeto de la fase (ciclo de espiral) en que nos encontremos. Como esta implementación está dirigida por el riesgo, el desarrollo podrá seguir las pautas de un prototipado evolutivo, las del ciclo de vida clásico, las orientadas a transformaciones automáticas, o cualquier otro enfoque del desarrollo. En definitiva, esto permite al modelo en espiral acomodarse a cualquier mezcla de estrategias de desarrollo.
- **Evaluación del cliente.** Una característica importante del modelo en espiral es que cada ciclo de espiral se completa con una revisión en la que participan aquellos que tienen relación con el producto (desarrolladores, usuarios, etc.). Esta revisión incluye todos los productos desarrollados durante el ciclo, los planes para el siguiente ciclo y los recursos necesarios para llevarlos a cabo.

Según esto, el modelo se puede representar mediante unos ciclos externos de espiral, que representan las fases en que se ha dividido el desarrollo del proyecto software, normalmente las del modelo clásico, y unos ciclos internos, iterativos para cada fase, en los que se llevan a cabo las cuatro actividades antes citadas. La dimensión radial indica los costes de desarrollo acumulativos, mientras que la dimensión angular indica el progreso hecho en cumplimentar cada fase.

La principal ventaja del modelo en espiral es el amplio rango de opciones a que puede ajustarse y que éstas permiten utilizar los modelos de proceso de construcción de software tradicionales; por otra parte, su orientación al riesgo evita, si no elimina, muchas de las posibles dificultades. Otras **ventajas** son:

- Concentra su atención en opciones que permiten la reutilización de software ya existente.
- Se centra en la eliminación de errores y alternativas poco atractivas.
- No establece procedimientos diferentes para el desarrollo del software y el mantenimiento del mismo.
- Proporciona un marco estable para desarrollos integrados hardware-software.
- Permite preparar la evolución del ciclo de vida del producto software, así como el crecimiento y cambios de éste.
- Permite incorporar objetivos de calidad en el desarrollo de productos software.
- Se adapta muy bien al diseño y programación orientada a objetos. Posiblemente con este método es cuando obtiene mejores resultados.

En cuanto a los **inconvenientes** que plantea la utilización del modelo en espiral, cabe citar:

- Dificultad para adaptar su aplicabilidad al software contratado, debido a la poca flexibilidad y libertad de éste.
- Dependencia excesiva de la experiencia que se tenga en la identificación y evaluación de riesgos.
- Necesidad de una elaboración adicional de los pasos del modelo, lo que depende también, en gran medida, de la experiencia del personal.

#### *31.4.5 Modelos basados en Transformaciones*

Estos modelos, también llamados *ciclo de vida con producción automática de diseño y código*, se han propuesto como solución al problema que plantean los modelos de desarrollo evolutivo de producir software mal estructurado. Se basan en la posibilidad de convertir automáticamente una especificación formal de un producto software en un programa que satisfaga las especificaciones, utilizando herramientas de cuarta generación. Para ello, los pasos típicos que siguen estos modelos son:

1. Especificación formal del producto tal como lo permita la comprensión inicial del problema.
2. Transformación automática de la especificación en código.
3. Realizar bucles iterativos para mejorar el rendimiento del código resultante.
4. Probar el producto resultante.
5. Reajustar las especificaciones para dejarlas en concordancia con el resultado de la experiencia operativa y volver a generar el código a partir de las especificaciones, volviendo a optimizar y probar el producto.

El modelo de transformación, por tanto, evita la dificultad de tener que modificar el código poco estructurado (por haber pasado por sucesivas reoptimizaciones), puesto que las modificaciones las aplica sobre la especificación de partida. Esto, también evita el tiempo adicional que se emplearía en los pasos intermedios de diseño, codificación y pruebas.

La dificultad que presentan estos modelos es que las posibilidades de transformación automática generalmente sólo están disponibles para productos relativamente pequeños y aplicados a unas áreas muy limitadas. También comparte algunas de las dificultades del modelo de desarrollo evolutivo tales como, por ejemplo, la suposición de que el sistema operacional del usuario final se prestará a evoluciones no planificadas con anterioridad.

Dentro de este tipo de modelos se encuentran:

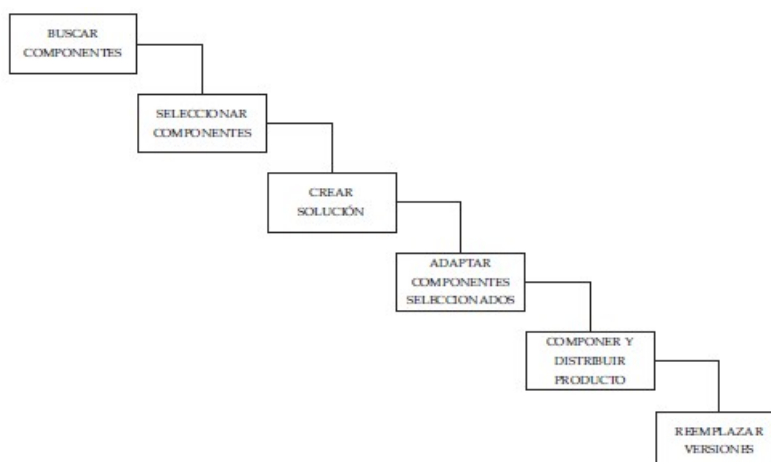


- Los que usan técnicas de cuarta generación (Roger Pressman): Suelen estar basados en herramientas de cuarta generación. Estos permiten la generación de código rápido. En ellos se indica qué se quiere obtener, no cómo.
- Basados en modelos de transformación (Carma McClure) => Basados en herramientas CASE que permiten, siguiendo el MCV clásico, pasar de una etapa a otra aplicando las transformaciones que dan las herramientas.

En ambos casos, la filosofía general es llegar a generar código a partir de unas especificaciones transformándolas por medio de herramientas.

#### *31.4.6 Desarrollo basado en componentes*

La complejidad de los sistemas computacionales actuales nos ha llevado a buscar la reutilización del software existente. El desarrollo de software basado en componentes permite reutilizar piezas de código preelaborado que permiten realizar diversas tareas, conllevando a diversos beneficios como las mejoras a la calidad, la reducción del ciclo de desarrollo y el mayor retorno sobre la inversión. En esencia, un componente es una pieza de código preelaborado que encapsula alguna funcionalidad expuesta a través de interfaces estándar. Los componentes son los "ingredientes de las aplicaciones", que se juntan y combinan para llevar a cabo una tarea. El paradigma de ensamblar componentes y escribir código para hacer que estos componentes funcionen se conoce como Desarrollo de Software Basado en Componentes.



Los pasos de que consta el ciclo de desarrollo para un sistema basado en componentes son:

1. Buscar componentes, tanto COTS (Comercial Off-The-Shelf) como no COTS.
2. Seleccionar los componentes más adecuados para el sistema.
3. Crear una solución compuesta que integre la solución previa.
4. Adaptar los componentes seleccionados de forma que se ajusten al modelo de componentes o a los requisitos de la aplicación.
5. Componer y distribuir el producto.
6. Reemplazar versiones anteriores o mantener las partes COTS y no COTS del sistema.

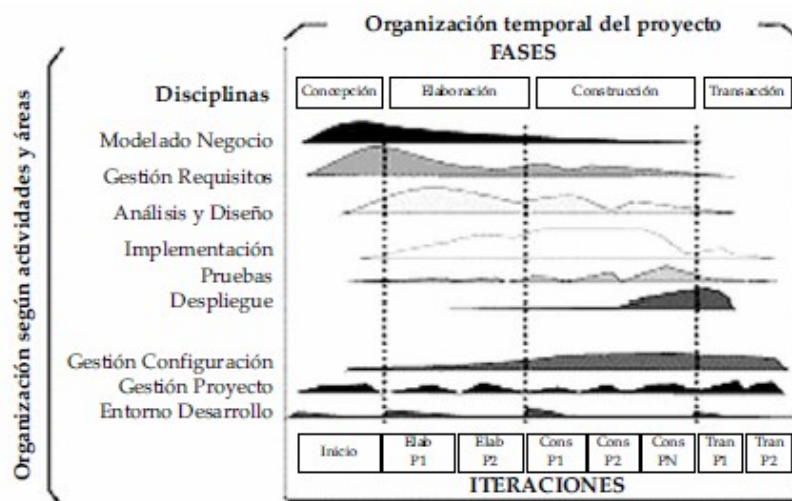
Además de los problemas inherentes a la reutilización del software, los productos COTS presentan problemas específicos como incompatibilidad, inflexibilidad (no existe código fuente), complejidad (esfuerzo de aprendizaje) o cambio de versiones, por lo que el establecimiento de métodos sistemáticos y repetibles para evaluar y seleccionar dichos componentes es un aspecto importante para el desarrollo del software basado en componentes y, en general, para la Ingeniería del Software Basada en Componentes (ISBC).

Entre las ventajas del Desarrollo basado en componentes tenemos que se reducen tiempos y costes de desarrollo y se aumenta la fiabilidad. Entre

los inconvenientes, tendremos la dificultad para reconocer los componentes potencialmente reutilizables, dificultad de catalogación y recuperación y los problemas de gestión de configuración.

### 31.4.7 Proceso Unificado de Desarrollo de software (PUDS)

En realidad es una metodología que propone un modelo de ciclo de vida. Está desarrollada por tres padres de la IS moderna: Yourdon, Booch y Rumbaugh. Plantea un modelo de ciclo de vida iterativo e incremental, centrado en una arquitectura que guía el desarrollo del sistema, cuyas actividades están dirigidas por casos de uso y soporta las técnicas orientadas a objetos. PUDS impulsa un control de calidad y una gestión de riesgos objetivos y continuos.



El PUDS se compone de fases, iteraciones y ciclos. Una fase es el intervalo de tiempo entre dos hitos importantes del proceso durante la cual se cumple un conjunto bien definido de objetivos, se completan entregables y se toman las decisiones sobre si pasar o no a la siguiente fase. Las **fases** son:

1. **Iniciación.** En esta fase se establece la visión del negocio, que incluye el contexto del negocio, los factores de éxito, y la previsión económica. Para completar la visión del negocio se genera un plan del proyecto, una descripción de los posibles riesgos y del propio proyecto (requisitos principales del proyecto, restricciones y características claves)
2. **Elaboración.** Es donde el proyecto comienza a tomar forma. En esta fase se hace el análisis del dominio del problema y se obtiene una idea básica de la arquitectura del sistema, además de revisarse los riesgos. En esta fase el proyecto todavía puede cancelarse o rediseñarse.
3. **Construcción.** En esta fase el enfoque se traslada al desarrollo de componentes y otras características del sistema que está siendo diseñado. Aquí se realiza el grueso de las tareas de codificación. En proyectos grandes, se puede dividir la fase en varias iteraciones para dividir los casos de uso en segmentos manejables que produzcan prototipos funcionales.
4. **Transición.** El producto se implanta en la organización del usuario final. Aquí se lleva a cabo la formación de los usuarios finales y las pruebas de aceptación del sistema para validarlo contra las expectativas del usuario.

En cada fase hay una o varias iteraciones. Una **iteración** ofrece como resultado un incremento del producto desarrollado que añade o mejora las funcionalidades del sistema en desarrollo. Cada fase e iteración se centra en disminuir algún riesgo y concluye con un hito bien definido. El paso a través de las 4 fases constituye un **ciclo** de desarrollo y produce una generación de software. El primer ciclo es el inicial y después serán ciclos de evolución del sistema.

Los flujos de trabajo del proceso son los siguientes:

- Modelado del negocio. El objetivo es establecer una mejor comprensión y un mejor canal de comunicación entre los clientes y los expertos en sistemas.
- Requisitos. El objetivo es describir lo que el sistema debe hacer.
- Análisis y diseño. Aquí se muestra la forma que tendrá el sistema en la fase de implementación.
- Implementación. Codificar y realizar pruebas unitarias.
- Pruebas. Se realizan pruebas de integración.
- Despliegue. Incluye una amplia variedad de actividades como la generación de versiones estables o la distribución e instalación del software.
- Configuración y gestión de cambios.
- Gestión del proyecto. Se realiza a 2 niveles, un nivel de grano grueso que trata la planificación de las fases y otro nivel de grano fino que trata la planificación de las iteraciones.
- Entorno.

#### *31.4.8 Modelo de métodos formales*

El modelo de métodos formales comprende un conjunto de actividades que conducen a la especificación matemática del software de computadora. Los métodos formales permiten que un ingeniero de software especifique, desarrolle y verifique un sistema basado en computadora aplicando una notación rigurosa y matemática. Algunas organizaciones de desarrollo del software actualmente aplican una variación de este enfoque, llamado **ingeniería del software de sala limpia**.

Cuando se utilizan métodos formales la ambigüedad, lo incompleto y la inconsistencia se descubren y se corrigen más fácilmente, no mediante una revisión a propósito para el caso, sino mediante la aplicación del análisis

matemático. Cuando se utilizan métodos formales durante el diseño, sirven como base para la verificación de programas y por consiguiente permiten que el ingeniero del software descubra y corrija errores que no se pudieron detectar de otra manera. Los modelos de métodos formales ofrecen la promesa de un software libre de defectos. Sin embargo, se ha hablado de una gran preocupación sobre su aplicabilidad en un entorno de gestión:

1. El desarrollo de modelos formales actualmente es bastante caro y lleva mucho tiempo.
2. Se requiere un estudio detallado porque pocos responsables del desarrollo de software tienen los antecedentes necesarios para aplicar métodos formales.
3. Es difícil utilizar los modelos como un mecanismo de comunicación con clientes que no tienen muchos conocimientos técnicos.

No obstante es posible que el enfoque a través de métodos formales tenga más partidarios entre los desarrolladores que deben construir software de mucha seguridad y robustez.

#### *31.4.9 Programación Extrema (eXtreme Programming)*

En la programación extrema, todos los requerimientos se expresan como escenarios (llamados historias de usuario), los cuales se implementan directamente como una serie de tareas. Los programadores trabajan en parejas y desarrollan pruebas para cada tarea antes de escribir el código. Todas las pruebas se deben ejecutar satisfactoriamente cuando el código nuevo se integre al sistema. Existe un pequeño espacio de tiempo entre las entregas del sistema. La programación extrema implica varias prácticas, que se ajustan a los principios de los métodos ágiles:

1. El desarrollo incremental se lleva a cabo a través de entregas del sistema pequeñas y frecuentes y por medio de un enfoque para la

- descripción de requerimientos basado en las historias de cliente o escenarios que pueden ser la base para el proceso de planificación.
2. La participación del cliente se lleva a cabo a través del compromiso a tiempo completo del cliente en el equipo de desarrollo. Los representantes de los clientes participan en el desarrollo y son los responsables de definir las pruebas de aceptación del sistema.
  3. El interés en las personas, en vez de en los procesos, se lleva a cabo a través de la programación en parejas, la propiedad colectiva del código del sistema, y un proceso de desarrollo sostenible que no implique excesivas jornadas de trabajo.
  4. El cambio se lleva a cabo a través de las entregas regulares del sistema, un desarrollo previamente probado y la integración continua.
  5. El mantenimiento de la simplicidad se lleva a cabo a través de la refactorización constante para mejorar la calidad del código y la utilización de diseños sencillos que no prevén cambios futuros en el sistema.

Los clientes del sistema son parte del equipo de desarrollo y discuten escenarios con otros miembros del equipo. Desarrollan conjuntamente una «tarjeta de historias» (story card) que recoge las necesidades del cliente. El equipo de desarrollo intentará entonces implementar ese escenario en una entrega futura del software. Una vez que se han desarrollado las tarjetas de historias, el equipo de desarrollo las divide en tareas y estima el esfuerzo y recursos requeridos para su implementación. El cliente establece entonces la prioridad de las historias a implementar.

El problema con la implementación de cambios imprevistos es que tienden a degradar la estructura del software, por lo que los cambios se hacen cada vez más difíciles de implementar. La programación extrema aborda este problema sugiriendo que se debe refactorizar constantemente el software. Esto significa que el equipo de programación busca posibles mejoras del

software y las implementa inmediatamente. Por lo tanto, el software siempre debe ser fácil de entender y cambiar cuando se implementen nuevas historias.

Otra práctica innovadora que se ha introducido es que los programadores trabajan en parejas para desarrollar el software. Las ventajas de esto son que apoya la idea de la propiedad y responsabilidad comunes del sistema, actúa como un proceso de revisión informal del código y ayuda en la refactorización

### **31.6 Metodologías de desarrollo de sistemas de información.**

Un proceso de software detallado y completo suele denominarse “Metodología”. Las metodologías se basan en una combinación de los modelos de proceso genéricos (cascada, evolutivo, incremental, etc.).

Adicionalmente una metodología debería definir con precisión los productos, roles y actividades involucrados, junto con prácticas y técnicas recomendadas, guías de adaptación de la metodología al proyecto, guías para uso de herramientas de apoyo, etc. Habitualmente se utiliza el término “método” para referirse a técnicas, notaciones y guías asociadas, que son aplicables a una (o algunas) actividades del proceso de desarrollo, por ejemplo, suele hablarse de métodos de análisis y/o diseño.

La comparación y/o clasificación de metodologías no es una tarea sencilla debido a la diversidad de propuestas y diferencias en el grado de detalle, información disponible y alcance de cada una de ellas. A grandes rasgos, si tomamos como criterio las notaciones utilizadas para especificar productos producidos en actividades de análisis y diseño, podemos clasificar las metodologías en dos grupos: Metodologías Estructuradas y Metodologías Orientadas a Objetos. Por otra parte, considerando su filosofía de desarrollo, aquellas metodologías con mayor énfasis en la planificación y control del proyecto, en especificación precisa de requisitos y modelado, reciben el apelativo de Metodologías Tradicionales (o peyorativamente



denominada Metodologías Pesadas, o Peso Pesado). Otras metodologías, denominadas Metodologías Ágiles, están más orientadas a la generación de código con ciclos muy cortos de desarrollo, se dirigen a equipos de desarrollo pequeños, hacen especial hincapié en aspectos humanos asociados al trabajo en equipo e involucran activamente al cliente en el proceso. A continuación se revisan brevemente algunas de estas categorías de metodologías.

- **Metodologías estructuradas:** Los métodos estructurados comenzaron a desarrollarse a fines de los 70's con la Programación Estructurada. A mediados de los 70's aparecieron técnicas para el Diseño (por ejemplo: el diagrama de Estructura) primero y posteriormente para el Análisis (por ejemplo: Diagramas de Flujo de Datos). Estas metodologías son particularmente apropiadas en proyectos que utilizan para la implementación lenguajes de 3ra y 4ta generación. Ejemplos de metodologías estructuradas de ámbito gubernamental: MERISE (Francia), MÉTRICA (España), SSADM (Reino Unido). Ejemplos de propuestas de métodos estructurados en el ámbito académico: Gane & Sarson, Ward & Mellor, Yourdon & DeMarco e Information Engineering.
- **Metodologías orientadas a objetos:** Su historia va unida a la evolución de los lenguajes de programación orientada a objetos. A fines de los 80's comenzaron a consolidarse algunos métodos Orientados a Objetos. En 1995 Booch y Rumbaugh proponen el Método Unificado con la ambiciosa idea de conseguir una unificación de sus métodos y notaciones, que posteriormente se reorienta a un objetivo más modesto, para dar lugar al Unified Modeling Language (UML), la notación OO más popular en la actualidad. Algunos métodos OO con notaciones predecesoras de UML son: OOAD (Booch), OOSE (Jacobson), Coad & Yourdon, Shaler & Mellor y OMT (Rumbaugh). Algunas metodologías orientadas a objetos que utilizan la notación

UML son: Rational Unified Process (RUP), OPEN, MÉTRICA (que también soporta la notación estructurada).

- **Metodologías tradicionales (no ágiles):** Las metodologías no ágiles son aquellas que están guiadas por una fuerte planificación durante todo el proceso de desarrollo; llamadas también metodologías tradicionales o clásicas, donde se realiza una intensa etapa de análisis y diseño antes de la construcción del sistema. Todas las propuestas metodológicas antes indicadas pueden considerarse como metodologías tradicionales..
- **Metodologías ágiles:** Un proceso es ágil cuando el desarrollo de software es **incremental** (entregas pequeñas de software, con ciclos rápidos), **cooperativo** (cliente y desarrolladores trabajan juntos constantemente con una cercana comunicación), **sencillo** (el método en sí mismo es fácil de aprender y modificar, bien documentado), y **adaptable** (permite realizar cambios de último momento). Algunas de las metodologías ágiles identificadas son Extreme Programming, Scrum, Familia de Metodologías Crystal, Feature Driven Development, Proceso Unificado Rational, Dynamic Systems Development Method, Adaptive Software Development. Se verán con más detalle en un apartado posterior.
- **Proceso Unificado de Rational:** Es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. El RUP no es un sistema con pasos firmemente establecidos, sino un conjunto de metodologías adaptables al contexto y necesidades de cada organización. Se verá en un apartado posterior.
- **Métrica V3:** La metodología MÉTRICA Versión 3 ofrece a las Organizaciones un instrumento útil para la sistematización de las actividades que dan soporte al ciclo de vida del software. Se verá en un apartado posterior.

- **Open Source Development Software:** Open Source es software desarrollado con la falta de coordinación, donde los programadores colaboran libremente, utilizando el código fuente distribuido y la infraestructura de comunicaciones de Internet. El código abierto se basa en la filosofía del software libre, sin embargo, extiende esta ideología ligeramente para presentar un enfoque más comercial que incluye tanto un modelo de negocio como una metodología de desarrollo.

### **31.7 Métrica Versión 3.**

La metodología MÉTRICA Versión 3 ofrece a las Organizaciones un instrumento útil para la sistematización de las actividades que dan soporte al ciclo de vida del software dentro del marco que permite alcanzar los siguientes objetivos:

- Proporcionar o definir Sistemas de Información que ayuden a conseguir los fines de la Organización mediante la definición de un marco estratégico para el desarrollo de los mismos.
- Dotar a la organización de productos software que satisfagan las necesidades de los usuarios dando una mayor importancia al análisis de requisitos.
- Mejorar la productividad de los departamentos de Sistemas y Tecnologías de la Información y las Comunicaciones, permitiendo una mayor capacidad de adaptación a los cambios y teniendo en cuenta la reutilización en la medida de lo posible.
- Facilitar la comunicación y entendimiento entre los distintos participantes en la producción de software a lo largo del ciclo de vida del proyecto, teniendo en cuenta su papel y responsabilidad, así como las necesidades de todos y cada uno de ellos.
- Facilitar la operación, mantenimiento y uso de los productos software obtenidos.

En la elaboración de MÉTRICA Versión 3 se han tenido en cuenta los métodos de desarrollo más extendidos, así como los últimos estándares de ingeniería del software y calidad, además de referencias específicas en cuanto a seguridad y gestión de proyectos. En una única estructura, la metodología MÉTRICA Versión 3 cubre distintos tipos de desarrollo: estructurado y orientado a objetos, facilitando a través de interfaces la realización de los procesos de apoyo u organizativos: Gestión de Proyectos, Gestión de Configuración, Aseguramiento de Calidad y Seguridad.

En lo que se refiere a estándares se ha tenido en cuenta como referencia el Modelo de Ciclo de Vida de Desarrollo propuesto en la norma ISO 12.207 *"Information technology - Software life cycle processes"*. Siguiendo este modelo se ha elaborado la estructura de MÉTRICA Versión 3 en la que se distinguen procesos principales (Planificación, Desarrollo y Mantenimiento) e interfaces (Gestión de Proyectos, Aseguramiento de la Calidad, Seguridad y Gestión de Proyectos) cuyo objetivo es dar soporte al proyecto en los aspectos organizativos. Además de la norma ISO 12.207, entre los estándares de referencia hay que destacar las normas ISO/IEC TR 15.504/SPIICE *"Software Process Improvement and Assurance Standards Capability Determination"*, UNE-EN-ISO 9001:2000 *Sistemas de Gestión de la Calidad. Requisitos*, UNE-EN-ISO 9000:2000 *Sistemas de Gestión de la Calidad. Fundamentos y Vocabulario* y el estándar IEEE 610.12-1.990 *"Standard Glossary of Software Engineering Terminology"*. Igualmente se han tenido en cuenta otras metodologías como SSADM, Merise, Information Engineering, MAGERIT. *Metodología de Análisis y Gestión de Riesgos de los Sistemas de Información* promovida por el Consejo Superior de Informática y EUROMÉTODO.

Se ha diferenciado entre la aplicación de Técnicas, como conjunto de heurísticas y procedimientos apoyados en estándares que utilizan notaciones específicas en términos de sintaxis y semántica, y de Prácticas cuya utilización no conlleva reglas preestablecidas con la misma rigidez.

Las nuevas técnicas están ampliamente soportadas por herramientas comerciales.

### *31.7.1 Procesos Principales de Métrica Versión 3*

MÉTRICA Versión 3 tiene un enfoque orientado al proceso y por ello, como ya se ha dicho, se ha enmarcado dentro de la norma ISO 12.207, que se centra en la clasificación y definición de los procesos del ciclo de vida del software. MÉTRICA Versión 3 cubre el Proceso de Desarrollo y el Proceso de Mantenimiento de Sistemas de Información.

MÉTRICA Versión 3 ha sido concebida para abarcar el desarrollo completo de Sistemas de Información sea cual sea su complejidad y magnitud, por lo cual su estructura responde a desarrollos máximos y deberá adaptarse y dimensionarse en cada momento de acuerdo a las características particulares de cada proyecto. La metodología descompone cada uno de los procesos en actividades, y éstas a su vez en tareas. Para cada tarea se describe su contenido haciendo referencia a sus principales acciones, productos, técnicas, prácticas y participantes. El orden asignado a las actividades no debe interpretarse como secuencia en su realización, ya que éstas pueden realizarse en orden diferente a su numeración o bien en paralelo. Sin embargo, no se dará por acabado un proceso hasta no haber finalizado todas las actividades del mismo determinadas al inicio del proyecto.

Así los procesos de la estructura principal de MÉTRICA Versión 3 son los siguientes:

- PLANIFICACIÓN DE SISTEMAS DE INFORMACIÓN.
- DESARROLLO DE SISTEMAS DE INFORMACIÓN.
- MANTENIMIENTO DE SISTEMAS DE INFORMACIÓN.

### *31.7.2 Planificación de sistemas de información*

El objetivo de un Plan de Sistemas de Información es proporcionar un marco estratégico de referencia para los Sistemas de Información de un determinado ámbito de la Organización. El resultado del Plan de Sistemas debe, por tanto, orientar las actuaciones en materia de desarrollo de Sistemas de Información con el objetivo básico de apoyar la estrategia corporativa, elaborando una arquitectura de información y un plan de proyectos informáticos para dar apoyo a los objetivos estratégicos. Por este motivo es necesario un proceso como el de Planificación de Sistemas de Información, en el que participen, por un lado los responsables de los procesos de la organización con una visión estratégica y por otro, los profesionales de SI capaces de enriquecer dicha visión con la aportación de ventajas competitivas por medio de los sistemas y tecnologías de la información y comunicaciones.

Como productos finales de este proceso se obtienen los siguientes:

- Catálogo de requisitos de PSI que surge del estudio de la situación actual en el caso de que sea significativo dicho estudio, del diagnóstico que se haya llevado a cabo y de las necesidades de información de los procesos de la organización afectados por el plan de sistemas.
- Arquitectura de información que se compone de los siguientes productos: modelo de información, modelo de sistemas de información, arquitectura tecnológica, plan de proyectos y plan de mantenimiento del PSI.

Este nuevo enfoque de alineamiento de los sistemas de información con la estrategia de la organización, la implicación directa de la alta dirección y la propuesta de solución presenta como ventajas:

- La implicación de la alta dirección facilita que se pueda desarrollar con los recursos necesarios y el calendario establecido.

- La perspectiva horizontal de los procesos dentro de la Organización facilita que se atienda a intereses globales y no particulares de unidades organizativas que puedan desvirtuar los objetivos del Plan.
- La prioridad del desarrollo de los sistemas de información de la organización por objetivos estratégicos.
- La propuesta de Arquitectura de Información que se hace en el plan es más estratégica que tecnológica.

### *31.7.3 Desarrollo de Sistemas de Información*

El proceso de Desarrollo de MÉTRICA Versión 3 contiene todas las actividades y tareas que se deben llevar a cabo para desarrollar un sistema, cubriendo desde el análisis de requisitos hasta la instalación del software. Además de las tareas relativas al análisis, incluye dos partes en el diseño de sistemas: arquitectónico y detallado. También cubre las pruebas unitarias y de integración del sistema. Este proceso es, sin duda, el más importante de los identificados en el ciclo de vida de un sistema y se relaciona con todos los demás.

En MÉTRICA Versión 3 se han abordado los dos tipos de desarrollo: estructurado y orientado a objeto, por lo que ha sido necesario establecer actividades a realizar en función del tipo de desarrollo elegido. Se han definido 5 subprocesos para este apartado:

- Estudio de Viabilidad del Sistema (EVS)
- Análisis Del Sistema De Información (ASI).
- Diseño Del Sistema De Información (DSI).
- Construcción Del Sistema De Información (CSI).
- Implantación Y Aceptación Del Sistema (IAS).

#### *31.7.3.1 Estudio de Viabilidad del Sistema (EVS)*

El propósito de este proceso es analizar un conjunto concreto de necesidades, con la idea de proponer una solución a corto plazo. Los criterios con los que se hace esta propuesta no serán estratégicos sino tácticos y relacionados con aspectos económicos, técnicos, legales y operativos. Los resultados del Estudio de Viabilidad del Sistema constituirán la base para tomar la decisión de seguir adelante o abandonar. Si se decide seguir adelante pueden surgir uno o varios proyectos que afecten a uno o varios sistemas de información. Se considerarán alternativas de solución basadas en soluciones "a medida", soluciones basadas en la adquisición de productos software del mercado o soluciones mixtas. Para valorar las alternativas planteadas y determinar una única solución, se estudiará el impacto en la organización de cada una de ellas, la inversión y los riesgos asociados. El resultado final de este proceso son los productos relacionados con la solución que se propone para cubrir la necesidad concreta que se planteó en el proceso, y que depende de si la solución conlleva desarrollo a medida o no.

#### *31.7.3.2 Análisis del Sistema de Información (ASI)*

El propósito de este proceso es conseguir la especificación detallada del sistema de información, a través de un catálogo de requisitos y una serie de modelos que cubran las necesidades de información de los usuarios para los que se desarrollará el sistema de información y que serán la entrada para el proceso de Diseño del Sistema de Información.

En primer lugar se describe inicialmente el sistema de información, a partir de los productos generados en el proceso Estudio de Viabilidad del Sistema (EVS). Se delimita su alcance, se genera un catálogo de requisitos



generales y se describe el sistema mediante unos modelos iniciales de alto nivel. Se recogen de forma detallada los requisitos funcionales que el sistema de información debe cubrir, catalogándolos, lo que permite hacer la traza a lo largo de los procesos de desarrollo. Además, se identifican los requisitos no funcionales del sistema, es decir, las facilidades que ha de proporcionar el sistema, y las restricciones a que estará sometido, en cuanto a rendimiento, frecuencia de tratamiento, seguridad, etc. Para facilitar el análisis del sistema se identifican los subsistemas de análisis, y se elaboran los modelos de Casos de Uso y de Clases, en desarrollos orientados a objetos, y de Datos y Procesos en desarrollos estructurados. Se especificarán todas las interfaces entre el sistema y el usuario, como formatos de pantallas, diálogos, formatos de informes y formularios de entrada. Finalizados los modelos, se realiza un análisis de consistencia. Una vez realizado dicho análisis de consistencia se elabora el producto **Especificación de Requisitos Software**, que constituye un punto de referencia en el desarrollo del software y la línea base de referencia para las peticiones de cambio sobre los requisitos inicialmente especificados. En este proceso se inicia también la especificación del Plan de Pruebas, que se completará en el proceso Diseño del Sistema de Información (DSI).

En este proceso es muy importante la participación de los usuarios, a través de técnicas interactivas, como diseño de diálogos y prototipos, que permiten al usuario familiarizarse con el nuevo sistema y colaborar en la construcción y perfeccionamiento del mismo.

#### *31.7.3.3 Diseño del Sistema de Información (DSI)*

El propósito del Diseño del Sistema de Información (DSI) es obtener la definición de la arquitectura del sistema y del entorno tecnológico que le va a dar soporte, junto con la especificación detallada de los componentes del sistema de información. A partir de dicha información, se generan todas las

especificaciones de construcción relativas al propio sistema, así como la especificación técnica del plan de pruebas, la definición de los requisitos de implantación y el diseño de los procedimientos de migración y carga inicial, éstos últimos cuando proceda.

Este proceso consta de un primer bloque de actividades, que se realizan en paralelo, y cuyo objetivo es obtener el diseño de detalle del sistema de información que comprende la partición física del sistema de información, independiente de un entorno tecnológico concreto, la organización en subsistemas de diseño, la especificación del entorno tecnológico sobre el que se despliegan dichos subsistemas y la definición de los requisitos de operación, administración del sistema, seguridad y control de acceso. En el caso de diseño orientado a objetos, conviene señalar que se ha contemplado que el diseño de la persistencia se lleva a cabo sobre bases de datos relacionales.

Un segundo bloque de actividades complementa el diseño del sistema de información, en el que se generan todas las especificaciones necesarias para la construcción del sistema de información.

#### *31.7.3.4 Construcción del Sistema de Información (CSI)*

La construcción del Sistema de Información (CSI) tiene como objetivo final la construcción y prueba de los distintos componentes del sistema de información, a partir del conjunto de especificaciones lógicas y físicas del mismo, obtenido en el Proceso de Diseño del Sistema de Información (DSI). Se desarrollan los procedimientos de operación y seguridad y se elaboran los manuales de usuario final y de explotación, estos últimos cuando proceda.

Para conseguir dicho objetivo, se recoge la información relativa al producto del diseño Especificaciones de construcción del sistema de

información, se prepara el entorno de construcción, se genera el código de cada uno de los componentes del sistema de información y se van realizando, a medida que se vaya finalizando la construcción, las pruebas unitarias de cada uno de ellos y las de integración entre subsistemas.

Si fuera necesario realizar una migración de datos, es en este proceso donde se lleva a cabo la construcción de los componentes de migración y procedimientos de migración y carga inicial de datos.

#### *31.7.3.5 Implantación y Aceptación del Sistema (IAS)*

Este proceso tiene como objetivo principal, la entrega y aceptación del sistema en su totalidad, que puede comprender varios sistemas de información desarrollados de manera independiente, según se haya establecido en el proceso de Estudio de Viabilidad del Sistema (EVS), y un segundo objetivo que es llevar a cabo las actividades oportunas para el paso a producción del sistema.

Se establece el plan de implantación, una vez revisada la estrategia de implantación y se detalla el equipo que lo realizará. Para el inicio de este proceso se toman como punto de partida los componentes del sistema probados de forma unitaria e integrados en el proceso Construcción del Sistema de Información (CSI), así como la documentación asociada. El Sistema se someterá a las Pruebas de Implantación con la participación del usuario de operación cuya responsabilidad, entre otros aspectos, es comprobar el comportamiento del sistema bajo las condiciones más extremas. También se someterá a las Pruebas de Aceptación cuya ejecución es responsabilidad del usuario final. En este proceso se elabora el plan de mantenimiento del sistema de forma que el responsable del mantenimiento conozca el sistema antes de que éste pase a producción. También se establece el acuerdo de nivel de servicio requerido una vez que se inicie la producción.

#### *31.7.4 Mantenimiento de Sistemas de Información (MSI)*

El objetivo de este proceso es la obtención de una nueva versión de un sistema de información desarrollado con MÉTRICA, a partir de las peticiones de mantenimiento que los usuarios realizan con motivo de un problema detectado en el sistema o por la necesidad de una mejora del mismo. Sólo se considerarán en MÉTRICA Versión 3 los tipos de Mantenimiento **Correctivo** y **Evolutivo**. Ante una petición de cambio de un sistema de información ya en producción, se realiza un registro de las peticiones, se diagnostica el tipo de mantenimiento y se decide si se le da respuesta o no, en función del plan de mantenimiento asociado al sistema afectado por la petición, y se establece con qué prioridad. La definición de la solución al problema o necesidad planteada por el usuario que realiza el responsable de mantenimiento, incluye un estudio del impacto, la valoración del esfuerzo y coste, las actividades y tareas del proceso de desarrollo a realizar y el plan de pruebas de regresión.

#### *31.7.5 Interfaces De Métrica Versión 3*

La estructura de MÉTRICA Versión 3 incluye también un conjunto de interfaces que definen una serie de actividades de tipo organizativo o de soporte al proceso de desarrollo y a los productos, que en el caso de existir en la organización se deberán aplicar para enriquecer o influir en la ejecución de las actividades de los procesos principales de la metodología y que si no existen habrá que realizar para complementar y garantizar el éxito del proyecto desarrollado con MÉTRICA Versión 3. Son cuatro:

- **Gestión de Proyectos:** Tiene como finalidad principal la planificación, el seguimiento y control de las actividades y de los recursos humanos y materiales que intervienen en el desarrollo de un Sistema de Información. Como consecuencia de este control es posible conocer en

todo momento qué problemas se producen y resolverlos o paliarlos lo más pronto posible, lo cual evitará desviaciones temporales y económicas. Las actividades de la Interfaz de Gestión de Proyectos son de tres tipos:

- o *Actividades de Inicio del Proyecto*, que permiten estimar el esfuerzo y establecer la planificación del proyecto.
  - o *Actividades de Seguimiento y Control*, supervisando la realización de las tareas por parte del equipo de proyecto y gestionando las incidencias y cambios en los requisitos que puedan presentarse y afectar a la planificación del proyecto.
  - o *Actividades de Finalización del Proyecto*, cierre y registro de la documentación de gestión.
- **Seguridad:** La interfaz de Seguridad hace posible incorporar durante la fase de desarrollo las funciones y mecanismos que refuerzan la seguridad del nuevo sistema y del propio proceso de desarrollo, asegurando su consistencia y seguridad, completando el plan de seguridad vigente en la organización o desarrollándolo desde el principio, utilizando MAGERIT como metodología de análisis y gestión de riesgos en el caso de que la organización no disponga de su propia metodología. Contempla dos tipos de actividades diferenciadas: las relacionadas con la seguridad intrínseca del sistema de información, y las que velan por la seguridad del propio proceso de desarrollo del sistema de información. Además se hace especial hincapié en la formación en materia de seguridad. Al ser finitos los recursos, no pueden asegurarse todos los aspectos del desarrollo de los sistemas de información, por lo que habrá que aceptar un determinado nivel de riesgo concentrándose en los aspectos más comprometidos o amenazados.
- **Gestión de la Configuración:** La interfaz de gestión de la configuración consiste en la aplicación de procedimientos administrativos y técnicos durante el desarrollo del sistema de

información y su posterior mantenimiento. Su finalidad es identificar, definir, proporcionar información y controlar los cambios en la configuración del sistema, así como las modificaciones y versiones de los mismos. Este proceso permitirá conocer el estado de cada uno de los productos que se hayan definido como elementos de configuración, garantizando que no se realizan cambios incontrolados y que todos los participantes en el desarrollo del sistema disponen de la versión adecuada de los productos que manejan. La gestión de configuración facilita además el mantenimiento del sistema, aportando información precisa para valorar el impacto de los cambios solicitados y reduciendo el tiempo de implementación de un cambio, tanto evolutivo como correctivo.

- **Aseguramiento de la Calidad:** El objetivo de la interfaz de Aseguramiento de la Calidad es proporcionar un marco común de referencia para la definición y puesta en marcha de planes específicos de aseguramiento de calidad aplicables a proyectos concretos. Las actividades están orientadas a verificar la calidad de los productos. Son actividades que evalúan la calidad y que son realizadas por un grupo de Asesoramiento de la Calidad independiente de los responsables de la obtención de los productos. Las actividades contempladas permitirán reducir, eliminar y prevenir las deficiencias de calidad de los productos a obtener, así como alcanzar una razonable confianza en que las prestaciones y servicios esperados por el cliente o el usuario queden satisfechas.

### **31.5 Metodologías Ágiles.**

En las **metodologías ágiles**, la creación de valor mediante la adaptación a las necesidades cambiantes aparece en un primer plano frente a la tradicional idea de diseñar un plan y cumplir unos calendarios/requerimientos estáticos. Los proyectos gestionados con metodologías ágiles se inician sin un detalle cerrado de lo que va a ser

construido. A nivel comercial, los proyectos pueden ser vendidos como servicios y no como productos. Las características básicas de los proyectos gestionados con metodologías ágiles son las siguientes:

- **Incertidumbre:** la dirección indica la necesidad estratégica que se desea cubrir (sin entrar en detalles), ofreciendo máxima libertad al equipo de trabajo.
- **Equipos auto-organizados:** no existen roles especializados
  - o Autonomía: libertad para la toma de decisiones.
  - o Auto-superación: de forma periódica se evalúa el producto que se esta desarrollando.
  - o Auto-enriquecimiento: transferencia del conocimiento.
- **Fases de desarrollo solapadas:** Las fases no existen como tal sino que se desarrollan tareas/actividades en función de las necesidades cambiantes durante todo el proyecto. De hecho, en muchas ocasiones no es posible realizar un diseño técnico detallado antes de empezar a desarrollar y ver algunos resultados. Por otra parte, las fases tradicionales efectuadas por personas diferentes no favorece el trabajo en equipo y pueden llegar a generar más inconvenientes que ventajas (por ej. un retraso en una fase, afecta a todo el proyecto).
- **Control sutil:** establecimientos de puntos de control para realizar un seguimiento adecuado sin limitar la libertad y creatividad del equipo. Así mismo, se recomienda:
  - o Evaluar el ambiente laboral, siendo fundamental la elección de personas que no generen conflictos.
  - o Reconocer los méritos mediante un sistema de evaluación justo y entender los errores como puntos de mejora y aprendizaje.
  - o Potenciar la interacción entre el equipo y el negocio, para que puedan conocer las necesidades de primera mano.

- **Difusión y transferencia del conocimiento:** alta rotación de los miembros de los equipos entre diferentes proyectos. Por otra parte, potenciar el acceso libre a la información y documentación.

Algunas de las metodologías ágiles más conocidas las veremos en los apartados siguientes.

### *31.5.1 Scrum*

Es un proceso de desarrollo de software iterativo e incremental utilizado comúnmente en entornos basados en la metodología Agile de desarrollo de software. Es un proceso marco que incluye un conjunto de prácticas y roles predefinidos. Los roles principales en Scrum son el ScrumMaster, que mantiene los procesos y trabaja de forma similar al director de proyecto, el ProductOwner, que representa a los stakeholders (clientes externos o internos), y el Team que incluye a los desarrolladores. Durante cada sprint, un periodo entre 15 y 30 días (la longitud es definida por el equipo), el equipo crea un incremento de software potencialmente entregable (utilizable). El conjunto de características que forma parte de cada sprint viene del **product backlog**, que es un conjunto de requisitos de alto nivel priorizados que dan forma al trabajo a realizar. Los elementos del backlog que forman parte del sprint se determinan durante la reunión de **sprint planning**. Durante esta reunión, el Product Owner informa al equipo de los elementos en el product backlog que quiere ver completados. El equipo entonces determina la cantidad de ese trabajo que puede comprometerse a completar durante el siguiente sprint. Durante el sprint, nadie puede cambiar el sprint backlog, lo que significa que los requisitos están congelados durante el sprint. Existen varias implementaciones de sistemas para gestionar el proceso de Scrum, que van desde notas amarillas "post-it" y pizarras hasta paquetes de software. Una de las mayores ventajas de Scrum es que es muy fácil de aprender, y requiere muy poco esfuerzo para comenzarse a utilizar.





### 31.5.2 Dynamic Systems Development Method (DSDM)

Provee un framework para el [desarrollo ágil de software](#), apoyado por la continua implicación del usuario en un [desarrollo iterativo y creciente](#). DSDM fue desarrollado en el [Reino Unido](#) en los [años 90](#). Como extensión del [Desarrollo rápido de aplicaciones](#) (RAD), DSDM se centra en los proyectos de sistemas de información que son caracterizados por presupuestos y agendas apretadas. DSDM trata los problemas que ocurren con frecuencia en el desarrollo de los sistemas de información en lo que respecta a pasar sobre tiempo y presupuesto y otras razones comunes para la falta en el proyecto tal como falta de implicación del usuario y de la comisión superior de la gerencia. DSDM consiste en 3 fases: fase del pre-proyecto, fase del ciclo de vida del proyecto, y fase del post-proyecto. La fase del ciclo de vida del proyecto se subdivide en 5 etapas: estudio de viabilidad, estudio de la empresa, iteración del modelo funcional, diseño e iteración de la estructura, e implementación. Tiene 9 principios fundamentales:

- **Involucrar al cliente** es la clave para llevar un proyecto eficiente y efectivo, donde ambos, cliente y desarrolladores, comparten un entorno de trabajo para que las decisiones puedan ser tomadas con precisión.
- **El equipo del proyecto debe tener el poder** para tomar decisiones que son importantes para el progreso del proyecto, sin esperar aprobación de niveles superiores.

- DSDM se centra en la **entrega frecuente de productos**, asumiendo que entregar algo temprano es siempre mejor que entregar todo al final. Al entregar el producto frecuentemente desde una etapa temprana del proyecto, el producto puede ser verificado y revisado allí donde la documentación de registro y revisión puede ser tomada en cuenta en la siguiente fase o iteración.
- El principal **criterio de aceptación** de entregables reside en entregar un sistema que satisface las actuales necesidades de negocio.
- El **desarrollo** es **iterativo e incremental**, guiado por la realimentación de los usuarios para converger en una solución de negocio precisa.
- Todos los **cambios** durante el desarrollo son reversibles.
- Requerimientos globales antes de comenzar el proyecto.
- Las pruebas son realizadas durante todo el ciclo vital del proyecto.
- La comunicación y cooperación entre todas las partes interesadas.

### *31.5.3 Extreme Programming (XP)*

La programación extrema se diferencia de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad. Los defensores de XP consideran que los cambios de requisitos sobre la marcha son un aspecto natural, inevitable e incluso deseable del desarrollo de proyectos. Creen que ser capaz de adaptarse a los cambios de requisitos en cualquier punto de la vida del proyecto es una aproximación mejor y más realista que intentar definir todos los requisitos al comienzo del proyecto e invertir esfuerzos después en controlar los cambios en los requisitos. XP construye un proceso de diseño evolutivo que se basa en refactorizar un sistema simple en cada iteración. Todo el diseño se centra en la iteración actual y no se hace nada anticipadamente para necesidades futuras. Los Valores originales de la programación extrema

son: **simplicidad** (de diseño, código y documentación), **comunicación** (la comunicación con el cliente es fluida ya que el cliente forma parte del equipo de desarrollo. El cliente decide que características tienen prioridad y siempre debe estar disponible para solucionar dudas), **retroalimentación** (*feedback*. Al estar el cliente integrado en el proyecto, su opinión sobre el estado del proyecto se conoce en tiempo real.), y **coraje** (Se requiere coraje para implementar las características que el cliente quiere ahora sin caer en la tentación de optar por un enfoque más flexible que permita futuras modificaciones). Un quinto valor, **respeto** (los miembros del equipo respetan el trabajo del resto no haciendo menos a otros, sino orientándolos a realizarlo mejor, obteniendo como resultado una mejor autoestima en el equipo y elevando el ritmo de producción en el equipo), fue añadido posteriormente.

Las características fundamentales de esta metodología son:

- Desarrollo iterativo e incremental
- Pruebas unitarias continuas
- Programación en parejas
- Integración del equipo de programación con el cliente
- Corrección de todos los errores
- Refactorización del código
- Propiedad del código compartida
- Simplicidad en el código

#### *31.5.4 Agile Modeling (AM)*

Se puede describir como una metodología basada en la práctica para el modelado efectivo de sistemas de software. No define procedimientos detallados de cómo crear un tipo de modelo dado. En lugar de eso, sugiere prácticas para que los modelos y documentación sean efectivos. Su secreto no está en las técnicas de modelado a usar, sino en como se aplican. No es

un desarrollo de software completo ya que no cubre actividades de programación, prueba, gestión de proyectos, implementación, soporte u otros elementos de la realización de proyectos que no sean la documentación y el modelado. Es necesario, por lo tanto, combinarlo con otras metodologías como pueden ser XP, DSDM, SCRUM o RUP. Los valores de esta metodología son la **comunicación** (entre participantes del equipo de trabajo, desarrolladores y analistas, etc.), **simplicidad**, **coraje** (para tomar decisiones importantes y ser capaces de cambiar de dirección cuando el camino tomado no es el correcto) y **humildad** (todos los interesados en el proyecto pueden contribuir en algo para la mejor realización).

#### 31.5.5 *Feature Driven Development (FDD):*

Se basa en un proceso iterativo con iteraciones cortas que producen un software funcional que el cliente y la dirección de la empresa pueden ver y monitorizar. Las iteraciones se deciden en base a funcionalidades, que son pequeñas partes del software con significado para el cliente. **No** cubre todo el ciclo de vida sino sólo las fases de diseño y construcción. No requiere un modelo específico de proceso y se complementa con otras metodologías. FDD consiste en cinco procesos secuenciales durante los cuales se diseña y construye el sistema:

- **Desarrollo de un modelo general:** Cuando comienza esta fase, los expertos del dominio ya tienen una idea del contexto y los requerimientos del sistema. El dominio global es dividido en diferentes áreas y se realiza informe detallado para cada una de ellas por parte de los expertos del dominio.
- **Construcción de la lista de funcionalidades** Los ensayos, modelos de objetos y documentación de requerimientos proporcionan la base para construir una amplia lista de funcionalidades. Estas

funcionalidades son pequeños ítems útiles a los ojos del cliente. La lista de funcionalidades es revisada por los usuarios y patrocinadores para asegurar su validez. Las funcionalidades que requieran de más de diez días se descomponen en otras más pequeñas.

- **Planeamiento por funcionalidades:** En esta etapa se incluye la creación de un plan de alto nivel, en el cual la lista de funcionalidades es ordenada en base a la prioridad y a la dependencia entre cada funcionalidad. Además, las clases identificadas en la primera etapa son asignadas a cada programador.
- **Diseño y construcción por funcionalidades:** El diseño y construcción de la funcionalidad es un proceso iterativo durante el cual las funcionalidades seleccionadas son producidas. Una iteración puede llevar desde unos pocos días a un máximo de dos semanas. Este proceso iterativo incluye tareas como inspección del diseño, codificación, pruebas unitarias, integración e inspección del código.

### *31.5.6 Familia Cristal*

Alistair Cockburn es el propulsor detrás de la serie de metodologías Crystal. Es una familia porque él cree que los tipos diferentes de proyectos requieren tipos diferentes de metodologías. Él mira esta variación a lo largo de dos ejes: el número de personas en el proyecto, y las consecuencias de los errores. Dispone un código de **color** para marcar la complejidad de cada metodología. Comparte con la XP una orientación humana, pero esta centralización en la gente se hace de una manera diferente. Alistair considera que las personas encuentran difícil seguir un proceso disciplinado, así que más que seguir la alta disciplina de la XP, Alistair explora la metodología menos disciplinada que aun podría tener éxito, intercambiando conscientemente productividad por facilidad de ejecución. Él considera que aunque Cristal es menos productivo que la XP, más personas serán capaces de seguirlo. Alistair también pone mucho peso en

las revisiones al final de la iteración, animando al proceso a ser “automejorable”. Defiende que el desarrollo iterativo está para encontrar los problemas temprano, y entonces permitir corregirlos. Esto pone más énfasis en la gente supervisando su proceso y afinándolo conforme desarrollan.

## **Bibliografía**

- Ingeniería del Software. Un enfoque práctico. ROGER S. PRESSMAN. Ed. McGraw Hill
- Ingeniería de Software 7 Edición - Ian Sommerville
- Metodologías para la gestión y desarrollo de software - ¿?
- <http://msdn.microsoft.com/es-es/library/bb972268.aspx> Desarrollo basado en Componentes.
- <http://alarcos.inf-cr.uclm.es/per/fruiz/cur/pso/trans/res.pdf> Curso de Proceso Software: Conceptos, Estándares, Modelos, Arquitecturas y Herramientas. Francisco Ruiz
- [http://administracionelectronica.gob.es/archivos/pae\\_000001027.pdf](http://administracionelectronica.gob.es/archivos/pae_000001027.pdf) Introducción a Métrica Versión 3. Ministerio de Administraciones Públicas
- Análisis, Diseño y Mantenimiento del Software. José Ramón Álvarez Sánchez y Manuel Arias Calleja. Dpto. de Inteligencia Artificial - ETSI Informática - UNED
- <http://www.marblestation.com/?p=661>. Metodologías ágiles de gestión de proyectos. Sergi Blanco Cuaresma.
- <http://es.wikipedia.org/wiki/DSDM>
- Agile Modeling (AM) Felipe Ferrada.
- [http://www.ort.edu.uy/fi/publicaciones/ingsoft/investigacion/ayudantias/metodologia\\_FDD.pdf](http://www.ort.edu.uy/fi/publicaciones/ingsoft/investigacion/ayudantias/metodologia_FDD.pdf). Metodología FDD. Cátedra de Ingeniería de Software. Luis Calabria. Universidad ORT Uruguay
- <http://www.programacionextrema.org/articulos/newMethodology.es.html> La Nueva Metodología. [Martin Fowler](#)

Autor: Hernán Vila Pérez  
Jefe del Servicio de Informática. Instituto Galego de Vivenda e Solo  
Vicepresidente del CPEIG

**32. ANÁLISES ESTRUCTURADA.  
ANÁLISES ORIENTADA A  
OBJETOS. LENGUAJE  
UNIFICADO DE MODELADO  
(UML). DISEÑO. DISEÑO  
ESTRUCTURADO. DISEÑO DE  
DATOS. DISEÑO DEL  
INTERFACE DE USUARIO.  
DISEÑO PROCEDIMENTAL.  
DISEÑO ORIENTADO A  
OBJETOS.**

Tema 32: Análisis estructurado. Análisis orientado a objetos. Lenguaje unificado de modelado (UML). Diseño. Diseño estructurado. Diseño de datos. Diseño de la interfaz de usuario. Diseño procedimental. Diseño orientado a objetos.

### 32.1 Análisis del sistema de información.

#### 32.1.1 Análisis estructurado.

#### 32.1.2 Análisis orientado a objetos.

#### 32.1.3 Lenguaje unificado de modelado.

### 32.2 Diseño de un sistema de información.

#### 32.2.1 Diseño estructurado.

#### 32.2.2 Diseño de datos.

#### 32.2.3 Diseño de la interfaz de usuario.

#### 32.2.4 Diseño procedimental.

#### 32.2.5 Diseño orientado a objetos.

### **32.1 Análisis del sistema de información.**

La ingeniería del software empieza con una serie de tareas de modelado que llevan a una especificación completa de los requisitos y a una representación del diseño general del software a construir. El *modelo de análisis*, realmente un conjunto de modelos, es la primera representación técnica de un sistema. Con los años se han propuesto muchos métodos para el modelado del análisis, pero el panorama actual está dominado por dos tendencias: el **análisis estructurado**, método de modelado clásico, cada vez más en desuso, y el **análisis orientado a objetos**, que es la tendencia que domina hoy en día.

#### **32.1.1 Análisis estructurado.**

El término análisis estructurado originalmente acuñado por Douglas Ross, fue popularizado por Tom DeMarco en su libro “Análisis Estructurado”, que



fue el primero en establecer los principios por los cuales debe guiarse un proceso de análisis de software. Se basa principalmente en el análisis de los flujos de información de un sistema, permitiendo al analista conocer un sistema o proceso de una forma lógica y manejable, al mismo tiempo que proporciona la base para asegurar que no se omite ningún detalle pertinente.

Según Pressman, el modelo de análisis debe lograr tres objetivos primarios:

1. Describir lo que requiere el cliente.
2. Establecer una base para la creación de un diseño de software
3. Definir un conjunto de requisitos que se pueda validar una vez que se construye el software.

Para lograr estos objetivos, el modelo de análisis extraído durante el análisis estructurado toma la forma ilustrada en la siguiente figura:



En el centro del modelo se encuentra el **diccionario de datos** (un almacén que contiene definiciones de todos los objetos de datos consumidos y producidos por el software). Tres diagramas diferentes rodean el núcleo.

- El **diagrama de entidad-relación** (DER) representa las relaciones entre los objetos de datos. El DER es la notación que se usa para

realizar la actividad de modelado de datos. Los atributos de cada objeto de datos señalados en el DER se puede describir mediante una descripción de objetos de datos.

- El **diagrama de flujo de datos** (DFD) sirve para dos propósitos:
  - a) Proporcionar una indicación de cómo se transforman los datos a medida que se avanza en el sistema
  - b) representar las funciones (y subfunciones) que transforman el flujo de datos.

El DFD proporciona información adicional que se usa durante el análisis del dominio de información y sirve como base para el modelado de función. En una especificación de proceso (EP) se encuentra una descripción de cada función presentada en el DFD.

- El **diagrama de transición de estados** (DTE) indica cómo se comporta el sistema como consecuencia de sucesos externos. Para lograr esto, el DTE representa los diferentes modos de comportamiento (llamados estados) del sistema y la manera en que se hacen las transiciones de estado a estado. El DTE sirve como la base del modelado de comportamiento. Dentro de la especificación de control (EC) se encuentra más información sobre los aspectos de control del software.

Estos tres diagramas permiten modelizar completamente el sistema desde tres puntos de vista:

- **Punto de Vista Funcional.** Se describe cómo se transforman los datos dentro del sistema. Para describir este modelo funcional se crea un modelo de procesos, el cual se construye usando la técnica de los Diagramas de Flujo de Datos.
- **Punto de Vista Estático.** Describe los objetos de información del sistema y cómo se relacionan estos objetos entre ellos. Para crear este modelo estático se construye el modelo de datos, el cual se puede construir usando la técnica del diagrama Entidad – Relación.

- **Punto de Vista Dinámico.** Describe los aspectos del sistema que cambian a lo largo del tiempo. Se describen los estados por los que va pasando el sistema y las condiciones y eventos que hacen que el sistema pase de un estado a otro. Una posible técnica para construir el modelo dinámico es la del Diagrama de Transición de Estados.

La metodología de **Análisis Estructurado Moderno** es una propuesta de Edward Yourdon que establece un nuevo modelo para el análisis estructurado e incorpora las ideas, criterios y herramientas que lo precedieron. Se basa en la construcción del **modelo esencial** del sistema, que indica lo que el sistema debe hacer para satisfacer los requisitos del usuario y debe mencionar lo mínimo posible como el sistema lo llevará a cabo. Está compuesto por dos componentes principales:

- **El Modelo Ambiental:** define la frontera entre el sistema y el resto del mundo (es decir, el ambiente donde el sistema reside). El modelo ambiental está compuesto de un Diagrama de Contexto, una Lista de Eventos y una descripción pequeña del propósito del sistema, la Declaración de Objetivos.
- **El Modelo de Comportamiento:** describe la conducta, del interior del sistema, necesaria para interactuar exitosamente con el ambiente. Esta compuesto de Diagramas de Flujo de Datos, Lista de Eventos, Diagramas de Entidades y Relaciones, Diagramas de Transición de Estados, Diccionario de Datos y Especificación de Procesos.

### **31.1.2 Análisis orientado a objetos.**

Los **objetos** son entidades que tienen atributos (datos) y formas de comportamiento (procedimientos o métodos) particulares. Las **clases** describen un conjunto de objetos diferentes con propiedades (atributos) similares y un comportamiento común. Las clases son un concepto estático definido en el programa fuente, son una abstracción de la esencia de un

objeto, mientras que los objetos son entes dinámicos que existen en tiempo y espacio y que ocupan memoria en la ejecución de un programa. Los objetos interaccionan entre si mediante **mensajes**, que no es más que una solicitud que hace un objeto a otro pidiéndole que se comporte de alguna forma determinada.

Los principios del modelo orientado a objetos son:

- **Identidad:** Cada objeto tiene su propia identidad inherente, es decir, dos objetos son distintos aunque tengan todas sus propiedades iguales.
- **Clasificación.** Se refiere a que los objetos que tienen la misma estructura de datos (atributos), y el mismo comportamiento (operaciones), están agrupados en una clase.
- **Herencia.** La herencia es el mecanismo mediante el cual una clase (subclase) adquiere las propiedades de otra clase jerárquicamente superior (superclase, clase base). La herencia proporciona el mecanismo para compartir automáticamente métodos y datos entre clases y subclases.
- **Abstracción:** Se trata de abstraer los datos y métodos comunes a un conjunto de objetos para almacenarlos en una clase.
- **Encapsulación:** Es el término que se utiliza para expresar que los datos de un objeto sólo pueden ser manipulados mediante los mensajes y métodos predefinidos.
- **Polimorfismo:** Es la propiedad por la cual un mismo mensaje puede originar conductas diferentes al ser recibido por objetos diferentes. También hablamos de polimorfismo cuando tenemos distintos métodos que tienen un comportamiento distinto en función del número o tipo de parámetros que reciben.
- **Reusabilidad:** Es la capacidad de producir componentes reutilizables para otros diseños o aplicaciones.

- **Persistencia:** La persistencia es la cualidad que se refiere a la permanencia del objeto, es decir, al tiempo durante el cual se le asigna espacio y permanece accesible en la memoria del ordenador (principal o secundaria).
- **Extensibilidad:** Es la capacidad de un programa para ser fácilmente alterado de forma que pueda tratar con nuevas clases de entrada.

El Análisis Orientado a Objetos es un método de análisis que examina los requisitos desde la perspectiva de las clases y objetos encontrados en el vocabulario de del dominio del problema (Booch). Debe ser comprendido por el cliente y servir de ayuda para encontrar los verdaderos requerimientos del sistema. Firesmith describe el **análisis del dominio** como “la identificación, análisis y especificación de requisitos comunes en un dominio de aplicación específico, normalmente para su reutilización en múltiples proyectos dentro del mismo dominio de aplicación. El análisis orientado a objetos del dominio es la identificación, análisis y especificación de capacidades comunes y reutilizables dentro de un dominio de aplicación específico, en términos de objetos, clases, submontajes y marcos de trabajo comunes”. Al igual que en los métodos estructurados tradicionales, en la etapa Análisis hay que establecer qué es lo que debe hacerse, dejando para etapas posteriores los detalles. El resultado del análisis debe ser una completa comprensión del problema. Las dos grandes etapas de que consta el Análisis son las siguientes:

1. La descripción o especificación del problema. Esta descripción no debe considerarse inmutable, sino más bien como la base para ir refinando las especificaciones reales. La especificación del problema debe comprender el establecer el ámbito del problema, describir las necesidades y requisitos, el contexto de la aplicación, los supuestos de que se parte o las necesidades de rendimiento del sistema. En estas especificaciones, el usuario del sistema debe indicar cuáles son obligadas y cuáles se pueden considerar opcionales. Asimismo, otros puntos a tratar pueden ser los estándares de Ingeniería del Software,

diseño de las pruebas a efectuar, previsión de futuras extensiones, etc.

2. La modelización del Análisis: Las características esenciales deben abstraerse en un modelo. Las especificaciones expresadas en lenguaje natural tienden a ser ambiguas, incompletas e inconsistentes, sin embargo, el Modelo de Análisis es una representación precisa y concisa del problema, que permite construir una solución. La etapa siguiente de diseño se remitirá a este modelo, en lugar de a las vagas especificaciones iniciales. El Modelo de Análisis se construye identificando las clases y objetos del dominio del problema (estructura estática), las interacciones entre los objetos y su secuenciamiento (estructura dinámica), y las acciones a realizar por el sistema que producen un resultado observable y valioso para los usuarios (estructura funcional).

### **32.1.3 Lenguaje unificado de modelado (UML).**

UML (Unified Modeling Language) es un lenguaje que permite modelar, construir y documentar los elementos que forman un sistema software orientado a objetos. Se ha convertido en el estándar de facto de la industria, debido a que ha sido impulsado por los autores de los tres métodos más usados de orientación a objetos: Grady Booch, Ivar Jacobson y Jim Rumbaugh. Es el estándar actual del llamado Object Management Group (OMG).

UML sirve para *especificar*, modelos concretos, no ambiguos y completos. Un modelo de UML representa a un sistema software desde una perspectiva específica. Cada modelo nos permite fijarnos en un aspecto distinto del sistema. Debido a su estandarización, y aunque no sea un lenguaje de programación, UML se puede conectar de manera directa a lenguajes de programación como Java, C++ o Visual Basic.

UML se expresa a través de *elementos de construcción*, de *relaciones* y de *diagramas* que contienen elementos y relaciones

Los **elementos de construcción** son los bloques básicos de construcción que serán de cuatro tipos:

1. **Elementos estructurales.** Son las partes estáticas del modelo. Son siete:
  - a) **Clase:** Descripción de un conjunto de objetos que comparten los mismos atributos, operaciones, relaciones y semántica.
  - b) **Interfaz:** colección de operaciones que especifican un servicio de una clase o componente, mostrando el comportamiento visible externamente de ese elemento.
  - c) **Colaboración:** define una interacción y representa un conjunto de elementos del modelo que colaboran para proporcionar un comportamiento cooperativo mayor que la suma de los comportamientos de sus elementos.
  - d) **Caso de uso:** descripción de un conjunto de secuencias de acciones que un sistema ejecuta y que produce un resultado observable de interés para un usuario particular.
  - e) **Clase activa:** Clase cuyos objetos tienen uno o más procesos o hilos de ejecución, y por lo tanto pueden dar origen a actividades de control.
  - f) **Componente:** Parte física y reemplazable de un sistema que representa típicamente el empaquetamiento físico de diferentes elementos lógicos, como clases, interfaces y colaboraciones.
  - g) **Nodo:** Elemento físico que existe en tiempo de ejecución y representa un recurso computacional que generalmente dispone de memoria y capacidad de procesamiento. (Impresoras, PCs, ...)
2. **Elementos de comportamiento:** Partes dinámicas de los modelos. Son dos:

- a) **Interacciones:** una interacción es un comportamiento que comprende un conjunto de mensajes intercambiados entre un conjunto de objetos, dentro de un contexto particular para alcanzar un propósito específico. El comportamiento de una sociedad de objetos o una operación individual puede especificarse mediante una interacción.
  - b) **Máquinas de estados:** una máquina de estados especifica las secuencias de estados por las que pasa un objeto o una interacción durante su vida en respuesta a eventos.
3. **Elementos de agrupación:** Son las partes organizativas de los modelos de UML. Solo existe una: el paquete, que es que es un mecanismo para organizar los elementos en grupos.
  4. **Elementos de anotación:** Son comentarios que se pueden aplicar para describir, clarificar y hacer observaciones sobre cualquier elemento de un modelo. El principal elemento de anotación es la nota, que es simplemente un símbolo para mostrar restricciones y comentarios junto a un elemento o una colección de elementos.

Las **relaciones** entre los elementos estructurales son de cuatro tipos:

- **Dependencia:** es una relación semántica entre dos elementos, en la cual un cambio a un elemento (el elemento independiente) puede afectar a la semántica del otro elemento (el elemento dependiente).
- **Asociación:** Una **asociación** es una relación estructural que describe un conjunto de enlaces, los cuales son conexiones entre objetos. La **agregación** es un tipo especial de asociación, que representa una relación estructural entre un todo y sus partes. La **composición** es un tipo de agregación en el que cada parte solo puede pertenecer a un todo y no puede existir la parte sin el todo.
- **Generalización:** Una generalización es una relación de especialización /generalización en la cual los objetos del elemento especializado (el hijo) pueden sustituir a los objetos del elemento



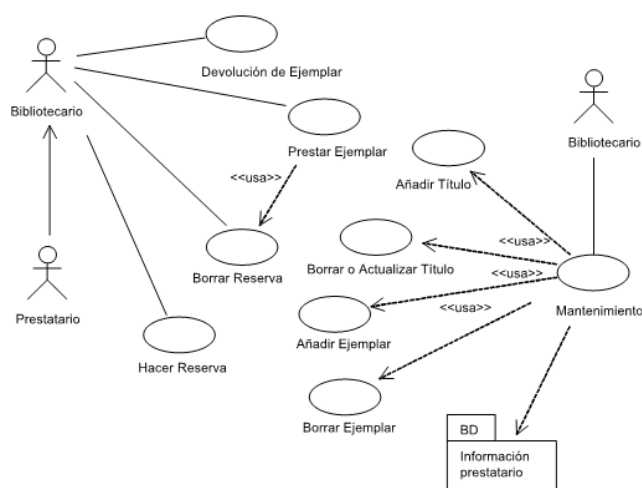
general (el padre). De esta forma, el hijo comparte la estructura y el comportamiento del padre.

- **Realización:** Es una relación semántica entre elementos, en donde un elemento especifica un contrato que otro elemento garantiza que cumplirá. Pueden ser entre interfaces y las clases y componentes que las realizan, y entre los casos de uso y las colaboraciones que los realizan.

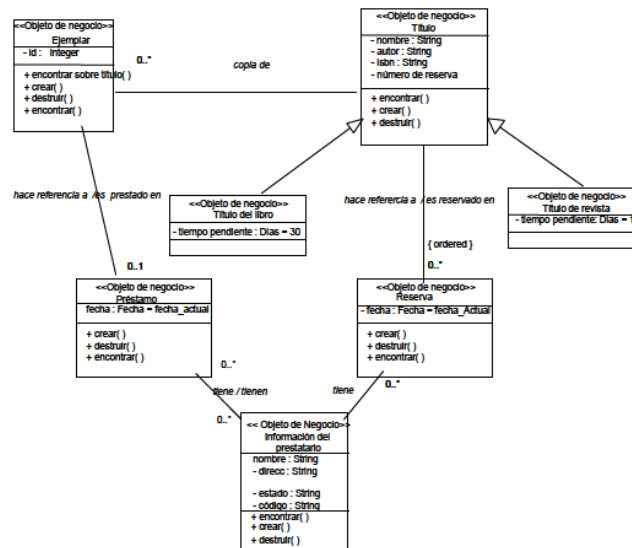
Por último, los **diagramas** son la representación gráfica de un conjunto de elementos, en general visualizado como un grafo conexo de nodos (elementos) y arcos (relaciones). Los diagramas se dibujan para visualizar un sistema desde diferentes perspectivas. Se pueden agrupar en dos bloques en función de si vemos el modelo de forma estática (estructural) o de forma dinámica (comportamiento). La primera incluye los diagramas de despliegue, componentes, clases y objetos, mientras que la segunda incluye los diagramas de estados, actividades, secuencia, colaboración y casos de uso.

1. **Diagrama de casos de uso.** Un caso de uso es una secuencia de acciones realizadas por el sistema, que producen un resultado observable y valioso para un usuario en particular, es decir, representa el comportamiento del sistema con el fin de dar respuestas a los usuarios. Los escenarios son los distintos caminos por los que puede evolucionar un caso de uso, dependiendo de las condiciones que se van dando en su realización. Los diagramas están formados por dos elementos: **actores**, que es algo o alguien que se encuentra fuera del sistema y que interactúa con él, y **casos de uso**, que representa el comportamiento que ofrece el sistema de información desde el punto de vista del usuario. Entre estos elementos se pueden dar tres tipos de relaciones: comunica (es la relación entre un actor y un caso de uso, que denota la participación del actor en dicho caso de uso), usa (relación de dependencia entre dos casos de uso que denota la inclusión del comportamiento de un

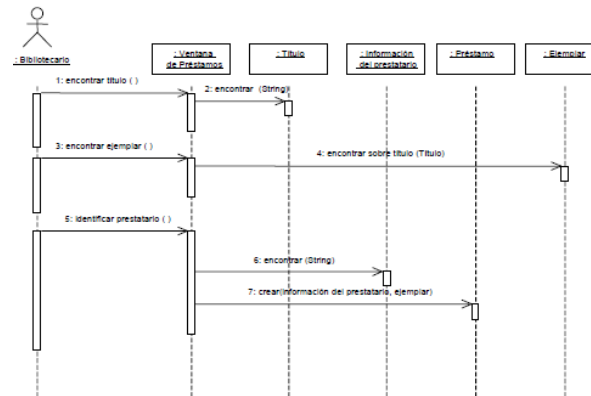
escenario en otro. Se usa cuando se quiere reflejar un comportamiento común en varios casos de uso) y extiende (relación de dependencia entre dos casos de uso en el que uno es una especialización del otro).



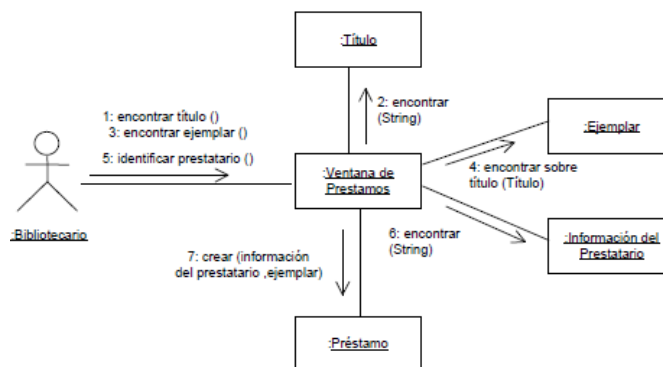
2. **Diagrama de clases:** Representa los aspectos estáticos del sistema, utilizando diversos mecanismos de abstracción (clasificación, generalización, agregación). Recoge las clases de objetos y sus asociaciones. En este diagrama se representa la estructura y el comportamiento de cada uno de los objetos del sistema y sus relaciones con los demás objetos. El modelo está formado por: **clases** (se representa como una caja dividida en tres zonas. En la zona superior se pone el nombre de la clase, en el centro se colocan los atributos (características) de la clase y en la zona inferior se incluye una lista con las operaciones que proporciona la clase), **relaciones** (asociaciones, agregaciones, composiciones, dependencias o herencia), **interfaces y paquetes**.



3. **Diagrama de objetos:** Modelan las instancias de elementos contenidos en los diagramas de clases. Muestra un conjunto de objetos y sus relaciones en un momento concreto. La representación gráfica de un objeto en UML es igual que la de una clase pero con el nombre subrayado.
4. **Diagramas de Interacción.** Se utilizan para modelar los aspectos dinámicos del sistema. Existen dos tipos de diagramas de interacción: **secuencia** y **colaboración**. Ambos son equivalentes. La diferencia entre ellos está en los aspectos que resaltan. Los diagramas de secuencia destacan el orden temporal de los mensajes, mientras que los diagramas de colaboración destacan la organización estructural de los objetos.
  - a. **Diagramas de secuencia:** Muestran las interacciones entre objetos ordenadas en **secuencia temporal**. Muestra los objetos que se encuentran en el escenario y la secuencia de mensajes intercambiados entre ellos para llevar a cabo la funcionalidad descrita por el escenario.

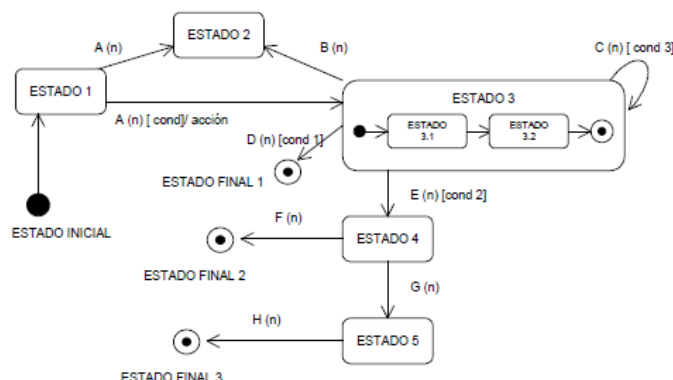


- b. **Diagramas de colaboración:** Es una forma alternativa al diagrama de secuencia para mostrar un escenario. Muestra la misma información pero de forma diferente. En los diagramas de colaboración no existe una secuencia temporal. Este diagrama resalta la **organización estructural** de los objetos que envían y reciben los mensajes. Este tipo de diagrama muestra un conjunto de objetos, enlaces entre ellos y los mensajes que intercambian.

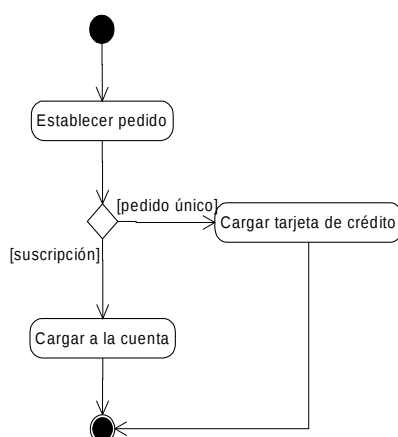


5. **Diagramas de estados:** Representan los estados que puede tomar un componente o un sistema y muestra los eventos que implican el cambio de un estado a otro. Los dos elementos principales en estos diagramas son los **estados** (representa algún comportamiento que es observable externamente y que perdura durante un periodo de tiempo finito. Viene dado por el valor de uno o varios atributos que lo caracterizan en un momento dado) y las posibles **transiciones** entre ellos. Una transición es un cambio de estado producido por un evento

y refleja los posibles caminos para llegar a un estado final desde un estado inicial. Un sistema sólo puede tener un **estado inicial**, que representa el punto de partida. El **estado final** representa que un componente ha dejado de tener cualquier interacción o actividad. Puede haber varios estados finales en un diagrama.



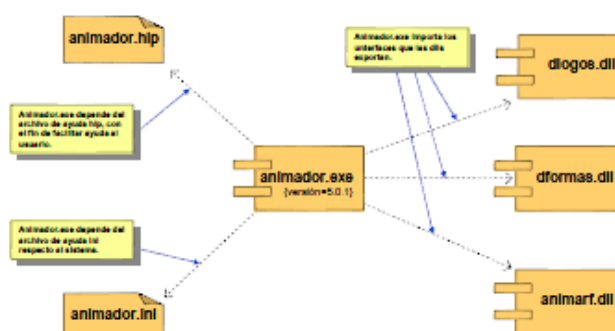
6. **Diagrama de actividades.** Puede considerarse un caso especial del diagrama de estados en el cual casi todos los estados son estados acción (identifican una acción que se ejecuta al entrar en él) y casi todas las transiciones evolucionan al término de dicha acción. Se suelen utilizar para modelar los pasos de un algoritmo.



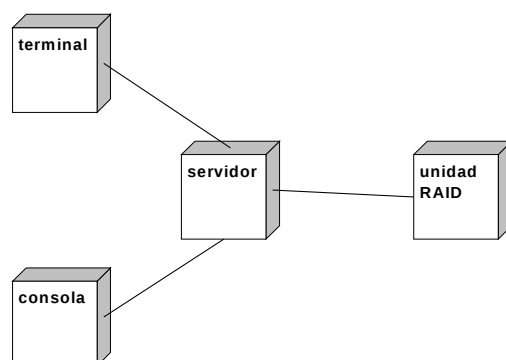
7. **Diagramas de implementación.** Un diagrama de implementación muestra las dependencias entre las partes de código del sistema (diagrama de componentes) o la estructura del sistema en ejecución (diagrama de despliegue). Los diagramas de componentes se utilizan para modelar la vista de implementación estática de un sistema,

mientras que los diagramas de despliegue se utilizan para modelar la vista de despliegue estática.

8. **Diagrama de componentes:** Muestra las organizaciones y dependencias lógicas entre componentes software, sean éstos componentes de código fuente, binarios, documentos, archivos o ejecutables. Los diagramas de componentes pueden contener paquetes para organizar los elementos.



9. **Diagrama de despliegue.** Muestra las relaciones físicas entre los componentes hardware y software en el sistema final, es decir, la configuración de los elementos de procesamiento en tiempo de ejecución y los componentes software (proceso y objetos que se ejecutan en ellos). Cubre principalmente la distribución, entrega e instalación de las partes que configuran el sistema físico, y se suelen utilizar para modelar sistemas empotrados, sistemas cliente-servidor y sistemas distribuidos.



## 32.2 Diseño de un sistema de información

Se define el **Diseño del Sistema** como *el proceso de definición de la arquitectura software: componentes, módulos, interfaces, procedimientos de prueba y datos de un sistema que se crean para satisfacer unos requisitos especificados y con los suficientes detalles como para permitir su implementación física*. Según la Metodología Métrica V3, el objetivo del proceso de Diseño del Sistema de Información es la definición de la arquitectura del sistema y del entorno tecnológico que le va a dar soporte, junto con la especificación detallada de los componentes del sistema de información.

### **32.2.1 Diseño estructurado.**

El diseño estructurado se ocupa de la identificación, selección y organización de los módulos y sus relaciones. Se decide que componentes son necesarios y la interconexión entre los mismos para solucionar un problema bien especificado. Se comienza con la especificación resultante del proceso de análisis y se realiza una descomposición del sistema en módulos estructurados en jerarquías de tal modo que permitan la implementación de un sistema que no requiera elevados costes de mantenimiento.

Los principios que se consideran generalmente aceptados para las técnicas de diseño estructurado son los siguientes:

- **Modularización.** Se persigue que la Arquitectura técnica del sistema se fundamente en módulos de pequeño tamaño.
- **Independencia modular.** Cada uno de los módulos del sistema debe orientarse a que realice una función bien definida e independiente, lo cual permita “aislar” el origen de un problema o el área del sistema a modificar en caso preciso.
- **Modelización conceptual.** La estructuración del sistema en módulos no debe atender a un criterio exclusivamente técnico en sentido reducido, sino que debe reflejar conceptualmente la

estructura de la organización a la que sirve, de modo que se facilite su comprensión.

- **Principio de “caja negra”**. Una correcta definición de las interfaces permitirá ignorar la estructura interna de los módulos (es decir, considerarlos como una “caja negra”) y centrar la atención exclusivamente en los resultados que devuelve.

La técnica más importante dentro del Diseño Estructurado es el Diagrama de Estructuras (Structure chart), también denominada Diagrama de Estructura de Cuadros. Los **diagramas de estructura** son una herramienta para modelar los módulos de un sistema y sus relaciones y, junto con las especificaciones de funcionalidad de los módulos y las estructuras de datos, componen un diseño inicial que deberá ser analizado y mejorado. Permiten modelar un programa como una jerarquía de módulos. Cada nivel de la jerarquía representa una descomposición más detallada del módulo del nivel superior. La notación usada se compone básicamente de tres símbolos:

- **Módulos**: un conjunto de instrucciones que ejecutan alguna actividad y se presenta como una caja negra. Desde un punto de vista práctico, un módulo es una colección de instrucciones de un programa con cuatro características básicas: *Entradas y Salidas*, es decir, lo que un módulo recibe en una invocación y que necesita y lo que retorna como resultado; *Función* que son las actividades que un módulo hace con la entrada para producir la salida; la *Lógica Interna* por la cual se ejecuta la función; y *Estado Interno* que es su área de datos privada, datos para los cuales sólo el módulo hace referencia. Las entradas y salidas y las funciones proveen una visión externa del módulo, mientras que la lógica interna representa la visión interna del módulo. Un módulo se presenta como una caja, con un nombre en el interior que representa su función y las entradas y salidas representadas por pequeñas flechas que entran y salen del módulo.



- **Invocaciones:** los diagramas de estructura muestran las invocaciones que un módulo hace a otros módulos. Estas invocaciones son diseñadas como una flecha que sale del módulo llamador y apunta al módulo llamado. Los diagramas de estructura no tienen especificado el orden de invocación de los módulos invocados.
- **Cuplas:** Son las comunicaciones entre módulos. Existen varios tipos, basados en lo que ellas pueden producir en el módulo receptor:
  - o Cupla de Datos. Transporta datos a un módulo. Son también conocidos como parámetros de entrada del módulo.
  - o Cupla Modificada. Con una flecha doble (apuntando al modulo llamador y al módulo llamado) se especifica un argumento enviado a un módulo que deberá modificar su valor y del volver este valor modificado. Parámetros de entrada/salida
  - o Cupla de Resultados. Existen módulos que retornan valores sin la necesidad de que estén inicializados en el momento que se invocan.

Como dijimos al principio al hablar de los diagramas de estructuras, estos constituyen una aproximación inicial, que debe ser analizada y mejorada. Entre los criterios que se utilizan para mejorar el diseño tenemos:

- **Acoplamiento:** El acoplamiento entre módulos clasifica el grado de independencia entre pares de módulos. El objetivo es minimizar el acoplamiento, es decir, maximizar la independencia entre módulos.
- **Cohesión:** es la medida de intensidad de asociación funcional de los elementos de un módulo. El objetivo del diseño estructurado es obtener módulos altamente cohesivos, cuyos elementos estén fuerte y genuinamente relacionados unos con otros. Por otro lado, los elementos de un módulo no deberían estar fuertemente relacionados con elementos de otros módulos, porque eso llevaría a un fuerte acoplamiento entre ellos.

- **Descomposición** (Factoring): La descomposición es la separación de una función contenida en un módulo, para un nuevo módulo. Puede ser hecha para reducir el tamaño del módulo, hacer el sistema más claro, minimizar la duplicación de código, separar el trabajo de la coordinación de trabajos, o crear módulos más generales y fácilmente reutilizables.
- **Fan-Out:** El fan-out de un módulo es usado como una medida de complejidad. Es el número de subordinados inmediatos de un módulo (cantidad de módulos invocados).
- **Fan-In:** El fan-in de un módulo es usado como una medida de reusabilidad. Es el número de superiores inmediatos de un módulo (la cantidad de módulos que lo invocan).

### **32.2.2 Diseño de datos**

El diseño de datos (a veces llamado arquitectura de datos) crea un modelo de datos y/o información que se representa con un alto nivel de abstracción (la visión de datos del cliente/usuario). Este modelo de datos es entonces refinado en progresivas representaciones específicas de la implementación, que pueden ser procesadas por un sistema basado en computadora. La estructura de datos ha sido siempre una parte importante del diseño de software.

Los objetos de datos definidos durante el análisis de requisitos del software son modelados utilizando diagramas entidad-relación y el diccionario de datos. La actividad de diseño de datos traduce esos elementos del modelo de requisitos en estructuras de datos a nivel de los componentes del software y, cuando es necesario, a arquitectura de base de datos a nivel de aplicación.

El diseño de datos a nivel de componentes se centra en la representación de estructuras de datos a las que se accede directamente a través de uno o más componentes del software. En realidad, el diseño de datos empieza

durante la creación del modelo de análisis. Wasserman ha propuesto un conjunto de principios que pueden emplearse para especificar y diseñar dicha estructura de datos, que pueden integrarse en las fases de análisis y diseño:

- 1) Los principios del análisis sistemático aplicados a la función y al comportamiento deberían aplicarse también a los datos. Invertimos mucho tiempo y esfuerzo en obtener, revisar y especificar los requisitos funcionales y el diseño preliminar. Las representaciones de flujo de datos y contenido deberían desarrollarse y revisarse, las de objetos de datos deberían identificarse, se deberían estudiar las organizaciones alternativas de datos y debería evaluarse el impacto del modelado de datos sobre el diseño del software. Una organización de datos alternativa nos podría llevar a obtener mejores resultados.
- 2) Todas las estructuras de datos y las operaciones a llevar a cabo en cada una de ellas deberían estar claramente identificadas. El diseño de una estructura de datos eficaz debe tener en cuenta las operaciones que se van a llevar a cabo sobre dicha estructura de datos. La especificación de los tipos de datos abstractos puede simplificar considerablemente el diseño del software.
- 3) Se debería establecer un diccionario de datos y usarlo para definir el diseño de los datos y del programa. Un diccionario de datos representa explícitamente las relaciones entre los objetos de datos y las restricciones de los elementos de una estructura de datos. Los algoritmos que deben aprovecharse de estas relaciones específicas pueden definirse más fácilmente si existe una especificación de datos tipo diccionario.
- 4) Las decisiones de diseño de datos de bajo nivel deberían dejarse para el final del proceso de diseño. Se puede usar un proceso de refinamiento paso a paso para el diseño de datos. Es decir, la organización general de datos puede definirse durante el análisis de requisitos; refinarse durante los trabajos de diseño de datos y especificarse en detalle durante el diseño a nivel de componentes. El

enfoque descendente del diseño de datos proporciona ventajas análogas al enfoque descendente del diseño de software: se diseñan y evalúan primeramente los atributos estructurales principales de manera que se pueda establecer la arquitectura de los datos.

- 5) La representación de las estructuras de datos deberían conocerla sólo aquellos módulos que deban hacer uso directo de los datos contenidos dentro de la estructura. El concepto de ocultación de información y el concepto relacionado de acoplamiento proporcionan una importante visión de la calidad del diseño del software. Además, se debe separar la visión lógica de un objeto de datos de su visión física.
- 6) Debería desarrollarse una biblioteca de estructuras de datos útiles y de las operaciones que se les pueden aplicar. Las estructuras de datos y las operaciones deberían considerarse como recursos para el diseño del software. Las estructuras de datos pueden diseñarse para que se puedan reutilizar. Una biblioteca de plantillas de estructuras de datos (tipos abstractos de datos) puede reducir el esfuerzo del diseño y de la especificación de datos.
- 7) Un diseño del software y un lenguaje de programación debería soportar la especificación y realización de los tipos abstractos de datos. La implementación de una estructura de datos sofisticada puede hacerse excesivamente difícil si no existen los medios de especificación directa de la estructura en el lenguaje de programación escogido para dicha implementación.

### ***32.2.3 Diseño de la interfaz de usuario.***

Todo equipo, y dentro de éste toda aplicación software, necesita poder comunicarse con el usuario de alguna forma. La idea fundamental en el concepto de interfaz es la de mediación entre hombre y máquina. Podemos definir entonces la Interfaz de Usuario (en adelante IU), como el conjunto de elementos hardware y software de un equipo que presentan información

al usuario y le permiten interactuar con la información y con el equipo. Es decir, es la cara que va a presentar el equipo (y dentro de ésta cada aplicación software) al usuario para poder comunicarse con él.

Dentro de la interfaz de usuario tendremos dos elementos:

- **Parte Hardware.** Es lo referente a la parte física del ordenador, los dispositivos utilizados para introducir, procesar y entregar los datos, básicamente: teclado, ratón y monitor.
- **Parte Software.** Es en la que suele pensarse cuando se habla de Interfaz de Usuario. Es la relativa a presentación que muestra, normalmente a través del monitor, una aplicación software.

La evolución de las interfaces va en paralelo al sistema operativo, pudiendo tener los siguientes tipos de interfaces:

- **Interfaz de línea de comandos:** el usuario escribe órdenes usando un lenguaje formal con vocabulario y sintaxis propia. Se usa un teclado, típicamente, y las órdenes están encaminadas a realizar una acción y mostrar los resultados al usuario.
- **Interfaz a través de Menús:** Un Menú es una lista de opciones que se muestran en pantalla o en una ventana de la pantalla para que los usuarios elijan la opción que deseen. Tenemos distintos tipos: Menús de pantalla completa, menús de barra, en cascada, paleta de herramientas o menús contextuales.
- **Interfaces gráficos de usuario.** La idea básica o paradigma de las interfaces gráficas es: *Lo que ves es lo que puedes conseguir* (en inglés, la conocida frase: *What you see is what you get*, WYSIWYG) y esto se consigue fundamentalmente con la manipulación directa del Interfaz de usuario a través de iconos y gráficos en general. De hecho, la interfaz de usuario gráfica de un sistema junto con sus dispositivos de la entrada a veces es llamado: *look-and-feel* (mirar y sentir). Un GUI es una representación gráfica en la pantalla del ordenador de los programas, datos y objetos, así como de la interacción con ellos. Proporciona al usuario las herramientas para

realizar sus operaciones, más que una lista de las posibles operaciones. Aunque el uso de los GUI es más sencillo que el de las interfaces de línea de comandos, la introducción de instrucciones es más lenta, por lo que las GUI suelen tener la opción de emplear un sistema equivalente al de línea de instrucciones como alternativa rápida para los usuarios más expertos.

Existen tres puntos de vista distintos, también llamados modelos, en el diseño de IU:

- Modelo del **Usuario**: El usuario tiene su visión personal del sistema, y espera que éste se comporte de una cierta forma.
- Modelo del **Programador**: El programador no tiene que considerar la parte visual de interacción con el usuario, sino el funcionamiento interno de la aplicación. Los objetos que manipula el programador, son distintos de los que trata el usuario.
- Modelo del **Diseñador**: El diseñador debe tener un punto de vista que mezcle las necesidades, ideas y deseos del usuario con los materiales de que dispone el programador para diseñar un producto de software. Es un intermediario entre ambos. El modelo del diseñador describe los objetos que utiliza el usuario, su presentación al mismo y las técnicas de interacción para su manipulación. Consta de tres partes: **presentación**, **interacción** y **relaciones** entre los objetos.

Existen varios principios relevantes para el diseño e implementación de IU, sobre todo para los IU gráficas, siendo algunos de los más importantes son los siguientes

- **Anticipación**. Las aplicaciones deberían intentar anticiparse a las necesidades del usuario y no esperar a que el usuario tenga que buscar la información, recopilarla o invocar las herramientas que va a utilizar.

- **Autonomía.** El ordenador, la IU y el entorno de trabajo deben estar a disposición del usuario. Se debe dar al usuario el ambiente flexible para que pueda aprender rápidamente a usar la aplicación. Sin embargo, está comprobado que el entorno de trabajo debe tener ciertas cotas, es decir, ser explorable pero no azaroso.
- **Percepción del color.** Aunque se utilicen convenciones de color en la IU, se deberían usar otros mecanismos secundarios para proveer la información a aquellos usuarios con problemas en la visualización de colores
- **Valores por defecto.** Los valores por defecto deberían ser opciones inteligentes y sensatas. Además, los mismos tienen que ser fáciles de modificar.
- **Consistencia.** Para lograr una mayor consistencia en la IU se requiere profundizar en diferentes aspectos que están catalogados en niveles. Se realiza un ordenamiento de mayor a menor consistencia:
  - o **Interpretación del comportamiento del usuario:** La IU debe comprender el significado que le atribuye un usuario a cada requerimiento. Ejemplo: mantener el significado de los comandos abreviados (shortcutkeys) definidos por el usuario.
  - o **Estructuras invisibles:** se requiere una definición clara de las mismas, ya que sino el usuario nunca podría llegar a descubrir su uso. Ejemplo: ampliación de ventanas mediante la extensión de sus bordes.
  - o **Pequeñas estructuras visibles:** se puede establecer un conjunto de objetos visibles capaces de ser controlados por el usuario, que permitan ahorrar tiempo en la ejecución de tareas específicas. Ejemplo: icono y/o botón para impresión.
  - o **Una sola aplicación o servicio:** la IU permite visualizar a la aplicación o servicio utilizado como un componente único. Ejemplo: La IU despliega un único menú, pudiendo además acceder al mismo mediante comandos abreviados.

- o **Un conjunto de aplicaciones o servicios:** la IU visualiza a la aplicación o servicio utilizado como un conjunto de componentes. Ejemplo: La IU se presenta como un conjunto de barras de comandos desplegadas en diferentes lugares de la pantalla, pudiendo ser desactivadas en forma independiente.
- o **Consistencia del ambiente:** la IU se mantiene en concordancia con el ambiente de trabajo. Ejemplo: La IU utiliza objetos de control como menús, botones de comandos de manera análoga a otras IU que se usen en el ambiente de trabajo.
- o **Consistencia de la plataforma:** La IU es concordante con la plataforma. Ejemplo: La IU tiene un esquema basado en ventanas, el cual es acorde al manejo del sistema operativo utilizado.
- **Eficiencia del usuario.** Se debe considerar la productividad del usuario antes que la productividad de la máquina. Si el usuario debe esperar la respuesta del sistema por un período prolongado, estas pérdidas de tiempo se pueden convertir en pérdidas económicas para la organización. Los mensajes de ayuda deben ser sencillos y proveer respuestas a los problemas. Los menús y etiquetas de botones deberían tener las palabras claves del proceso.
- **Ley de Fitt.** Según dice la ley de Fitt: “El tiempo para alcanzar un objetivo es una función de la distancia y tamaño del objetivo”. Es por ello que es conveniente usar objetos grandes para las funciones importantes.
- **Interfaces explorables.** Siempre que sea posible se debe permitir que el usuario pueda salir ágilmente de la IU, dejando una marca del estado de avance de su trabajo, para que pueda continuarlo en otra oportunidad. Para aquellos usuarios que sean novatos en el uso de la aplicación, se deberá proveer de guías para realizar tareas que no sean habituales. Siempre se debe contar con un comando



“deshacer”. Este suprimirá la necesidad de tener que contar con diálogos de confirmación para cada acción que realice en sistema. El usuario debe sentirse seguro de poder salir del sistema cuando lo desee. La IU debe tener un objeto fácil de accionar con el cual poder finalizar la aplicación.

- **Uso de metáforas.** Las buenas metáforas crean figuras mentales fáciles de recordar. La IU puede contener objetos asociados al modelo conceptual en forma visual, con sonido u otra característica perceptible por el usuario que ayude a simplificar el uso del sistema.
- **Curva de aprendizaje.** El aprendizaje de un producto y su usabilidad no son mutuamente excluyentes. El ideal es que la curva de aprendizaje sea nula, y que el usuario principiante pueda alcanzar el dominio total de la aplicación sin esfuerzo.
- **Reducción de latencia.** Siempre que sea posible, el uso de tramas (multi-threading) permite colocar la latencia en segundo plano (background). Las técnicas de trabajo multitarea posibilitan el trabajo ininterrumpido del usuario, realizando las tareas de transmisión y computación de datos en segundo plano.
- **Protección del Trabajo.** Se debe poder asegurar que el usuario nunca pierda su trabajo, ya sea por error de su parte, problemas de transmisión de datos, de energía, o alguna otra razón inevitable.
- **Legibilidad.** Para que la IU favorezca la usabilidad del sistema de software, la información que se exhiba en ella debe ser fácil de ubicar y leer. Para lograr obtener este resultado se deben tener en cuenta algunas premisas como: el texto que aparezca en la IU debería tener un alto contraste, se debe utilizar combinaciones de colores como el texto en negro sobre fondo blanco o amarillo suave. El tamaño de las fuentes tiene que ser lo suficientemente grande como para poder ser leído en monitores estándar. Es importante hacer clara la presentación visual (colocación/agrupación de objetos, evitar la presentación de excesiva información).

#### **32.2.4 Diseño procedimental.**

El **diseño procedimental** se realiza después de que se ha establecido la estructura del programa y de los datos. Transforma los elementos estructurales en una descripción de los elementos del software. Define los algoritmos de procesamiento necesarios. Debe especificar los detalles de los procedimientos sin ambigüedad y además debe facilitar la codificación, de forma que el código se obtenga de forma natural a partir del diseño.

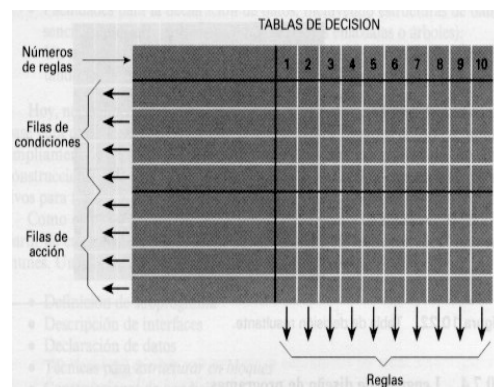
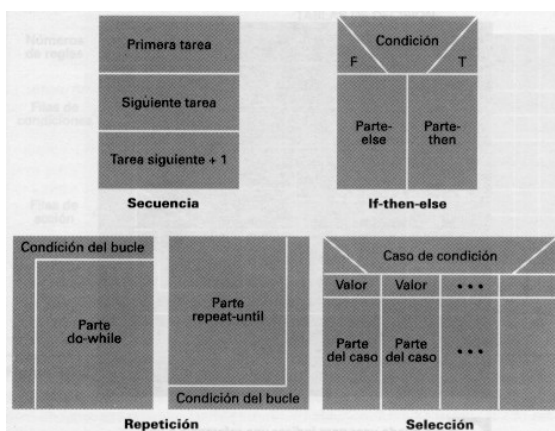
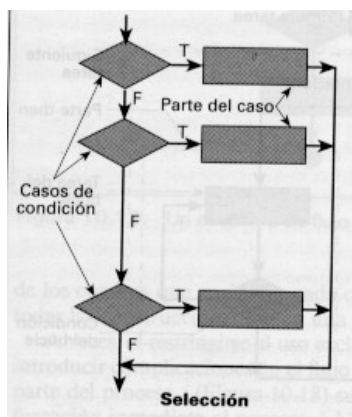
A finales de los 60 se propuso la utilización de un conjunto de construcciones lógicas con las que podía formarse cualquier programa. Estas construcciones serían la **secuencia**, la **condición** y la **repetición**, y cualquier programa podría construirse utilizando solo este tipo de construcciones. Estas son uno de los fundamentos de la programación estructurada. Las construcciones estructuradas se propusieron para limitar el diseño procedimental del software a un número pequeño de operaciones predecibles, lo que facilita la legibilidad, prueba y mantenimiento del software.

Para la especificación procedimental que define los algoritmos se podría utilizar el lenguaje natural, pero debido a la cantidad de ambigüedades que este lenguaje acarrea, es necesario utilizar una forma más restringida de representación. Y entre estas formas de representación tenemos:

- **Diagrama de flujo:** Es la representación gráfica que más se utiliza en el diseño procedimental. Para representar un paso de procesamiento se utiliza un cuadro, para representar una condición se utiliza un rombo, y para representar el flujo de control se utilizan flechas.
- **Diagrama de cajas (N-S):** Esta notación surgió del deseo de desarrollar una representación para el diseño procedimental que no permitiera la violación de construcciones estructuradas. Estos

diagramas fueron desarrollados por Nassi y Schneiderman y perfeccionados por Chapin, y tienen las características siguientes:

- o El ámbito funcional está bien definido y es claramente visible.
  - o La transferencia de control arbitraria es imposible.
  - o Es fácil determinar el ámbito de los datos locales y globales.
  - o La recursividad es fácil de representar
- **Tablas de decisión.** Constituyen una notación que traduce las acciones y condiciones a una forma tabular. Los pasos para la creación de una tabla de decisión son los siguientes:
    - o Listar todas las acciones que se pueden asociar a un módulo.
    - o Listar todas las condiciones necesarias para la ejecución de un procedimiento.
    - o Asociar conjuntos de condiciones específicas a acciones específicas, eliminando combinaciones imposibles. Desarrollar las posibles combinaciones de condiciones.
    - o Definir reglas indicando las acciones que ocurren para una serie de condiciones.
  - **Lenguaje de diseño de programas.** Un lenguaje de diseño de programas (LDP), también conocido como lenguaje estructurado o pseudocódigo es un lenguaje que utiliza el vocabulario de un lenguaje y la sintaxis de otro. Independientemente de su origen, un LDP tiene que tener las características siguientes:
    - o Una sintaxis fija de palabras clave que permitan construir todas las construcciones estructuradas, declarar datos y establecer características de modularidad.
    - o Una sintaxis libre en lenguaje natural para describir las características de procesamiento.
    - o Facilidades para la declaración de datos, incluyendo estructuras de datos simples y complejas.
    - o Un mecanismo de definición de subprogramas y de llamada a éstos.



**32.2.5**

**Diseño**

### ***orientado a objetos.***

El diseño orientado a objetos transforma el modelo de análisis creado usando análisis orientado a objetos, en un modelo de diseño que sirve para la construcción de software que podrá ser implementado en un lenguaje de programación. Se deben identificar los objetos pertinentes, clasificarlos dentro de las clases en la granularidad correcta, definir interfaces de clases y jerarquías de herencia y establecer relaciones clave entre ellos, de tal forma que se obtenga la solución al problema del usuario. El diseño debe ser específico al problema que se tiene entre manos, pero suficientemente general para adaptarse a problemas y requerimientos futuros.

Los métodos estructurados y sus correspondientes notaciones definen un sistema como un conjunto secuencial de módulos interdependientes que comparten datos. En cambio, los métodos orientados a objetos definen un conjunto de módulos independientes relacionados y con visibilidad limitada entre ellos.

Esencialmente, el DOO consta de los tres siguientes pasos:

- **Identificación y definición de objetos y clases.** El principal problema del desarrollo de un sistema orientado a objetos es encontrar los objetos en la fase de AOO y DOO. Un posible método para encontrarlos es el método gramatical propuesto por Booch que

analiza la descripción textual del sistema y propone a los sustantivos como posibles identificadores de las clases y los verbos como posibles métodos de las clases. La lista resultante de clases (nombres) y métodos (verbos) se utilizará para comenzar el proceso de diseño. Una vez identificados los objetos, se detallan los atributos de cada uno. Con el **método de Booch** para encontrar clases es difícil conseguir un resultado de alta calidad, pues la abstracción de clases que consigamos, depende de una estructuración inteligente de la descripción del problema en elementos independientes e intuitivamente correctos. Otro modelo que utiliza el método gramatical para la identificación de clases y objetos es el **modelo CRC** (Clases - Responsabilidades - Colaboraciones). Un modelo CRC es una colección de tarjetas que representan clases. Las tarjetas están divididas en tres secciones. A lo largo de la cabecera de la tarjeta se escribe el nombre de la clase. En el cuerpo se listan las responsabilidades de la clase a la izquierda y a la derecha los colaboradores. Las **responsabilidades** estarían formadas por los atributos y operaciones de las clases. Las clases cumplen con sus responsabilidades de dos formas: O bien una clase puede usar sus propias operaciones para manipular sus propios atributos, cumpliendo por lo tanto con una responsabilidad particular, o puede colaborar con otras clases.

Se han definido una serie de directrices que facilitan la identificación y definición de clases y métodos:

- o Modelar con clases las entidades que ocurren de forma natural en el problema.
- o Diseñar métodos con una sola finalidad.
- o Diseñar un método nuevo antes de ampliar uno existente.
- o Evitar métodos extensos.
- o Guardar como instancia los datos necesitados por más de un método o por una subclase.

- o El diseñador debe trabajar para obtener una biblioteca de clases, y no para él mismo, ni para desarrollar el sistema actual.

Además, para evitar la creación de clases innecesarias o declaración de clases que no lo sean, una clase debe ofrecer una serie de servicios a objetos de un tipo determinado. Una clase se debería crear cuando:

- o La nueva clase represente una abstracción significativa del problema
  - o Es probable que los servicios que proporciona sean utilizados por otras clases.
  - o Su comportamiento sea complejo.
  - o Si se representara como un método de otra clase, pocos usuarios de ésta lo invocarían.
- **Organización de relaciones entre clases.** Una vez identificadas las clases deberíamos encontrar las relaciones entre estas. Para ello podríamos usar distintas alternativas, como el modelo Clases-Responsabilidades-Colaboraciones (CRC) visto anteriormente. Cada tarjeta del modelo CRC contiene una clase con una lista de responsabilidades. El siguiente paso es definir aquellas clases colaboradoras que ayudan en la realización de cada responsabilidad. Esto establece las **conexiones o relaciones** entre clases. Las **relaciones** pueden derivarse a partir del examen de los verbos en la descripción del problema. Una vez conectadas las clases con sus colaboradores, etiquetamos cada una de estas conexiones, les añadimos una dirección, en función de qué clase llama a qué otra y por último se evalúa cada extremo de la conexión para determinar la cardinalidad. Este modelo de clases conectadas da lugar al **modelo Objeto-Relación**.
  - **Extracción de estructuras en una jerarquía de clases.** Una vez definidos los objetos y sus relaciones, el paso siguiente consiste en

observar características comunes para crear abstracciones a nivel de clase, pero no existe ninguna metodología formal para la realización de estas abstracciones. Uno de los objetos de un diseño orientado a objetos es guardar propiedades y métodos de clases en el nivel de abstracción más alto posible, de forma que los compartan la mayoría de clases y se fomente la reutilización. Para este paso se emplean los diagramas de generalización o herencia.

Aunque desde los inicios del desarrollo de software orientado a objetos han existido múltiples metodologías, como por ejemplo Object-Oriented Design (OOD, Booch), Object-Modeling Technique (OMT, Rumbaugh), Object Oriented Analysis (OOA, Coad/Yourdon), Hierarchical Object Oriented Design (HOOD, ESA), Object Oriented Structured Design (OOSD, Wasserman), etc, el Lenguaje Unificado de modelado (UML) puso fin a la guerra de metodologías. Actualmente en la industria del desarrollo de software tenemos al UML como un estándar para el modelado de sistemas orientados a objetos que es usado tanto para la fase de análisis como desarrollo.

Durante el proceso de DOO, un ingeniero del software debe observar cada oportunidad en la que pueda reutilizar patrones de diseño existentes (cuando cumplen las necesidades del diseño), en vez de crear otros nuevos. Los patrones de diseño son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces. Un **patrón de diseño** es una solución a un problema de diseño. Para que una solución sea considerada un patrón debe poseer ciertas características. Una de ellas es que debe haber comprobado su efectividad resolviendo problemas similares en ocasiones anteriores. Otra es que debe ser reutilizable, lo que significa que es aplicable a diferentes problemas de diseño en distintas circunstancias.

## **Bibliografía**

- Herramientas y Metodologías de Análisis y Diseño Estructurado. Apunte de la Cátedra Metodologías de Desarrollo de Software I. Claudia Marcos - Edgardo Belloni
- Ingeniería del Software. Un enfoque práctico. ROGER S. PRESSMAN. Ed. McGraw Hill
- Aprendiendo UML en 24 horas. Joseph Schmuller
- Métrica 3 - Técnicas y Prácticas. Ministerio de Administraciones Públicas
- Curso de OO dirigido por la introducción a la ambigüedad. Anexo 1 : UML  
[http://is.ls.fi.upm.es/docencia/proyecto/docs/curso/12Anexo\\_1\\_UML.doc](http://is.ls.fi.upm.es/docencia/proyecto/docs/curso/12Anexo_1_UML.doc)
- Introducción a la programación I. Diagramas de estructuras. <http://www.exa.unicen.edu.ar/catedras/prog1/introprog1/sites/default/files/ApuntesDiagramaEstructura.pdf>
- Fundamentos de diseño del software y Diseño orientado a los objetos Asignatura Laboratorio de proyectos. Univ. de Almería. <http://indalog.ual.es/mtorres/LP/FundamentosDiseno.pdf>  
<http://indalog.ual.es/mtorres/LP/DOO.pdf>
- [http://es.wikipedia.org/wiki/Patrón\\_de\\_diseño](http://es.wikipedia.org/wiki/Patrón_de_diseño)

**Autor: Hernán Vila Pérez**

**Jefe del Servicio de Informática. Instituto Galego de Vivenda e Solo**

**Vicepresidente del CPEIG**



# **33. INGENIERÍA DE REQUISITOS. VERIFICACIÓN. VALIDACIÓN. ESPECIFICACIÓN DE REQUISITOS. GESTIÓN DE REQUISITOS.**

### **33- Ingeniería de requisitos. Verificación. Validación. Especificación de requisitos. Gestión de requisitos.**

#### **INDICE**

- 33.1 Ingeniería de requisitos
  - 33.1.1 Tipos de requisitos
- 33.2 Identificación de los requisitos del software
  - 33.2.1 Entrevistas
  - 33.2.2 JAD (Joint Application Design)
  - 33.2.3 Prototipos
  - 33.2.4 Análisis de factores críticos de éxito
  - 33.2.5 Brainstorming
  - 33.2.6 Escenarios y casos de uso
  - 33.2.7 Etnografía
- 33.3 Verificación - Validación
- 33.4 Especificación de Requisitos
  - 33.4.1 IEEE/ANSI 830-1998
- 33.5 Gestión de requisitos
  - 33.5.1 Planificación de la gestión requisitos.
  - 33.5.2 Gestión del cambio.

#### **35.1 Ingeniería de requisitos**

En la ingeniería de software, la Ingeniería de requisitos comprende todas las tareas relacionadas con la determinación de las necesidades o de las condiciones a satisfacer para un software nuevo o modificado, tomando en cuenta los diversos requisitos de los inversores. Así, los requisitos se generan a partir de la interacción entre los usuarios y los ingenieros del

software, representando las características del sistema a construir, es decir, las necesidades de los usuarios.

Hay múltiples definiciones del término requisito, algunas de ellas son:

- Según IEEE un **requisito** es:
  - o Una condición o capacidad necesitada por un usuario para resolver un problema o alcanzar un objetivo.
  - o Una condición o capacidad que debe cumplir o poseer un sistema o un componente del mismo para satisfacer un contrato, un estándar, una especificación, u otro documento impuesto de una manera formal.
  - o Una representación documentada de una condición o capacidad tal como las expresadas en los dos puntos anteriores.
- Desde el punto de vista del usuario, los requisitos son las condiciones o capacidades necesarias para que un usuario pueda resolver un problema o alcanzar un objetivo
- Desde el punto de vista del equipo de desarrollo, los requisitos son las capacidades o condiciones que debe reunir un sistema para satisfacer un contrato, estándar o cualquier otro documento impuesto formalmente.

En resumen, los requisitos son las características que debe cumplir el sistema para que cubra las necesidades de los usuarios. Muchas veces se habla de requerimientos en vez de requisitos. Esto se debe a una mala traducción del inglés. La palabra *requirement* debe ser traducida como requisito, mientras que requerimiento se traduce al inglés como *request*.

Sommerville divide el proceso de ingeniería de requisitos en cuatro subprocesos que son la evaluación de si el sistema es útil para el negocio (**estudio de viabilidad**); el **descubrimiento de requisitos** (obtención y análisis); la transformación de estos requisitos en formularios estándar

(**especificación**), y la verificación de que los requisitos realmente definen el sistema que quiere el cliente (**validación**).

Para Thayer, *“La ingeniería de requisitos proporciona el mecanismo apropiado para entender lo que el cliente quiere, analizar las necesidades, evaluar la factibilidad, negociar una solución razonable, especificar la solución sin ambigüedades, validar la especificación, y administrar los requisitos conforme se transforman en un sistema operacional”* y establece las siguientes fases para el proceso:

- **Inicio.** Se establece una comprensión básica del problema por parte de los analistas.
- **Obtención.** Se obtienen los requisitos del software mediante la interacción entre los analistas y el cliente.
- **Elaboración.** Se refina la información obtenida en el paso anterior y se enfoca a la construcción de un modelo de análisis que represente al sistema a construir.
- **Negociación.** Hay requisitos que no son implementables o son difíciles de hacerse, por esta razón los analistas negocian estos requisitos para llegar a un entendimiento y lograr un sistema factible de desarrollarse en un plazo y coste.
- **Especificación.** Se confecciona un conjunto de documentos (descripciones en lenguaje natural, diagramas, etc.) que definan lo que el sistema debe hacer.
- **Validación.** Se examina la especificación para asegurar que todos los requisitos software se han establecido de una manera precisa, que no hay inconsistencias, omisiones ni errores, además de cumplirse los estándares de calidad establecidos para el proyecto.
- **Gestión de Requisitos.** Esta actividad permite tratar con el inevitable problema de los cambios de especificaciones, identificando, controlando y determinando el impacto del cambio de requisitos en el resto.

Otros autores modifican el número de etapas y las dividen en:

- **Educción de Requisitos.** En ella los analistas obtienen las necesidades del cliente a partir de todas las fuentes de información que tienen disponibles (documentación, entrevistas, estudio de los procesos de la organización, etc.). Términos equivalentes usados por los ingenieros de software para esta actividad son Extracción de Requisitos, Identificación de Requisitos, Determinación de Requisitos, etc. Vemos que estas se corresponderían con las dos primeras del modelo anterior.
- **Análisis de Requisitos:** se procede a trabajar sobre los requisitos educidos en el paso anterior. Se estudian los requisitos en busca de conflictos e incoherencias, implicaciones, información no obtenida y aspectos no resueltos. Después se clasifican, evalúa su viabilidad y se integran los nuevos requisitos con los ya existentes. El objetivo final es lograr una lista de requisitos que defina las necesidades del cliente. Corresponderían con las fases de Elaboración y Negociación del modelo anterior.
- **Representación de los Requisitos.** Actividad en la que se representan los requisitos de una o más formas, utilizando para ello diferentes técnicas como por ejemplo el lenguaje formal, el lenguaje natural, representaciones gráficas, etc. Para la representación existen múltiples técnicas, las más usadas son, entre otras, los Diagramas de Flujo de Datos, Modelo Entidad Relación, Casos de Uso o Diagramas de Clases. Una vez que están los requisitos representados, es necesario que se reúnan los diversos participantes en el desarrollo para revisarlos y aprobarlos. El producto final con el que culmina esta fase es la Especificación de Requisitos Software, en donde está descrito con exactitud todo lo que el sistema debe hacer. Corresponde a las fases de Especificación y Validación del modelo anterior.

- **Validación de Requisitos** Se procede a definir una serie de criterios y técnicas que permitirán, cuando el sistema esté construido, comprobar que éste cumple los requisitos.

Independientemente del modelo que usemos, lo importante es tener claro que el objetivo de la ingeniería de requisitos es determinar con claridad y precisión qué es lo que hay que hacer y para ello será necesario identificar los requisitos clave.

Para pensar en la importancia de los requisitos en la Ingeniería del Software, algunos estudios reflejan que el 45 por 100 de los errores tienen su origen en los requisitos y en el diseño preliminar, otros señalan que el 56 por 100 de los errores que tienen lugar en un proyecto software se deben a una mala especificación de requisitos, etc.

Los factores principales que conducen al fracaso en los proyectos software son: la falta de comunicación con los usuarios, los requisitos incompletos y los cambios en los requisitos; y la evidencia demuestra que los requisitos contienen demasiados errores, que muchos de estos errores no se detectan al principio, pero podrían ser detectados, y que no detectar estos errores incrementa grandemente los costes del proyecto y su duración. La consecuencia es que el sistema no satisfará a los usuarios, se producirán desacuerdos entre usuarios y desarrolladores, y se gastará tiempo y dinero en construir un sistema equivocado.

Quizá no sea posible elaborar los requisitos con absoluta perfección, pero se debe intentar minimizar el impacto de los errores en los requisitos y organizar mejor las tareas relacionadas con ellos. Para remediar esta situación es para lo que surge la **Ingeniería de Requisitos**.

### *33.1 Tipos de requisitos*

La mayoría de los autores distinguen entre:

- **Requisitos funcionales:** Son declaraciones de los servicios que debe proporcionar el sistema, de la manera en que éste debe reaccionar a entradas particulares y de cómo se debe comportar en situaciones concretas. En algunos casos, los requisitos funcionales de los sistemas también pueden declarar explícitamente lo que el sistema no debe hacer. Los requisitos funcionales de un sistema describen lo que el sistema debe hacer.
- **Requisitos no funcionales:** Son restricciones de los servicios o funciones ofrecidos por el sistema. Incluyen restricciones de tiempo, sobre el proceso de desarrollo y estándares. Como su nombre sugiere, no se refieren directamente a las funciones específicas que proporciona el sistema, sino a las propiedades de éste como la fiabilidad, el tiempo de respuesta y la capacidad de almacenamiento. Pueden venir de las características requeridas del software (requisitos del producto), de la organización que desarrolla el software (requisitos organizacionales) o de fuentes externas.
  - o **Requisitos del producto.** Estos requisitos especifican el comportamiento del producto. Algunos ejemplos son los requisitos de rendimiento en la rapidez de ejecución del sistema y cuánta memoria se requiere; los requisitos de fiabilidad que fijan la tasa de fallos para que el sistema sea aceptable; los requisitos de portabilidad, y los requisitos de usabilidad.
  - o **Requisitos organizacionales.** Estos requisitos se derivan de políticas y procedimientos existentes en la organización del cliente y en la del desarrollador. Algunos ejemplos son los estándares en los procesos que deben utilizarse; los requisitos de implementación, como los lenguajes de programación o el método de diseño a utilizar, y los requisitos de entrega que

especifican cuándo se entregará el producto y su documentación.

- o **Requisitos externos.** Este gran apartado incluye todos los requisitos que se derivan de los factores externos al sistema y de su proceso de desarrollo. Éstos pueden incluir los requisitos de interoperabilidad que definen la manera en que el sistema interactúa con sistemas de otras organizaciones; los requisitos legislativos que deben seguirse para asegurar que el sistema funcione dentro de la ley, y los requisitos éticos. Estos últimos son puestos en un sistema para asegurar que será aceptado por sus usuarios y por el público en general.

Sommerville distingue entre **requisitos del usuario**, que son declaraciones, en lenguaje natural y en diagramas, de los servicios que se espera que el sistema proporcione y de las restricciones bajo las cuales debe funcionar, y **requisitos del sistema**, que establecen con detalle las funciones, servicios y restricciones operativas del sistema. Estos diferentes niveles de especificación serían de utilidad debido a que comunican la información del sistema a diferentes tipos de lectores.

### **33.2 Identificación de los requisitos del software**

La educación, identificación o determinación de requisitos es el paso durante el cual los requisitos del software son obtenidos de fuentes tales como: la gente implicada (usuarios, clientes, expertos en la materia, etc.), las necesidades que ha de satisfacer el sistema, el entorno físico que rodea al sistema, el entorno organizacional, etc.

Los problemas de la obtención de requisitos se pueden agrupar en tres categorías:

- 1- Problemas de alcance, ya que los requisitos pueden implicar demasiada o muy poca información.



- 2- Problemas de comprensión, como consecuencia de una pobre comunicación entre usuario y analista. En este caso los requisitos obtenidos son ambiguos, incompletos, inconsistentes e incorrectos, porque no responden a las verdaderas necesidades de los usuarios.
- 3- Problemas de volatilidad, ya que los requisitos evolucionan con el tiempo. En efecto, a medida que avanza el desarrollo del sistema, las necesidades del usuario pueden madurar a causa del conocimiento adicional fruto del desarrollo, o de necesidades del entorno o de la organización no previstas.

La solución al primer problema pasa por determinar claramente el contexto del sistema, es decir, los límites y objetivos del mismo. Si no se contempla el contexto donde va a funcionar el sistema, se puede llegar a requisitos incompletos, no verificables, innecesarios y no utilizables. La solución del segundo es que exista una buena comunicación entre usuarios, desarrolladores y clientes, a fin de que los requisitos se puedan escribir de manera que permitan, tanto que el desarrollador pueda distinguir si dichos requisitos se pueden implementar, como que el personal de control pueda comprobar si la implementación cumple con los requisitos. Por último la solución al tercer problema es incorporar estos cambios a los requisitos originales, pues si no, éstos serán incompletos e inconsistentes con la nueva situación.

Son numerosas las estrategias y técnicas que se han desarrollado para la obtención de los requisitos. Las más importantes las veremos a continuación.

### *33.2.1 Entrevista*

Se entiende por entrevista el encuentro que se realiza “cara a cara” entre un usuario y la persona responsable de obtener la información. Para realizar la entrevista solo es necesario designar a las personas que deben

participar en ella y determinar el lugar en el que poder llevarla a cabo. Es importante identificar a qué tipo de perfil va dirigida la entrevista, a quiénes se va a entrevistar y cuál es el momento más oportuno, con el fin de evitar situaciones embarazosas y conseguir que la entrevista sea eficaz y productiva.

Como paso previo a la realización de la entrevista se deben tener en cuenta una serie de reglas generales o directrices básicas:

- Desarrollar un plan global de la entrevista.
- Asegurarse de que se cuenta con la aprobación para hablar con los usuarios.
- Preparar la entrevista previamente.
- Realizar la entrevista.
- Consolidar el resultado de la entrevista.

Además, es conveniente planificar las entrevistas estudiando la secuencia en que se van a llevar a cabo, en función de los distintos perfiles implicados y las relaciones existentes entre los entrevistados. Según la información a obtener y dependiendo de las distintas fuentes que pueden proporcionarla, puede ser necesario realizar una entrevista conjunta con varias personas. Durante la preparación de la entrevista es imprescindible remitir al usuario un guión previo sobre los puntos a tratar, para que pueda estudiarlo con tiempo y solicitar la información que estime conveniente para la entrevista. Se debe pensar bien el tipo de guión, según el perfil y las responsabilidades del entrevistado y su extensión, de forma que se pueda conseguir la suficiente información, sin provocar rechazo en el entrevistado. Si se considera apropiado se pueden utilizar herramientas automatizadas.

Una vez que se dispone de la aprobación para hablar con los usuarios, se hace la convocatoria de la entrevista enviando la información oportuna y fijando los objetivos, el método de trabajo que se va a seguir y el tiempo del que se dispone.

Para realizar la entrevista, es importante hacer un resumen general de los temas a tratar, utilizar un estilo apropiado y crear desde su inicio un clima de confianza entre los asistentes. Es posible que el entrevistado se resista a aportar información, siendo útil en estos casos utilizar técnicas específicas de comunicación.

Antes de finalizar la entrevista es importante que el entrevistador sintetice las conclusiones y compruebe que todos los asistentes están de acuerdo, dejando siempre abierta la posibilidad de volver a contactar para aclarar temas que surjan al estudiar la información recopilada.

Finalmente, el responsable depura y consolida el resultado de las entrevistas, elaborando un informe de conclusiones. En algunos casos puede ser conveniente elaborar un acta que refleje estas conclusiones y remitirla a los entrevistados con el objetivo de asegurar que se han comprendido bien las especificaciones dadas.

### *33.2.2 JAD (Joid Application Design)*

Las características de una sesión de trabajo tipo JAD se pueden resumir en los siguientes puntos:

- Se establece un equipo de trabajo cuyos componentes y responsabilidades están perfectamente identificados y su fin es conseguir el consenso entre las necesidades de los usuarios y los servicios del sistema en producción.
- Se llevan a cabo pocas reuniones, de larga duración y muy bien preparadas.
- Durante la propia sesión se elaboran los modelos empleando diagramas fáciles de entender y mantener, directamente sobre herramientas CASE.

- Al finalizar la sesión se obtienen un conjunto de modelos que deberán ser aprobados por los participantes.

Es importante definir claramente el perfil y las responsabilidades de los participantes de una sesión JAD. Se pueden distinguir los siguientes perfiles:

- Moderador (líder) con amplios conocimientos de la metodología de trabajo, dinámica de grupos, psicología del comportamiento, así como de los procesos de la organización objeto del estudio.
- Promotor, persona que ha impulsado el desarrollo.
- Jefe de proyecto, responsable de la implantación del proyecto.
- Especialista en modelización, responsable de la elaboración de los modelos en el transcurso de la sesión.
- Desarrolladores, aseguran que los modelos son correctos y responden a los requisitos especificados.
- Usuarios, responsables de definir los requisitos del sistema y validarlos.

Para llevar a cabo una sesión JAD, es necesario realizar una serie de actividades antes de su inicio, durante el desarrollo y después de su finalización. Estas actividades se detallan a continuación:

- **Inicio:** se define el ámbito y la estructura del proyecto, los productos a obtener, se prepara el material necesario para la sesión, se determina el lugar donde se van a llevar a cabo, se seleccionan los participantes y se sugiere una agenda de trabajo.
- **Desarrollo:** se identifican las salidas del proyecto y se debe conseguir el consenso entre los participantes de modo que se materialice en los modelos.
- **Finalización:** se valida la información de la sesión y se generan los productos de la metodología de trabajo propuesta. Si fuera necesario se integran los productos de salida.

### *33.2.3 Prototipado*

Los prototipos son sistemas software que sólo implementan una parte del sistema. Normalmente, los prototipos se emplean para obtener requisitos del usuario cuando éstos no están completamente claros. Es mucho más fácil que el usuario entienda y apruebe un sistema, que aunque es reducido pero también es operativo y puede interaccionar con él, a revisar una larga lista de texto con los requisitos que describen dicho sistema. Al fin y al cabo, como se suele decir, el prototipo permite salvar la situación siguiente: “No se lo que quiero, pero lo sabré cuando lo vea”. Y en efecto, será mucho más fácil para un usuario saber si lo que quiere es lo que tiene delante y que puede interaccionar con él, que si es un montón de hojas con todos los requisitos escritos uno a uno.

La aplicación de la técnica del prototipado consta de los siguientes pasos:

- 1- Estudio preliminar de los requisitos del usuario.
- 2- Proceso iterativo consistente en:
  - a. Construir un prototipo.
  - b. Evaluarlo con los usuarios.
  - c. Formular nuevos requisitos.
  - d. Desechar el prototipo.

Los aspectos clave en el diseño de prototipos son: la identificación de los usuarios a los que va dirigido, teniendo en cuenta que debe responder a diferentes individualidades, con distintos conocimientos y habilidades, qué funciones tienen asignadas, y qué tipo de información requerirán para llevar a cabo dichas funciones.

### *33.2.4 Análisis de factores críticos de éxito.*

Esta técnica consistente en identificar y concentrarse en un pequeño conjunto de factores críticos de los que depende el éxito y efectividad del sistema. Aunque la identificación de los factores críticos correctos es, a veces, una labor compleja, esta técnica es bastante útil cuando el sistema es técnicamente complejo.

### *33.2.5 Brainstorming*

Esta técnica se usa principalmente para la generación de ideas en los casos en los que la generación de éstas no es obvia. Se trata de una técnica sencilla en la que se reúnen en grupo de 4 a 10 personas para generar ideas, sin restricciones, en un ambiente libre de críticas. Uno de los principales puntos fuertes de la técnica es que ideas que en un principio pueden ser descabelladas tienen cabida. Primero se proporcionan las ideas y en una segunda vuelta se refinan. El Brainstorming permite aportar a la obtención de requisitos los siguientes aspectos:

- **Genera múltiples puntos de vista de un problema.** Cada uno da su visión del problema, que no tiene porqué ser la misma. Una vez vistos todos los puntos de vista se puede atacar el problema de una forma más efectiva.
- **Formular un problema de distintas formas.** Cada participante puede ver el problema desde una óptica distinta, por lo que a la hora de formularlo lo hará de una forma distinta. Estas diferencias pueden aportar gran valor a la hora del estudio del problema, identificando los puntos clave del problema de cada participante.

La técnica de Brainstorming, para que sea efectiva, debe ser previamente preparada. Las fases por las que pasa la ejecución de la técnica son:

- 1- **Fase de Preparación:** Se identifican los participantes de la sesión (clientes, usuarios, analistas, etc.), se designa un líder que lleva la sesión, se planifica la sesión y se busca una sala adecuada.

- 2- **Fase de Generación:** Se comienza exponiendo las ideas de forma libre, por turnos o espontáneamente. Las ideas se van apuntando en una pizarra que todos puedan ver.
- 3- **Fase de Consolidación:** Se revisan las ideas obtenidas en el paso anterior, se clarifican si no están claras, se rescriben si es necesario, se descartan las que no son utilizables, se discuten las restantes y se priorizan.

### *33.2.6 Escenarios y Casos de uso*

Normalmente, las personas encuentran más fácil dar ejemplos de la vida real que descripciones abstractas. Los **escenarios** pueden ser especialmente útiles para agregar detalle a un esbozo de la descripción de requisitos. Son descripciones de ejemplos de las sesiones de interacción. Cada escenario abarca una o más posibles interacciones. El escenario comienza con un esbozo de la interacción y, durante la obtención se agregan detalles para crear una descripción completa de esta interacción. De forma general, un escenario puede incluir:

- 1- Una descripción de lo que esperan el sistema y los usuarios cuando el escenario comienza.
- 2- Una descripción del flujo normal de eventos en el escenario.
- 3- Una descripción de lo que puede ir mal y cómo manejarlo.
- 4- Información de otras actividades que se podrían llevar a cabo al mismo tiempo.
- 5- Una descripción del estado del sistema cuando el escenario termina.

Los escenarios se pueden redactar como texto, complementados por diagramas, fotografías de las pantallas, etcétera. De forma alternativa, se puede adoptar un enfoque más estructurado, como los escenarios de evento o los casos de uso.

Los casos de uso son una técnica que se basa en escenarios para la obtención de requisitos. En su forma más simple, un caso de uso identifica el tipo de interacción y los actores involucrados. Los actores en el proceso se representan como figuras delineadas, y cada clase de interacción se representa como una elipse con su nombre. El conjunto de casos de uso representa todas las posibles interacciones a representar en los requisitos del sistema. Actualmente se han convertido en una característica fundamental de la notación de UML, que se utiliza para describir modelos de sistemas orientados a objetos.

Los escenarios y los casos de uso son técnicas eficaces para obtener requisitos para los puntos de vista de los interactuadores, donde cada tipo de interacción se puede representar como un caso de uso. Sin embargo, debido a que se centran en las interacciones, no son tan eficaces para obtener restricciones y requisitos de negocio y no funcionales de alto nivel de puntos de vista indirectos o para descubrir requisitos del dominio.

### *33.2.7 Observación directa e investigación contextual (Etnografía)*

La **etnografía** es una técnica de observación que se puede utilizar para entender los requisitos sociales y organizacionales. Un analista se sumerge por sí solo en el entorno laboral donde se utilizará el sistema. Observa el trabajo diario y anota las tareas reales en las que los participantes están involucrados. El valor de la etnografía es que ayuda a los analistas a descubrir los requisitos implícitos que reflejan los procesos reales más que los formales en los que la gente está involucrada.

La etnografía es especialmente efectiva para descubrir dos tipos de requisitos:

- Los requisitos que se derivan de la forma en la que la gente trabaja realmente más que de la forma en la que las definiciones de los procesos establecen que debería trabajar.



- Los requisitos que se derivan de la cooperación y conocimiento de las actividades de los demás.

Los estudios etnográficos pueden revelar los detalles de los procesos críticos que otras técnicas de obtención de requisitos a menudo olvidan. Sin embargo, puesto que se centran en el usuario final, este enfoque no es apropiado para descubrir los requisitos organizacionales o del dominio. Los estudios etnográficos no siempre pueden identificar nuevas propiedades que se deban agregar al sistema. Por lo tanto, la etnografía no es un enfoque completo para la obtención de requisitos por sí mismo, y debe utilizarse para complementar otros enfoques, como el análisis de casos de uso.

### **33.3 Verificación - Validación**

La **validación** de requisitos trata de mostrar que éstos realmente definen el sistema que el cliente desea. La validación de requisitos es importante debido a que los errores en los requisitos pueden conducir a importantes costes al repetir el trabajo cuando son descubiertos durante el desarrollo o después de que el sistema esté en uso. El coste de arreglar un problema en los requisitos haciendo un cambio en el sistema es mucho mayor que reparar los errores de diseño o los de codificación.

La razón de esto es que un cambio en los requisitos normalmente significa que el diseño y la implementación del sistema también deben cambiar y que éste debe probarse nuevamente.

Durante el proceso de validación de requisitos, se deben llevar a cabo las siguientes **verificaciones**:

- 1- **Verificaciones de validez.** Un usuario puede pensar que se necesita un sistema para llevar a cabo ciertas funciones. Sin embargo, el razonamiento y el análisis pueden identificar que se

requieren funciones adicionales o diferentes. También se deben tener en cuenta que en un mismo sistema suele haber diferentes usuarios, con diferentes puntos de vista, algunas veces contrapuestos, y que éstos deben llegar a un compromiso a la hora de definir los requisitos del sistema.

- 2- **Verificaciones de consistencia.** Los requisitos no deben contradecirse. Esto es, no debe haber restricciones o descripciones contradictorias de la misma función del sistema.
- 3- **Verificaciones de completitud.** Los requisitos deben definir todas las funciones y restricciones propuestas por el usuario del sistema.
- 4- **Verificaciones de realismo.** Utilizando el conocimiento de la tecnología existente, los requisitos deben verificarse para asegurar que se pueden implementar. Estas verificaciones también deben tener en cuenta el presupuesto y la confección de agendas para el desarrollo del sistema.
- 5- **Verificabilidad.** Para reducir la posibilidad de discusiones entre el cliente y el desarrollador, los requisitos del sistema siempre deben redactarse de tal forma que sean verificables. Esto significa que debe poder escribir un conjunto de pruebas que demuestren que el sistema a entregar cumple cada uno de los requisitos especificados.

Pueden utilizarse, en conjunto o de forma individual, varias técnicas de validación de requisitos:

- 1- **Revisiones de requisitos.** Los requisitos son analizados manual y sistemáticamente por un equipo de revisores formado por personas tanto de la organización del cliente como de la desarrolladora. Se verifican los requisitos en cuanto a anomalías y omisiones. Pueden ser informales o formales. Las informales sencillamente implican que los desarrolladores deben tratar los requisitos con tantos usuarios del sistema como sea posible. En la revisión formal de requisitos, el equipo de desarrollo debe «conducir» al cliente a través de los

requisitos del sistema, explicándole las implicaciones de cada requisito. Los revisores deben comprobar la:

- a. **Verificabilidad.** ¿Puede probarse el requisito de modo realista?
- b. **Comprensibilidad.** ¿Las personas que adquieren el sistema o los usuarios finales comprenden correctamente el requisito?
- c. **Rastreabilidad.** ¿Está claramente establecido el origen del requisito? Puede tener que volver a la fuente del requisito para evaluar el impacto del cambio. La rastreabilidad es importante ya que permite evaluar el impacto del cambio en el resto del sistema.
- d. **Adaptabilidad.** ¿Es adaptable el requisito? Es decir, ¿puede cambiarse el requisito sin causar efectos de gran escala en los otros requisitos del sistema?

Los conflictos, contradicciones, errores y omisiones en los requisitos deben ser señalados por los revisores y registrarse formalmente en el informe de revisión. Queda en los usuarios, la persona que adquiere el sistema y el desarrollador de éste negociar una solución para estos problemas identificados.

- 2- **Construcción de prototipos** En este enfoque de validación, se muestra un modelo ejecutable del sistema a los usuarios finales y a los clientes. Éstos pueden experimentar con este modelo para ver si cumple sus necesidades reales.
- 3- **Generación de casos de prueba.** Los requisitos deben poder probarse. Si las pruebas para éstos se conciben como parte del proceso de validación, a menudo revela los problemas en los requisitos. Si una prueba es difícil o imposible de diseñar, normalmente significa que los requisitos serán difíciles de implementar y deberían ser considerados nuevamente.

Es difícil demostrar que un conjunto de requisitos cumple las necesidades del usuario. Como consecuencia, rara vez se encuentran todos los problemas en los requisitos durante el proceso de validación de éstos. Es inevitable que haya cambios adicionales de requisitos para corregir las omisiones y las malas interpretaciones después de que el documento de requisitos haya sido aprobado.

### **33.4 Especificación de requisitos**

El **documento de requisitos del software** (algunas veces denominado *especificación de requisitos del software* o **ERS**) es la declaración oficial de qué deben implementar los desarrolladores del sistema. Tiene un conjunto diverso de usuarios que va desde los altos cargos de la organización que pagan por el sistema, hasta los ingenieros responsables de desarrollar el software. Los objetivos que pretende el ERS son los siguientes:

- Proporcionar los medios de comunicación entre todas las partes implicadas en el sistema: clientes, usuarios, analistas y diseñadores.
- Servir como base para las actividades de prueba y verificación.
- Ayudar al control de la evolución del sistema software.

El documento ERS debe incluir una descripción completa y concisa de toda la interfaz externa del sistema con su entorno, incluido el resto del software, puertos de comunicación, hardware y usuarios. Es decir, debe incluir tanto los requisitos de comportamiento del sistema (funcionales), que son aquellos que definen lo que hace éste y la información que maneja, como los requisitos que no son de comportamiento, esto es, aquellos que definen los atributos del sistema según realiza su trabajo (eficiencia, fiabilidad, seguridad, etc.).

Por contra, un documento ERS no debe incluir los elementos de gestión del proyecto (planificaciones, hitos, etc.), ni el diseño, ni los planes de control

del producto (gestión de configuración, garantía de calidad, etc.). Un documento ERS debe reunir las siguientes características:

- **Correcto.** Cada requisito establecido debe representar algo requerido para el sistema.
- **No Ambiguo.** Cada requisito establecido tiene una sola interpretación.
- **Completo.** Debe incluir todo lo que el software tiene que hacer.
- **Verificable.** Se ha de poder comprobar, mediante un proceso efectivo y de coste limitado, que el producto reúne cada requisito establecido.
- **Consistente.** Cada requisito no puede estar en conflicto con otros requisitos.
- **Modificable.** La estructura y estilo del documento debe hacer fáciles los cambios.
- **Conciso,** comprensible por el usuario y organizado.
- **Referenciado.** Cada requisito debe estar calificado y debidamente referenciado.

#### *33.4.1 IEEE/ANSI 830-1998*

Varias organizaciones grandes, como el Departamento de Defensa de los Estados Unidos y el IEEE, han definido estándares para los documentos de requisitos. El estándar más ampliamente conocido es el IEEE/ANSI 830-1998 (IEEE, 1998). Este estándar IEEE sugiere la siguiente estructura para los documentos de requisitos:

##### **1. Introducción**

1.1 Propósito del documento de requisitos: Propósito y a quién va dirigido.

1.2 Alcance del producto

1.3 Definiciones, acrónimos y abreviaturas

#### 1.4 Referencias

#### 1.5 Descripción del resto del documento

### 2. Descripción general

#### 2.1 Perspectiva del producto

#### 2.2 Funciones del producto

#### 2.3 Características del usuario

#### 2.4 Restricciones generales

#### 2.5 Suposiciones y dependencias

### 3. **Requisitos específicos.** Este será el grueso del documento.

**3.1. Interfaces Externas:** Se describirán los requisitos que afecten a la interfaz de usuario, interfaz con otros sistemas (hardware y software) e interfaces de comunicaciones.

**3.2. Funciones:** Esta subsección (quizá la más larga del documento) deberá especificar todas aquellas acciones (funciones) que deberá llevar a cabo el software. Si se considera necesario, podrán utilizarse notaciones gráficas y tablas, pero siempre supeditadas al lenguaje natural, y no al revés. El estándar permite organizar esta subsección de múltiples formas, y sugiere, entre otras, las siguientes:

- **Por tipos de usuario:** Distintos usuarios poseen distintos requisitos. Para cada clase de usuario que exista en la organización, se especificarán los requisitos funcionales que le afecten o tengan mayor relación con sus tareas.
- **Por objetos:** Los objetos son entidades del mundo real que serán reflejadas en el sistema. Para cada objeto, se detallarán sus atributos y sus funciones. Los objetos pueden agruparse en clases. Esta organización de la ERS no quiere decir que el diseño del sistema siga el paradigma de Orientación a Objetos.
- **Por objetivos:** Un objetivo es un servicio que se desea que ofrezca el sistema y que requiere una determinada entrada para obtener su resultado. Para cada objetivo o subobjetivo

que se persiga con el sistema, se detallarán las funciones que permitan llevarlo a cabo.

- **Por estímulos:** Se especificaran los posibles estímulos que recibe el sistema y las funciones relacionadas con dicho estímulo.
- **Por jerarquía funcional:** Si ninguna de las anteriores alternativas resulta de ayuda, la funcionalidad del sistema se especificara como una jerarquía de funciones que comparten entradas, salidas o datos internos. Se detallaran las funciones (entrada, proceso, salida) y las subfunciones del sistema. Esto no implica que el diseño del sistema deba realizarse según el paradigma de Diseño Estructurado.

**3.3. Requisitos de Rendimiento:** Se detallarían los requisitos relacionados con la carga que se espera tenga que soportar el sistema. Por ejemplo, el número de terminales, el número esperado de usuarios simultáneamente conectados, número de transacciones por segundo que deberá soportar el sistema, etc. También, si es necesario, se especificarán los requisitos de datos, es decir, aquellos requisitos que afecten a la información que se guardará en la base de datos. Por ejemplo, la frecuencia de uso, las capacidades de acceso y la cantidad de registros que se espera almacenar (decenas, cientos, miles o millones).

**3.4. Restricciones de Diseño:** Todo aquello que restrinja las decisiones relativas al diseño de la aplicación: Restricciones de otros estándares, limitaciones del hardware, etc.

**3.5. Atributos del Sistema:** Se detallaran los atributos de calidad del sistema: Fiabilidad, mantenibilidad, portabilidad, y, muy importante, la seguridad. Debería especificarse que tipos de usuario están autorizados, o no, a realizar ciertas tareas, y como se implementarían los mecanismos de seguridad

**3.6. Otros Requisitos**

## **4. Apéndices**

## **5. Índice**

El estándar IEEE es un marco general que se puede transformar y adaptar para definir un estándar ajustado a las necesidades de una organización en particular. La información que se incluya en un documento de requisitos debe depender del tipo de software a desarrollar y del enfoque de desarrollo que se utilice.

El documento de requisitos es fundamental cuando un desarrollador exterior está construyendo el sistema software. Sin embargo, los métodos de desarrollo ágiles sostienen que los requisitos cambian tan rápidamente que un documento de requisitos se queda desfasado en cuanto se redacta, por lo que el esfuerzo en gran parte se malgasta.

### **33. 5 Gestión de requisitos**

Los requisitos para sistemas software grandes son siempre cambiantes. Debido a que el problema no puede definirse completamente, es muy probable que los requisitos del software sean incompletos. Durante el proceso del software, la comprensión del problema por parte de los clientes cambia, y los requisitos deben entonces evolucionar para reflejar esto. Además, una vez que un sistema se ha instalado, inevitablemente surgen nuevos requisitos. Cuando los usuarios finales trabajan con un sistema, descubren nuevas necesidades y prioridades debidas a:

- Normalmente, los sistemas grandes tienen una comunidad de usuarios diversa donde los usuarios tienen diferentes requisitos y prioridades. Los que se usaron para definir el sistema pudieron no ser los mejores.
- Las personas que pagan por el sistema y los usuarios de éste raramente son la misma persona. Los clientes del sistema imponen requisitos debido a las restricciones organizacionales y de presupuesto. Éstos pueden estar en conflicto con los requisitos de los



usuarios finales y, después de la entrega, pueden tener que añadirse nuevas características de apoyo al usuario si el sistema tiene que cumplir sus objetivos.

- El entorno de negocios y técnico del sistema cambia después de la instalación, y estos cambios se deben reflejar en el sistema. Se puede introducir nuevo hardware, puede ser necesario que el sistema interactúe con otros sistemas, las prioridades de negocio pueden cambiar con modificaciones consecuentes en la ayuda al sistema, y puede haber una nueva legislación y regulaciones que deben ser implementadas por el sistema.

La **gestión de requisitos** es el proceso de comprender y controlar los cambios en los requisitos del sistema. Es necesario poder evaluar el impacto de los cambios en los requisitos. Hay que establecer un proceso formal para implementar las propuestas de cambios. El proceso de gestión de requisitos debería empezar en cuanto esté disponible una versión preliminar del documento de requisitos, pero se debería empezar a planificar cómo gestionar los requisitos que cambian durante el proceso de obtención de requisitos.

Existen **requisitos duraderos**, que son requisitos relativamente estables que se derivan de la actividad principal de la organización y que están relacionados directamente con el dominio del sistema. Estos requisitos se pueden derivar de los modelos del dominio que muestran las entidades y relaciones que caracterizan un dominio de aplicación. Por otro lado, existen **requisitos volátiles**, que son requisitos que probablemente cambian durante el proceso de desarrollo del sistema o después de que éste se haya puesto en funcionamiento, debido a cambios del entorno, legislación,...

#### *33.5.1 Planificación de la gestión requisitos.*

Para cada proyecto, la etapa de planificación establece el nivel de detalle necesario en la gestión de requisitos. Habrá que decidir sobre:

1. **La identificación de requisitos.** Cada requisito se debe identificar de forma única de tal forma que puedan ser remitidos por otros requisitos de manera que pueda utilizarse en las evaluaciones de rastreo.
2. **Un proceso de gestión del cambio.** Éste es el conjunto de actividades que evalúan el impacto y coste de los cambios. Lo veremos en un apartado posterior.
3. **Políticas de rastreo o trazabilidad.** Estas políticas definen las relaciones entre los requisitos, y entre éstos y el diseño del sistema que se debe registrar y la manera en que estos registros se deben mantener.
4. **Selección de herramientas CASE.** La gestión de requisitos comprende el procesamiento de grandes cantidades de información sobre los requisitos. Las herramientas que se pueden utilizar van desde sistemas de gestión de requisitos especializados hasta hojas de cálculo y sistemas sencillos de bases de datos.

El concepto de trazabilidad hace referencia a la posibilidad de determinar cómo se ha llegado a un cierto elemento del software a partir de otros. Para poder llevar a cabo esta trazabilidad es especialmente importante el poder relacionar unos requisitos con otros y también relacionar requisitos con elementos del sistema a los que da lugar. Se habla de trazabilidad *hacia delante* cuando partiendo de un requisito se llega a todos los elementos que materializan dicho requisito, o *hacia atrás*, cuando partiendo de un elemento del sistema se llega al requisito que lo generó.

Un requisito se debe poder relacionar con otros requisitos similares para así evitar repeticiones en los mismos. Además, cuando se hace un cambio en uno, es más fácil localizar aquellos requisitos en los que hay relación, para ver el impacto del cambio. Por otro lado, es útil poder relacionar un

requisito con los elementos del sistema a los que éste da lugar, por si hay que cambiar un elemento del sistema, o simplemente, para poder saber qué requisitos dieron lugar a un determinado componente del sistema. Esta relación no sólo debe ser entre requisitos, sino también entre cualquier elemento que se haya derivado posteriormente, ya sea del análisis, del diseño, de la construcción, de pruebas, etc.

La gestión de requisitos necesita ayuda automatizada. Las herramientas CASE para esto deben elegirse durante la fase de planificación. Se precisan herramientas de ayuda para:

1. **Almacenar requisitos.** Los requisitos deben mantenerse en un almacén de datos seguro y administrado que sea accesible a todos los que estén implicados en el proceso de ingeniería de requisitos.
2. **Gestionar el cambio.**
3. **Gestionar el rastreo.** Las herramientas de ayuda para el rastreo permiten que se descubran requisitos relacionados. Algunas herramientas utilizan técnicas de procesamiento del lenguaje natural para ayudarle a descubrir posibles relaciones entre los requisitos.

Para sistemas pequeños, es posible que no sea necesario utilizar herramientas de gestión de requisitos especializadas, pero si es muy conveniente para sistemas grandes.

### *33.5.2 Gestión del cambio*

La ventaja de utilizar un proceso formal para gestionar el cambio es que todos los cambios propuestos son tratados de forma consistente y que los cambios en los requisitos se hacen de forma controlada. Existen varias etapas principales en un proceso de gestión de cambio:

- **Propuesta de Cambios.** El afectado por un requisito que no le satisface debe rellenar un formulario de propuesta de cambio

indicando cual es el cambio que hay que realizar. Este cambio se remitirá al equipo de gestión de requisitos para que lo estudie.

- **Análisis de Impactos.** La propuesta de cambio se evalúa para determinar el impacto del cambio en el resto de los requisitos. Hay propuestas cuyo impacto es mínimo, y otras cuyo impacto hace que se deba modificar una parte sustancial del sistema. Además del impacto se deberá estudiar la oportunidad o necesidad del cambio. Ocurre muchas veces que a un usuario no le parece bien un determinado aspecto del sistema pero a otros si les satisface. En estos casos se debe llegar a un consenso entre todas las partes. Otras veces el cambio es interesante pero se considera que no es oportuno hacerlo en dicho momento, quizás porque se quiera lanzar el sistema cuanto antes o porque el cambio es considerable y sería mejor atacarlo en una fase posterior de desarrollo.
- **Toma de Decisiones.** Dependiendo del impacto, de la necesidad u oportunidad del cambio y de otras cuestiones específicas que puedan surgir alrededor del cambio, se debe determinar si el cambio debe ser realizado o no. Si se decide realizarlo se deben hacer las modificaciones pertinentes en el sistema para integrar el cambio con el resto, empezando por los requisitos y finalizando por las partes más avanzadas del sistema. Si por el contrario se decide no hacerlo se puede optar por dos soluciones:
  - o Rechazar el cambio, porque se considera que no tiene sentido.
  - o Posponer su realización para un futuro, se considera el cambio necesario pero el momento no es oportuno. Se pospone para cuando se pueda acometer.
- **Comunicación.** Tanto si se acepta como si no, se debe notificar los efectos de la propuesta de cambio a todos los afectados.
- **Incorporación.** Se hacen las modificaciones pertinentes que se identificaron en el análisis de impacto en los diferentes elementos afectados por el cambio.

- **Medición de la Estabilidad de los Requisitos.** Se evalúan los parámetros que definen la estabilidad de los requisitos tras las modificaciones que se hayan realizado.

**Bibliografía:**

Ingeniería de Software 7 Ed - Ian Sommerville

El Proceso Unificado de Desarrollo de Software - Jacobson - Booch - Rumbaugh

Métrica 3 - Técnicas y Prácticas. Ministerio de Administraciones Públicas

Especificación de Requisitos según el estándar de IEEE 830

<http://www.fdi.ucm.es/profesor/gmendez/docs/is0809/ieee830.pdf>

**Autor: Hernán Vila Pérez**

**Jefe del Servicio de Informática. Instituto Galego de Vivenda e Solo**

**Vicepresidente del CPEIG**



**34. TÉCNICAS DE  
PROGRAMACIÓN.  
PROGRAMACIÓN  
ESTRUCTURADA.  
PROGRAMACIÓN ORIENTADA A  
OBJETOS. INGENIERÍA  
INVERSA Y REINGENIERÍA.**

Tema 34. Técnicas de programación. Programación Estructurada.  
Programación orientada a objetos. Ingeniería inversa y reingeniería.

## **INDICE**

- 34.1 Técnicas de programación
  - 34.1.1 Clasificación y evolución de los lenguajes de programación
- 34.2 Programación Estructurada
  - 34.2.1 Recursos abstractos
  - 34.2.2 Diseño descendente
  - 34.2.3 Estructuras básicas
- 34.3 Programación orientada a objetos
- 34.4 Ingeniería inversa
  - 34.4.1 Modelo cíclico
  - 34.4.2 Modelo de herradura
  - 34.4.3 Modelo del IEEE
  - 34.4.4 Método Análisis de Opciones para Reingeniería (OAR)

### **34.1 Técnicas de programación**

Un **lenguaje de programación** es un lenguaje artificial diseñado para representar expresiones e instrucciones de forma que las entienda un ordenador. Aunque más sencillos que los lenguajes naturales, tienen un alfabeto (símbolos básicos) con los que se construye el vocabulario (tokens, palabras reservadas), que se combinan según unas reglas sintácticas formando expresiones y sentencias, cuyo significado viene dado por la semántica del lenguaje. Se llama **programa** al conjunto de instrucciones escritas en un lenguaje de programación, destinadas a realizar una tarea. De forma general tomará unos datos de entrada y devolverá unos datos de salida. Se dice que lo que hace el programa es implementar un **algoritmo**, que es un conjunto finito de instrucciones que se deben seguir para realizar una determinada tarea.

El diseño de un programa que se realiza sin seguir una metodología puede funcionar, pero se debe tener en cuenta que con el tiempo se convertirá en un conjunto de instrucciones difíciles de tocar y mantener. Esto hará que los programas sean difíciles de adaptar a una nueva configuración, se

pierda mucho tiempo con la corrección de errores, sean difíciles de entender para otros programadores, etc.

Es de suma importancia poder prevenir las modificaciones que puedan realizarse en el futuro, así como también tener la documentación siempre actualizada. La creación de programas debe tener la flexibilidad suficiente para ser modificables en el momento en que se requiera. Estos deben ser claros, simples, con el fin de poder ser leídos e interpretados de forma fácil. Para la programación deberán asumirse ciertas normas que permitan la estandarización, implicando una disminución en costos, independencia del programador y seguridad.

Entendemos pues la **programación** como *la planificación, proyección, desarrollo e implementación de la resolución de un problema*, la que abarca obviamente a la creación del algoritmo. Un profesional en la programación debe encarar la solución del problema de forma tal que su producto sea útil ahora y en el futuro, estando o no él en el centro de desarrollo. Para lograr esto debe tener muy presentes las futuras posibles modificaciones del mismo.

Las características, entonces, que debe tener un programa son:

- 1) **Claridad algorítmica:** Que sea claro significa, que su resolución algorítmica sea sencilla, que esté correctamente estructurado, resultando de fácil comprensión.
- 2) **Legibilidad:** Que sea legible significa que cuando se codificó se eligieron bien los nombres de los objetos utilizados, se agregaron comentarios para indicar lo que se iba haciendo y se estructuró bien el código para resaltar el contenido semántico sobre lo sintáctico.
- 3) **Modificabilidad:** Que sea fácilmente modificable, implica que cualquier modificación del problema que genere un agregado, supresión o cambio de alguna de sus partes, no debe obligar a cambiar todo el programa, sino sólo una parte.



En sus inicios la programación no seguía ninguna metodología, y teníamos cientos o miles de líneas de código, hechas por un programador, y que solo él se atrevía a tocar. En esta época los profesionales programadores solían estar atados a la empresa y no era frecuente que abandonaran la misma. Los programas no tenían estructura definida. Las modificaciones en el programa suponían un coste importante ya que había que revisar el código casi línea a línea para buscar donde hacer las modificaciones. Además, estos cambios no dejaban de ser un riesgo importante debido a los posibles efectos colaterales. Esto acabó en lo que se denominó crisis del software.

Para tratar de dar respuesta a la crisis del software, en los años 60 surgieron técnicas de programación como la **programación modular** y la **programación estructurada**. Ambas técnicas parten de la idea del diseño descendente (también llamado refinamiento sucesivo), que consiste en dividir un problema complejo en varios problemas menores, más sencillos de resolver. Inicia planteando el problema y su solución a un nivel alto, superior, y luego se sigue descomponiendo el problema general en otros problemas más sencillos de resolver. Este proceso continúa hasta que llegamos a pequeños problemas, fácilmente implementables en código llamadas módulos. Un módulo es un conjunto de acciones (un bloque del código del algoritmo total) junto a un conjunto de especificaciones de datos para realizar una tarea específica. Cuando se utiliza esta técnica, la solución queda dividida en varias partes: un algoritmo principal y uno o varios subalgoritmos (los módulos). La ejecución se inicia en el algoritmo principal y desde este se invoca a los módulos. La programación estructurada hace uso de la programación modular, además de otras características que se verán en un apartado posterior.

También como respuesta a la crisis del software surgieron en los años 60 los conceptos de la orientación a objetos, que tenían como principal objetivo reducir la complejidad del desarrollo y mantenimiento del software. Dado que la tecnología no estaba acorde con su implementación,

se mantuvo sólo como concepto hasta su gran expansión a finales de los 80. La **programación orientada a objetos** no supone una ruptura radical frente al paradigma de la programación estructurada / imperativa predominante hasta su aparición, sino que supone una evolución. Frente a la programación estructurada, cuyos programas separan datos y estructuras de datos de funciones y procedimientos, la programación orientada a objetos encapsula en una misma abstracción estos dos elementos clásicos de los sistemas de información: datos y los procesos. Esto lo hace a través de una nueva abstracción: el objeto. La orientación a objetos consiste en una visión de los objetos como entidades activas que ejecutan acciones sobre sus propios datos en respuesta a peticiones externas. No considera a unos y a otros (datos y procesos) como realidades aisladas susceptibles de analizarse e implantarse por separado. Los objetos tratan de abstraer características comunes que podrán compartirse entre varias aplicaciones y reutilizarse todo lo posible. La creación de un nuevo sistema consiste esencialmente en una labor de ensamblado de objetos preexistentes, completada con el desarrollo de un porcentaje reducido de nuevos objetos, que a su vez alimentarán las correspondientes librerías para poder ser utilizados en los próximos sistemas.

El paradigma de ensamblar componentes y escribir código para hacer que estos componentes funcionen se conoce como **Desarrollo de Software Basado en Componentes**. Un componente es una pieza de código preelaborado que encapsula alguna funcionalidad expuesta a través de interfaces estándar. Cada componente es diseñado para acoplarse perfectamente con sus pares, las conexiones son estándar y el protocolo de comunicación está ya preestablecido. Al unirse las partes, obtenemos un sistema completo.

Por último, la **programación orientada a aspectos** surge para resolver el problema de la "dispersión de código" existente en la programación orientada a objetos para aquellos aspectos de un programa que no tienen

que ver directamente con el dominio de negocio del problema: gestión de conexiones, logs, trazas de mensajes, sincronización y concurrencia, manejo de errores y excepciones, etc. Este código está disperso a lo largo de diferentes objetos del programa (siendo buena parte de las veces el mismo código repetido) lo que dificulta su mantenimiento. Esta es una nueva metodología de programación que intenta separar los componentes y los aspectos unos de otros, proporcionando mecanismos que hagan posible abstraerlos y componerlos para formar todo el sistema. Es un desarrollo que sigue al paradigma de la orientación a objetos, y como tal, soporta la descomposición orientada a objetos, además de la procedimental y la descomposición funcional. Pero, a pesar de esto, no se puede considerar como una extensión de la POO, ya que puede utilizarse con los diferentes estilos de programación ya mencionados.

#### *34.1.1 Clasificación y evolución de los lenguajes de programación*

Para explicar la evolución de los lenguajes de programación hablamos de generaciones:

- **Primera generación.** Lenguaje máquina. Los programas están hechos de secuencias de unos y ceros que el computador es capaz de interpretar. Es el único que es directamente entendible por el ordenador, sin necesidad de traducción.
- **Segunda generación.** Ensamblador. Se usan nemotécnicos que sustituyen los unos y ceros. Siguen siendo dependientes de la máquina. El código fuente es traducido al lenguaje máquina mediante traductores.
- **Tercera generación.** Lenguajes de alto nivel. Lenguajes estructurados con comandos cercanos al lenguaje natural. Fueron creados para facilitar el proceso de programación. Los comandos se asemejan a palabras de uso común.

- **Cuarta generación:** Nacen los lenguajes 4G. Son lenguajes de propósito especial, orientados a resolver problemas específicos, como generación de informes, pantallas, consultas de bases de datos,... Se caracterizan por tener una mayor facilidad de uso comparado con los de tercera generación, permitiendo la creación de prototipos rápidamente. Generan el código fuente automáticamente a través de asistentes, plantillas, etc.
- **Quinta generación:** Lenguajes Naturales. Son lenguajes orientados a la inteligencia artificial y los sistemas expertos. En lugar de sólo ejecutar órdenes, el objetivo de los sistemas es anticipar las necesidades de los usuarios. Están aún en desarrollo.

Los lenguajes de programación se pueden clasificar según muchos criterios:

- Según el **nivel de abstracción**:
  - o **Lenguajes máquina y de bajo nivel:** el lenguaje o código máquina está formado por cadenas binarias entendibles directamente por la máquina. No necesitan traducción y son muy rápidos. El lenguaje ensamblador permite al menos escribir las instrucciones utilizando una notación simbólica, utilizar direcciones de memoria relativas y no absolutas, insertar comentarios. Necesitan de un traductor para convertir a código máquina, traducción que resulta casi trivial debido a su bajo nivel. Estos lenguajes son completamente dependientes de la máquina.
  - o **Lenguajes de medio nivel:** aquí se encontraría el lenguaje C. El lenguaje C permite un manejo abstracto e independiente de la máquina (lo que lo acercaría al alto nivel), pero sin perder su característica de potencia, eficiencia y cercanía a la máquina (uso directo de punteros, etc.)
  - o **Lenguajes de alto nivel:** son independientes de la arquitectura de la máquina, disponen de gran diversidad de

instrucciones potentes, y usan una sintaxis similar al lenguaje natural lo que facilita su comprensión. Aquí estarían básicamente el resto de los lenguajes conocidos.

- Según la **forma de ejecución**
  - o **Compilados:** una vez escrito el programa, éste se traduce a partir de su código fuente por medio de un compilador en un archivo ejecutable para una determinada plataforma. El compilador lee el código fuente y almacena el ejecutable resultado de la traducción para posibles ejecuciones futuras. Ejemplos: C, C++, Pascal, Kilix, Delphi, ...
  - o **Interpretados:** necesitan de un intérprete para ejecutar el código escrito en los programas. Las instrucciones se traducen o interpretan una a una en tiempo de ejecución a un lenguaje intermedio o lenguaje máquina. En la ejecución de un lenguaje interpretado se van leyendo las líneas del código fuente y traduciéndolas a medida que va siendo necesario. Cada vez que se lee una instrucción se interpreta y se ejecuta, generando su código máquina “al vuelo”. Ejemplos: Java, JavaScript, PHP, lenguajes de .NET,...
- Según el **paradigma de programación:** Con esto nos referimos al enfoque a la hora de afrontar el problema que tratamos de programar.
  - o **Imperativos:** se fundamentan en la utilización de variables para almacenar valores y en la realización de instrucciones para operar con los datos almacenados. Aparece el concepto de algoritmo, el CÓMO conseguir el objetivo. Dentro de los lenguajes imperativos, podemos clasificarlos en:
    - **Procedurales o procedimentales:** agrupan código en subprogramas que pueden llamarse desde distintos puntos del programa. Ej. C, Pascal, Fortran, Basic, etc.

- **Orientados a objetos:** ven el programa como una colección de objetos que interactúan los unos con los otros a través de mensajes. Es el más utilizado en la actualidad. Ej. Smalltalk, C++, Java, C#, Visual Basic.NET, Eiffel, etc.

En la actualidad, casi todas las nuevas versiones de los lenguajes incorporan características de orientación a objetos. De hecho, muchos autores proponen el paradigma orientado a objetos como paradigma propio, evolución del paradigma imperativo.

- o **Declarativos:** se declara o se describe las propiedades o características del problema, dejando que la solución la encuentre la máquina. Realmente se declaran condiciones, proposiciones, hechos, reglas, afirmaciones, restricciones, ecuaciones, transformaciones, etc. que debe cumplir el conjunto de valores que constituyen la solución. Dentro de este tenemos tres tipos:
  - **Funcional o aplicativo:** Todas las construcciones son funciones matemáticas. No hay instrucciones, y tampoco hay instrucción de asignación. El programa se define por composición de funciones más simples. Para ejecutarlo, se llama a una función con los datos de entrada y se obtiene un resultado. Ej. Haskell, LISP, CAML, etc.
  - **Lógico:** Se basa en la definición de reglas lógicas para luego interrogar al sistema y resolver problemas. La programación lógica trata con relaciones (predicados) entre objetos (datos), en lugar de hacerlo con funciones. Ej. Prolog
  - **Algebraicos o Relacionales:** Algunos autores hablan de este otro paradigma para clasificar el SQL, que es el lenguaje utilizado para interrogar BBDD relacionales.

Especificamos el resultado que queremos y el SQL nos lo proporciona.

Hay que decir que algunos lenguajes son **multiparadigma**. Por ejemplo el C++ es un lenguaje orientado a objetos, pero también podemos usarlo como lenguaje imperativo, sin hacer uso de la orientación a objetos, o el Prolog, que es lógico, pero también cuenta con estructuras repetitivas propias del paradigma imperativo.

### **34.2 Programación Estructurada.**

La programación estructurada es un concepto que surge como repuesta a la crisis del software de los años 60, cuando el coste en el desarrollo de programas era cada vez mayor, y el mantenimiento de los mismos se hacía cada vez menos manejable.

Fue desarrollada en sus principios por Edsgar W. Dijkstra en sus “Notes on Structures Programming” y se basa en el denominado Teorema de la Estructura o de *Böhm-Jacopini* en honor de Corrado Böhm y Giuseppe Jacopini.

Edsgar Dijkstra definió a la programación estructurada como *aquella que utiliza recursos abstractos, se basa en el diseño descendente y respeta un conjunto de estructuras básicas llamadas Estructuras de Control: Estructura secuencial, estructuras selectivas y estructuras repetitivas.*

#### **34.2.1 Recursos abstractos**

Todos los lenguajes de programación poseen un conjunto de recursos que podríamos denominar recursos concretos: instrucciones, palabras reservadas, tipos de datos, funciones, reglas, etc. Sin embargo, estos recursos no son suficientes para escribir programas para distintas aplicaciones, por lo tanto, el programador deberá valerse de artificios para

implementar sus algoritmos utilizando solo los recursos concretos. Estos artificios se denominan recursos abstractos.

Según Dijkstra, *escribir un programa en términos de recursos abstractos consiste en descomponer las acciones complejas en acciones simples, capaces de ser ejecutadas por una máquina*. Esto significa que es posible escribir programas complejos utilizando un conjunto limitado de instrucciones muy simples.

### **34.2.2 Diseño Descendente**

Existen dos técnicas para escribir los algoritmos: el diseño ascendente y el diseño descendente.

Cuando se utiliza diseño ascendente, se parte desde los detalles particulares de implementación de la solución hacia la solución general del problema. La solución del problema se logra con la integración de todas las soluciones particulares planteadas al principio. Esto significa que primero se resuelven partes particulares del problema y por último el problema en sí. Esta técnica presenta dos dificultades: lograr que las soluciones particulares funcionen en conjunto y lograr que varios programadores trabajen en coordinación.

El diseño descendente consiste en comprender el problema a solucionar y luego descomponerlo en un conjunto de problemas menores. Cuando se usa esta técnica, primero se plantea la solución de forma general para luego pasar a los detalles particulares. Esto se realiza en varios pasos llamados refinamientos. Pueden existir varios niveles de refinamiento hasta llegar a los detalles de implementación de la solución (subalgoritmos independientes llamados módulos). Una ventaja de esta técnica es que permite la división de tareas entre varios programadores donde cada uno podrá escribir un algoritmo que obtenga una parte de la solución.



La programación estructurada hace pues uso de la **programación modular**. Un módulo es un conjunto de acciones que realiza una tarea específica. Puede realizar las mismas acciones que un programa: aceptar datos, realizar cálculos y devolver resultados. Sin embargo los módulos se utilizan para un fin específico. Cuando se utiliza esta técnica, toda la solución (el algoritmo) queda dividida en varias partes: un algoritmo principal y uno o varios subalgoritmos (los módulos). La ejecución se inicia en el algoritmo principal y desde este se invoca a los módulos. Los módulos también pueden ser invocados desde otros módulos y el control de la ejecución siempre se retorna al punto desde donde se lo invocó por última vez.

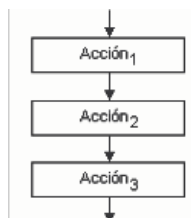
### 34.2.3 Estructuras básicas

Bohm y Jacopini demostraron que es posible resolver problemas escribiendo programas denominados **propios**. Un programa se define como propio si cumple con:

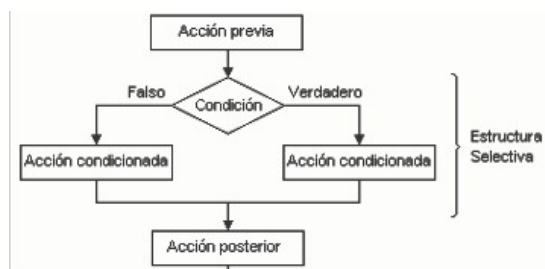
- tiene un solo punto de entrada y uno solo de salida (inicio y fin),
- todas las acciones del algoritmo son accesibles, es decir, para cada acción existe un conjunto de datos que hará que ésta sea accesible desde el inicio y además se podrá llegar al fin.
- no posee lazos o bucles infinitos.

Pues bien, el **teorema de la estructura** establece que *cualquier programa propio puede ser escrito utilizando solamente las siguientes estructuras lógicas de control: secuencia, selección e iteración*. Un algoritmo correctamente implementado poseerá un único punto de entrada, un único punto de salida, y permitirá la ejecución de todas sus instrucciones en función de los parámetros de entrada. La programación estructurada se basa en el concepto de programa propio y utiliza solamente estas tres estructuras básicas:

- **Estructura Secuencial:** Se caracteriza porque una acción se escribe y ejecuta a continuación de otra. Está representada por una sucesión de operaciones que se ejecutan secuencialmente.

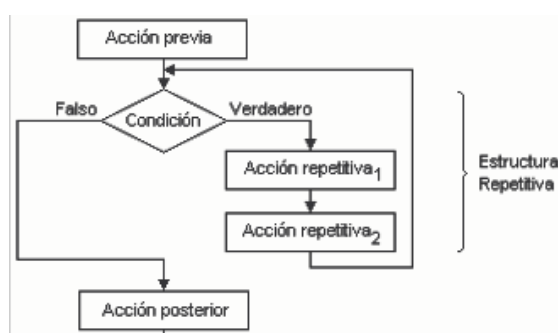


- **Estructura selectiva:** Se caracteriza porque existen dos o más secuencias alternativas de acciones en el programa. La selección de una u otra se realiza de acuerdo a una *condición* que se debe cumplir para que el conjunto de acciones sea ejecutado. Se dice que las estructuras selectivas *bifurcan* la ejecución del programa. Una instrucción de bifurcación evalúa una *condición* y, en función del resultado de esa evaluación, la ejecución se bifurca a un determinado punto. Una *condición* se construye con *expresiones lógicas*.



- **Estructura repetitiva o bucle:** Se caracteriza porque existe un conjunto de acciones cuya ejecución se debe repetir un determinado número de veces llamado bucle (o lazo). La implementación del bucle debe ser tal que sea posible acceder a él y también salir de él. No deben existir lazos de ejecución infinita. Para que un bucle cumpla con las condiciones mencionadas (poder acceder y poder salir de él) es necesario el uso de una condición que lo controle. En cada ciclo de ejecución del bucle se deberá evaluar la condición para decidir si se vuelve a ejecutar o si se sale de él, así, todo bucle deberá poseer un

controlador (un contador o un centinela), una decisión (una condición para decidir la repetición o la salida del bucle) y un cuerpo (conjunto de acciones que se repiten). Un contador es una variable numérica entera cuyo contenido se incrementa o decrementa sucesivamente con un valor constante. En un proceso de introducción de datos, el centinela es el valor cuya lectura como dato, indica la finalización del bucle.



Como ejemplos de lenguajes de programación estructurados tenemos FORTRAN, PASCAL, MODULA, ADA, C, etc.

### 34.3 Programación orientada a objetos

La programación orientada a objetos es lo que se conoce como un paradigma o modelo de programación. Esto significa que no es un lenguaje específico, o una tecnología, sino una forma de programar, una manera de plantearse la programación. Lo que caracteriza a la programación orientada a objetos es que intenta llevar al mundo del código lo mismo que encontramos en el mundo real. Cuando miramos a nuestro alrededor ¿qué vemos? pues, cosas, **objetos**, y podemos reconocer estos objetos porque **cada objeto pertenece a una clase** de objeto. Eso nos permite distinguir, por ejemplo, un perro de un coche (porque son de clases diferentes) y también un TV de otro (porque, aunque sean iguales, cada uno es un objeto distinto). Éste es el modelo que la programación orientada a objetos intenta seguir para estructurar un sistema.

La construcción de software orientado a objetos es el método de desarrollo de software que basa la arquitectura de cualquier sistema software en módulos deducidos de los tipos de objetos que manipula (en lugar de basarse en la función o funciones a las que el sistema está destinado a asegurar). Los desarrolladores analizarán los tipos de los objetos del sistema. El diseño irá pasando por las sucesivas mejoras de su comprensión de estas clases de objetos. Es un proceso ascendente, frente a la programación estructurada que era descendente, de construcción de soluciones robustas y extensibles para ciertas partes del problema, y de combinación de esas soluciones en montajes cada vez más potentes, hasta que el montaje final da una solución del problema original; además, los mismos componentes ensamblados de manera diferente y combinados posiblemente con otros, deberán ser lo suficientemente generales para producir también otros subproductos software, es decir, deben ser reutilizables.

Así, un sistema se concibe como un conjunto de objetos, pertenecientes a alguna clase, que se comunican entre sí mediante mensajes. Los objetos se distinguen por:

- Poseer un estado que puede cambiar,
- tener una identidad única,
- soportar interrelaciones con otros objetos,
- poseer un comportamiento,
- poder recibir y emitir mensajes.

Un **objeto** se describe por sus propiedades, también llamadas atributos (definen la estructura del objeto), y por los servicios (métodos u operaciones) que puede proporcionar (comportamiento del objeto). El estado de un objeto viene determinado por los valores que toman sus atributos. El objeto es la unidad fundamental de descomposición en la programación orientada a objetos.

Además, todo objeto es una instancia de alguna **clase**. Todos los objetos se crean de acuerdo a una definición de clase de objetos. Dicha clase incluye declaraciones de todos los atributos y operaciones que deberían asociarse con un objeto de dicha clase. Una clase es una abstracción de las propiedades comunes y comportamiento de un conjunto de objetos.

Así pues, un programa orientado a objetos consta de una o más clases interdependientes. Las clases permiten describir las propiedades (atributos o campos) y habilidades (métodos) de los objetos con los que el programa tiene que tratar. El conjunto de atributos y métodos de una clase reciben también el nombre de **miembros** de esa clase.

La definición de los miembros pueden ir precedidas de un **modificador de acceso**, que es algo que indica desde que puntos del código es accesible ese miembro. Los tres principales modificadores de acceso son:

- **Public:** Se puede acceder desde cualquier punto.
- **Protected:** Es accesible desde los métodos de su clase y de las clases que de ella pudieran derivarse.
- **Private:** Solo se puede utilizar desde los métodos de la propia clase.

Estos son los modificadores de acceso más comunes, aunque dependiendo del lenguaje podríamos tener otros modificadores que varían los accesos en función de si las clases están en el mismo paquete,...

El **comportamiento** de un objeto viene determinado por el conjunto de métodos que éste proporciona. Nos referiremos indistintamente a ellos como **métodos, servicios u operaciones**. Un método está formado por un conjunto de sentencias (acciones) que deben llevarse a cabo para hacer efectivo un comportamiento de un objeto. Un objeto proporciona un conjunto de métodos que pueden utilizarse desde otros objetos. Dicho conjunto de métodos constituye lo que se denomina **interfaz** del objeto. Mediante la interfaz de un objeto podremos acceder a los valores de sus atributos (estado) para su consulta y/o modificación y activar un comportamiento del mismo. Cuando se hace una llamada a un método de

la interfaz de un objeto decimos que estamos enviando un mensaje a dicho objeto. Así, un **mensaje** se puede definir como una petición a un objeto para la obtención de algún comportamiento deseado del mismo. Una llamada a un método puede contener **parámetros**, o conjunto de datos de entrada del método. Los métodos pueden devolver valores, objetos o nada (void).

Se dice que la información acerca de un objeto está **encapsulada** por su comportamiento. A un objeto se le deben pedir sus datos, o pedir que los cambie con un mensaje. Al encapsular u ocultar información se separan los aspectos externos de un objeto (los accesibles para todos) de los detalles de implementación (los accesibles para nadie). Con esto se trata de lograr que al tener algún cambio en la implementación de un objeto no se tengan que modificar los programas que utilizan tal objeto.

Una de las propiedades más características de la orientación a objetos es la **herencia**, que es la capacidad de crear nuevas clases (subclases) a partir de otra clase ya existente, especificando únicamente las diferencias con la clase padre. Aunque algunos lenguajes permiten la herencia múltiple (una clase que deriva de más de una clase), en general, la mayoría de ellos no lo hacen y sólo permiten que se herede de una clase. Para simular la herencia múltiple se utilizan los interfaces, que si son soportados por todos los lenguajes orientados a objetos. Los interfaces especifican (no implementan) conjuntos de métodos y atributos para que las clases los implementen. Una clase que implementa un interfaz está obligada a tener una implementación para cada método definido en el interfaz.

Cuando se diseña un modelo orientado a objetos, es útil introducir clases a cierto nivel que pueden no existir en realidad, pero que permiten actuar como un depósito de métodos y atributos compartidos para las subclases de nivel inferior. Estas clases se denominan **clases abstractas**, y no pueden tener ninguna instancia. No se pueden crear objetos de dicha

clase. También tenemos **métodos abstractos**, que son aquellos en los que no se especifica ninguna implementación. Son las clases derivadas las que tienen que hacerlo. Una **clase** se dice que es **sellada** cuando no puede tener clases derivadas. Igualmente, un **método sellado** es aquel que no puede ser redefinido en las clases derivadas.

Los atributos de objeto son aquellos que almacenan un valor para cada objeto de la clase. Los atributos de clase son aquellos que almacenan un valor accesible para todos los objetos de la clase. Los métodos de objeto son aquellos que acceden al estado de un objeto concreto, mientras que los métodos de clases no necesitan acceder a ningún atributo de ningún objeto. Los miembros (atributos y métodos) de clase también son llamados estáticos o compartidos.

Un **constructor** es un método especial que se ejecuta automáticamente cuando se crea el objeto. Su propósito es inicializar el objeto con, al menos, los datos mínimos que necesita. Igual que para el resto de métodos, puede haber varios constructores sobrecargados. Es decir, podemos crear objetos de distintas formas. Se llama constructor por defecto a aquel que no tiene argumentos. Un **destructor** es un método especial que suele liberar la memoria y otros recursos cuando un objeto deja de ser usado. Puede ser llamado automáticamente, liberando así de la responsabilidad al programador.

Por último hablaremos de otra de las características fundamentales de la programación orientada a objetos, el **polimorfismo**. El polimorfismo se refiere a dos aspectos diferentes: por un lado la sobrecarga de métodos y operadores (métodos polimórficos) y por otro lado, la ligadura dinámica. El primero es la capacidad de tener distintos métodos en la misma clase con el mismo nombre, aunque con distinta firma (número de argumentos, tipos de los argumentos, orden). Lo que no pueden es devolver un resultado de distinto tipo. El otro aspecto, la ligadura dinámica, se refiere a que cuando se envía un mensaje a un objeto, el código que se llama no se determina

hasta el momento de la ejecución. El compilador asegura que la función existe, pero no conoce el código exacto a ejecutar. Para ello, el compilador inserta un código especial en lugar de una llamada absoluta. Este código calcula en tiempo de ejecución la dirección real del método que hay que ejecutar utilizando la información almacenada en el propio objeto. Esta ligadura dinámica está relacionada con la herencia. En términos prácticos, el polimorfismo permite definir una referencia a una clase padre, que en tiempo de ejecución apuntará a un objeto concreto de alguna clase hija (no conocida en tiempo de compilación). La llamada a los métodos virtuales (los métodos de la clase padre que serán redefinidos en la clase hija) a través de dicha referencia se ejecutarán correctamente, pues durante la ejecución se verifica el tipo del objeto almacenado y se llama a la versión correcta del método.

Ejemplos de lenguajes de programación orientados a objetos son: C++, Objective C, Java, Smalltalk, Eiffel, Ruby, Python, OCAML, Object Pascal, CLIPS, Actionscript, Perl, C#, Visual Basic.NET, PHP, Simula, Delphi, PowerBuilder

### **34.4 Ingeniería inversa y reingeniería**

La Ingeniería Inversa se ocupa de estudiar un sistema de información en el orden inverso establecido en el ciclo de vida habitual; esto es, partiendo del código fuente, se trata de identificar los componentes del sistema y las relaciones existentes entre ellos. Hasta su llegada, el ciclo de vida del software era, en teoría, en una sola dirección; ahora, se puede hablar de dos direcciones: forward o hacia adelante, que es la tradicional, y reverse o hacia atrás, que es la de la Ingeniería Inversa. La Ingeniería Inversa también es conocida como modernización de caja blanca (White-Box Modernization).

Daremos un par de definiciones de ingeniería inversa:



“El análisis de un sistema para identificar sus componentes actuales y las dependencias que existen entre ellos, para extraer y crear abstracciones de dicho sistema e información de su diseño” [Chifofsky, 1990].

“El proceso de analizar el código, documentación y comportamiento de un sistema para identificar sus componentes actuales y sus dependencias para extraer y crear una abstracción del sistema e información de diseño. El sistema en estudio no es alterado, sino que se produce conocimiento adicional acerca del sistema” [SEI, 2004].

A partir de estas definiciones podemos declarar que la ingeniería inversa tiene la misión de desentrañar los misterios y secretos de los sistemas en uso. Consiste principalmente en recuperar el diseño de una aplicación a partir del código. Esto se realiza principalmente mediante herramientas que extraen información de los datos, procedimientos y arquitectura del sistema existente. El objetivo primordial es proporcionar una base para el mantenimiento y futuros desarrollos. Este objetivo general se puede traducir en los siguientes objetivos parciales:

- Reducir los errores y los costes del mantenimiento.
- Hacer los sistemas más fáciles de entender, cambiar y probar.
- Proteger y extender la vida del sistema.
- Facilitar la reutilización de componentes del sistema existentes.
- Proporcionar documentación que no existe, o actualizar la existente.
- Migrar a otra plataforma hardware o software, cuando sea necesario.
- Llevar el sistema bajo el control de un entorno CASE.

A la vista de estos objetivos, los sistemas candidatos a aplicarles la Ingeniería Inversa reúnen algunas de las siguientes características:

- Las especificaciones de diseño y la documentación, no existen o están incompletas.
- El código no es estructurado.

- Inexistencia de documentación interna en los programas, o bien ésta es incomprensible o está desfasada.
- El sistema necesita un excesivo mantenimiento correctivo.
- Algunos módulos se han hecho excesivamente complejos debido a los sucesivos cambios realizados en ellos.
- Se necesita una migración hacia una nueva plataforma de hardware o de software.
- La aplicación está sujeta a cambios frecuentes, que pueden afectar a parte del diseño.
- Se prevé que la aplicación pueda tener aún larga vida.

Aplicar la Ingeniería Inversa supone un enorme esfuerzo y, por tanto, se hace necesario evaluar exhaustivamente y siendo muy realistas los casos en los que es rentable su aplicación. Su resultado varía fuertemente en función de los siguientes elementos:

- **El nivel de abstracción** del proceso de Ingeniería Inversa, y de las herramientas que se usen. Esto alude a la sofisticación de la información de diseño que se puede extraer del código fuente. El nivel de abstracción ideal deberá ser lo más alto posible. Esto es, el proceso de ingeniería inversa deberá ser capaz de derivar sus representaciones de diseño de procedimientos (con un bajo nivel de abstracción); y la información de las estructuras de datos y de programas (un nivel de abstracción ligeramente más elevado); modelos de flujo de datos y de control (un nivel de abstracción relativamente alto); y modelos de entidades y de relaciones (un elevado nivel de abstracción). A medida que crece el nivel de abstracción se proporciona al ingeniero del software información que le permitirá comprender más fácilmente estos programas.
- **La completitud** del proceso. La completitud de un proceso de ingeniería inversa alude al nivel de detalle que se proporciona en un determinado nivel de abstracción. En la mayoría de los casos, la

completitud decrece a medida que aumenta el nivel de abstracción. Por ejemplo, dado un listado del código fuente, es relativamente sencillo desarrollar una representación de diseño de procedimientos completa. También se pueden derivar representaciones sencillas del flujo de datos, pero es mucho más difícil desarrollar un conjunto completo de diagramas de flujo de datos o un diagrama de transición de estados. La completitud mejora en proporción directa a la cantidad de análisis efectuado por la persona que está efectuando la ingeniería inversa.

- **La interactividad del proceso.** La interactividad alude al grado con el cual el ser humano se “integra” con las herramientas automatizadas para crear un proceso de ingeniería inversa efectivo. En la mayoría de los casos, a medida que crece el nivel de abstracción, la interactividad deberá incrementarse, o sino la completitud se verá reducida.
- **La direccionalidad** del proceso. Si la direccionalidad del proceso de ingeniería inversa es monodireccional, toda la información extraída del código fuente se proporcionará a la ingeniería del software que podrá entonces utilizarla durante la actividad de mantenimiento. Si la direccionalidad es bidireccional, entonces la información se suministrará a una herramienta de reingeniería que intentará reestructurar o regenerar el viejo programa.

La Ingeniería inversa no implica la modificación del sistema, ni la generación de nuevos sistemas, sin embargo, existen una serie de técnicas intrínsecamente relacionadas con ella:

- **Redocumentación:** es la producción de una representación semántica de un sistema a cualquier nivel de abstracción que se requiera. Las herramientas usadas parten del código fuente existente, para producir diagramas de flujo de datos, modelos de datos, etc. Si la redocumentación toma la forma de modificación de

comentarios en el código fuente, puede ser considerada una forma suave de reestructuración. Si se piensa en ella como una transformación desde el código fuente a pseudocódigo y/o prosa, esta última es considerada como de más alto nivel de abstracción que la primera.

- **Recuperación del diseño:** es un subconjunto de la ingeniería inversa, en el cual, aparte de las observaciones del sistema, se añaden conocimientos sobre su dominio de aplicación, información externa, y procesos deductivos con el objeto de identificar abstracciones significativas a un mayor nivel.
- **Reestructuración:** La transformación desde una forma de representación a otra en el mismo nivel de abstracción, preservando las características externas del sistema (funcionalidad y semántica). [Chifofsky, 1990]. La reestructuración del software modifica el código fuente y/o los datos en un intento de adecuarlo a futuros cambios. En general, la reestructuración no modifica la arquitectura global del programa. Tiene a centrarse en los detalles de diseño de módulos individuales y en estructuras de datos locales definidas dentro de los módulos. Si el esfuerzo de la reestructuración se extiende más allá de los límites de los módulos y abarca la arquitectura del software, la reestructuración pasa a ser ingeniería directa (forward engineering). Arnold indica que los beneficios que se pueden lograr con la reestructuración del software son el obtener programas de mayor calidad, mejorar la productividad de los ingenieros del software, reducir el esfuerzo requerido para llevar a cabo actividades de mantenimiento, y hacer que el software sea más sencillo de comprobar y de depurar.
- **Reingeniería:** La reingeniería parte de los resultados obtenidos en la ingeniería inversa para reconstruir el sistema mediante ingeniería hacia adelante. La Reingeniería no sólo recupera la información de diseño de un software existente, sino que usa ésta para alterar o

reconstruir el sistema existente, en un esfuerzo por mejorar la calidad general.

Chikofsky define la **reingeniería** como el *examen y alteración de un sistema para reconstruirlo de una nueva forma y la subsiguiente implementación de esta nueva forma*». Arnold, por su parte, señala que reingeniería es *cualquier actividad que mejore nuestro entendimiento sobre el software y prepare o mejore el propio software, normalmente para su facilidad de mantenimiento, reutilización o evolución*. La definición dada por el Reengineering Center del Software Engineering Institute es *la transformación sistemática de un sistema existente a una nueva forma para realizar mejoras de la calidad en operación, capacidad del sistema, funcionalidad, rendimiento o capacidad de evolución a bajo coste, con un plan de desarrollo corto y con bajo riesgo para el cliente*.

La importancia de las técnicas de reingeniería del software estriban en que reducen los riesgos evolutivos de una organización, ayudan a las organizaciones a recuperar sus inversiones en software, hacen el software más fácilmente modificable, amplían la capacidad de las herramientas CASE y son un catalizador para la automatización del mantenimiento del software.

Para algunos autores, la reingeniería de sistemas puede clasificarse según los niveles de conocimientos requeridos para llevar a cabo el proyecto. La reingeniería que requiere conocimientos a bajos niveles de abstracción (código fuente) se llama **Ingeniería Inversa o Modernización de Caja Blanca** y aquella que sólo requiere el conocimiento de las interfaces del sistema se llama **Reingeniería** propiamente dicha o **Modernización de Caja Negra**.

Para realizar la reingeniería en los sistemas existentes (también llamados **legacy systems** o **sistemas heredados**) se emplean técnicas métricas, de visualización de programas, de abstracción y reformulación del código.

Tanto para reingeniería como para ingeniería inversa, se crean patrones para la resolución de problemas relacionados con estas técnicas. En base al conocimiento del sistema, los datos, las funcionalidades y las interfases, se desarrollan nuevas técnicas de reingeniería no basadas en el conocimiento del código sino en el examen del comportamiento de las entradas y salidas del sistema, desarrollando nuevos patrones de reingeniería y sentando las bases de la reingeniería basada en wrapping. Idealmente, **wrapping** es una reingeniería en la que sólo se analizan las interfases (las entradas y salidas) del sistema existente ignorando los detalles internos. Esta solución no es aplicable siempre y a veces requiere el concurso de la ingeniería inversa para el conocimiento interno del sistema.

Veremos en los siguientes apartados distintos modelos de reingeniería:

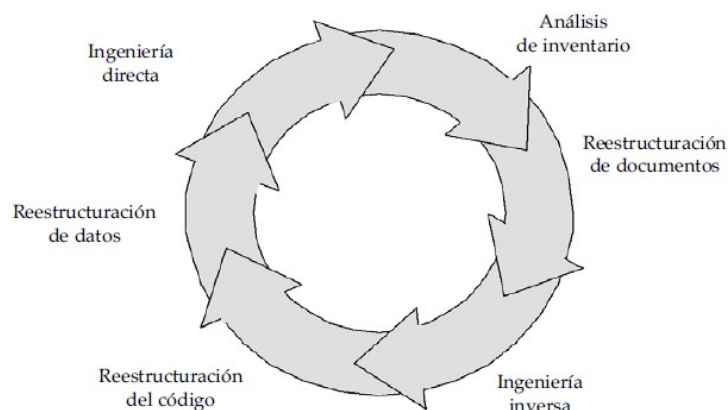
#### *34.4.1 Modelo Cíclico*

El modelo cíclico, debido a Pressman, concibe la reingeniería como un proceso compuesto por seis actividades las cuales se producen, generalmente, de forma secuencial y lineal. Las actividades que se definen en el modelo cíclico son:

- **Análisis de inventario.** Todas las organizaciones de software deberán disponer de un inventario de todas sus aplicaciones, a fin de poder identificar los candidatos a la reingeniería.
- **Reestructuración de documentos.** Una documentación escasa es la marca de muchos sistemas de información heredados. Ante ello será necesario, o bien crear la documentación, o actualizar la existente o hacerla nueva por completo.
- **Ingeniería Inversa.** La Ingeniería Inversa del software es el proceso de análisis de un programa con el fin de crear una representación de programa con un nivel de abstracción más elevado que el código fuente. La Ingeniería Inversa se extraerá del programa existente

información del diseño arquitectónico y de proceso, e información de los datos.

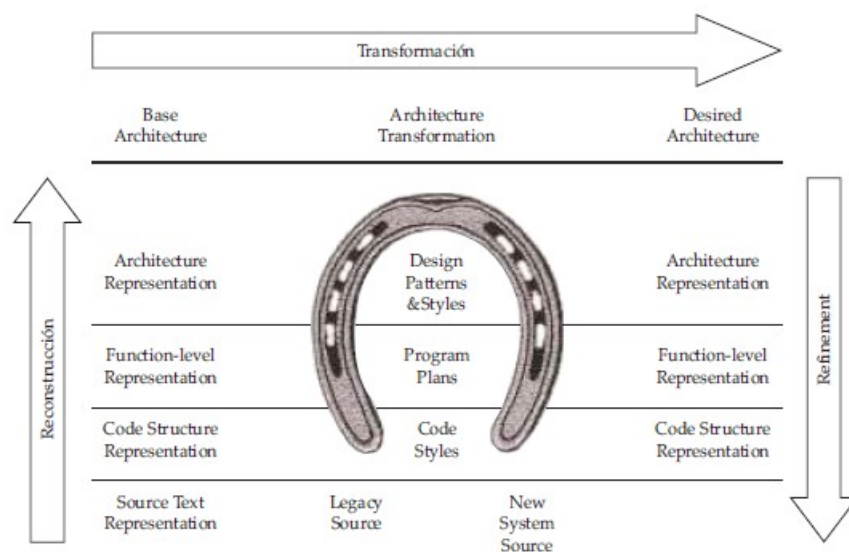
- **Reestructuración del código.** El tipo más común de reingeniería es la reestructuración del código. Algunos sistemas heredados tienen una arquitectura de programa relativamente sólida, pero los módulos individuales han sido codificados de una forma que hace difícil comprenderlos, comprobarlos y mantenerlos. En estos casos, se puede reestructurar el código ubicado dentro de los módulos sospechosos.
- **Reestructuración de datos.** Un programa que posea una estructura de datos débil será difícil de adaptar y de mejorar. De hecho, para muchas aplicaciones, la arquitectura de datos tiene más que ver con la viabilidad a largo plazo del programa que el propio código fuente. A diferencia de la reestructuración de código, que se produce en un nivel relativamente bajo de abstracción, la estructuración de datos es una actividad de reingeniería a gran escala.
- **Ingeniería directa (forward engineering).** La ingeniería directa no solamente recupera la información de diseño de un software ya existente, sino que, además, utiliza esta información en un esfuerzo por mejorar su calidad global. En la mayoría de los casos, el software procedente de una reingeniería vuelve a implementar la funcionalidad del sistema existente, y añade además nuevas funciones y/o mejora el rendimiento global.



### 34.4.2 El modelo de herradura

El modelo de herradura se fundamenta en considerar tres niveles de abstracción en todo sistema y que la reingeniería está formada por tres procesos básicos:

- **Análisis de un sistema existente:** sube el extremo izquierdo de la herradura, y recupera la arquitectura por medio de la extracción de artefactos desde el código fuente.
- **Transformación lógica:** cruza la parte superior y es la transformación de arquitectura. La arquitectura antes construida es recuperada y se reingenieriza para hacer la nueva arquitectura deseable
- **Desarrollo de un nuevo sistema:** baja por el extremo derecho de la herradura, y construye la nueva arquitectura deseable.



La riqueza del modelo de herradura son los tres niveles de abstracción que pueden ser adoptados para las descripciones lógicas, las cuales pueden ser artefactos tan concretos y simples como el código fuente del sistema o tan complejos y abstractos como la arquitectura del sistema. Los tres niveles que adopta el modelo de herradura son:



- **Representación de la estructura de código**, el cual incluye código fuente y artefactos tales como árboles de sintaxis abstractos y diagramas de flujo.
- **Representación del nivel funcional**, el cual describe la relación entre las funciones del programa (llamadas), datos (funciones y relaciones de datos), y archivos (agrupamiento de funciones y datos).
- **Nivel conceptual**, el cual representa grupo tanto de funciones y artefactos del nivel de código que son ensamblados dentro de subsistemas de componentes relacionados o conceptos.

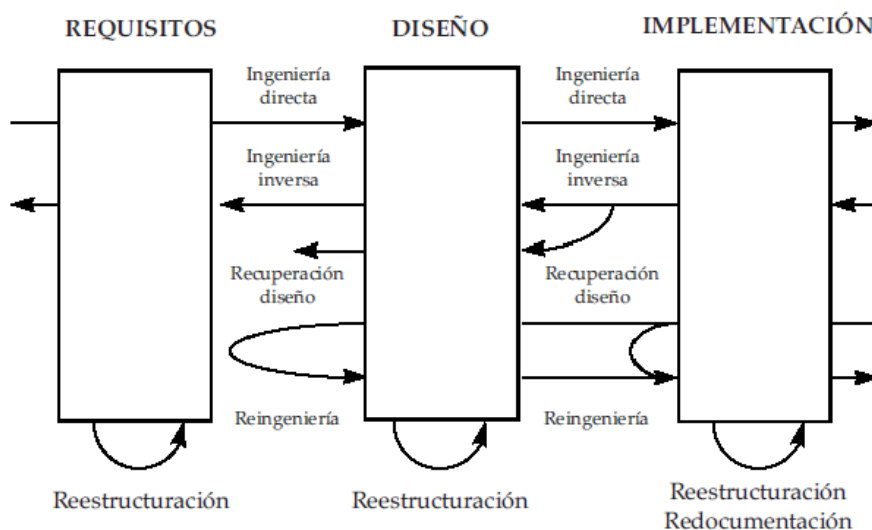
#### 34.4.3 El modelo del IEEE

Este modelo se basa en considerar tres niveles de abstracción en todo sistema, el nivel requisitos, el nivel diseño y el nivel implementación, y en fijar una terminología. Los conceptos que define el IEEE, basados en las definiciones de Chifofsky y Cross, son los siguientes:

- **Ingeniería Inversa (Reverse Engineering)**: es el proceso de analizar un sistema para identificar los componentes y las interrelaciones entre ellos, creando representaciones del sistema en otra forma distinta a la original o bien a un nivel superior de abstracción.
- **Reingeniería (Reengineering)**: es el examen y modificación de un sistema para ser reconstruido de una forma nueva y además realizar la implantación derivada de esa nueva forma. La Reingeniería normalmente incluye alguna forma de Ingeniería Inversa y va seguida de alguna forma de Ingeniería «hacia adelante» o también de una Reestructuración.
- **Reestructuración (Restructuring)**: es la transformación de una forma de representación del sistema en otra distinta, pero del mismo nivel de abstracción, sin modificar el comportamiento externo del sistema.

- **Ingeniería Hacia Adelante (Forward Engineering):** es el proceso que va desde un alto nivel de abstracción, que es independiente de la implementación concreta, hasta la propia implementación física del sistema. Es decir, es la Ingeniería del Software en su vertiente restringida al nuevo desarrollo.
- **Reingeniería de Empresas (Business Process Reengineering):** es la aplicación del concepto de Reingeniería al campo económico, y se desarrolla alrededor de tres actividades clave: rediseñando los procesos básicos de trabajo para alcanzar los objetivos del negocio; utilizando las nuevas tecnologías para concebir, diseñar y poner en marcha nuevas actividades; y cambiando la forma en la que trabajan los empleados.

Las relaciones entre las definiciones y los niveles de abstracción son los que se ven en la figura.

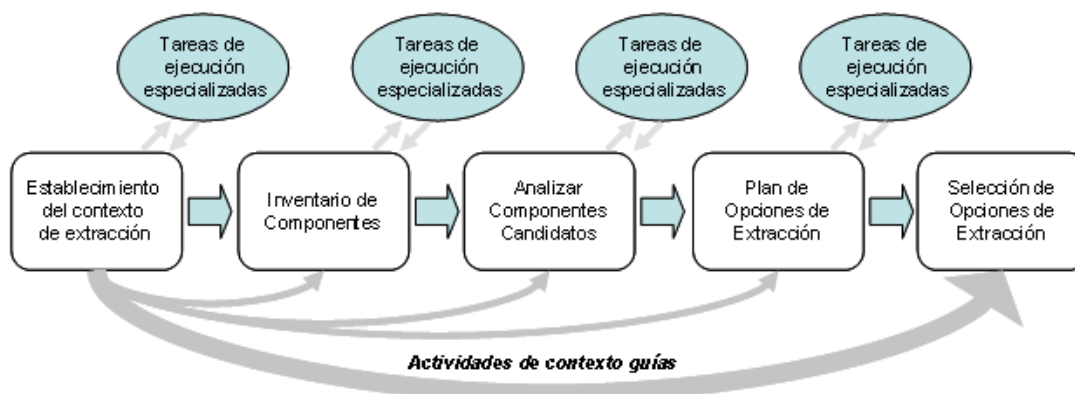


#### 34.4.4 El método Análisis de Opciones para Reingeniería ("Options Analysis for Reengineering" (OAR))

Es un método sistemático para la identificación y extracción de componentes dentro de grandes y complejos sistemas de software. OAR identifica componentes de arquitectura potencialmente relevantes y

analiza los cambios requeridos para usarlos en una línea de producción de software o nuevas arquitecturas de software. En esencia, OAR proporciona un conjunto de opciones de extracción junto con estimación de costos, esfuerzo y riesgos asociados con estas opciones. El método OAR consiste en cinco actividades principales:

- **Establecimiento del contexto de extracción:** consiste en entrevistar a los accionistas y estudiar la línea de producción de la organización o nuevos requerimientos de sistema, base heredada y expectativas para la extracción de componentes heredados. Estos esfuerzos establecen una línea base de un conjunto de metas, expectativas y necesidades de componentes.
- **Inventario De Componentes:** el equipo OAR identifica los componentes del sistema heredado que potencialmente pueden ser extraídos para usarlos en una línea de producción o en una nueva arquitectura de software. Esta actividad resulta en un inventario de los componentes heredados candidatos.
- **Análisis de componentes candidatos:** El siguiente paso de los miembros del equipo es analizar el conjunto de candidatos de componentes heredados para extraer los tipos de cambios que son requeridos.
- **Plan de opciones de extracción:** Dado el conjunto de componentes candidatos analizados, el equipo desarrollará alternativas para la extracción basada en consideraciones de calendario, costo, esfuerzo, riesgo y recursos. El equipo OAR también filtra una vez más los componentes candidatos y analiza el impacto de agregación de diferentes componentes.
- **Selección de opciones de extracción:** Finalmente, los miembros del equipo seleccionan la mejor opción de extracción o combinación de opciones. Después de evaluar cada opción de extracción, ellos preparan un resumen en el que presentan y justifican sus elecciones.



## Bibliografía

- Ingeniería del Software. Un enfoque práctico.
- Algoritmos y estructuras de datos.
- Reverse engineering and design recovery: A taxonomy. E. Chikofsky y J. Cross. Ed. IEEE Software.
- Software Reengineering. R. S. Arnold. Ed. IEEE Computer Society Press.
- Técnicas de programación. Instituto nacional de estadística e informática.
- <http://www.authorstream.com/Presentation/verarex-33544-evolucion-de-los-lenguajes-programaci-evolucionlp-education-ppt-powerpoint/>. Evolución de los lenguajes de programación. Denis Cedeño.

Reconstrucción de la arquitectura: Una actividad de la reingeniería de software. Flores Carmona Nicolás Johnatan

**[http://www.cybertesis.edu.pe/sisbib/2007/acevedo\\_rj/pdf/acevedo\\_rj.pdf](http://www.cybertesis.edu.pe/sisbib/2007/acevedo_rj/pdf/acevedo_rj.pdf)**. Ingeniería inversa aplicado a sistemas desarrollados con programación orientada a objetos para obtener la documentación.  
Jessica Jahany Acevedo Ricse

**Autor: Hernán Vila Pérez**

**Jefe del Servicio de Informática. Instituto Galego de Vivenda e Solo**

**Vicepresidente del CPEIG**

# **35. MÉTODOS DE PRUEBA DEL SOFTWARE. FUNDAMENTOS. ESTRATEGIA DE PRUEBA DEL SOFTWARE: VERIFICACIÓN Y VALIDACIÓN. CAJA NEGRA Y CAJA BLANCA. PRUEBAS DE CONTORNO Y APLICACIONES ESPECIALIZADAS. PRUEBAS FUNCIONALES. PRUEBAS DE INTEGRACIÓN. PRUEBAS DE REGRESIÓN. PRUEBAS DE VALIDACIÓN.**

**Tema 35. Métodos de prueba del software. Fundamentos. Estrategia de prueba del software. Verificación y validación. Caja negra y caja blanca. Pruebas de entorno y aplicaciones especializadas. Pruebas funcionales. Pruebas de integración. Pruebas de regresión. Pruebas de validación.**

### **35.1 Métodos de prueba del software. Fundamentos. Verificación y validación**

#### **35.1.1 Principios de las pruebas**

### **35.2 Estrategias de prueba del software**

#### **35.2.1 Pruebas de Unidad**

#### **35.2.2 Pruebas de integración.**

#### **35.2.3 Pruebas de regresión.**

#### **35.2.4 Pruebas del sistema**

#### **35.2.5 Pruebas de implantación**

#### **35.2.6 Pruebas de Validación.**

### **35.3 Pruebas de Caja negra y Caja blanca**

#### **35.3.1 Pruebas de Caja blanca**

##### **35.3.1.1 Prueba del camino básico**

##### **35.3.1.2 Prueba de la estructura de control**

##### **Prueba de condiciones**

##### **Prueba de flujo de datos**

##### **Prueba de bucles**

#### **35.3.2 Pruebas de Caja Negra**

##### **35.3.2.1 Partición equivalente**

##### **35.3.2.2 Análisis de valores límite**

##### **35.3.2.3 Valores típicos de error y valores imposibles**

##### **35.3.2.4 Basados en grafos**

##### **35.3.2.5 Tabla Ortogonal**

##### **35.3.2.6 Prueba de comparación**

### **35.4 Pruebas de entorno y aplicaciones especializadas**

#### **35.4.1. Prueba de interfaces gráficas de usuario**

#### **35.4.2 Prueba de arquitecturas cliente/servidor**

#### **35.4.3 Prueba de la documentación y facilidades de ayuda**

#### **35.4.4 Prueba de sistemas de tiempo-real**

### **35.5 Pruebas funcionales**

#### **35.1 Fundamentos de las pruebas del software. Verificación y Validación**

La prueba del software es un elemento de un tema más amplio que, a menudo, es conocido como **verificación** y **validación** (V&V). La **verificación** se refiere al conjunto de actividades que aseguran que el software implementa correctamente una función específica. La **validación** se refiere a un conjunto diferente de actividades que aseguran que el software construido se ajusta a los requisitos del cliente.

- **Verificación:** ¿Estamos construyendo el producto correctamente
- **Validación:** ¿Estamos construyendo el producto correcto?

La verificación y la validación abarcan una amplia lista de actividades de Garantía de Calidad del Software (SQA) que incluye: revisiones técnicas formales, auditorías de calidad y de configuración, monitorización de rendimientos, simulación, estudios de factibilidad, revisión de la documentación, revisión de la base de datos, análisis algorítmico, pruebas de desarrollo, pruebas de validación y pruebas de instalación. A pesar de que las actividades de prueba tienen un papel muy importante en la Verificación y Validación, muchas otras actividades son también necesarias. La aplicación adecuada de los métodos y de las herramientas, las revisiones técnicas formales efectivas y una sólida gestión y medición, conducen a la calidad, que se confirma durante las pruebas.

La prueba presenta una anomalía para el ingeniero del software, ya que, mientras en las fases anteriores de definición y desarrollo, el ingeniero intenta construir, al llegar las pruebas, el ingeniero diseña una serie de casos de prueba que pretenden “demoler” lo construido. Por eso los autores dicen que la prueba se puede ver como destructiva, en lugar de constructiva.

El proceso de pruebas del software tiene dos objetivos distintos:

- Para demostrar al desarrollador y al cliente que el software satisface sus requerimientos. Para el software a medida, esto significa que debería haber al menos una prueba para cada requerimiento de los documentos de requerimientos del sistema y del usuario. Para productos de software genéricos, significa que debería haber pruebas para todas las características del sistema que se incorporarán en la entrega del producto.
- Para descubrir errores en el software en que el comportamiento de éste es incorrecto, no deseable o no cumple su especificación. La prueba de errores está relacionada con la eliminación de todos los tipos de comportamientos del sistema no deseables, tales como caídas del sistema, interacciones no permitidas con otros sistemas, cálculos incorrectos y corrupción de datos.

De las muchas definiciones válidas de la “prueba del software” podemos quedarnos con:

- La prueba es el proceso de ejecución de un programa con la intención de descubrir errores (Glenford Myers).
- La prueba es cualquier actividad dirigida a evaluar un la capacidad de un programa y determinar que alcanza los resultados requeridos.

### ***35.1.1 Principios de las pruebas***

Conocida la definición de prueba, algunos de los principios que les afectan son:



- La prueba es el proceso de ejecutar un programa con la intención de descubrir errores, por lo que un buen caso de prueba es aquel que tiene una alta probabilidad de descubrir un error no encontrado hasta entonces.
- Las pruebas deben planificarse antes de que se empiece a codificar. Así la planificación comienza con el modelo de requisitos, y la definición detallada de los casos de prueba una vez que el diseño del sistema está consolidado.
- El 80% de los errores estarán localizados en el 20% de los módulos.
- La prueba completa es imposible.
- Para ser más eficaces, las pruebas deberían ser hechas por un equipo independiente.
- La probabilidad de la existencia de más errores en una parte del software es proporcional al número de errores ya encontrados en dicha parte.

### **35.2 Estrategias de prueba del software.**

Una estrategia de prueba del software integra las técnicas de diseño de casos de prueba en una serie de pasos bien planificados que dan como resultado una correcta construcción del software. La estrategia proporciona un mapa que describe los pasos que hay que llevar a cabo como parte de la prueba, cuándo se deben planificar y realizar esos pasos, y cuánto esfuerzo, tiempo y recursos se van a requerir. Por tanto, cualquier estrategia de prueba debe incorporar la planificación de la prueba, el diseño de casos de prueba, la ejecución de las pruebas y la agrupación y evaluación de los datos resultantes.

Una estrategia de prueba del software debe ser suficientemente flexible para promover la creatividad y la adaptabilidad necesarias para adecuar la prueba a todos los grandes sistemas basados en software. Al mismo

tiempo, la estrategia debe ser suficientemente rígida para promover un seguimiento razonable de la planificación y la gestión a medida que progresa el proyecto.

Se han propuesto varias estrategias de prueba del software en distintos libros. Todas proporcionan al ingeniero del software una plantilla para la prueba y todas tienen las siguientes características generales:

- Las pruebas comienzan a nivel de módulo y trabajan «hacia fuera», hacia la integración de todo el sistema.
- Según el momento, son apropiadas diferentes técnicas de prueba.
- La prueba la lleva a cabo el responsable del desarrollo del software y (para grandes proyectos) un grupo independiente de pruebas.
- La prueba y la depuración son actividades diferentes, pero la depuración se debe incluir en cualquier estrategia de prueba.

Una estrategia de prueba del software debe incluir pruebas de bajo nivel que verifiquen que todos los pequeños segmentos de código fuente se han implementado correctamente, así como pruebas de alto nivel que validen las principales funciones del sistema frente a los requisitos del cliente. Una estrategia debe proporcionar una guía al profesional y proporcionar un conjunto de hitos para el jefe de proyecto.

Se deben abordar los siguientes puntos si se desea implementar con éxito una estrategia de prueba del software:

- Especificar los requisitos del producto de manera cuantificable mucho antes de que comiencen las pruebas.
- Establecer los objetivos de la prueba de manera explícita.
- Comprender qué usuarios van a manejar el software y desarrollar un perfil para cada categoría de usuario.
- Desarrollar un plan de prueba que haga hincapié en la «prueba de ciclo rápido»
- Construir un software «robusto» diseñado para probarse a sí mismo

- Usar revisiones técnicas formales efectivas como filtro antes de la prueba.
- Llevar a cabo revisiones técnicas formales para evaluar la estrategia de prueba y los propios casos de prueba.
- Desarrollar un enfoque de mejora continua al proceso de prueba

Si consideramos el proceso desde el punto de vista procedimental, la prueba, en el contexto de la ingeniería del software, realmente es una serie de cinco pasos que se llevan a cabo secuencialmente. Inicialmente, la prueba se centra en cada módulo individualmente, asegurando que funcionan adecuadamente como una unidad. De ahí el nombre de *prueba de unidad*. La prueba de unidad hace un uso intensivo de las técnicas de prueba de caja blanca (se verán con posterioridad en otro apartado), ejercitando caminos específicos de la estructura de control del módulo para asegurar un alcance completo **y** una detección máxima de errores. A continuación, se deben ensamblar o integrar los módulos para formar el paquete de software completo. La *prueba de integración* se dirige a todos los aspectos asociados con el doble problema de verificación **y** de construcción del programa. Durante la integración, las técnicas que más prevalecen son las de diseño de casos de prueba de caja negra (se verán con posterioridad en otro apartado), aunque se pueden llevar a cabo algunas pruebas de caja blanca con el fin de asegurar que se cubren los principales caminos de control. Después de que el software se ha integrado (construido), se dirigen un conjunto de *pruebas de alto nivel*. Se debe comprobar que el software, al combinarlo con otros elementos del sistema (por ejemplo, hardware, gente, bases de datos), cada elemento encaja de forma adecuada y que se alcanza la funcionalidad y el rendimiento exigido. Esta es la base de *la prueba de sistemas*. Posteriormente se asegurará mediante la *prueba de implantación* el funcionamiento correcto del sistema integrado de hardware y software en el entorno de operación, y permitir al usuario que, desde el punto de vista de operación, realice la aceptación del sistema una vez instalado en su entorno real y en base al cumplimiento de

los requisitos no funcionales especificados. La *prueba de aceptación o de validación* proporciona una seguridad final de que el software satisface todos los requisitos funcionales, de comportamiento y de rendimiento. Durante las últimas tres pruebas (*sistemas, implantación y aceptación*) se usan exclusivamente técnicas de prueba de caja negra.

### **35.2.1 Pruebas de Unidad**

Las pruebas unitarias tienen como objetivo verificar la funcionalidad y estructura de cada componente individualmente una vez que ha sido codificado.

La **prueba de unidad** es un proceso para probar los subprogramas, las subrutinas, los procedimientos individuales o las clases en un programa. Es decir, es mejor probar primero los bloques desarrollados más pequeños del programa, que inicialmente probar el software en su totalidad. Las motivaciones para hacer esto son tres. Primera, las pruebas de unidad son una manera de manejar los elementos de prueba combinados, puesto que se centra la atención inicialmente en unidades más pequeñas del programa. En segundo lugar, la prueba de una unidad facilita la tarea de eliminar errores (el proceso de establecer claramente y de corregir un error descubierto), puesto que, cuando se encuentra un error, se sabe que existe en un módulo particular. Finalmente, las pruebas de unidad introducen paralelismo en el proceso de pruebas del software presentándose la oportunidad de probar los múltiples módulos simultáneamente.

Se necesita dos tipos de información al diseñar los casos de prueba para una prueba de unidad: la especificación para el módulo y el código fuente del módulo. La especificación define típicamente los parámetros de entrada y de salida del módulo y su función.

Las pruebas de unidad son en gran parte orientadas a caja blanca. Una razón es que como en pruebas de entidades más grandes tales como programas enteros (es el caso para los procesos de prueba subsecuentes), la prueba de caja blanca llega a ser menos factible. Una segunda razón es que los procesos de prueba subsecuentes están orientados a encontrar diversos tipos de errores. Por lo tanto, el procedimiento para el diseño de casos de prueba para una prueba de unidad es la siguiente: analizar la lógica del módulo usando uno o más de los métodos de caja blanca y después completar los casos de prueba aplicando métodos de caja negra a la especificación del módulo.

### ***35.2.2 Pruebas de integración.***

El objetivo de las **pruebas de integración** es verificar el correcto ensamblaje entre los distintos componentes una vez que han sido probados unitariamente con el fin de comprobar que interactúan correctamente a través de sus interfaces, tanto internas como externas, cubren la funcionalidad establecida y se ajustan a los requisitos no funcionales especificados en las verificaciones correspondientes.

En las pruebas de integración se examinan las interfaces entre grupos de componentes o subsistemas para asegurar que son llamados cuando es necesario y que los datos o mensajes que se transmiten son los requeridos. Debido a que en las pruebas unitarias es necesario crear módulos auxiliares que simulen las acciones de los componentes invocados por el que se está probando y a que se han de crear componentes "conductores" para establecer las precondiciones necesarias, llamar al componente objeto de la prueba y examinar los resultados de la prueba, a menudo se combinan los tipos de prueba unitarias y de integración.

Los tipos fundamentales de integración son los siguientes:

- *Integración incremental:* se combina el siguiente componente que se debe probar con el conjunto de componentes que ya están probados y se va incrementando progresivamente el número de componentes a probar. Con el tipo de prueba incremental lo más probable es que los problemas que surjan al incorporar un nuevo componente o un grupo de componentes previamente probado, sean debidos a este último o a las interfaces entre éste y los otros componentes.
- *Integración no incremental:* se prueba cada componente por separado y posteriormente se integran todos de una vez realizando las pruebas pertinentes. Este tipo de integración se denomina también **Big-Bang**.

Dentro de la *integración incremental*, tenemos tres tipos de estrategias:

- *Estrategia Descendente (top-down):* El primer componente que se prueba es el primero de la jerarquía. Los componentes de nivel más bajo se sustituyen por componentes auxiliares llamados **resguardos** para simular a los componentes invocados. Luego se van sustituyendo los resguardos subordinados por los componentes reales. Ventajas: Las interfaces entre los distintos componentes se prueban en una fase temprana y con frecuencia. Verifica los puntos de decisión o de control principales al principio del proceso de prueba.
- *Estrategia Ascendente (bottom-up):* En este caso se crean primero los componentes de más bajo nivel y se crean componentes **conductores o controladores** para simular a los componentes que los llaman. A continuación se sustituyen los controladores por los módulos desarrollados de más alto nivel y se prueban.
- *Estrategia combinada:* A menudo es útil aplicar las estrategias anteriores conjuntamente. De este modo, se prueban las partes principales del sistema con un enfoque **top-down**, mientras que las

partes de nivel más bajo se prueban siguiendo un enfoque **bottom-up**.

### ***35.2.3 Pruebas de regresión***

Cada vez que se añade un nuevo módulo como parte de una prueba de integración, el software cambia. Se establecen nuevos caminos de flujo de datos, pueden ocurrir nuevas E/S y se invoca una nueva lógica de control. Estos cambios pueden causar problemas con funciones que antes trabajaban perfectamente. En este contexto, la **prueba de regresión** consiste en volver a ejecutar un subconjunto de pruebas que se han llevado a cabo anteriormente para asegurarse de que los cambios no han propagado efectos colaterales no deseados.

En un contexto más amplio, las pruebas con éxito (de cualquier tipo) dan como resultado el descubrimiento de errores, y los errores hay que corregirlos. Cuando se corrige el software, se cambia algún aspecto de la configuración del software (el programa, su documentación o los datos que lo soportan). La prueba de regresión es la actividad que ayuda a asegurar que los cambios (debidos a las pruebas o por otros motivos) no introducen un comportamiento no deseado o errores adicionales. La prueba de regresión se puede hacer manualmente, volviendo a realizar un subconjunto de todos los casos de prueba o utilizando herramientas automáticas de reproducción de captura. Las herramientas de reproducción de captura permiten al ingeniero del software capturar casos de prueba y los resultados para la subsiguiente reproducción y comparación.

### ***35.2.4 Pruebas de sistema***

Las pruebas del sistema tienen como objetivo ejercitar profundamente el sistema comprobando la integración del sistema de información

globalmente, verificando el funcionamiento correcto de las interfaces entre los distintos subsistemas que lo componen y con el resto de sistemas de información con los que se comunica.

Una vez que se han probado los componentes individuales y se han integrado, se prueba el sistema de forma global. En esta etapa pueden distinguirse los siguientes tipos de pruebas, cada uno con un objetivo claramente diferenciado:

- **Pruebas funcionales.** Dirigidas a asegurar que el sistema de información realiza correctamente todas las funciones que se han detallado en las especificaciones dadas por el usuario del sistema.
- **Pruebas de comunicaciones.** Determinan que las interfaces entre los componentes del sistema funcionan adecuadamente, tanto a través de dispositivos remotos, como locales. Asimismo, se han de probar las interfaces hombre/máquina.
- **Pruebas de rendimiento.** Consisten en determinar que los tiempos de respuesta están dentro de los intervalos establecidos en las especificaciones del sistema.
- **Pruebas de volumen.** Consisten en examinar el funcionamiento del sistema cuando está trabajando con grandes volúmenes de datos, simulando las cargas de trabajo esperadas.
- **Pruebas de sobrecarga.** Consisten en comprobar el funcionamiento del sistema en el umbral límite de los recursos, sometiéndole a cargas masivas. El objetivo es establecer los puntos extremos en los cuales el sistema empieza a operar por debajo de los requisitos establecidos.
- **Pruebas de disponibilidad de datos.** Consisten en demostrar que el sistema puede recuperarse ante fallos, tanto de equipo físico como lógico, sin comprometer la integridad de los datos.



- **Pruebas de facilidad de uso.** Consisten en comprobar la adaptabilidad del sistema a las necesidades de los usuarios, tanto para asegurar que se acomoda a su modo habitual de trabajo, como para determinar las facilidades que aporta al introducir datos en el sistema y obtener los resultados.
- **Pruebas de operación.** Consisten en comprobar la correcta implementación de los procedimientos de operación, incluyendo la planificación y control de trabajos, arranque y re arranque del sistema, etc.
- **Pruebas de entorno.** Consisten en verificar las interacciones del sistema con otros sistemas dentro del mismo entorno.
- **Pruebas de seguridad.** Consisten en verificar los mecanismos de control de acceso al sistema para evitar alteraciones indebidas en los datos.
- **Pruebas de configuración.** Programas tales como sistemas operativos, sistemas de gerencia de base de datos, y programas de conmutación de mensajes soportan una variedad de configuraciones de hardware, incluyendo varios tipos y números de dispositivos de entrada-salida y líneas de comunicaciones, o diversos tamaños de memoria. A menudo el número de configuraciones posibles es demasiado grande para probar cada uno, pero en lo posible, se debe probar el programa con cada tipo de dispositivo de hardware y con la configuración mínima y máxima. Si el programa por sí mismo se puede configurar para omitir componentes, o si puede funcionar en diversas computadoras, cada configuración posible de este debe ser probada.
- **Pruebas de instalación.** Al funcionar incorrectamente el programa de instalación podría evitar que el usuario tenga una experiencia acertada con el sistema. La primera experiencia de un usuario es cuando él o ella instala la aplicación. Si esta fase se realiza mal, entonces el usuario/el

cliente puede buscar otro producto o tener poca confianza en la validez de la aplicación.

- **Pruebas de documentación.** La documentación del usuario debe ser el tema de una inspección, comprobándola para saber si hay exactitud y claridad. Cualquiera de los ejemplos ilustrados en la documentación se deben probar y hacer parte de los casos y alimentarlos al programa.

### ***35.2.5 Prueba de implantación***

El objetivo de las **pruebas de implantación** es comprobar el funcionamiento correcto del sistema integrado de hardware y software en el entorno de operación, y permitir al usuario que, desde el punto de vista de operación, realice la aceptación del sistema una vez instalado en su entorno real y en base al cumplimiento de los requisitos no funcionales especificados.

Una vez que hayan sido realizadas las pruebas del sistema en el entorno de desarrollo, se llevan a cabo las verificaciones necesarias para asegurar que el sistema funcionará correctamente en el entorno de operación. Debe comprobarse que responde satisfactoriamente a los requisitos de rendimiento, seguridad, operación y coexistencia con el resto de los sistemas de la instalación para conseguir la aceptación del usuario de operación.

Las pruebas de seguridad van dirigidas a verificar que los mecanismos de protección incorporados al sistema cumplen su objetivo; las de rendimiento a asegurar que el sistema responde satisfactoriamente en los márgenes establecidos en cuanto tiempos de respuesta, de ejecución y de utilización de recursos, así como los volúmenes de espacio en disco y capacidad; por último con las pruebas de operación se comprueba que la planificación y control de trabajos del sistema se realiza de acuerdo a los procedimientos

establecidos, considerando la gestión y control de las comunicaciones y asegurando la disponibilidad de los distintos recursos.

Asimismo, también son llevadas a cabo las pruebas de gestión de copias de seguridad y recuperación, con el objetivo de verificar que el sistema no ve comprometido su funcionamiento al existir un control y seguimiento de los procedimientos de salvaguarda y de recuperación de la información, en caso de caídas en los servicios o en algunos de sus componentes. Para comprobar estos últimos, se provoca el fallo del sistema, verificando si la recuperación se lleva a cabo de forma apropiada. En el caso de realizarse de forma automática, se evalúa la inicialización, los mecanismos de recuperación del estado del sistema, los datos y todos aquellos recursos que se vean implicados.

Las verificaciones de las pruebas de implantación y las pruebas del sistema tienen muchos puntos en común al compartir algunas de las fuentes para su diseño como pueden ser los casos para probar el rendimiento (pruebas de sobrecarga o de stress).

El responsable de implantación junto al equipo de desarrollo determina las verificaciones necesarias para realizar las pruebas así como los criterios de aceptación del sistema. Estas pruebas las realiza el equipo de operación, integrado por los técnicos de sistemas y de operación que han recibido previamente la formación necesaria para llevarlas a cabo.

### **35.2.6 Pruebas de Validación**

El objetivo de las **pruebas de validación**, también conocidas como **pruebas de aceptación** es validar que un sistema cumple con el funcionamiento esperado y permitir al usuario de dicho sistema que determine su aceptación desde el punto de vista de su funcionalidad y rendimiento.

Las pruebas de validación son definidas por el usuario del sistema y preparadas por el equipo de desarrollo, aunque la ejecución y aprobación final corresponden al usuario. Estas pruebas van dirigidas a comprobar que el sistema cumple los requisitos de funcionamiento esperado, recogidos en el catálogo de requisitos y en los criterios de aceptación del sistema de información, y conseguir así la aceptación final del sistema por parte del usuario.

El responsable de usuarios debe revisar los criterios de aceptación que se especificaron previamente en el plan de pruebas del sistema y, posteriormente, dirigir las pruebas de aceptación final. La validación del sistema se consigue mediante la realización de pruebas de caja negra que demuestran la conformidad con los requisitos y que se recogen en el plan de pruebas, el cual define las verificaciones a realizar y los casos de prueba asociados. Dicho plan está diseñado para asegurar que se satisfacen todos los requisitos funcionales especificados por el usuario teniendo en cuenta también los requisitos no funcionales relacionados con el rendimiento, seguridad de acceso al sistema, a los datos y procesos, así como a los distintos recursos del sistema.

La formalidad de estas pruebas dependerá en mayor o menor medida de cada organización, y vendrá dada por la criticidad del sistema, el número de usuarios implicados en las mismas y el tiempo del que se disponga para llevarlas cabo, entre otros.

Si el software se desarrolla como un producto que va a ser usado por muchos clientes, no es práctico realizar pruebas de aceptación formales para cada uno de ellos. La mayoría de los desarrolladores de productos de software llevan a cabo un proceso denominado prueba alfa y beta para descubrir errores que parezca que sólo el usuario final puede descubrir.

La **prueba alfa** se lleva a cabo, por un cliente, en el lugar de desarrollo. Se usa el software de forma natural con el desarrollador como observador del

usuario y registrando los errores y los problemas de uso. Las pruebas alfa se llevan a cabo en un entorno controlado.

La **prueba beta** se lleva a cabo por los usuarios finales del software en los lugares de trabajo de los clientes. A diferencia de la prueba alfa, el desarrollador no está presente normalmente. Así, la prueba beta es una aplicación «en vivo» del software en un entorno que no puede ser controlado por el desarrollador. El cliente registra todos los problemas (reales o imaginarios) que encuentra durante la prueba beta e informa a intervalos regulares al desarrollador. Como resultado de los problemas informados durante la prueba beta, el desarrollador del software lleva a cabo modificaciones y así prepara una versión del producto de software para toda la clase de clientes.

### **35.3 Pruebas de Caja Negra y Caja Blanca.**

Cualquier producto de ingeniería se puede probar de una de estas dos formas:

- Conociendo la función específica para la que fue diseñado el producto, se pueden llevar a cabo pruebas que demuestren que cada función es completamente operativa.
- Conociendo el funcionamiento del producto, se pueden desarrollar pruebas que aseguren que «todas las piezas encajan», o sea, que la operación interna se ajusta a las especificaciones y que todos los componentes internos se han comprobado de forma adecuada.

El primer enfoque de prueba se denomina prueba de **caja negra** y el segundo, prueba de **caja blanca**.

La prueba de **caja negra** se refiere a las pruebas que se llevan a cabo sobre la interfaz del software. O sea, los casos de prueba pretenden demostrar que las funciones del software son operativas, que la entrada se

acepta de forma adecuada y que se produce un resultado correcto, así como que la integridad de la información externa (por ejemplo, archivos de datos) se mantiene. Una prueba de caja negra examina algunos aspectos del modelo fundamental del sistema sin tener mucho en cuenta la estructura lógica interna del software.

La prueba de **caja blanca** del software se basa en el minucioso examen de los detalles procedimentales. Se comprueban los caminos lógicos del software proponiendo casos de prueba que ejerciten conjuntos específicos de condiciones y/o bucles. Se puede examinar el «estado del programa» en varios puntos para determinar si el estado real coincide con el esperado o mencionado.

### **35.3.1 Pruebas de caja blanca**

El enfoque de la estrategia de pruebas conocido por el nombre de método de «**caja blanca**» o, también, «**conducido por la lógica**» o «**logic driven**», se centra en probar el comportamiento interno y la estructura del programa, examinando la lógica interna del mismo y sin considerar los aspectos de rendimiento. El objetivo de este enfoque es ejecutar, al menos una vez, todas las sentencias y ejecutar todas las condiciones tanto en su vertiente verdadera como falsa, teniendo en cuenta que la única información de entrada con que se cuenta es el diseño del programa y el código fuente. Las técnicas específicas más usuales que siguen el método de «caja blanca» son:

- Prueba del camino básico
- Prueba de la estructura de control
  - o Prueba de condiciones
  - o Prueba de flujo de datos
  - o Prueba de bucles

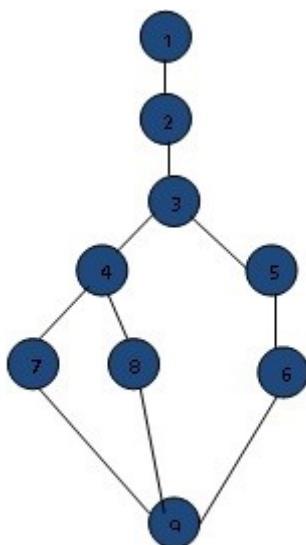
#### **35.3.1.1. Prueba del camino básico**

Fue propuesta inicialmente por Tom McCabe. Permite al diseñador de casos de prueba obtener una medida de la complejidad lógica de un diseño procedimental y usar esa medida como guía para la definición de un conjunto básico de caminos de ejecución. Los casos de prueba obtenidos del conjunto básico garantizan que durante la prueba se ejecuta por lo menos una vez cada sentencia del programa.

De la técnica del camino básico se derivan dos pruebas complementarias y no excluyentes:

- *Prueba de cobertura de sentencias.* Consiste en generar casos de prueba que permitan probar todas y cada una de las sentencias de un módulo una vez. Esta prueba es necesaria pero no es suficiente.
- *Prueba de cobertura de condiciones.* Consiste en diseñar juegos de prueba que consideren todos los valores posibles de cada una de las condiciones. Esta prueba también es necesaria pero no es suficiente ya que no garantiza que todos los caminos sean cubiertos, por lo que debe ser complementada con la anterior.

La técnica de prueba del camino básico utiliza una convención de representación denominada grafos de flujo para la determinación de caminos y para ello se apoya en el concepto de complejidad ciclomática propuesto por Tom McCabe.



La notación de **grafo de flujo**, propuesta por McCabe se utiliza para simplificar el desarrollo del conjunto básico de caminos de ejecución. Su objetivo es ejemplificar el flujo de control del módulo que se está probando y para ello utiliza tres elementos:

- **Nodos** o tareas de procesamiento (N). Representan cero, una o varias sentencias procedimentales. Cada nodo comprende como máximo una sentencia de decisión (bifurcación). Cada nodo que contiene una condición se denomina **nodo predicado** y está caracterizado porque dos o más aristas emergen de el.
- **Aristas**, flujo de control o conexiones (A). Unen dos nodos, incluso aunque el nodo no represente ninguna sentencia procedimental.
- **Regiones** (R). Son las áreas delimitadas por las aristas y nodos. Cuando se contabilizan las regiones debe incluirse el área externa como una región más.

La **complejidad ciclomática** es una métrica del software basada en la teoría de grafos, que proporciona una medición cuantitativa de la complejidad lógica de un programa. Cuando se usa en el contexto del método de prueba del camino básico, el valor calculado como complejidad ciclomática define el número de caminos independientes del conjunto



básico de un programa y nos da un límite superior para el número de pruebas que se deben realizar para asegurar que se ejecuta cada sentencia al menos una vez. Un camino independiente es cualquier camino del programa que introduce, por lo menos, un nuevo conjunto de sentencias de proceso o una nueva condición. En términos del grafo de flujo, un camino independiente está constituido por lo menos por una arista que no haya sido recorrida anteriormente a la definición del camino. El valor de la complejidad se puede obtener de tres formas:

- 1- El número de regiones del grafo. (R)
- 2- El número de aristas menos el número de nodos + 2. ( $A - N + 2$ .)
- 3- Número de nodos predicado + 1. ( $P + 1$ )

La prueba del camino básico nos permite obtener los casos de prueba de la siguiente forma:

- 1- Usando el diseño o el código como base, dibujamos el correspondiente grafo de flujo.
- 2- Determinamos la complejidad ciclomática del grafo de flujo resultante.
- 3- Determinamos un conjunto básico de caminos linealmente independientes.
- 4- Preparamos los casos de prueba que forzarán la ejecución de cada camino del conjunto básico.

### **35.3.1.2 Pruebas de la estructura de control**

Dentro de éste tipo de prueba se contempla el método del camino básico mencionado anteriormente pero además existen otras pruebas asociadas que permiten ampliar la cobertura de la prueba y mejorar su calidad.

#### **35.3.1.2.1 Prueba de condición.**

Es un método de diseño de casos de prueba que ejercita las condiciones lógicas contenidas en el módulo de un programa. Algunos conceptos empleados alrededor de esta prueba son los siguientes:

- *Condición simple*: es una variable lógica o una expresión relacional ( $E_1 < \text{operador - relacional} > E_2$ ).
- *Condición compuesta*: esta formada por dos o más condiciones simples, operadores lógicos y paréntesis.

En general los tipos de errores que se buscan en una prueba de condición, son los siguientes:

- *Error en operador lógico* (existencia de operadores lógicos incorrectos, desaparecidos, sobrantes).
- *Error en variable lógica*.
- *Error en paréntesis lógico*.
- *Error en operador relacional*.
- *Error en expresión aritmética*.

#### **35.3.1.2.2 Prueba del flujo de datos.**

Selecciona caminos de prueba de un programa de acuerdo con la ubicación de las definiciones y los usos de las variables del programa.

#### **35.3.1.2.3. Prueba de bucles**

La prueba de bucles es una técnica de prueba de caja blanca que se centra exclusivamente en la validez de las construcciones de bucles. Se pueden definir cuatro clases diferentes de bucles:

- **Bucles simples.** A los bucles simples se les debe aplicar el siguiente conjunto de pruebas, donde  **$n$**  es el número máximo de pasos permitidos por el bucle:
  1. pasar por alto totalmente el bucle
  2. pasar una sola vez por el bucle
  3. pasar dos veces por el bucle
  4. hacer  $m$  pasos por el bucle con  $m < n$
  5. hacer  $n - 1$ ,  $n$  y  $n+1$  pasos por el bucle
- **Bucles anidados.** Si extendiéramos el enfoque de prueba de los bucles simples a los bucles anidados, el número de posibles pruebas aumentaría geométricamente a medida que aumenta el nivel de anidamiento. Esto llevaría un número impracticable de pruebas. Se sugiere un enfoque que ayuda a reducir el número de pruebas:
  1. Comenzar por el bucle más interior. Establecer o configurar los demás bucles con sus valores mínimos.
  2. Llevar a cabo las pruebas de bucles simples para el bucle más interior, mientras se mantienen los parámetros de iteración (por ejemplo, contador del bucle) de los bucles externos en sus valores mínimos. Añadir otras pruebas para valores fuera de rango o excluidos.
  3. Progresar hacia fuera, llevando a cabo pruebas para el siguiente bucle, pero manteniendo todos los bucles externos en sus valores mínimos y los demás bucles anidados en sus valores «típicos».
  4. Continuar hasta que se hayan probado todos los bucles.
- **Bucles concatenados.** Los bucles concatenados se pueden probar mediante el enfoque anteriormente definido para los bucles simples, mientras cada uno de los bucles sea independiente del resto. Sin embargo, si hay dos bucles concatenados y se usa el controlador del bucle 1 como valor inicial del bucle 2, entonces los bucles no son

independientes. Cuando los bucles no son independientes, se recomienda usar el enfoque aplicado para los bucles anidados.

- **Bucles no estructurados.** En este caso, ante la complejidad que puede representar la comprensión del flujo de control, es más práctico rediseñar el módulo a probar de forma que se codifique mediante bucles estructurados.

### **35.3.2 Pruebas de caja negra.**

El enfoque de la estrategia de pruebas conocido por el nombre de método de «**caja negra**» o, también, «**conducido por los datos**» («**data driven**») o «**conducido por la entrada/salida**» («**input-output driven**»), o también **pruebas de comportamiento**, no considera el detalle procedimental de los programas y se centra en buscar situaciones donde el programa no se ajusta a su especificación, utilizando ésta como entrada para derivar los casos de prueba.

Si por las especificaciones funcionales se sabe lo que tiene que hacer un módulo, es más sencillo comprobarlo que desmenuzarlo y examinarlo internamente en todas las circunstancias posibles. Por ello, las pruebas de «caja negra» son el enfoque más simple de prueba del software, y en ella diseñaremos los casos de prueba a partir de las especificaciones funcionales.

En este tipo de pruebas los casos de prueba consisten en conjuntos de datos de entrada que deberán generar una salida acorde con la especificación. La atención se centra, pues, en los datos de entrada y salida ignorando intencionadamente el conocimiento del código del programa. Si con esta técnica se quisieran encontrar todos los errores del programa, habría que recurrir a probar todas las posibles combinaciones de casos de entrada, lo que supondría generar todas las posibles

combinaciones de valores para todas las posibles variables de entrada, y ello, en la realidad, es imposible. De esta imposibilidad se pueden extraer la conclusión de que mediante el método de la «caja negra» no es posible asegurar que un programa esté libre de errores.

Dado que no se pueden probar todos los casos posibles, el método de «caja negra» contempla una serie de técnicas encaminadas a simplificar los casos de prueba. Éstas son:

- Partición equivalente
- El análisis de valores límite.
- Los valores típicos de error y los valores imposibles.
- Métodos de prueba basados en grafos.
- Prueba de la tabla ortogonal
- Las pruebas de comparación.

#### ***35.3.2.1 Partición equivalente.***

La **partición equivalente** es un método de prueba de caja negra que divide el campo de entrada de un programa en clases de datos de los que se pueden derivar casos de prueba. Un caso de prueba ideal descubre de forma inmediata una clase de errores (por ejemplo, proceso incorrecto de todos los datos de carácter) que, de otro modo, requerirían la ejecución de muchos casos antes de detectar el error genérico. La partición equivalente se dirige a la definición de casos de prueba que descubran clases de errores, reduciendo así el número total de casos de prueba que hay que desarrollar.

El diseño de casos de prueba para la partición equivalente se basa en una evaluación de las clases de equivalencia para una condición de entrada. Una **clase de equivalencia** representa un conjunto de estados válidos o no válidos para condiciones de entrada. Típicamente, una condición de

entrada es un valor numérico específico, un rango de valores, un conjunto de valores relacionados o una condición lógica. Las clases de equivalencia se pueden definir de acuerdo con las siguientes directrices:

- 1- Si una condición de entrada especifica un rango, se define una clase de equivalencia válida y dos no válidas.
- 2- Si una condición de entrada requiere un valor específico, se define una clase de equivalencia válida y dos no válidas.
- 3- Si una condición de entrada especifica un miembro de un conjunto, se define una clase de equivalencia válida y una no válida.
- 4- Si una condición de entrada es lógica, se define una clase de equivalencia válida y una no válida.

#### **35.3.2.2 Análisis de valores límite.**

El **análisis de valores límite** es una técnica de diseño de casos de prueba que complementa a la partición de equivalencia y se justifica en la constatación de que para una condición de entrada que admite un rango de valores, es más fácil que existan errores en los límites que en el centro. Por tanto, la diferencia entre esta técnica y la partición de equivalencia estriba en que en el análisis de valores límite no se selecciona un elemento representativo de la clase de equivalencia, sino que se seleccionan uno o más elementos de manera que los límites de cada clase de equivalencia son objeto de prueba. Otra diferencia es que con esta técnica también se derivan casos de prueba para las condiciones de salida.

Al igual que en el caso anterior, la técnica de análisis de valores límite no asegura la prueba completa, ya que es imposible probar exhaustivamente todos los conjuntos de datos de entrada tanto en su vertiente válida como inválida. Sin embargo, la ventaja que presenta esta técnica es que maximiza el número de errores encontrados con el menor número de casos de prueba posibles, lo que rentabiliza la inversión efectuada en la prueba.

Las directrices de AVL son similares en muchos aspectos a las que proporciona la partición equivalente:

- 1- Si una condición de entrada especifica un rango delimitado por los valores a y b, se deben diseñar casos de prueba para los valores a y b, y para los valores justo por debajo y justo por encima de a y b, respectivamente.
- 2- Si una condición de entrada especifica un número de valores, se deben desarrollar casos de prueba que ejerciten los valores máximo y mínimo. También se deben probar los valores justo por encima y justo por debajo del máximo y del mínimo.
- 3- Aplicar las directrices 1 y 2 a las condiciones de salida.
- 4- Si las estructuras de datos internas tienen límites preestablecidos (por ejemplo, una matriz que tenga un límite definido de 100 entradas), hay que asegurarse de diseñar un caso de prueba que ejercite la estructura de datos en sus límites.

### ***35.3.2.3. Valores típicos de error y valores imposibles.***

Un buen complemento de las dos técnicas fundamentales de pruebas tipo «caja negra» (particiones de equivalencia y análisis de valores límite) consiste en incluir en los casos de prueba ciertos valores de los datos de entrada susceptibles de causar problemas, esto es, valores típicos de error, y valores especificados como no posibles, es decir, valores imposibles.

La determinación de los valores típicos de error se realiza en función de la naturaleza y funcionalidad del programa a probar, por lo que depende en buena medida de la experiencia del diseñador de la prueba.

Asimismo, dentro de las especificaciones del sistema o del programa a probar puede haber valores de datos especificados como no posibles. El hecho de probar estos valores imposibles se debe a que dichos valores podrían haber sido generados internamente por el sistema o el programa

provocando un mal funcionamiento del mismo. La prueba de valores imposibles debe realizarse siempre que dichos valores puedan ser detectados y el programa pueda manejarlos adecuadamente sin provocar errores irreparables.

#### **35.3.2.4 Métodos de prueba basados en grafos.**

En este método se debe entender los objetos (objetos de datos, objetos de programa tales como módulos o colecciones de sentencias del lenguaje de programación) que se modelan en el software y las relaciones que conectan a estos objetos. Una vez que se ha llevado a cabo esto, el siguiente paso es definir una serie de pruebas que verifiquen que todos los objetos tienen entre ellos las relaciones esperadas. En este método:

1. Se crea un grafo de objetos importantes y sus relaciones.
2. Se diseña una serie de pruebas que cubran el grafo de manera que se ejerciten todos los objetos y sus relaciones para descubrir errores.

Boris Beizer describe un número de modelados para pruebas de comportamiento que pueden hacer uso de los grafos:

- *Modelado del flujo de transacción.* Los nodos representan los pasos de alguna transacción (por ejemplo, los pasos necesarios para una reserva en una línea aérea usando un servicio en línea), y los enlaces representan las conexiones lógicas entre los pasos (por ejemplo, vuelo.información.entrada es seguida de validación /disponibilidad.procesamiento).
- *Modelado de estado finito.* Los nodos representan diferentes estados del software observables por el usuario (por ejemplo, cada una de las pantallas que aparecen cuando un telefonista coge una petición por teléfono), y los enlaces representan las transiciones que ocurren para moverse de estado a estado (por ejemplo, petición-información se



verifica durante inventario-disponibilidad-búsqueda y es seguido por cliente-factura-información-entrada).

- *Modelado de flujo de datos.* Los nodos objetos de datos y los enlaces son las transformaciones que ocurren para convertir un objeto de datos en otro.
- *Modelado de planificación.* Los nodos son objetos de programa y los enlaces son las conexiones secuenciales entre esos objetos. Los pesos de enlace se usan para especificar los tiempos de ejecución requeridos al ejecutarse el programa.
- *Gráfica Causa-efecto.* La gráfica Causa-Efecto representa una ayuda gráfica al seleccionar, de una manera sistemática, un gran conjunto de casos de prueba. Tiene un efecto secundario beneficioso en precisar estados incompletos y ambigüedades en la especificación. Un gráfico de causa-efecto es un lenguaje formal al cual se traduce una especificación. El gráfico es realmente un circuito de lógica digital (una red combinatoria de lógica), pero en vez de la notación estándar de la electrónica, se utiliza una notación algo más simple. No hay necesidad de tener conocimiento de electrónica con excepción de una comprensión de la lógica booleana (entendiendo los operadores de la lógica y, o, y no).

### **35.3.2.5 Prueba de la tabla ortogonal**

Hay aplicaciones donde el número de parámetros de entrada es pequeño y los valores de cada uno de los parámetros están claramente delimitados. Cuando estos números son muy pequeños (por ejemplo, 3 parámetros de entrada tomando 3 valores diferentes), es posible considerar cada permutación de entrada y comprobar exhaustivamente el proceso del dominio de entrada. En cualquier caso, cuando el número de valores de entrada crece y el número de valores diferentes para cada elemento de dato se incrementa, la prueba exhaustiva se hace impracticable.

La prueba de la **tabla ortogonal** puede aplicarse a problemas en que el dominio de entrada es relativamente pequeño pero demasiado grande para posibilitar pruebas exhaustivas. El método de prueba de la tabla ortogonal es particularmente útil al encontrar errores asociados con fallos localizados -una categoría de error asociada con defectos de la lógica dentro de un componente software-. La prueba de tabla ortogonal permite proporcionar una buena cobertura de pruebas con bastantes menos casos de prueba que en la estrategia exhaustiva.

#### **35.3.2.6 Prueba de comparación.**

Hay situaciones en las que la fiabilidad del software es algo absolutamente crítico. En ese tipo de aplicaciones, a menudo se utiliza hardware y software redundante para minimizar la posibilidad de error. Cuando se desarrolla software redundante, varios equipos de ingeniería del software separados desarrollan versiones independientes de una aplicación, usando las mismas especificaciones. En esas situaciones, se deben probar todas las versiones con los mismos datos de prueba, para asegurar que todas proporcionan una salida idéntica. Luego, se ejecutan todas las versiones en paralelo y se hace una comparación en tiempo real de los resultados, para garantizar la consistencia.

Esas versiones independientes son la base de una técnica de prueba de caja negra denominada **prueba de comparación o prueba mano a mano**.

### **35.4 Pruebas de entorno y aplicaciones especializadas**

A medida que el software de computadora se ha hecho más complejo, ha crecido también la necesidad de enfoques de pruebas especializados. Los métodos de prueba de caja blanca y de caja negra son aplicables a todos

los entornos, arquitecturas y aplicaciones pero a veces se necesitan unas directrices y enfoques más específicos para las pruebas.

#### ***35.4.1. Prueba de interfaces gráficas de usuario (IGUs)***

Con el paso del tiempo la complejidad de las IGUs ha aumentado, originando más dificultad en el diseño y ejecución de los casos de prueba. Dado que las IGUs modernas tienen la misma apariencia y filosofía, se pueden obtener una serie de pruebas estándar. Los grafos de modelado de estado finito pueden ser utilizados para realizar pruebas que vayan dirigidas sobre datos específicos y programas objeto que sean relevantes para las IGUs. Considerando el gran número de permutaciones asociadas con las operaciones IGU, sería necesario utilizar herramientas automáticas para realizar las pruebas.

#### ***35.4.2 Prueba de arquitecturas cliente/servidor***

La naturaleza distribuida de los entornos cliente/servidor, los aspectos de rendimiento asociados con el proceso de transacciones, la presencia potencial de diferentes plataformas hardware, las complejidades de las comunicaciones de red, la necesidad de servir a múltiples clientes desde una base de datos centralizada (o en algunos casos, distribuida) y los requisitos de coordinación impuestos al servidor se combinan todos para hacer las pruebas de la arquitectura **C/S** y el software residente en ellas, considerablemente más difíciles que la prueba de aplicaciones individuales.

#### ***35.4.3 Prueba de la documentación y facilidades de ayuda***

Los errores en la documentación pueden ser tan destructivos para la aceptación del programa, como los errores en los datos o en el código fuente. Nada es más frustrante que seguir fielmente el manual de usuario y obtener resultados o comportamientos que no coinciden con los anticipados por el documento. La prueba de la documentación se puede enfocar en dos fases. La primera fase, la revisión e inspección, examina el

documento para comprobar la claridad editorial. La segunda fase, la prueba en vivo, utiliza la documentación junto al uso del programa real.

#### ***35.4.4 Prueba de sistemas de tiempo-real***

La naturaleza asíncrona **y** dependiente del tiempo de muchas aplicaciones de tiempo real, añade un nuevo **y** potencialmente difícil elemento a la complejidad de las pruebas (el tiempo). El responsable del diseño de casos de prueba no sólo tiene que considerar los casos de prueba de caja blanca **y** de caja negra, sino también el tratamiento de sucesos (por ejemplo, procesamiento de interrupciones), la temporización de los datos **y** el paralelismo de las tareas (procesos) que manejan los datos. En muchas situaciones, los datos de prueba proporcionados al sistema de tiempo real cuando se encuentra en un determinado estado darán un proceso correcto, mientras que al proporcionárselos en otro estado llevarán a un error. Además, la estrecha relación que existe entre el software de tiempo real **y** su entorno de hardware también puede introducir problemas en la prueba. Se puede proponer una estrategia en cuatro pasos como método de prueba para sistemas de tiempo real:

- **Prueba de tareas.** El primer paso de la prueba de sistemas de tiempo real consiste en probar cada tarea independientemente.
- **Prueba de comportamiento.** Utilizando modelos del sistema creados con herramientas CASE, es posible simular el comportamiento del sistema en tiempo real y examinar su comportamiento como consecuencia de sucesos externos. Estas actividades de análisis pueden servir como base del diseño de casos de prueba que se llevan a cabo cuando se ha construido el software de tiempo real.
- **Prueba intertareas.** Una vez que se han aislado los errores en las tareas individuales y en el comportamiento del sistema, la prueba se dirige hacia los errores relativos al tiempo. Se prueban las tareas asíncronas que se sabe que comunican con otras, con diferentes

tasas de datos y cargas de proceso para determinar si se producen errores de sincronismo entre las tareas. Además, se prueban las tareas que se comunican mediante colas de mensajes o almacenes de datos, para detectar errores en el tamaño de esas áreas de almacenamiento de datos.

- **Prueba del sistema.** El software y el hardware están integrados, por lo que se lleva a cabo una serie de pruebas completas del sistema para intentar descubrir errores en la interfaz software/hardware.

### **35.5 Pruebas funcionales**

En las pruebas funcionales se hace una verificación dinámica del comportamiento de un sistema, basada en la observación de un conjunto seleccionado de ejecuciones controladas o casos de prueba. Las pruebas funcionales son aquellas que se aplican al producto final, y permiten detectar en qué puntos el producto no cumple sus especificaciones, es decir, comprobar su funcionalidad. Para realizarlas se debe hacer una planificación que consiste en definir los aspectos a examinar y la forma de verificar su correcto funcionamiento, punto en el cual adquieren sentido los casos de prueba. Se usan sobre todo en el campo de la orientación a objetos.

Las formas actuales usadas para derivar casos de prueba funcionales son:

- La metodología SCENT permite derivar casos de prueba tomando como partida la definición de escenarios y actores que interactúan con el sistema, para luego definir prioridades, pasando por la elaboración de diagramas de dependencia, diagramas de estados y por último generar casos de prueba.
- Heumann desarrolla un método para generar casos de prueba tomando como base casos de uso, e identificando dentro de cada uno

los posibles escenarios, o caminos de ejecución, y por último definir los valores a probar de cada caso de prueba. Finalmente se obtiene una lista de casos de prueba, con los valores que deben probar y los resultados esperados para cada caso.

- La propuesta de Riebisch está centrada en la transformación automática de un modelo de casos de uso a un modelo de uso que sirve como entrada para realizar pruebas estadísticas automáticas, que mejoran el nivel de cobertura, partiendo de que diferentes partes de un software no necesitan ser probadas con la misma minuciosidad. El método comienza con el refinamiento de los casos de uso ampliándolos con precondiciones y postcondiciones, alternativas al camino de ejecución principal y referencia a otros casos de uso relacionados. Después se traducen a diagramas de estado y se elabora el modelo de uso donde se indica la probabilidad de que ocurra una transición y se identifican los caminos de ejecución más frecuentes. Por último, se extraen los modelos de prueba a partir de los modelos de uso y se generan recorridos aleatorios sobre cada modelo de uso. Cada camino aleatorio será un caso de prueba.
- Por último, Hartman propone una metodología centrada en dos productos: el primero compuesto por un modelo del sistema escrito en el lenguaje de modelado IF y un conjunto de diagramas UML de clases y estados que van a permitir la generación automática del conjunto de pruebas. El segundo, conformado por un conjunto de objetos de casos de prueba ejecutables tanto en el modelo del sistema como en la implantación, lo que permite comparar los resultados esperados y los obtenidos. Para obtener los productos enunciados, en primer lugar se construye un modelo de comportamiento del sistema a partir de sus especificaciones. Este modelo está compuesto por diagramas UML de clases y un diagrama UML de estados por cada clase que describe el comportamiento de los objetos de dicha clase. A continuación se elaboran los objetivos de

las pruebas (pruebas de casos de uso con datos concretos, pruebas de carga del sistema, etc.) y se traducen a un conjunto de directivas de generación y ejecución de pruebas. En el siguiente paso, una herramienta genera automáticamente una serie de pruebas que satisfacen los objetivos de prueba anteriores y se ejecuta automáticamente. Por último, se analizan los resultados y se repiten los pasos hasta que se alcanzan los objetivos deseados.

### Bibliografía

- Ingeniería del Software. Un enfoque práctico. Roger Pressman. Ed. Mc Graw Hill.
- Ingeniería del Software. Ian Sommerville. Ed. PEARSON ADDISON WESLEY.
- Metodología MÉTRICA versión 3. Técnicas y Prácticas. Ministerio de Administraciones Públicas.
- Grupo ARQUISOFT - Johanna Rojas - Emilio Barrios – 2007
- Método para generar casos de prueba funcional en el desarrollo de software. Liliana Gonzalez Palacio.

**Autor: Hernán Vila Pérez**

**Jefe del Servicio de Informática. Instituto Galego de Vivenda e Solo**  
**Vicepresidente del CPEIG**

# **SEGURIDAD EN LOS SISTEMAS DE INFORMACIÓN**



**36. SEGURIDAD DE LA  
INFORMACIÓN.  
CONFIDENCIALIDAD,  
INTEGRIDAD Y  
DISPONIBILIDAD. MEDIDAS DE  
SEGURIDAD FÍSICAS,  
TÉCNICAS, ORGANIZATIVAS Y  
LEGALES. IDENTIFICACIÓN Y  
AUTENTICACIÓN. CONTROL DE  
ACCESOS FÍSICOS Y LÓGICOS.  
CONTROL DE FLUJO DE  
DATOS.**

**Tema 36. Seguridad de la información. Confidencialidad, integridad y disponibilidad. Medidas de seguridad físicas, técnicas, organizativas y legales. Identificación y autenticación. Control de accesos físicos y lógicos. Control de flujo de datos.**

INDICE

1.1. Introducción.....	2
1.2. Las dimensiones de la seguridad.....	3
1.2. Amenazas a la seguridad.....	5
1.2.1. Personas.....	6
1.2.2. Amenazas lógicas.....	6
1.2.3. Catástrofes.....	8
2. MEDIDAS DE SEGURIDAD FISICAS, TECNICAS, ORGANIZATIVAS Y LEGALES.	
CONTROL DE ACCESO FISICO Y LÓGICO.....	8
2.1. Plan de seguridad y plan de contingencias.....	11
2.1.1. El plan de seguridad.....	11
2.1.2. El plan de contingencias.....	12
2.2. Seguridad Física.....	13
2.3. Seguridad Lógica.....	15
2.3.1. Ataques contra la seguridad lógica .....	16
2.3.2. La protección de la seguridad lógica.....	22
3. IDENTIFICACION Y AUTENTICACION.....	25
4. CONTROL DEL FLUJO DE DATOS.....	27
4.1. Técnicas de control de flujo de datos.....	27
4.1.1. Paridad simple (paridad horizontal).....	27
4.1.2. Paridad cruzada (paridad horizontal-vertical).....	28
4.1.3. Códigos de redundancia cíclica (CRC):.....	28
4.1.4. Suma de comprobación (checksum).....	30
4.1.5. Distancia de Hamming.....	30
5. REFERENCIAS.....	32

SEGURIDAD DE LA INFORMACION: AUTENTICIDAD, INTEGRIDAD, CONFIDENCIALIDAD, DISPONIBILIDAD Y TRAZABILIDAD.

### **1.1. Introducción**

Definimos un sistema informático como el conjunto formado por recursos físicos denominados hardware, recursos lógicos denominados software y recursos humanos, que interactúan entre sí con el objetivo de obtener, almacenar y procesar la información.

En este contexto, la seguridad informática será una característica de los sistemas informáticos que indicará si el sistema está libre de peligro, daño o riesgo. A menudo, la seguridad absoluta es difícil, o incluso imposible, de alcanzar y los sistemas informáticos no son una excepción.

En un sistema informático los esfuerzos se centrarán en proteger el software, el hardware y los datos. Éstos últimos suelen ser el elemento más valioso por lo que requieren de una mayor inversión en seguridad al tratarse del recurso más amenazado y el más difícil de recuperar.

Sin embargo, teniendo en cuenta que la seguridad absoluta de los sistemas de información es inalcanzable, se debe buscar siempre un compromiso entre el nivel de riesgo asumido y el coste de las medidas de seguridad de tal modo que dicho coste no supere nunca el valor de lo que se pretende proteger.

La seguridad en los sistemas de información atañe a todos los miembros de la organización. Su naturaleza dinámica hace que deba ser planificada, diseñada, ejecutada y mejorada de forma continua ya que todos los días surgen nuevas amenazas. Para protegernos contra ellas no es suficiente con implementar medidas técnicas, sino que tendremos que preparar una política de seguridad de la organización, planes de actuación, medidas de seguridad física, formación, concienciación, etc.

## **1.2. Las dimensiones de la seguridad**

Las dimensiones de la seguridad, también denominadas servicios o factores de seguridad, hacen referencia a las propiedades que la información debe cumplir para considerar un sistema como seguro. Un sistema informático se encuentra en óptimas condiciones de seguridad para operar cuando es capaz de garantizar la seguridad de todos sus componentes en cinco dimensiones:

- **Confidencialidad:** es la propiedad de la información por la que se garantiza que únicamente está accesible para los usuarios y procesos.

Si la información se encuentra almacenada en un sistema propio, la confidencialidad se basa en primer lugar en garantizar la autenticidad de cualquier usuario que intente acceder a ella. Una vez autenticado, debe controlarse que el usuario está autorizado, es decir, que cuenta con los permisos para acceder a esa información en concreto.

Sin embargo, cuando la información se transmite entre un emisor y un receptor a través de un medio externo y por tanto inseguro, la confidencialidad debe garantizarse mediante el uso de técnicas de cifrado, con el fin de evitar que un tercero que pueda interceptar el mensaje sea capaz de extraer cualquier información inteligible del mismo.

- **Integridad:** es la propiedad de la información por la que se garantiza que ésta no ha sido manipulada intencionada o accidentalmente por usuarios no autorizados.

La integridad puede protegerse mediante técnicas de validación de datos que permitan detectar cualquier alteración de los mismos como sumas de validación (*checksum*), uso de bits de paridad,

chequeos de redundancia cíclica (CRC), algoritmos de resumen (SHA1, MD5, etc.) y cualquier otro algoritmo destinado a detectar cualquier cambio no autorizado en la información.

- **Disponibilidad:** es la propiedad de la información por la que se garantiza que ésta se encontrará a disposición de las personas, procesos o aplicaciones que deban acceder a ella y dispongan de autorización para ello.

La disponibilidad implica que el sistema, tanto a nivel hardware como a nivel software, debe mantenerse funcionando eficientemente y que es capaz de recuperarse rápidamente en caso de fallo. Esto se consigue diseñando los sistemas de manera apropiada en términos de capacidad y escalabilidad para poder crecer de forma ordenada en caso de ser necesario. Además de esto, existen técnicas destinadas a evitar problemas de disponibilidad como el balanceo de carga entre servidores, virtualización de servidores, redundancia de sistemas de comunicaciones, alimentación eléctrica y de almacenamiento (RAID) o los sistemas de respaldo.

- **Autenticidad:** es la propiedad de la información por la que se garantiza su genuinidad y que permite identificar a su autor o generador.

Existe una propiedad relacionada con la autenticidad conocida como imposibilidad de rechazo o 'no repudio'. Esta propiedad permite demostrar ante terceros que una información ha sido enviada o consignada por una entidad sin que esta pueda negarlo.

La autenticidad implica la necesidad de dar pruebas de identidad de los participantes en una comunicación. El modo en que un usuario puede demostrar su identidad recae en uno o más de los siguientes factores: algo que el usuario sabe (por ejemplo una

clave de acceso), algo que tiene (por ejemplo una tarjeta de acceso) o algo que es parte del propio usuario (por ejemplo medidas biométricas como la huella digital). Se considera que para que una autenticación sea realmente segura debe involucrar al menos elementos de dos de los tres (si no de los tres) factores.

Por ejemplo, la arquitectura de firma electrónica con certificados emitidos por entidades confiables (Autoridades de Certificación) es un sistema de autenticación multifactor, ya que primero debemos poseer un documento que nos identifique para poder obtener un certificado avalado por un tercero y conocer un código personal para realizar las firmas.

- **Trazabilidad:** la trazabilidad en un sistema de información hace referencia a su capacidad para seguir y conservar la secuencia de todas las acciones (o al menos de aquellas que puedan afectar a la seguridad del sistema) que ocurran en el sistema para determinar su origen. Un sistema con procesos trazables facilitará la respuesta a incidencias en la seguridad. También será de gran ayuda a la hora de realizar auditorías de seguridad y diseñar sistemas de repuesta basados en el estudio de patrones de situaciones, como los sistemas de detección de intrusos (IDS).

## **1.2. Amenazas a la seguridad**

Los sistemas informáticos se encuentran expuestos a diversos agentes externos o internos capaces de causarles daño. Dependiendo de su origen, distinguiremos tres tipos de amenazas: personas, amenazas lógicas y catástrofes.

### **1.2.1. Personas**

Tanto el propio personal de la organización como curiosos o intrusos suponen una amenaza para la seguridad de la información. Los primeros, aunque normalmente lo harán de forma no intencionada, pueden causar daños al sistema durante el ejercicio de su actividad diaria. Estos daños serán proporcionales al nivel de exposición de la información ya que no todos los usuarios tendrán el mismo nivel de acceso a la información. Por su parte, los intrusos suponen a menudo una amenaza mayor ya que atacan de forma consciente con la intención de hacer el mayor daño posible. Dentro de este grupo podemos incluir a ex-empleados, crackers, terroristas, intrusos remunerados etc.

### **1.2.2. Amenazas lógicas**

En esta categoría se incluyen todo tipo de programas que de una forma u otra pueden dañar nuestro sistema, ya sea intencionadamente o por error. Podemos enumerar en esta categoría las siguientes amenazas:

- **Software incorrecto:** programas que debido a defectos en su código pueden causar daños al sistema o facilitar ataques desde el exterior.
- **Herramientas de seguridad:** pueden ser una excelente herramienta de ayuda para gestionar la seguridad y configurar nuestros sistemas correctamente, pero constituyen un arma de doble filo, ya que también son útiles para descubrir vulnerabilidades en nuestro sistema. Ejemplos de este tipo de herramientas son los *sniffers*<sup>1</sup> (como por ejemplo *Wireshark*) o las herramientas de escaneo de puertos (como por ejemplo *nmap*).

---

<sup>1</sup> Programas que permiten registrar y analizar toda la información transmitida a través de una red.



- **Bombas lógicas:** son partes del código de ciertos programas que permanecen inactivos hasta que se cumple cierta condición (ausencia o presencia de ciertos ficheros, fechas, etc.). Cuando esta condición se activa, se ejecuta el código malicioso que ataca al sistema.
- **Canales de comunicación ocultos:** permiten que un proceso transfiera información, violando la política de seguridad.
- **Puertas traseras:** son 'atajos' dejados por los programadores para acelerar los procesos de pruebas del software, pero que si no son corregidos antes del paso a producción pueden convertirse en peligrosos agujeros en la seguridad.
- **Virus:** son secuencias de código que se insertan en algún lugar del sistema (habitualmente en archivos ejecutables, aunque pueden residir en otros lugares) para realizar su tarea maliciosa e intentar extenderse a otras partes del mismo sistema o incluso a otros sistemas.
- **Gusanos:** programas capaces de ejecutarse y propagarse por sí mismos a través de redes.
- **Trojanos:** son instrucciones de código escondidas en programas que se alojan en el sistema y que realizan funciones ocultas, generalmente destinadas a tomar el control de un sistema (Por ejemplo, los *rootkit*).
- **Programas conejo** (también conocidos como programas bacteria): programas que no hacen nada salvo reproducirse hasta agotar los recursos del sistema y conseguir bloquearlo.
- **Técnicas Salami:** robo automatizado y sistemático de pequeñas cantidades de bienes de una gran cantidad original.



### **1.2.3. Catástrofes**

Las catástrofes, ya sean naturales o artificiales, constituyen la amenaza menos probable pero también la más desastrosa para los sistemas de información. La defensa contra ellas se basa en recursos físicos y diseño adecuado de la sede física de nuestros sistemas.

- **Incendios:** El centro de proceso de datos debe contar con un sistema de extinción de incendios adecuado que permita sofocar cualquier conato sin que la propia extinción cause daños a los sistemas.
- **Inundaciones:** Además de buscar un emplazamiento con el mínimo riesgo de inundación, es recomendable instalar sensores que detecten posibles fugas de agua y alerten sobre ellas antes de que puedan causar daño.
- **Terremotos:** En las zonas con actividad sísmica es un riesgo a tener en cuenta.
- **Tormentas eléctricas:** La instalación eléctrica debe contar con todas las garantías proporcionando alimentación ininterrumpida y neutralizando los picos de tensión que pudiesen producirse.
- **Temperaturas extremas:** Los equipos informáticos, como aparatos electrónicos que son, necesitan unas condiciones de temperatura y humedad adecuadas para un funcionamiento óptimo. Estas condiciones suelen conseguirse mediante la instalación de equipos de climatización.

## **2. MEDIDAS DE SEGURIDAD FISICAS, TECNICAS, ORGANIZATIVAS Y LEGALES. CONTROL DE ACCESO FISICO Y LÓGICO**

La seguridad de un sistema de información requiere acciones en diferentes ámbitos, que podemos diferenciar en los niveles físico, técnico (o lógico), organizativo y legal.

Las medidas de seguridad en el nivel físico pasan por controlar los accesos físicos al sistema de información, diseñar las infraestructuras adecuadamente conforme a las actividades de la organización y al entorno físico de su ubicación.

Las medidas técnicas son todas aquellas destinadas a garantizar la seguridad de los datos y los equipos desde el punto de vista del software y las comunicaciones, entendidas estas últimas como los protocolos y los datos, y no como los medios de transmisión físicos (equipos de transmisión, cableado, etc.).

La seguridad informática no es un problema que deba ser abordado en solitario por el departamento de sistemas de información. El ámbito organizativo debe participar activamente en la definición de las políticas de seguridad de la organización. Se trata de definir procedimientos, políticas y normas destinadas a garantizar la seguridad.

Algunas de las acciones organizativas sirven en realidad para ayudar a diseñar las medidas físicas y técnicas. Toda organización debería contar con:

- **Análisis de riesgos:** el análisis de riesgos debe realizarse siempre como paso inicial en el diseño de la seguridad de nuestros sistemas de información. Supone identificar los activos a proteger y el daño que sufriría la organización en caso de ser afectados por un ataque. Veremos este punto en profundidad en el próximo capítulo.

- **Identificación de amenazas:** consiste en identificar cada una de las amenazas y vulnerabilidades que pueden afectar a los recursos del sistema.
- **Políticas de seguridad:** es fundamental contar con una política de seguridad, diseñada a medida para la organización y conocida por todos los empleados y/o usuarios. En este sentido, el RFC 2196 *Site Security Handbook* es una guía para el desarrollo de políticas y procedimientos de seguridad aplicables a sistemas de información. Está destinado principalmente a sistemas que trabajan en el ámbito de Internet, pero también a aquellos sistemas que simplemente se comunican con otros. Y en forma general también puede ser utilizado en sistemas aislados. El contenido incluye políticas, conceptos de seguridad en redes y sistemas, y respuestas a los incidentes de seguridad.
- **Estrategia de seguridad:** una estrategia adecuada debe ser concebida de modo que abarque varios niveles de seguridad: seguridad física, seguridad lógica, el personal de la organización y la interacción que existe entre estos factores. El plan de seguridad es un documento fundamental en la organización que debe incluir una estrategia de previsión de ataques para minimizar los puntos vulnerables existentes en la directiva de seguridad. Debe también desarrollar planes de contingencias. Estos servirán como una estrategia reactiva de respuesta al ataque que ayude al personal de seguridad a evaluar el daño causado y a recuperar los niveles de servicio necesarios.
- **Equipos de respuesta a incidencias:** es aconsejable formar un equipo de respuestas a incidencias que dé apoyo al responsable de seguridad y que actúe siguiendo los planes de contingencia en caso de ser necesario.

El nivel de medidas legales viene evidentemente establecido por las administraciones autonómicas, estatales y europeas competentes en la materia. Además de la legislación general, relacionada con violaciones de privacidad, robos, etc., existe cierta legislación específica íntimamente ligada con la seguridad de la información.

Podemos citar la Ley Orgánica 15/1999 de Protección de Datos de Carácter Personal (LOPD) y el Real Decreto 1720/2007 que la desarrolla; la Ley 59/2003, de 19 de diciembre, de firma electrónica; la Ley 11/2007, de 22 de junio, de Acceso Electrónico de los Ciudadanos a los Servicios Públicos, que introduce en su artículo 42 el Esquema Nacional de Seguridad, desarrollado en el Real Decreto 3/2010. A nivel autonómico gallego, se ha desarrollado el Plan Director de Seguridad de la Información y el Decreto 230/2008 de buenas prácticas.

## **2.1. Plan de seguridad y plan de contingencias**

La seguridad informática debe ser una cuestión que abarque todos los ámbitos de la organización, con un gran componente técnico basado en tecnologías de la información, pero también político, de concienciación y formación de todo el personal.

### **2.1.1. El plan de seguridad**

El Plan de Seguridad Informática establece los principios organizativos y funcionales de la actividad de seguridad informática en una organización. Recoge claramente las políticas de seguridad y las responsabilidades de cada uno de los participantes en el proceso informático, así como las medidas y procedimientos que permitan prevenir, detectar y responder a las amenazas que gravitan sobre el mismo.

Una vez que la política de seguridad es aprobada, debe ponerse a disposición de todos los miembros de la organización ya que serán ellos los responsables finales de su éxito. Las políticas deben ser revisadas y actualizadas anualmente (o si es posible cada seis meses) para reflejar los cambios en la organización.

No debería haber dos políticas de seguridad iguales puesto que cada empresa es diferente y los detalles de la política dependen de las necesidades exclusivas de cada una. Sin embargo, se puede comenzar con un conjunto general de políticas de seguridad y luego personalizarlo de acuerdo con los requerimientos específicos, limitaciones de financiación e infraestructura existente.

Un plan de seguridad informática completo es un recurso valioso que justifica la dedicación de tiempo y esfuerzo a su elaboración.

### **2.1.2. El plan de contingencias**

Un plan de contingencias es un conjunto de procedimientos alternativos al orden normal de la organización, cuyo fin es permitir el normal funcionamiento de esta, aun cuando alguna de sus funciones se viese dañada por un fallo de seguridad.

Que una organización prepare sus planes de contingencias, no significa que reconozca la ineficacia de su plan de seguridad, sino que supone un avance a la hora de superar cualquier eventualidad que puedan acarrear pérdidas importantes.

Los planes de contingencias se deben hacer de cara a futuros acontecimientos para los que hace falta estar preparado.

La función principal de un plan de contingencias es la continuidad de las operaciones de la empresa. Dividimos su elaboración en cuatro etapas:

1. Evaluación.
2. Planificación.
3. Pruebas de viabilidad.
4. Ejecución.

Las tres primeras hacen referencia al componente preventivo y la última a la ejecución del plan una vez ocurrido el siniestro.

Se deben crear planes de contingencias para todos los riesgos de seguridad conocidos para prevenir la aparición de nuevas amenazas o vulnerabilidades desconocidas que dañen a nuestros sistemas.

Los planes de contingencias de seguridad deben especificar claramente las acciones a realizar si se produce un incidente para minimizar las consecuencias y la repercusión en los activos de la organización. Lo ideal es contar con planes de respuesta a incidentes y planes de continuación de negocio para garantizar una reacción eficaz durante y después de un ataque.

## **2.2. Seguridad Física**

La seguridad física se encarga de la protección contra amenazas a las instalaciones, equipamientos, sistemas de comunicaciones y personal que forman parte de un sistema de información. Consiste en la aplicación de barreras físicas y procedimientos de control ante amenazas a los recursos del sistema.

Resulta habitual que las organizaciones se centren en la seguridad lógica descuidando a veces la seguridad física, lo que puede ser fuente de problemas ya que la seguridad física afecta a los tres tipos de recursos: hardware, software y datos, y los ataques con acceso físico exitosos pueden ser muy dañinos. Por ejemplo, un acceso no autorizado que tiene

como resultado la desaparición de un equipo con datos sensibles de la organización.

Las amenazas a la seguridad física pueden clasificarse atendiendo a diversos aspectos. Establecemos aquí tres tipos:

- Amenazas ocasionadas involuntariamente por personas: se trata accidentes causados por personas de forma fortuita por accidente o negligencia en el uso del sistema. Estas personas no tienen por qué ser exclusivamente trabajadores técnicos. Incluye también a personal de limpieza y mantenimiento, visitantes, etc. En la práctica se traduce en derrames de líquidos, desconexiones bruscas de la red, caídas y roturas de material, etc.
- Acciones hostiles deliberadas: se trata de amenazas que pueden ser realmente peligrosas en función de la capacidad e intenciones del atacante. Son acciones deliberadas y usualmente planificadas contra nuestro sistemas o las personas que trabajan en el. Incluye accesos no autorizados a las instalaciones, robo, secuestros, fraudes, sabotajes, etc.
- Desastres naturales, incendios, humedad e inundaciones: son agentes externos que de afectar a nuestros sistemas suelen causar graves daños. Aunque son infrecuentes, es necesario preverlos y tenerlos en cuenta a lo largo de todo el diseño y la vida útil de nuestro sistema.

Las medidas de seguridad física permiten limitar el alcance de las amenazas citadas mediante el uso de controles y procedimientos de seguridad. El estándar TIA-942, aprobado por la *Telecommunications Industry Association* y por ANSI<sup>2</sup>, proporciona una guía de recomendaciones y normas para el diseño e instalación de infraestructuras de centros de

---

<sup>2</sup> Siglas en inglés de American National Standards Institute.

procesamiento de datos (CPD) que contribuyen a un considerable aumento de la seguridad física del sistema de información. Establece cuatro niveles (*tiers*) de disponibilidad implementados mediante medidas de carácter arquitectónico, de telecomunicaciones, eléctricas y mecánicas.<sup>3</sup>

### **2.3. Seguridad Lógica**

La seguridad lógica hace referencia al conjunto de operaciones y técnicas destinadas a la protección de la información, procesos y programas contra la destrucción, la modificación, la divulgación indebida o el retraso en su gestación.

Es poco frecuente que un ataque lógico afecte al hardware, aunque existe la posibilidad de que algún ataque de este tipo pueda llegar a dañar algún componente.

En un sistema de información, los datos constituyen uno de los recursos más importantes y valiosos. La gran variedad de los ataques, el posible origen remoto de los mismos y el impacto que pueden causar nos obliga a establecer medidas de protección similares o superiores a las asumidas en la seguridad física.

La seguridad lógica plantea una serie de requerimientos íntimamente relacionados con las precitadas dimensiones de la seguridad entre los que podemos mencionar los siguientes:

- Asegurar que los operadores puedan trabajar sin necesidad de una supervisión minuciosa y que no puedan modificar los programas ni los archivos que no les correspondan.

---

<sup>3</sup> Dejamos para el tema 56, dedicado al diseño de centros de procesamiento de datos, el desarrollo de las medidas seguridad física.



- Asegurar que se estén utilizando los datos y los programas correctos por el procedimiento correcto.
- Restringir el acceso al software y a los datos.
- Que la información transmitida sea recibida por el destinatario al que ha sido enviada y no por otro.
- Proveer técnicas que permitan garantizar que la información no resulta alterada durante un proceso de transmisión.
- Procurar que los sistemas de comunicación entre los distintos componentes del sistema de información estén redundados.
- Mantener un registro de las operaciones llevadas a cabo en el sistema junto con el usuario que las realiza para poder realizar su seguimiento en caso de ser necesario.

### **2.3.1. Ataques contra la seguridad lógica**

La seguridad lógica puede ser vulnerada por medio de ataques muy heterogéneos y que además se encuentran bajo continua evolución. Dada la gran diversidad de sistemas operativos, aplicaciones y protocolos, es imposible determinar su número, que sigue aumentando día a día.

Por otro lado, la transmisión de información a través de sistemas de ajenos a la organización, como Internet, resulta cada vez más frecuente y el volumen de información transmitido aumenta también de un modo notable. Estas operaciones de transmisión de información a través de un canal no controlado por la organización y compartido con terceros son especialmente vulnerables a los ataques de estos.

Podemos clasificar estos ataques atendiendo al objetivo de los mismos de la siguiente manera:

1. **Ataques Pasivos:** la información no resulta alterada sino que el atacante únicamente la captura o monitoriza para obtener los datos que están siendo transmitidos.

Los objetivos de estos ataques son la interceptación de datos y el análisis de tráfico con las siguientes finalidades:

- o Recopilación de datos sobre cuentas de usuarios y claves de acceso para utilizar más tarde en ataques activos.
  - o Obtención del origen y destinatario de la comunicación, a través de la lectura de las cabeceras de los paquetes monitorizados.
  - o Monitorización del volumen de tráfico intercambiado entre las entidades objeto del ataque, obteniendo así información acerca de actividad o inactividades inusuales.
  - o Monitorización de las horas habituales de intercambio de datos entre las entidades de la comunicación, para extraer información acerca de los periodos de actividad.
2. **Ataques activos:** implican algún tipo de modificación de los datos o la creación de datos falsos. Suelen ser intencionados y realizados por personas con conocimientos y consciencia de lo que están haciendo.

Se los pueden subdividir en varias categorías atendiendo a las acciones que se realizan durante el ataque:

- o **Interceptación:** un elemento no autorizado consigue un acceso a un determinado objeto del sistema, pero este no es modificado en ningún modo. Si se trata de una comunicación, esta llegará a su destino sin constancia de la interceptación.

- o **Destrucción:** algunos autores consideran un caso especial de la modificación la destrucción, entendiéndola como una modificación que inutiliza al objeto afectado.
- o **Modificación:** Si además de lograrse una interceptación el ataque consigue modificar el objeto de datos.
- o **Interrupción:** un ataque se clasifica como interrupción si hace que un objeto del sistema se pierda, quede inutilizable o no disponible.
- o **Fabricación:** modificación destinada a suplantar al objeto real.

Como decíamos, existe un elevado número de tipos de ataques contra la seguridad lógica. Citamos a continuación algunos de ellos:<sup>4</sup>

- **Ingeniería Social:** consiste en manipular a las personas del entorno para obtener acceso a los sistemas o a información confidencial que facilite otros ataques técnicos.
- **Ingeniería Social Inversa:** el intruso da a conocer de alguna manera que es capaz de brindar ayuda a los usuarios, y estos le llaman ante algún imprevisto. El intruso aprovechará la oportunidad para pedir información necesaria para solucionar el problema consiguiendo información útil para realizar ataques.
- **Shoulder-surfing**<sup>5</sup>: consiste en espiar físicamente a los usuarios mientras introducen su nombre de usuario y clave de acceso correspondiente o cuando están accediendo a información restringida.

---

<sup>4</sup> El RFC 4949 (*Internet Security Glossary, Version 2*) proporciona un listado más extenso y descriptivo.

<sup>5</sup> Expresión en inglés que se refiere a mirar por encima del hombro.

- **Piggybacking**: relacionado con el anterior. Hoy en día hace referencia al uso de redes wireless ajenas, pero su significado original era el de ‘colarse’ en un lugar detrás de otra persona.
- **Masquerading/spoofing** (Suplantación): suplantación electrónica o física de personas autorizadas para acceder al sistema u obtener información relevante sobre el mismo. Dependiendo del objeto de la implantación podríamos hablar de IP spoofing, DNS spoofing, web spoofing, etc.
- **Basureo**: paradójicamente, uno de los ataques más efectivos. Consiste en inspeccionar los deshechos en papeleras, contenedores, etc. en busca de información sensible.
- **Exploits**: aprovechamiento de errores conocidos (bugs) en determinadas versiones del software instalado en el sistema de información que pueden ser usados como puerta de entrada de ataques.
- **Escaneo de puertos**: técnicas de ataque pasivas que analizan las máquinas de un sistema para determinar cual es el estado de los puertos (abierto/cerrado). En algunos casos permite incluso conocer cual es el sistema operativo de la máquina e incluso que software y que versión del mismo se encuentra escuchando en cada puerto.
- **Wiretapping**: un tipo de ataque que intercepta y accede a información que se transmite por un canal de comunicaciones. El nombre del término (que podríamos traducir como ‘pinchar cables’) tiene como origen el pinchazo de teléfonos que se hacía de forma mecánica. Hoy en día hace referencia a la captura de datos mediante cualquier técnica con o sin cables de por medio.



- ***Eavesdropping-packet sniffing***: un tipo de ataque wiretapping pasivo. Es la interceptación pasiva del tráfico de red. Esto se realiza con aplicaciones llamadas sniffers, que son programas que capturan paquetes de datos que circulan por la red. Esto puede hacerse colocando el sniffer en uno de los equipos pertenecientes a la red o consiguiendo conectar un equipo externo a la misma. Este último caso es la forma más habitual en redes wireless.
- ***Man-in-the-middle***: un tipo de ataque wiretapping activo. El atacante intercepta y modifica selectivamente los paquetes de datos capturados para simular ser uno de los participantes en una comunicación ajena.
- **Denegación de servicio**: conocido habitualmente por sus siglas en inglés DOS (Denial of Service), este tipo de ataques consiste en conseguir que el objetivo del ataque deje de realizar su función de modo temporal o definitivo. El objetivo del ataque puede ser una máquina, una red de comunicaciones, un servicio concreto, etc. Se trata de uno de los tipos de ataques más comunes y efectivos. En la práctica existen múltiples variedades de este ataque de las que podemos nombrar: Flooding, ICMP flood, Syn flood, ping of death, land, smurf, teardrop, etc.
- **Denegación de Servicio Distribuida**: un caso especial del anterior, también conocida por sus siglas en inglés DDOS (Distributed Denial of Service). En este caso el ataque de DoS no proviene de un único atacante, sino que proviene de muchos a la vez de forma coordinada. El conjunto de atacantes puede no ser realmente un conjunto de personas, ya que es habitual el uso de máquinas secuestradas (máquinas zombies) que participan en el ataque muchas veces sin que su legítimo dueño tenga conocimiento de ello.

- **Ataques de secuestro (*Hijack*):** se centran en el secuestro de algún elemento en una comunicación previamente establecida por la víctima o de un recurso vital de una máquina. Los objetivos del secuestro pueden ser la sesión de un usuario ya autenticado en un sistema, el propio navegador de la víctima o incluso una página ofrecida por un servidor para modificarla y hacer que los datos insertados en ella sean enviados a una máquina bajo el control del atacante, por ejemplo.
- ***Tamper*:** ataque consistente en realizar modificaciones en la configuración de una máquina o sistema objetivo que degraden el nivel de seguridad de la misma.
- ***Phishing*:** es en realidad un tipo de ataque de tipo masquerading que ha crecido en frecuencia en los últimos años. Es un ataque que intenta adquirir información sensible del objetivo (número de cuentas bancarias, nombres de usuario y claves de acceso, etc.) mediante una solicitud fraudulenta en un correo electrónico o página web que el atacante ha construido para simular ser una entidad o persona de confianza del objetivo.
- ***SQL injection*:** es un tipo de ataque por inserción de código dirigido a la base de datos en lenguaje SQL. La técnica consiste en insertar secuencias concretas que son sintácticamente correctas en SQL en campos de texto de aplicaciones (usualmente web) para ejecutar consultas fraudulentas. Aunque es un ataque peligroso es fácilmente contrarrestable filtrando adecuadamente las entradas.
- ***Cross-site request forgery*:** abreviado habitualmente como CSRF ó XSRF y también conocido como 'ataque de un click'. Se basa en aprovechar la confianza que un sitio web tiene en el navegador web de un usuario. La víctima es engañado para usar

un hipere enlace manipulado por el atacante. La manipulación consiste en la construcción de un objeto de petición (request) que realiza directamente una acción fraudulenta e involuntaria para el usuario que la ejecuta. La clave reside en que el sitio web debe confiar en el usuario (porque ya se haya autenticado previamente, por ejemplo) y ejecutar la petición directamente, de ahí el nombre de ‘ataque en un click’.

- **Cross site scripting**: abreviado habitualmente como XSS. Se trata de cualquier ataque que permita ejecutar código de scripting (VBScript, Javascript, etc.) insertado por el atacante en el contexto un sitio web.

### 2.3.2. La protección de la seguridad lógica

Existen medidas de protección que pueden y deben ser utilizadas para aumentar la seguridad de nuestros sistemas de información.

Más adelante veremos que los niveles de seguridad pueden ser controlados y gestionados mediante al análisis de riesgos, por lo que en este apartado nos limitaremos a enunciar las medidas organizativas y técnicas que podemos aplicar para aumentar nuestra seguridad.

Los ataques a la seguridad lógica, suelen basarse en realidad en fallos de diseño inherentes a Internet (o sus protocolos), o a los sistemas operativos utilizados. La continua aparición de nuevas tecnologías hace que el número de tipos de ataques aumente también.

Por lo tanto, los responsables de seguridad y administradores de los sistemas deben mantenerse actualizados en cuanto a nuevos ataques y como protegerse contra ellos. Y por supuesto es de vital importancia que mantengan actualizado el sistema operativo y todo el software usado en las máquinas del sistema.

Una máquina que contenga información que no sea considerada valiosa, debe ser igualmente tenida en cuenta a la hora de definir las políticas de seguridad ya que puede resultar útil para un atacante a la hora de ser empleada en un ataque de denegación de servicio distribuido (DDoS) o ser utilizada como intermediaria (*proxy*) para ocultar la verdadera dirección del atacante.

Se deben realizar auditorías de seguridad de forma periódica y valorar la posibilidad de implantar sistemas de gestión de la seguridad de la información (SGSI) que garanticen la calidad de nuestros sistemas y medidas de seguridad.

Uno de los ataques con mayor tasa de éxito de todos los citados es el de la ingeniería social. De nada sirve que tengamos nuestro sistema perfectamente actualizado y que contemos con las medidas de seguridad físicas y lógicas más avanzadas si mediante una llamada telefónica un atacante puede conseguir que un usuario le transmita su usuario y clave de acceso.

La defensa contra ataques de ingeniería social pasa por mantener a los usuarios del sistema informados y formados en materia de seguridad. Hay que mantenerlos alerta y generarles un espíritu de desconfianza que permita evitar ataques basados en la ingeniería social o el *phishing*.

Algunas herramientas y técnicas de protección de la seguridad que podemos aplicar a nuestros sistemas de información son:

- **Uso de sistemas operativos seguros:** el mercado ofrece diferentes sistemas operativos que pueden ser usados en nuestros sistemas. Como es de suponer, no todos ellos ofrecen el mismo nivel de seguridad. Es conveniente usar sistemas operativos con niveles de seguridad acordes a las necesidades de nuestra estrategia de seguridad. Existe una catalogación de los niveles de seguridad que ofrece un sistema operativo



determinada por el estándar unificado *Common Criteria for Information Technology Security Evaluation* (simplificado habitualmente por *Common Criteria*).

- **Copias de respaldo** (*backups*): el backup de datos permite tener disponible una copia íntegra de los datos en caso de pérdida o corrupción de los mismos tras un ataque o un accidente. Es conveniente diseñar cuidadosamente la política y el procedimiento de creación, transporte y almacenamiento de las copias de respaldo de los datos y también de los programas usados en la organización.
- **Cortafuegos** (*firewalls*): son sistemas, hardware o software, destinados a filtrar el tráfico que circula por los sistemas de comunicaciones. Es habitual utilizarlos para controlar el tráfico que entra y sale de nuestra organización para evitar así algunos tipos de ataques basados en manipulación de paquetes. Además, permiten también controlar las aplicaciones y protocolos que son utilizados por los empleados.
- **Sistemas de Detección de Intrusos**: conocidos por sus siglas en inglés IDS. Un IDS es capaz de recoger y usar información de los eventos ocurridos en el sistema para detectar patrones de ataques y alertar al administrador de posibles ataques. Algunos tipos de IDS son incluso capaces de llevar a cabo acciones reactivas destinadas a abortar los ataques detectados.
- **Programas antivirus**: son programas destinados a la detección y eliminación de virus informáticos. Es conveniente contar con antivirus en todos los equipos del sistema de información. Para que los antivirus sean efectivos es vital que se encuentren actualizados.

- **Herramientas de seguridad:** como ya se ha comentado, pueden ser de gran utilidad para gestionar la seguridad y configurar nuestros sistemas correctamente aumentando así la protección de nuestro sistema.
- **Encriptación de la información:** dado que por muchas medidas de seguridad que tomemos seguiremos corriendo el riesgo de sufrir accesos no autorizados a nuestra información, es una buena práctica usar la encriptación para aumentar la protección de la información sensible. La encriptación debe aplicarse tanto a la información que se transmite a través de sistemas de comunicación como a aquella de especial importancia aunque ésta no salga de nuestro sistema.

### **3. IDENTIFICACION Y AUTENTICACION**

Un aspecto importante de la seguridad informática es la identificación de cualquier persona, máquina o servicio que pretenda acceder al sistema restringiendo los privilegios de acceso y ejecución a aquellos que previamente hubiesen sido definidos por el equipo de administración.

Podemos dividir el proceso de autenticación en un sistema informático en dos fases:

- La **identificación** proporciona una identidad a cada usuario del sistema asignándole un identificador único. En el momento de ingresar en el sistema, se buscarán entre los identificadores de usuario registrados los privilegios de acceso del que pretende identificarse.
- La **autenticación** es el procedimiento de comprobación de la identidad de un usuario que deberá presentar pruebas de que es quién dice ser.

El modo en que un usuario puede demostrar su identidad recae en uno o más de los siguientes factores:

- o algo que el usuario sabe (por ejemplo una clave de acceso)
- o algo que tiene (por ejemplo una tarjeta de acceso)
- o algo que es parte del propio usuario (por ejemplo medidas biométricas como la huella digital).

Se considera que para que una autenticación sea realmente segura debe involucrar al menos elementos de dos de los tres (si no de los tres) factores.

Por ejemplo, la arquitectura de firma electrónica con certificados emitidos por entidades confiables (Autoridades de Certificación) es un sistema de autenticación multifactor, ya que primero debemos poseer un documento que nos identifique para poder obtener un certificado avalado por un tercero y conocer un código personal para realizar las firmas. El uso de contraseñas asociadas a nombres de usuario se considera jurídicamente válido a los efectos de cumplir con los requisitos de identificación y autenticación.

La autorización o control de acceso es el proceso por el cual una identidad, debidamente autenticada, recibe los privilegios de acceso concretos y asociados al sistema en el que se ha autenticado.

Los llamados sistemas *Single Sign-On* (SSO) permiten que una identidad que se ha autenticado correctamente pueda ser compartida por varios sistemas, de modo que el usuario pueda pasar a trabajar de uno a otro sin tener que autenticarse varias veces mientras su sesión continúe abierta. Ejemplos de este tipo de sistemas son el sistema *Kerberos*, el proyecto *OpenID* o el proyecto *CAS* (*Central Authentication Service*).

## **4. CONTROL DEL FLUJO DE DATOS**

Al transmitir información a través de un canal de comunicaciones se produce un flujo de datos que puede ser alterado por interferencias o ataques que hagan que los datos recibidos en un extremo difieran de los emitidos desde el otro.

Los mecanismos de control del flujo de datos permiten detectar y corregir estas alteraciones manteniendo así la integridad del mensaje transmitido.

### **4.1. Técnicas de control de flujo de datos**

El control de flujo de datos pasa por aumentar el tamaño de la información enviada añadiendo datos adicionales calculados a partir de los originales que permitan realizar comprobaciones en el destino. Existen dos estrategias básicas para realizar este control:

- **Códigos de detección de errores:** la información transmitida se codifica incluyendo sólo la información redundante necesaria en cada bloque de datos para detectar los errores. En este caso el número de bits de redundancia es menor.
- **Códigos de corrección de errores:** la información se codifica añadiendo bits de redundancia en cada bloque que, no sólo permiten detectar errores, sino también corregirlos.

#### **4.1.1. Paridad simple (paridad horizontal)**

Consiste en añadir un bit a cada bloque a enviar que tomará el valor 0 o 1 dependiendo de si el número de bits con valor 1 es par o impar, respectivamente. El bit añadido se denomina bit de paridad.



El receptor, al recibir el bloque, contará el número de bits con valor 1 comprobará si coincide con el valor de bit de paridad asumiendo que si coincide significa que no ha habido error. Este sistema detectará únicamente errores que afecten a un número impar de bits.

#### **4.1.2. Paridad cruzada (paridad horizontal-vertical)**

Este método trata de mejorar el anterior enviando un bloque de paridad cada N bloques de datos. El bloque de paridad sirve para comprobar la paridad de los n bloque de datos enviados inmediatamente antes que él. La técnica consiste en calcular la paridad de los bits que ocupan la posición P en cada uno de los bloques y realizar la comprobación de paridad contra el bit que ocupa la posición P del bloque de paridad.

Bloque 1	1	0	1	1	0	1	0	1	<b>1</b>
Bloque 2	0	1	1	0	1	0	0	1	<b>0</b>
Bloque 3	1	0	0	1	0	1	0	1	<b>0</b>
Bloque 4	1	0	0	0	0	1	1	0	<b>1</b>
Bloque 5	1	0	1	0	1	1	1	0	<b>1</b>
Bloque 6	1	1	1	0	1	0	0	1	<b>1</b>
Bloque 7	1	0	0	0	0	1	0	1	<b>1</b>
Bloque 8	0	0	0	1	0	1	1	1	<b>0</b>
Bloque de	<b>0</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>1</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>1</b>

#### **4.1.3. Códigos de redundancia cíclica (CRC):**

Los códigos cíclicos aparecen intentando mejorar los códigos que sólo controlan la paridad a nivel de bit. Estos códigos utilizan la aritmética modular para detectar una mayor cantidad de errores. Se usan operaciones en módulo 2 y las sumas y restas se realizan sin acarreo (convirtiéndose en operaciones de tipo Or-Exclusivo o XOR).

Además, para facilitar los cálculos se trabaja, aunque sólo teóricamente, con polinomios. La finalidad de este método es crear una parte de

redundancia, la cual se añade al final del código a transmitir, que siendo lo más pequeña posible, detecte el mayor número de errores.

El método debe ser sistemático, es decir, que con un mismo código a transmitir (y un mismo polinomio generador) se genere siempre el mismo código final. El polinomio generador es un polinomio elegido previamente y que tiene como propiedad minimizar la redundancia. Suele tener una longitud de 16 bits, para mensajes de 128 bytes, lo que indica que la eficiencia es buena ya que sólo incrementa la longitud en un aproximado 1,6%.

El polinomio usado habitualmente en las redes WAN es  $g(x) = x^{16} + x^{12} + x^5 + x$ , y los cálculos que realiza el equipo transmisor para calcular su CRC son:

1. Añadir tantos ceros por la derecha al mensaje original como el grado del polinomio generador.
2. Divide el mensaje con los ceros incluidos entre el polinomio generador.
3. El resto que se obtiene de la división se suma al mensaje con los ceros incluidos.
4. Se envía el resultado obtenido.

El receptor del mensaje realiza las siguientes operaciones:

5. Mediante el protocolo usado determina el polinomio generador utilizado.
6. Divide el código recibido entre el polinomio.
7. Comprueba el resto de dicha operación. Si el resto es cero, no se han producido errores. En caso de que el resto sea uno, el código recibido contiene errores.

Este método puede llegar a detectar gran cantidad de errores:

- o Errores simples: todos
- o Errores dobles: todos
- o Errores en las posiciones impares de los bits: todos
- o Errores en ráfagas con una longitud menor que el grado del polinomio generador: todos
- o Otras ráfagas: un porcentaje elevado y cercano al 100%

#### **4.1.4. Suma de comprobación (*checksum*)**

Es un método sencillo, pero eficiente sólo con bloques de una longitud pequeña. Es por ello que se suele utilizar en cabeceras de tramas u otros bloques importantes y en combinación con otros métodos. Consiste en agrupar el mensaje a transmitir en bloques de una longitud determinada  $L$  no muy grande, por ejemplo 16 bits. Considerando a cada cadena como un número entero numerado según el sistema de numeración  $2^L - 1$ . A continuación se suma el valor de todas las palabras en las que se divide el mensaje, y se añade el resultado al mensaje a transmitir, pero cambiado de signo. Con esto, el receptor lo único que tiene que hacer es sumar todas las cadenas, y si el resultado es 0 no hay errores.

#### **4.1.5. Distancia de Hamming**

Los códigos de Hamming se basan, como los anteriores, en incluir información redundante que permite detectar errores e incluso corregir algunos de ellos.

La Distancia de Hamming consiste en asignar valores a las operaciones (sustitución, borrado, añadir, etc.) necesarias para transformar un conjunto de bits (una palabra) en otro.

Si queremos detectar  $d$  bits erróneos en una palabra de  $n$  bits, podemos añadir a cada palabra de  $n$  bits  $d+1$  bits predeterminados al final, de forma que quede una palabra de  $n+d+1$  bits con una distancia mínima de Hamming de  $d+1$ . De esta manera, si uno recibe una palabra de  $n+d+1$  bits que no encaja con ninguna palabra del código (con una distancia de Hamming  $x \leq d+1$  la palabra no pertenece al código) se detecta correctamente que es una palabra errónea. Aún más,  $d$  o menos errores nunca se convertirán en una palabra válida debido a que la distancia de Hamming entre cada palabra válida es de al menos  $d+1$ , y tales errores conducen solamente a las palabras inválidas que se detectan correctamente. Dado un conjunto de  $m*n$  bits, podemos detectar  $x \leq d$  bits erróneos correctamente usando el mismo método en todas las palabras de  $n$  bits. De hecho, podemos detectar un máximo de  $m*d$  errores si todas las palabras de  $n$  bits son transmitidas con un máximo de  $d$  errores.

Otros códigos detectores y correctores de errores son la *corrección de errores hacia adelante* (en inglés, *Forward Error Correction* o *FEC*) o el *Código binario de Golay*.



## 5. REFERENCIAS<sup>6</sup>

- Telecommunications Industry Association (TIA)  
<http://www.tiaonline.org/>
- RFC 4949 Internet Security Glossary, Version 2  
<http://tools.ietf.org/html/rfc4949>
- RFC 2196 Site Security Handbook  
<http://tools.ietf.org/html/rfc2196>
- *Common Criteria* for Information Technology Security Evaluation  
<http://www.commoncriteriaportal.org/>
- *Mechanics of User Identification and Authentication: Fundamentals of Identity Management*, Dobromir Todorov, Auerbach Publications.
- **STALLINGS, W.** (2011). *Network Security Essentials. Applications and Standards. Fourth Edition*. Prentice Hall.

**Autor: Juan Otero Pombo**

**Ingeniero en Informática en el Concello de Ourense**

**Colegiado del CPEIG**

---

<sup>6</sup> Todos los enlaces fueron verificados en noviembre de 2011.



# **37. SEGURIDAD EN REDES: REDES PERIMETRALES. DMZ. CORTAFUEGOS. INTRUSIONES. ACCESOS NO AUTORIZADOS. TÉCNICAS DE SEGURIDAD PREVENTIVAS Y REACTIVAS.**

**Tema 37. Seguridad en redes: redes perimetrales. DMZ. Cortafuegos. Intrusiones. Accesos no autorizados. Técnicas de seguridad preventivas y reactivas.**

INDICE

<b><u>1 SEGURIDAD EN REDES: REDES PERIMETRALES. DMZ.....</u></b>	<b><u>2</u></b>
<b><u>1.1 Arquitectura de seguridad en el modelo OSI.....</u></b>	<b><u>3</u></b>
<b><u>1.2 Redes perimetrales.....</u></b>	<b><u>9</u></b>
<b><u>2 TECNICAS DE SEGURIDAD PREVENTIVAS Y REACTIVAS. CORTAFUEGOS. INTRUSIONES. ACCESOS NO AUTORIZADOS.....</u></b>	<b><u>16</u></b>
<b><u>2.1 Cortafuegos.....</u></b>	<b><u>16</u></b>
<b><u>2.2 Sistemas de detección de intrusiones (IDS/IPS).....</u></b>	<b><u>21</u></b>
<b><u>2.3 Redes privadas virtuales (RPV).....</u></b>	<b><u>23</u></b>
<b><u>3 REFERENCIAS.....</u></b>	<b><u>28</u></b>

## **1 SEGURIDAD EN REDES: REDES PERIMETRALES. DMZ**

La irrupción de los sistemas informáticos en las organizaciones provocado un cambio importante en el modo de garantizar la seguridad de la información. Antes del uso extendido de los sistemas informáticos, la documentación sensible se encontraba en soporte papel por lo que la seguridad de la información para la organización, era provista en un sentido físico y administrativo. Un claro ejemplo de esto eran, los armarios con cerradura de combinación utilizados para almacenar los documentos con información sensible y los procesos de selección utilizados para incorporar nuevo personal.

La implantación de los sistemas informáticos trajo consigo la necesidad de herramientas automatizadas para la protección de archivos y otra información almacenada en los sistemas. En la actualidad, la información se encuentra almacenada en sistemas compartidos y distribuidos que ofrecen la posibilidad de acceder a ella a través de una red privada de ordenadores o a través de Internet. Las **medidas de seguridad en la red** son el conjunto de medidas adoptadas para proteger los datos durante la transmisión a través de redes no seguras. Sin embargo, debido a que la práctica totalidad de empresas, gobiernos y organizaciones académicas tienen interconectados sus sistemas informáticos con una colección de redes que a su vez están interconectadas entre sí, dando lugar a lo que conocemos como Internet, es más frecuente el uso del término **seguridad en Internet**. La seguridad en internet abarca la prevención, detección y corrección de violaciones de la seguridad que puedan afectar a la transmisión de información.

El *Computer Security Handbook* del NIST [NIST95] define el término **computer security** como la protección conferida a un sistema informático con el fin de alcanzar los objetivos de preservar la integridad, disponibilidad

y confidencialidad de los recursos de información del sistema (incluyendo hardware, software, firmware, información/datos y telecomunicaciones). Esta definición abarca los objetivos centrales de la seguridad de los sistemas de información: confidencialidad, integridad, disponibilidad, autenticación y trazabilidad.

### **1.1 Arquitectura de seguridad en el modelo OSI**

Si la seguridad en entornos de procesamiento de datos cerrados es compleja, el uso de redes de área local y extensa aumenta esa complejidad de forma considerable. Es por esto que el administrador responsable de la seguridad de una organización necesita alguna metodología que le permita definir los requisitos de seguridad e identificar los mecanismos que le permitirán cumplirlos. Esta metodología deberá facilitar la cobertura efectiva de las necesidades de seguridad de la organización así como la evaluación y posterior elección de los distintos productos y políticas.

UIT-T<sup>1</sup> Recomendación X.800, *Security Architecture for OSI*, describe los servicios de seguridad básicos que pueden ser aplicados cuando es necesario proteger la comunicación entre sistemas. Aunque se trata de un modelo genérico definido en los años 90, sus conceptos y definiciones aún siguen vigentes en el día a día de los administradores de la seguridad. La arquitectura de seguridad del modelo OSI es útil para los administradores, ya que establece un protocolo para organizar la tarea de proporcionar seguridad. Además, como esta arquitectura fue desarrollada como un estándar internacional, los fabricantes de computadoras y sistemas de comunicación, han añadido características de seguridad a sus productos y servicios que se relacionan con esta definición estructurada de mecanismos y servicios. El modelo OSI de arquitectura de la seguridad se centra en los siguientes conceptos: **mecanismos de seguridad, servicios de**

---

<sup>1</sup> UIT-T son las siglas de la Unión Internacional de las Telecomunicaciones, Sector de Estandarización de las Telecomunicaciones que es una agencia patrocinada por las Naciones Unidas, que desarrolla estándares, llamadas Recomendaciones, relacionadas con las Telecomunicaciones.

**seguridad y ataques contra la seguridad.** Se pueden definir de manera resumida:

- **Mecanismos de seguridad:** serán los procesos que permiten detectar, prevenir, o recuperarse frente a un ataque contra la seguridad.
- **Servicios de seguridad:** un servicio de comunicación o procesado que incrementa la seguridad de los sistemas de información y las transferencias de datos realizadas por una organización. Los servicios intentan prevenir los ataques contra la seguridad haciendo uso de uno o varios mecanismos de seguridad.
- **Ataques contra la seguridad:** cualquier acción que atenta contra la seguridad de la información de la organización.

### 1.1.1 Mecanismos de seguridad

Los mecanismos de seguridad en X.800 se dividen en los que se aplican a una capa de protocolo específico y los que no son específicos de una capa de protocolo o servicio de seguridad (conocidos también como mecanismos de seguridad persistentes).

**Mecanismos específicos de seguridad:** pueden ser incorporados en una de las capas del protocolo con el fin de proporcionar alguno de los servicios de seguridad OSI:

- **Autenticación:** corrobora que una entidad, ya sea origen o destino de la información, es la deseada, por ejemplo, A envía un número aleatorio cifrado con la clave pública de B, B lo descifra con su clave privada y se lo reenvía a A, demostrando así que es quien pretende ser. Por supuesto, hay que ser cuidadoso a la hora de diseñar estos protocolos, ya que existen ataques para desbaratarlos.

- **Control de accesos:** esfuerzo para que sólo aquellos usuarios autorizados accedan a los recursos del sistema o a la red, como por ejemplo mediante las contraseñas de acceso.
- **Firma digital:** consiste en adjuntar una serie de datos a un mensaje o realizar una transformación criptográfica que permita al receptor comprobar el origen de un mensaje y verificar su integridad.
- **Cifrado:** consiste en la transformación de la información, por medio del uso de algoritmos matemáticos, a un formato que no es inteligible. La transformación y recuperación de la información depende de un algoritmo de cifrado y/o del uso claves de cifrado.
- **Notarización:** el uso de un tercero de confianza para asegurar ciertas propiedades en un intercambio de datos.
- **Integridad:** conjunto de mecanismos para asegurar la corrección y completitud de una unidad de datos o un flujo de datos.
- **Tráfico de relleno:** consiste en enviar tráfico espurio junto con los datos válidos para que el atacante no sepa si se está enviando información, ni qué cantidad de datos útiles se está transmitiendo.
- **Control de enrutado:** permite la selección, para ciertos datos, de rutas físicas seguras y la variación de las mismas, especialmente cuando se sospecha de una violación de la seguridad.

Dentro de los mecanismos de seguridad persistentes tenemos:

- **Registro de auditoría de seguridad:** datos recogidos y potencialmente usables para realizar una auditoría de seguridad.
- **Etiquetas de seguridad:** los atributos o propiedades de seguridad asociadas a un recurso o unidad de datos.

- **Funcionalidad de confianza:** lo que se debe percibir como correcto con respecto a algún criterio (por ejemplo: según lo establecido por una política de seguridad).
- **Detección de eventos:** detección de eventos relevantes de seguridad.
- **Recuperación de la seguridad:** se ocupa de las peticiones de los mecanismos, tales como manejo de eventos y gestión de funciones, y lleva a cabo tareas de recuperación.

### 1.1.2 Servicios de Seguridad

X.800 define un Servicio de Seguridad como un servicio que es provisto por una capa del protocolo de comunicación y que garantiza la adecuada seguridad de los sistemas o transferencias de datos. Tal vez, una definición más clara se encuentra en la RFC 4949, que presenta la siguiente definición: Un servicio de información o comunicación que es proporcionado por un sistema para una clase específica de protección de recursos informáticos. X.800 divide estos servicios en cinco categorías:

- **Confidencialidad:** requiere que la información sea accesible únicamente por las entidades autorizadas. La confidencialidad de datos se aplica a todos los datos intercambiados por las entidades autorizadas o tal vez a sólo porciones o segmentos seleccionados de los datos, por ejemplo mediante cifrado. La confidencialidad de flujo de tráfico protege la identidad del origen y destino(s) del mensaje, por ejemplo enviando los datos confidenciales a muchos destinos además del verdadero, así como el volumen y el momento de tráfico intercambiado, por ejemplo produciendo una cantidad de tráfico constante al añadir tráfico espurio al significativo, de forma que sean indistinguibles para un intruso. La desventaja de estos métodos es que incrementan drásticamente el volumen de tráfico intercambiado,



repercutiendo negativamente en la disponibilidad del ancho de banda bajo demanda.

- **Servicio de autenticación:** requiere una identificación correcta del origen del mensaje, asegurando que la entidad no es falsa. Se distinguen dos tipos: de entidad, que asegura la identidad de las entidades participantes en la comunicación, mediante biométrica (huellas dactilares, identificación de iris, etc.), tarjetas de banda magnética, contraseñas, o procedimientos similares; y de origen de información, que asegura que una unidad de información proviene de cierta entidad, siendo la firma digital el mecanismo más extendido.
- **Control de accesos:** en el contexto de seguridad informática, el control de acceso es la capacidad de controlar y limitar el acceso a través de la red a los sistemas y aplicaciones. Para lograr esto, cada entidad que intenta conseguir acceso debe ser autenticada, por lo que los derechos de acceso se pueden adaptar a cada usuario.
- **Integridad:** de la misma manera que la confidencialidad, la integridad se puede aplicar a un flujo de mensajes, a un único mensaje, o a un conjunto de campos seleccionados de un mensaje. De nuevo, el enfoque más sencillo y útil es la protección total del flujo de comunicación. Un servicio de **integridad orientado a conexión** trabaja con flujos de mensajes y garantiza que los mensajes son recibidos tal y como son enviados, sin ser duplicados, modificados, reordenados o repetidos.
- **No repudio:** el no repudio evita que el emisor o receptor de un mensaje puedan negar la transmisión. Así, cuando se envía el mensaje, el receptor puede probar que el supuesto emisor ha hecho el envío. Del mismo modo, cuando un mensaje es recibido, el emisor puede probar el hecho de que el mensaje, efectivamente ha sido recibido.

Ambos X.800 y RFC 4949 definen la **disponibilidad** como la propiedad de un sistema o recurso de ser usado y disponible bajo un sistema de autorización de entidad, de acuerdo con las especificaciones de rendimiento para el sistema. Es decir, el sistema está disponible si proporciona sus servicios de acuerdo con el diseño del sistema cada vez que los usuarios lo solicitan. Una gran variedad de ataques pueden producir la pérdida o reducción de la disponibilidad. Algunos de esos ataques pueden ser evitados con medidas automáticas, tales como la autenticación y el cifrado, mientras que otros requieren algún tipo de acción física para prevenir o recuperarse de una pérdida de disponibilidad en los elementos de un sistema distribuido.

La *Tabla 1* indica la relación entre los servicios de seguridad y los mecanismos de seguridad.

### **1.1.3 Ataques contra la seguridad**

Una forma útil de clasificar los ataques contra la seguridad, utilizada tanto en X.800 como en la RFC 4949, es por medio de los términos *ataque activo* y *ataque pasivo*.

Un ataque pasivo intenta conocer o hacer uso de la información del sistema pero sin afectar a sus recursos. Los ataques pasivos son muy difíciles de detectar, ya que no provocan ninguna alteración de los datos. Sin embargo, es posible evitar su éxito mediante el cifrado de la información y otros mecanismos que se verán más adelante.

Un ataque activo intenta cambiar los recursos del sistema o alterar su modo de funcionamiento. Los esfuerzos contra los ataques pasivos se centran en la prevención más que en la detección, mientras que por el contrario, frente a los ataques activos lo más importante es recuperarse lo antes posible de cualquier interrupción o retraso causado.

## **1.2 Redes perimetrales**

El concepto de red perimetral hace referencia al conjunto de dispositivos y/o mecanismos técnicos que son utilizados para separar la red interna de una organización (conocida como *intranet* o red corporativa) del resto de redes externas (habitualmente Internet).

### **1.2.1 Segmentación de redes**

En organizaciones con una infraestructura de red moderada, se recomienda segmentar las redes en subredes con el objetivo de aislar al máximo los daños que puedan producirse por la intrusión de un atacante en el sistema.

Si un atacante consigue acceso a la red de la organización, y ésta no está segmentada, tendrá acceso a la totalidad de equipos de la misma, aumentando la gravedad de los daños que podrá causar.

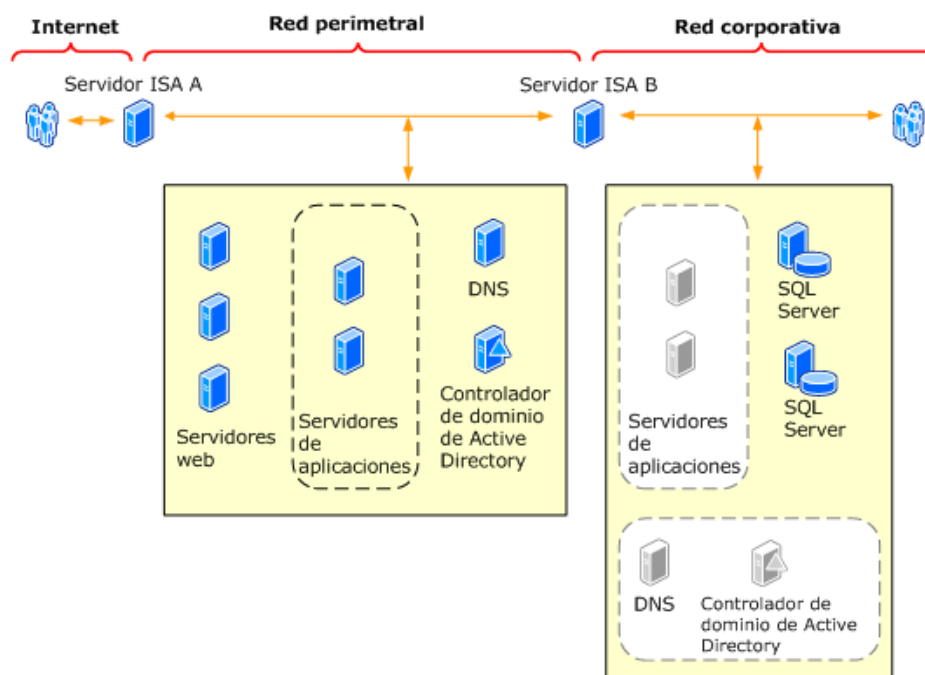
Si, por el contrario, se dispone de una red segmentada en subredes, un atacante se quedará en un segmento, pudiendo acceder a todos los recursos contenidos en él, pero no al resto de los recursos de la organización lo que minimizará los daños.

Parece lógico segmentar la red en función del diferente nivel de control que requiere la red interna en comparación con la comunicación con el exterior, y de la necesidad de establecer distintos niveles de control de acceso en una y otras redes.

Sin embargo, determinar dónde termina exactamente la red interna puede no ser una tarea trivial. El auge de las comunicaciones y la dispersión geográfica de muchas organizaciones complican la tarea de determinar y proteger la red propia.

La siguiente figura ilustra el concepto de red perimetral y su contexto de red interna y externa. En ella se puede observar que el concepto básico sobre el que gira la seguridad perimetral consiste en separar aquellos

servidores que pueden ser accedidos desde el exterior de aquellos que son accesibles solo desde el interior, de modo que los controles de seguridad de cada subred puedan ser diseñados a medida de las necesidades, habitualmente muy diferentes, de cada una de ellas.



Fuente: <http://technet.microsoft.com>

El término seguridad perimetral es muy amplio y ha tenido diversas atribuciones a lo largo del tiempo. El perímetro como tal, está formado por las máquinas y los dispositivos que se sitúan en la frontera de nuestra red, donde ésta interactúa con el exterior, con otras redes.

Sin embargo, a menudo el perímetro no está circunscrito a una sala y la seguridad física resulta insuficiente para proteger nuestra red. Es necesario incorporar ciertas medidas lógicas para controlar los accesos.

El establecimiento de VLAN (Virtual Local Area Network) permite crear redes lógicamente independientes dentro de una misma red física. Varias VLANs pueden coexistir en un único equipo físico o en una única red física. Son útiles para reducir el tamaño del dominio de difusión (broadcast) y

ayudan en la administración de la red separando segmentos de la red interna mediante un criterio concreto (por ejemplo por Departamentos). De esta forma, se puede afirmar, que los trabajadores acceden únicamente a los recursos de su red, denegando el acceso a los recursos del resto.

Por otra parte en la medida en que aumenta el uso de las arquitecturas orientadas al servicio o SaaS (del inglés *Software as a Service*) (además de la virtualización y la computación en la nube o *cloud computing*) y en la medida en que se hace más común el uso de herramientas dirigidas a la web 2.0 (redes sociales, blogs, wikis, etc.), las políticas basadas en puertos y protocolos, tradicionalmente utilizadas para el control del perímetro, son cada vez son menos efectivas. Por eso son necesarias nuevas herramientas para hacer frente a la evolución del comportamiento de los usuarios, a las nuevas formas en que los procesos de negocio utilizan las TIC y a los cambios en los mecanismos de los ataques que comprometen los sistemas de información.

### **1.2.2 El perímetro difuso**

En la actualidad, el perímetro se ha extendido y se ha convertido en una frontera dinámica (lo que algunos denominan perímetro difuso), en el que también se hace necesario proteger y protegerse de cada dispositivo que puede ocasionalmente formar parte del mismo.

Las medidas lógicas de control de acceso y vigilancia de las comunicaciones son en la actualidad más complejas siendo además necesario autenticar los dispositivos que acceden a la red para monitorizar el uso que hacen de los recursos de la red corporativa o doméstica.

Con este concepto actual de perímetro difuso podemos distinguir los siguientes elementos destinados a conseguir un nivel de seguridad perimetral adecuado:

- **Cortafuegos (firewalls):** dispositivos (hardware o software) de la red que se sitúan normalmente entre 2 redes (o subredes) para filtrar el tráfico en función de diferentes criterios.
- **IPS/IDS<sup>2</sup>:** los sistemas de prevención y detección de intrusiones permiten detectar intrusiones habitualmente en tiempo real. También pueden detectar patrones de ataques y tomar acciones contra ellos.
- **Redes privadas virtuales (RPV)<sup>3</sup>:** una red privada virtual es una tecnología de red que permite extender una red local sobre una red pública o no controlada, como por ejemplo Internet, garantizando las condiciones técnicas y de seguridad de una red local común.
- **Gestión y control de acceso e identidad:** se trata de mecanismos que permiten gestionar usuarios y sus datos de identificación. A partir de ello es posible asociar roles, perfiles, políticas de seguridad y controlar el acceso a los recursos. En el contexto de la seguridad en red podemos distinguir:
  - o Herramientas de control de acceso a la red corporativa: en inglés NAC (*Network Access Control*), son herramientas destinadas a proporcionar mecanismos para administrar y controlar el acceso de usuarios y de otras redes a los servicios de la red corporativa. Suelen incluir una función preventiva ante intrusiones y usos indebidos y una función de refuerzo de políticas basada en la identidad, roles y permisos de los usuarios.

---

<sup>2</sup> Siglas en inglés de *Intrusion Prevention Systems/ Intrusion Detection Systems*

<sup>3</sup> Es frecuente la denominación en inglés VPN (*Virtual Private Network*)

- o Gestión de identidad y autenticación: son herramientas centradas en la gestión de la identidad, que proveen un repositorio centralizado de usuarios y permiten realizar una autenticación y autorización centralizada a los sistemas y recursos de una organización. Aplican a los usuarios perfiles, privilegios, roles y políticas de uso de los recursos. Así mismo los servidores de autenticación.
- o Herramientas *Single Sign-On*: son herramientas que permiten el acceso a distintos sistemas o ubicaciones con un mecanismo de identificación común. Esto se realiza mediante la propagación de una identidad única y su asociación a los diversos servicios y recursos de una organización.

### **1.2.3 Zona desmilitarizada (DMZ)**

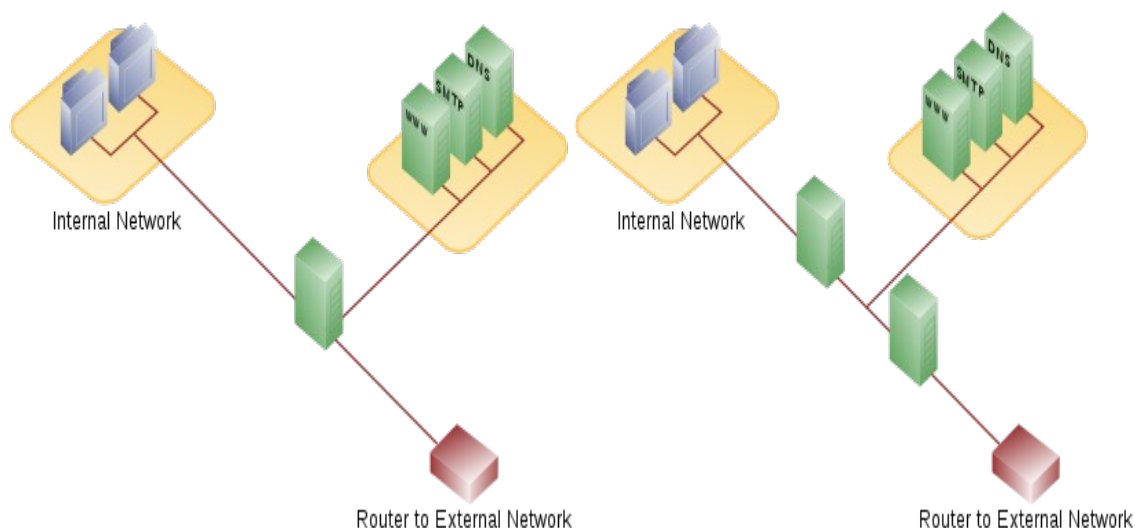
Una DMZ es una red local que se ubica entre la red interna de una organización y una red externa, generalmente Internet.

El objetivo de una DMZ es que las conexiones desde la red interna y la externa a la DMZ estén permitidas, mientras que las conexiones desde la DMZ sólo se permitan a la red externa, es decir, los equipos en la DMZ no pueden conectar con la red interna.

Esto permite que los equipos de la DMZ puedan dar servicios a la red externa a la vez que protegen la red interna en el caso de que intrusos comprometan la seguridad de los equipos (host) situados en la zona desmilitarizada.

El resultado es que la DMZ se convierte en una línea de protección para la red interna, ya que los agujeros de seguridad explotables desde el exterior jamás podrán alcanzar la red interna.

El concepto de DMZ es muy sencillo. Se trata de incluir la DMZ entre Internet y la red interna, separándola por uno o dos cortafuegos que garantizan las restricciones de acceso descritas.



DMZ implementadas con uno o con dos cortafuegos. Fuente: Wikipedia

La DMZ se usa habitualmente para ubicar servidores que sean necesariamente accesibles desde el exterior y que por ello sean especialmente vulnerables a ataques. Se trata de servidores de correo electrónico, Web y DNS. Cada uno de los equipos situados en la DMZ suele denominarse *Bastion Host*<sup>4</sup>, y se consideran como equipos especialmente sensibles a ataques, cuya seguridad debe ser gestionada con especial atención.

Las conexiones que se realizan desde la red externa hacia la DMZ se controlan generalmente utilizando *Port Address Translation* (PAT). El PAT es parte del estándar Network Address Translation (NAT), que traduce conexiones TCP y UDP hechas por un host y un puerto en una red externa a otra dirección y puerto de la red interna. Permite que una sola dirección IP

---

<sup>4</sup> En analogía con los bastiones de las fortalezas medievales, que eran puntos de ellas especialmente defendidos.



sea utilizada por varias máquinas de la intranet. Con PAT, una IP externa puede responder hasta a 64000 direcciones internas.

A la inversa, cuando un ordenador de la intranet manda un paquete hacia afuera, queremos ocultar su dirección real. El servicio NAT remplacea la IP interna con la nueva IP del propio servicio. Luego asigna a la conexión un puerto de la lista de puertos disponibles, inserta el puerto en el campo apropiado del paquete de datos y envía el paquete.

El servicio NAT crea una entrada en su tabla de direcciones IP internas, puertos internos y puertos externos. A partir de entonces, todos los paquetes que provengan del mismo hosts serán traducidos con los mismos puertos. Si hay alguna respuesta para ese paquete, el mecanismo PAT será capaz ahora de determinar cuál es la máquina a la que corresponde el puerto de destino.

Una DMZ se crea a menudo a través de las opciones de configuración del cortafuegos, donde cada red se conecta a un puerto distinto de éste. Esta configuración se llama cortafuegos en trípode (*three-legged firewall*). Un planteamiento más seguro es usar dos cortafuegos, donde la DMZ se sitúa en medio y se conecta a ambos cortafuegos, uno conectado a la red interna y el otro a la red externa. Esta configuración ayuda a prevenir configuraciones erróneas accidentales que permitan el acceso desde la red externa a la interna. Este tipo de configuración también es llamado cortafuegos de subred monitoreada (*screened-subnet firewall*).

## **2 TÉCNICAS DE SEGURIDAD PREVENTIVAS Y REACTIVAS. CORTAFUEGOS. INTRUSIONES. ACCESOS NO AUTORIZADOS**

La seguridad de una red no sólo consiste en prevenir los posibles ataques que esta pueda sufrir sino que se deben implementar técnicas que permitan reaccionar ante estos. Hablamos así de técnicas preventivas y reactivas.

Las técnicas preventivas nos permitirán reducir el número de ataques con impacto en nuestra red mientras que las reactivas nos ayudarán reaccionar para conseguir evitar un ataque y a recuperar la operatividad de la red en caso de que un ataque llegue a tener éxito y dañe nuestros.

En este contexto, los mecanismos de seguridad más habituales son: cortafuegos, sistemas de detección de intrusos y redes privadas virtuales.

### **2.1 Cortafuegos**

El mayor nivel de seguridad que se podría adoptar en un sistema informático consistiría en desenchufarlo de la red. Sin embargo, esta no suele ser una opción aceptable ya que, a día de hoy, la mayor parte de las empresas necesitan intercambiar información, ya sea a través de una red local o a través de internet.

Aunque se pueden tener todos los servidores y estaciones de trabajo protegidas con fuertes medidas de seguridad, no suele ser suficiente con la seguridad basada en host. Una medida complementaria, que es ampliamente aceptada, consiste en complementar la seguridad del host con un **servicio de cortafuegos**<sup>5</sup>. Un cortafuegos es un elemento que se sitúa entre la red interna y los intrusos potenciales externos constituyendo una barrera de protección. Normalmente se establece entre la red local e Internet, haciendo así de muro de protección exterior de la red local. El *cortafuegos* puede ser un software en un equipo o puede ser un conjunto

---

<sup>5</sup> Es muy común el uso del término en inglés *Firewall*.

formado por varios sistemas específicos destinados a controlar el acceso a la red interna de la organización.

### 2.1.1 Características de un cortafuegos

Un servicio de cortafuegos presenta una serie de características entre las que se pueden destacar las siguientes:

- Define un punto único de control, que permite alejar a los usuarios no autorizados de la red protegida, y proporciona protección contra ataques de *spoofing* y enrutado. Esto simplifica la gestión, porque las características de seguridad están centralizadas en un único sistema o conjunto de sistemas.
- Proporciona una localización para monitorizar eventos relacionados con la seguridad. En un sistema *cortafuegos* se pueden implementar auditorías de seguridad y alarmas.
- Proporciona una plataforma para otras funciones de Internet que no están relacionadas con la seguridad: traducción de direcciones, funciones de gestión para auditoría y registros de log para medir el uso de Internet.
- Puede ser utilizado para implementar redes privadas virtuales (RPV).

### 2.1.2 Tipos de cortafuegos

Atendiendo a la función desempeñada por el sistema cortafuegos podemos establecer la siguiente clasificación:

- **Cortafuegos de filtrado de paquetes:** aplica un conjunto de reglas para cada paquete IP de entrada y salida y a continuación reenvía o descarta el paquete. Las reglas de filtrado están basadas en la información contenida en cada paquete: IP origen, IP destino, protocolo, interfaz, etc.
- **Cortafuegos de inspección de estados:** Además de hacer el filtrado en base a paquetes guarda información de las sesiones y

conexiones abiertas. Esto permite políticas de seguridad más avanzadas y evitar muchos ataques.

- **Proxy de aplicaciones:** Actúa como intermediario en el tráfico de la capa de aplicación permitiendo acceder a ciertas características de las aplicaciones y reenviando la información.

### 2.1.3 Localizaciones del Cortafuegos y configuraciones

Atendiendo a la localización del cortafuegos se pueden obtener distintas configuraciones:

- **Redes DMZ:** se utilizan para aislar una o varias subredes (conocidas como “zonas desmilitarizadas”) los principales servicios de la organización que tienen que ser accedidos desde el exterior (Web, correo, DNS). A este segmento se le aplican unas reglas de filtrado para garantizar una conectividad controlada desde el exterior. Al resto de la red interna se le cierra el acceso desde el exterior.
- **Redes privada virtuales (RPV):** es una solución que ofrece grandes ventajas a los gestores de red. Consiste en dar acceso mediante una conexión segura a un equipo o segmento de una red privada a través de una red considerada insegura como puede ser Internet.
- **Cortafuegoss Distribuidos:** consiste en agrupar bajo un mismo mecanismo de control centralizado la gestión de dispositivos *cortafuegos* y la gestión de *cortafuegos* basados en host. Con estas herramientas, el administrador puede establecer políticas de seguridad y aplicarlas a equipos *cortafuegos*, servidores y estaciones de trabajo tanto locales como remotas. Además proporcionan capacidades de monitorización y alertas de seguridad.

#### 2.1.4 Netfilter/iptables

Los sistemas Linux incluyen en su *kernel* una aplicación que permite desarrollar nuestro propio cortafuegos personalizado.

**Netfilter** es un framework disponible en el núcleo Linux que permite interceptar y manipular paquetes de red. Dicho *framework* permite realizar el manejo de paquetes en diferentes estados del procesamiento. *Netfilter* es también el nombre que recibe el proyecto que se encarga de ofrecer herramientas libres para cortafuegos basados en Linux.

El componente más popular construido sobre *Netfilter* es *iptables* (anteriormente conocido como *ipchains*), una herramienta de cortafuegos que permite no solamente definir políticas de filtrado del tráfico que circula por la red, sino también realizar traducción de direcciones de red (NAT) para IPv4 o mantener registros de *log*.

Iptables se configura mediante reglas que determinan que hacer con cada paquete analizado por el *firewall*. Las reglas se agrupan en *cadena*s, de modo que cada cadena es una lista ordenada de reglas. Y a su vez, las cadenas se agrupan en *tabla*s, estando cada tabla asociada a un tipo diferente de procesamiento de paquetes. El contenido de las reglas es sencillo, ya que se basa en especificar qué paquetes la cumplen y cuál es el destino del paquete si es que cumple la regla.

Las tablas que se incluyen por defecto (es posible incorporar más) son:

- **Tabla *Filter*** (tabla de filtros): es la encargada de filtrar los paquetes permitiendo o no su paso. Contiene tres cadenas y todo paquete pasará por una de ellas: INPUT (entrada), OUTPUT (salida) y FORWARD (redirección).
- **Tabla *NAT*** (tabla de traducción de direcciones de red): se encarga de configurar las reglas de reescritura de direcciones o

de puertos de los paquetes, ya que es habitual que el *firewall* se encuentre en un punto de la red en el que decida sobre el enrutado de los paquetes hacia redes internas (mediante NAT y/o PAT). Cuenta también con tres cadenas: PREROUTING, POSTROUTING Y OUTPUT.

- **Tabla *Mangle*** (tabla de destrozo): esta tabla es la responsable de ajustar las opciones de los paquetes, como por ejemplo la [calidad de servicio](#). Todos los paquetes pasan por esta tabla. Contiene todas las cadenas predefinidas en las anteriores.

En lo que respecta al destino de los paquetes, *iptables* incluye por defecto los siguientes destinos como consecuencia de que una regla se cumpla:

- **ACCEPT:** el paquete se acepta, y por lo tanto se realizará la acción determinada por el paquete y la cadena por la que entró. Por ejemplo, si fuese INPUT se permitirá que sea recibido por el host de destino.
- **DROP:** el paquete no es aceptado y se descarta sin ningún otro tipo de acción ni notificación al origen ni al destino.
- **QUEUE:** el paquete se envía a una cola que puede ser manipulada por otra aplicación. Si ninguna aplicación lee esta cola, su semántica es la misma que la de DROP.
- **RETURN:** el paquete deja de circular por la cadena en cuya regla se ejecutó el destino RETURN. Si dicha cadena es una subcadena de otra, el paquete continuará por la cadena superior como si nada hubiera pasado. Si por el contrario la cadena es una cadena principal (por ejemplo la cadena INPUT), al paquete se le aplicará la política por defecto de la cadena en cuestión (ACCEPT, DROP o similar).

Un ejemplo de la sintaxis para añadir una regla que descartaría todos los paquetes UDP recibidos sería la siguiente: `iptables -A INPUT -p udp -j DROP`

## 2.2 Sistemas de detección de intrusiones (IDS/IPS)

Tal y como hemos apuntado, un *firewall* es un mecanismo de seguridad, que permite cerrar todos aquellos puertos de servicios que no se estén utilizando y así reducir la posibilidad de ataques por parte de intrusos. Pero aunque se permita solamente el acceso a los servicios básicos y teóricamente seguros, estos a veces tienen vulnerabilidades que pueden ser aprovechadas por un atacante con fines maliciosos para saltarse esta medida de protección. Haciendo un símil con la protección de una casa, el *firewall* se correspondería con la puerta de entrada, pero es necesario un sistema de alarma que se encargue de avisar en caso de que un intruso logre entrar. El elemento correspondiente al sistema de alarma en el mundo de la seguridad informática es el IDS (*Intrusion Detection System*). Se puede definir un IDS como un sistema que se encarga de vigilar la red, absorbiendo todo el tráfico e inspeccionándolo en busca de patrones de ataque. Las características principales de los IDS son las siguientes:

- Añade un alto nivel de integridad al resto de la red, ya que, en cierta forma, sabemos que el resto de sistemas están bien porque el IDS no avisa de lo contrario.
- Puede monitorizar la actividad de un atacante. Dependiendo de la infraestructura de IDS, se podrá monitorizar esta actividad en un único segmento o en varios.
- Alerta ante patrones de ataque comunes conocidos.
- Automatiza la búsqueda de nuevos patrones de ataque, ya que proporcionan herramientas estadísticas de búsqueda y monitorización de tráfico anómalo.

- Puede detectar ataques en tiempo real.
- Puede detectar errores de configuración en los equipos.

Los sistemas IDS constan de un equipo con una consola central de administración y una o varias “sondas” que se encargan de capturar el tráfico que se debe analizar. Dependiendo de la arquitectura de red de la organización se pueden seguir diversos criterios para ubicar las sondas: en la DMZ, detrás del *firewall*, en los accesos de usuario, entre la Extranet e Internet, etc.

Una variante de los sistemas de detección de intrusiones son los IPS (*Intrusion Prevention System*). La diferencia con respecto a los IDS estriba en que los IPS además de detectar un ataque y generar la correspondiente alerta son capaces de actuar e intentar neutralizar el ataque, por lo que, a pesar de su nombre, se trata de un mecanismo de defensa reactivo. Un ejemplo claro sería cuando el IPS detecta actividad maliciosa por parte de un usuario conectado a un servidor a través de una conexión remota. Una de las acciones drásticas que podría tomar el IPS es cortar la conexión.

Los sistemas IPS/IDS se pueden instalar como un software en un servidor (ej. Snort) o puede ser un equipo hardware con su software completo e independiente proporcionado por un fabricante de dispositivos de seguridad.

Un IDS consigue su objetivo de detección usando una de las dos siguientes técnicas:

- Heurística: determina actividad normal de red, como el orden de ancho de banda usado, protocolos, puertos y dispositivos que generalmente se interconectan, y alerta a un administrador o usuario cuando este varía del considerado como normal, clasificándolo como anómalo.



- **Patrón:** analiza paquetes en la red, y los compara con patrones de ataques conocidos, y preconfigurados. Estos patrones se denominan firmas. Esta técnica provoca que exista un periodo de tiempo entre el descubrimiento del ataque y su patrón, hasta que este es finalmente configurado en un IDS. Durante este tiempo, el IDS será incapaz de identificar el ataque.

## 2.3 **Redes privadas virtuales (RPV)**

Las redes privadas virtuales (RPV<sup>6</sup>) son un elemento cada vez más habitual en muchas organizaciones que contribuye al concepto de perímetro difuso de la red que introducíamos al principio de este tema.

Proporcionan una forma segura de conectarse desde una ubicación remota con una red de área local privada (LAN) a través de Internet o cualquier otra red pública no segura. Una RPV es una conexión que tiene la apariencia y muchas de las ventajas de un enlace dedicado pero trabajando sobre una red pública.

Para esto se utiliza una técnica llamada *tunneling* que permite enrutar los paquetes de datos por la red pública en un túnel privado que simula una conexión punto a punto. Las RPV son utilizadas frecuentemente por los trabajadores remotos o empleados en las delegaciones de la organización, para compartir datos y recursos de la red privada.

### 2.3.1 **Beneficios de las redes privadas virtuales**

Una red privada virtual puede proporcionar los siguientes beneficios para la organización:

1. **Rendimiento predecible:** en RPVs con canales dedicados se puede garantizar el ancho de banda entre los sitios y el rendimiento de la red se hace más predecible.

---

<sup>6</sup> Es frecuente la denominación en inglés VPN (*Virtual Private Network*)

2. **Independencia en la elección de las tecnologías de transporte para las redes de usuarios:** las posibilidades están limitadas por la elección de un proveedor o fabricante. Así la organización puede usar Ethernet, Frame Relay, IP y otras tecnologías de red para conectar sus sitios.
3. **Seguridad mejorada:** al reducir el número de conexiones con el mundo exterior se reduce considerablemente la posibilidad de un ataque. Además también se reduce la posibilidad de interceptación del tráfico.
4. **Espacio de direcciones IP independiente:** en las redes privadas es posible utilizar cualquier direccionamiento. Por ejemplo, casi todos los servicios de RPV permiten el uso de direcciones IP privadas tales como 10.0.0.1 o 192.168.0.3, que no pueden ser enrutadas a través de las redes públicas.

Estas características serán de utilidad para algunos usuarios, pero de importancia relativa para otros. Las vulnerabilidades y bajo rendimiento de las redes públicas pueden hacer que la “seguridad mejorada” y “rendimiento predecible” sean las características más deseables de una RPV.

Recientemente, la “independencia de elección de tecnología” y el “espacio de direcciones independiente” parece que se han vuelto menos importantes: el primero debido a la dominación de las tecnologías Ethernet en capa 2 e IP en capa 3. La segunda debido a que con la implantación de IPv6 se espera acabar con el déficit de direcciones.

De todas formas, tener un espacio de direcciones independiente mejora la seguridad, utilizando rangos de direcciones para separar sitios dentro de la organización y restringir accesos.

A la hora de crear una RPV para proporcionar acceso remoto, hay que escoger la tecnología que mejor se adapte al escenario en cuestión. Esta elección de tecnología implica técnicas de *tunneling*, autenticación, control de acceso y seguridad de datos.

### 2.3.2 Implementación de una RPV

Generalmente se definen tres arquitecturas principales de RPV:

- **RPV de acceso remoto:** en este tipo de RPVs se ubicarían los usuarios que desde un host remoto crean un túnel para conectarse a la red privada de la organización. El dispositivo remoto puede ser un equipo personal con un software cliente para crear RPV, y usar una conexión conmutada o una conexión de banda ancha permanente.
- **Extranet RPV:** este tipo de arquitecturas permiten que ciertos recursos de la red privada de la organización sean accedidos por redes de otras compañías, tales como clientes o proveedores. En este escenario es fundamental el control de acceso.
- **Intranet RPV (LAN-to-LAN RPV):** en este escenario las redes remotas de la organización son conectadas entre sí utilizando la red pública, convirtiéndose de esta manera en una única red LAN corporativa global.

Hay una amplia variedad de tecnologías que se pueden utilizar para la implementación de RPVs. Los criterios que deben cumplir estas tecnologías son:

- **Eficiencia:** los tiempos de respuesta deben ser adecuados y comparables con las redes consideradas “no seguras.

- **Facilidad de administración:** los usuarios y administradores de este tipo de redes, pueden hacer su trabajo de una manera rápida y efectiva.
- **Seguridad:** las conexiones deben ser brindadas cumpliendo con los requisitos de autenticación, autorización, privacidad, integridad y contabilidad.
- **Cumplimiento de estándares e interoperabilidad:** hay muchas tecnologías que son estándares y que participan en la creación de RVPs: IPSec, MD5, SOCKSv5, IKE, ISAKMP, Diffie-Hellman, X.509, RADIUS, etc.

Para clasificar las tecnologías que se pueden utilizar en la implementación de una arquitectura de RVP, se puede tomar como referencia el modelo OSI y ubicarlas en función del nivel en el que son implementadas. La Figura 3 muestra los principales protocolos utilizados para el establecimiento de conexiones RVP y su ubicación en el modelo de referencia OSI.

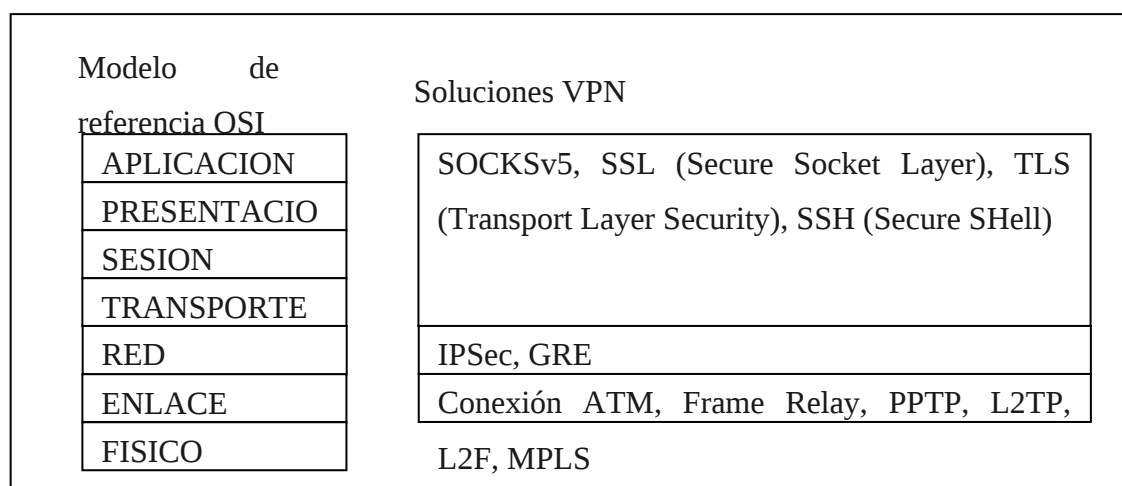


Figura 3: Ubicación de las soluciones RVP en el modelo de referencia OSI

A la hora de implementar una arquitectura de red basada en RVP hay dos posibles opciones. Hacer una implementación por hardware o por software.

Las implementaciones por Hardware realizan el proceso de encriptación y desencriptación del tráfico a nivel físico entre los extremos de la línea de comunicación. Los dispositivos utilizados normalmente son routers con capacidades de RPV incorporadas.

Como ventajas de esta solución se puede decir que la instalación y configuración son relativamente sencillas, no se necesita personal especializado y su mantenimiento es mínimo. Por el contrario presentan el inconveniente de que el sistema de encriptación viene impuesto por el fabricante y se depende del mismo para las actualizaciones.

Por otro lado las implementaciones basadas Software se están imponiendo cada día más. La explicación radica en que, la necesidad de los usuarios pequeños y medianos de implantar sistemas de seguridad en el acceso a sus máquinas va en aumento.

Las implementaciones software son mucho más baratas que comprar hardware preparado para RPVs y existe un gran número de RPVs desarrolladas por software. Como inconvenientes de esta aproximación se puede decir que es necesaria una máquina para dar soporte a la solución, el sistema de claves y certificados reside en máquinas potencialmente inseguras y en los casos que se utilice software de libre distribución puede ser que tenga puertas traseras u otras deficiencias de seguridad.

### 3 REFERENCIAS<sup>7</sup>

- **STALLINGS, W.** (2011). Network Security Essentials. Applications and Standards. Fourth Edition. Prentice Hall.
- Instituto Nacional de Tecnologías de la Comunicación – Centro de Respuestas a Incidentes de Seguridad.  
<http://cert.inteco.es>
- TLDP-ES/LuCAS: servicios editoriales para la documentación libre en español de Hispalinux.  
<http://es.tldp.org/>

**Autor: Juan Otero Pombo**

**Ingeniero en Informática en el Concello de Ourense**

**Colegiado del CPEIG**

---

<sup>7</sup> Todos los enlaces fueron verificados en noviembre de 2011.

# **38. CERTIFICADOS DIGITALES. TARJETAS CRIPTOGRÁFICAS. FIRMA DIGITAL. TÉCNICAS DE CIFRADO. INFRAESTRUCTURA DE CLAVE PÚBLICA (PKI).**

## **Tema 38: Certificados digitales. Tarjetas criptográficas. Firma digital. Técnicas de cifrado. Infraestructura de clave pública (PKI).**

### **INDICE**

<b>1. TECNICAS DE CIFRADO.....</b>	<b>2</b>
1.1. CRIPTOGRAFÍA Y CRIPTOANÁLISIS.....	2
1.2. TÉCNICAS CRIPTOGRÁFICAS CLÁSICAS.....	3
1.3. TÉCNICAS CRIPTOGRÁFICAS MODERNAS.....	3
1.4. CRIPTOGRAFÍA DE CLAVE PRIVADA O SIMÉTRICA.....	5
1.4.1. <i>Sustitución monoalfabeto</i> .....	5
1.4.2. <i>Sustitución polialfabeto</i> .....	6
1.4.3. <i>Cifrado en bloque</i> .....	7
1.4.4. <i>Cifrado en flujo</i> .....	13
1.4.5. <i>Cifrado en base a funciones resumen</i> .....	15
1.5. CRIPTOGRAFÍA DE CLAVE PÚBLICA O ASIMÉTRICA.....	16
1.5.1. <i>Ejemplos de algoritmos de clave pública: Diffie-Hellman</i> .....	18
1.5.2. <i>Ejemplos de algoritmos de clave pública: RSA</i> .....	18
<b>2. FIRMA DIGITAL.....</b>	<b>18</b>
<b>3. INFRAESTRUCTURA DE CLAVE PÚBLICA (PKI).....</b>	<b>20</b>
3.1. CERTIFICADOS DIGITALES.....	22
3.1.1. <i>Autoridad de Certificación</i> .....	24
3.1.2. <i>Autoridad de Registro</i> .....	27
3.1.3. <i>Autoridad de Validación</i> .....	28
3.1.4. <i>Autoridad de Sellado de Tiempos</i> .....	28
3.1.5. <i>Directorio de Certificados</i> .....	29
3.2. HARDWARE CRIPTOGRÁFICO .....	30
3.3. TARJETAS Y CHIP CRIPTOGRÁFICOS.....	30
3.4. 3.3. MARCO LEGAL Y ESTÁNDARES.....	31
<b>4. REFERENCIAS.....</b>	<b>34</b>



## 1. TECNICAS DE CIFRADO

La palabra **criptografía** es una palabra de origen griego (krypto -oculto- y graphos -escribir-) y se define como el arte de escribir con clave secreta o de un modo enigmático.

### 1.1. Criptografía y criptoanálisis

La historia de la criptografía se remonta a miles de años atrás y tiene una larga tradición en las escrituras religiosas que podrían ofender a la cultura dominante o a las autoridades políticas. La finalidad de esta técnica ha sido siempre enviar mensajes confidenciales con la garantía de que sólo el destinatario de los mismos pudiera acceder a la información contenida en el mensaje.

El método consiste en la aplicación de una transformación al mensaje conocida como **cifrado**, con el objetivo de que las personas que desconozcan la transformación realizada sean incapaces de acceder a la información contenida en el mensaje.

El estudio de técnicas destinadas a encontrar el sentido de una información cifrada, sin tener acceso a la información secreta requerida, es el **criptoanálisis**. La finalidad del criptoanálisis es, por tanto, descubrir la clave de cifrado. La vulnerabilidad de los algoritmos de cifrado dependerá de la dificultad de la tarea de descubrimiento de la clave. Una ataque por fuerza bruta consiste en buscar la clave de cifrado probando uno a uno todos los posibles valores de la.

La **criptología** es la disciplina que abarca la criptografía y el criptoanálisis.

Otro concepto relacionado es la **esteganografía**. Al igual que la criptografía lo que busca es ocultar un mensaje ante un posible atacante, pero la diferencia estriba en cómo ocultan la información ambas técnicas:

mientras que la criptografía pretende que la información no sea descifrada, la esteganografía lo que pretende es que la información pase desapercibida (por ejemplo un código secreto tatuado en el cuero cabelludo y oculto por el pelo)

Las técnicas criptográficas se pueden clasificar siguiendo varios criterios. Siguiendo un **criterio temporal** se pueden clasificar en clásicas o extemporáneas y modernas o contemporáneas.

## **1.2. Técnicas criptográficas clásicas**

Las técnicas criptográficas clásicas, realizan el cifrado en base a la sustitución y transposición de los caracteres del mensaje. El secreto está en el algoritmo aplicado al mensaje, por lo que tienen el inconveniente de que si un atacante lo descubre, será capaz de interpretar todos los mensajes cifrados que capture. Como ejemplos podemos citar:

- **Sustitución monoalfabeto:** consiste en la sustitución de símbolos uno a uno. Como ejemplo se puede citar el algoritmo de César.
- **Sustitución polialfabeto:** consiste en la sustitución de un símbolo por uno de un conjunto. Como ejemplo se puede citar el cifrado de Vigenère.
- **Transposición:** consiste en cambiar el orden de los símbolos.
- **Combinación de sustitución y transposición** (máquinas rotoras).

## **1.3. Técnicas criptográficas modernas**

Las técnicas criptográficas modernas, a diferencia de las clásicas, utilizan claves de cifrado para cifrar la información. Una premisa fundamental de la criptografía moderna es que la seguridad del método debe depender únicamente de la clave de cifrado, debiendo ser los algoritmos conocidos.

Esta premisa hace que estas técnicas resulten mucho más seguras y efectivas ya que resulta más sencillo mantener el secreto de la clave y además cambiar la clave de cifrado siempre será menos costoso que idear un nuevo algoritmo resultando frecuente que la clave de cifrado se genere de forma automática.

Podemos clasificar los algoritmos de cifrado atendiendo a las claves que utilizan o al modo en que procesan la información.

Si nos fijamos en el tipo de claves que utilizan tenemos dos tipos de algoritmos:

- Algoritmos de **cifrado simétrico o de clave privada**: Utilizan la misma clave para el cifrado y el descifrado por lo que debe ser secreta y compartida por el emisor y el receptor. Ejemplos de algoritmos de este tipo son DES, 3DES, AES, IDEA, RC5, etc.
- Sistemas de cifrado **asimétrico o de clave pública**: Utilizan un par de claves generadas por el emisor. Una de las claves es pública, es decir, conocida por todo el mundo y la otra es privada o secreta de forma que lo que se cifra con una clave es descifrado por la otra y viceversa. Ejemplos de algoritmos de este tipo son RSA, DSA, Diffie-Hellman, ElGamal, etc.

Si nos fijamos en el modo en que procesan tenemos 3 tipos de algoritmos:

- Técnicas criptográficas de cifrado en **modo flujo** (*stream cipher*): estos algoritmos de cifrado se basan en la combinación de un texto en claro con un texto de cifrado obtenido a partir de una clave. La característica fundamental es que se va cifrando un flujo de datos bit a bit. Ejemplos: RC4, SEAL.
- Técnicas criptográficas de cifrado en **modo bloque** (*block cipher*): se caracterizan porque el algoritmo de cifrado o descifrado se aplica separadamente a bloques de longitud  $l$ , y para cada uno de ellos el

resultado es un bloque de la misma longitud: Ejemplos: DES, 3DES, AES.

- Técnicas criptográficas basadas en funciones resumen (**hash functions**): la característica principal de estos algoritmos es que permiten obtener una cadena de bits de longitud fija a partir de un mensaje de longitud arbitraria: Ejemplos: MD5, familia SHA.

#### 1.4. Criptografía de clave privada o simétrica

La criptografía de clave simétrica, se caracteriza porque la clave de descifrado  **$k$** , es idéntica a la clave de cifrado o se puede obtener a partir de esta, residiendo de este modo la fortaleza del algoritmo en el secreto de la misma.

Si  **$M$**  es el mensaje en claro que se quiere proteger, al cifrarlo con un algoritmo en base a una clave privada  **$E_k(M)$**  se obtiene otro mensaje llamado texto cifrado  **$C$** . Para que este cifrado sea útil, existe otra función  **$D_k(C)$**  que a partir del texto cifrado por el emisor permite obtener de nuevo el mensaje en claro  **$M$** .

$$C = E_k(M)$$

$$M = D_k(C) = D_k(E_k(M))$$

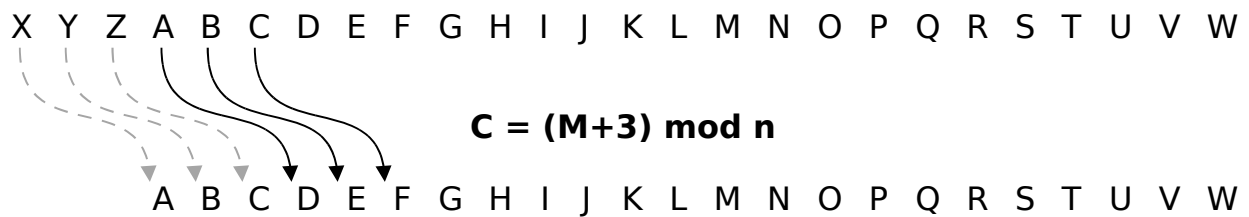
La seguridad del sistema recae pues en mantener en secreto la clave  **$k$** . El principal inconveniente de la criptografía simétrica, es el **intercambio de claves**. Este problema se soluciona con ayuda de la criptografía asimétrica.

##### 1.4.1. Sustitución monoalfabeto

Un ejemplo de algoritmo de cifrado por sustitución monoalfabeto es el cifrado César. Es un tipo de cifrado por sustitución en el que cada letra en el texto original es reemplazada por otra letra que se encuentra un número



fijo de posiciones más adelante en el alfabeto. Por ejemplo, con un desplazamiento de 3, la A sería sustituida por la D (situada 3 lugares a la derecha de la A), la B sería reemplazada por la E, etc. Este método debe su nombre a Julio César, que lo usaba para comunicarse con sus generales.



### **Ejemplo:**

<b>Mensaje en</b>	<b>H O L A C E S A R</b>
<b>claro:</b>	
<b>Mensaje</b>	<b>K R O D G H V D U</b>
<b>cifrado:</b>	

El descifrado de un mensaje consistiría en sustituir cada letra del texto por la que hay tres posiciones delante en el alfabeto.

La principal condición que debe cumplir la clave es que ha de ser una permutación del alfabeto, es decir, no puede haber letras repetidas ni faltar ninguna. Si no, la transformación no sería invertible en general.

### **1.4.2. Sustitución polialfabeto**

El inconveniente de los algoritmos de sustitución monoalfabeto, es que el texto cifrado mantiene la misma distribución de frecuencia de caracteres que tiene el texto claro original, lo que hace que sean criptoanalizables por métodos estadísticos sencillos. Una posible mejora de los cifrados por sustitución es intentar métodos que destruyan esa correspondencia de frecuencias entre el mensaje en claro y el criptograma. Por ejemplo, utilizando varios alfabetos a la vez para el cifrado. En los cifrados

polialfabéticos la sustitución aplicada a cada carácter varía en función de la posición que ocupe este dentro del texto claro. En realidad corresponde a una aplicación cíclica de  $n$  cifrados de sustitución monoalfabeto. Un ejemplo típico de cifrado polialfabético es el Cifrado **de Vigenère**.

### **1.4.3. Cifrado en bloque**

Un algoritmo de cifrado en bloque toma como entrada un bloque de longitud fija y una clave y genera un nuevo bloque cifrado de la misma longitud que el bloque de entrada.

La técnica consiste en dividir el texto a cifrar (con longitud  $L$ ) en bloques de tamaño  $b$  y a continuación cifrar cada uno de los bloques. Si  $L$  no es múltiplo de  $b$ , se agregan bits adicionales para conseguir que todos los bloques estén completos. Para descifrar el mensaje se procede de manera análoga.

Muchos de los algoritmos de cifrado en bloque se basan en la combinación de dos operaciones básicas: sustitución y transposición.

- La **sustitución** consiste en traducir cada bloque de bits que llegan como entrada a otro de salida siguiendo una permutación determinada. El cifrado César sería un ejemplo simple de sustitución en el que cada grupo de bits correspondería a una letra.
- La **transposición** consiste en reordenar la información del texto en claro según un patrón determinado. Un ejemplo podría ser la formación de grupos de cinco letras, incluidos los espacios en blanco, y rescribir cada grupo (1, 2, 3, 4, 5) en el orden (3, 1, 5, 2, 4). Por ejemplo:

Texto en claro: "HOLA MUNDO"

Texto cifrado: "LH OANMOUD"

La transposición no dificulta extraordinariamente el criptoanálisis, pero puede combinarse con otras operaciones para añadir complejidad a los algoritmos de cifrado.

El **producto de cifras**, o combinación en cascada de distintas transformaciones criptográficas es una técnica muy efectiva para implementar algoritmos bastante seguros de forma sencilla. Por ejemplo, muchos algoritmos de cifrado en bloque se basan en una serie de iteraciones de productos sustitución-transposición.

Dos propiedades deseables en un algoritmo criptográfico son la **confusión y la difusión**. La confusión consiste en esconder la relación entre la clave y las propiedades estadísticas del texto cifrado. La difusión propaga la redundancia del texto en claro a lo largo del texto cifrado para que no sea fácilmente reconocible.

La confusión consigue que, cambiando un solo bit de la clave, cambien muchos bits del texto cifrado, y la difusión implica que el cambio de un solo bit del texto en claro afecte también a muchos bits del texto cifrado.

### ***Modos de operación del cifrado en bloque***

Un aspecto que hay que tener en cuenta cuando se utiliza el cifrado es que, aunque se puede conseguir que un atacante no descubra directamente los datos transmitidos, en ocasiones es posible que se pueda deducir información indirectamente. Por ejemplo, en un protocolo que utilice mensajes con una cabecera fija, la aparición de los mismos datos cifrados varias veces en una transmisión puede indicar dónde empiezan los mensajes. Para intentar contrarrestar esto, el cifrado bloque opera en varios modos:

- El modo **ECB** (Electronic Codebook): consiste en dividir el texto en bloques y cifrar cada uno de ellos de forma separada. El

inconveniente de este método es que bloques idénticos de mensaje sin cifrar producirán idénticos textos cifrados.

- En el modo **CBC** (Cipher Block Chaining), antes de ser cifrado, a cada bloque de texto se le aplica una operación **XOR** bit a bit, con el previo bloque ya cifrado. De este modo, cada bloque es dependiente de todos los bloques de texto previos hasta ese punto. Además, para hacer cada mensaje único se puede usar un [vector de inicialización](#). CBC es el modo usado más a menudo. Su principal contrapartida es que es secuencial y no puede funcionar en paralelo.
- En el modo **CFB** (Cipher Feedback), el algoritmo de cifrado no se aplica directamente al texto en claro sino a un vector auxiliar (inicialmente igual al IV). Del resultado del cifrado se toman  $n$  bits que se suman a  $n$  bits del texto en claro para obtener  $n$  bits de texto cifrado. Estos bits cifrados se utilizan también para actualizar el vector auxiliar. El número  $n$  de bits generados en cada iteración puede ser menor o igual que la longitud de bloque  $b$ . Tomando como ejemplo  $n=8$ , tenemos un cifrado que genera un byte cada vez sin que sea necesario esperar a tener un bloque entero para poderlo descifrar.
- El modo **OFB** (Output Feedback) opera como el CFB pero en lugar de actualizar el vector auxiliar con el texto cifrado, se actualiza con el resultado obtenido del algoritmo de cifrado. La propiedad que distingue este modo de los demás consiste en que un error en la recuperación de un bit cifrado afecta solamente al descifrado de este bit.

### ***Ejemplos de algoritmos de cifrado en bloque: DES***



DES<sup>1</sup> es uno de los algoritmos de cifrado más usados en el mundo. Fue publicado en 1977 en el documento **FIPS**<sup>2</sup> PUB 46 del Instituto Nacional de Estándares y Tecnología (NIST).

El algoritmo fue controvertido al principio, con algunos elementos de diseño clasificados, una [longitud de clave](#) relativamente corta, y las continuas sospechas sobre la existencia de alguna [puerta trasera](#) para la NSA<sup>3</sup>. Posteriormente DES fue sometido a un intenso análisis académico y motivó el concepto moderno del [cifrado por bloques](#) y su [criptoanálisis](#).

Hoy en día, DES se considera inseguro para muchas aplicaciones. Esto se debe principalmente a que el tamaño de clave de 56 bits es corto. A finales de 2001 el algoritmo ha sido sustituido por el nuevo [AES](#)<sup>4</sup>.

DES es el prototipo de algoritmo de [cifrado por bloques](#): toma un texto en claro de una longitud fija de bits y lo transforma mediante una serie de operaciones básicas en otro texto cifrado de la misma longitud, dividiendo para ello el mensaje, en bloques de 64 bits. El algoritmo DES utiliza una [clave criptográfica](#) para modificar la transformación, de modo que el descifrado sólo puede ser realizado por aquellos que conozcan la clave concreta utilizada en el cifrado. La longitud de la clave es de 64 bits, aunque en realidad, sólo 56 de ellos son empleados por el algoritmo. Los ocho bits restantes se utilizan únicamente para comprobar la [paridad](#), y después son descartados.

La parte central del algoritmo consiste en dividir el mensaje de entrada en grupos de bits, hacer una sustitución distinta sobre cada grupo y, a continuación una transposición de todos los bits. Esta transformación se repite dieciséis veces: en cada iteración, la entrada es una transposición

---

<sup>1</sup> Siglas en inglés de Data Encryption Standard.

<sup>2</sup> Siglas en inglés de *Federal Information Processing Standard*.

<sup>3</sup> Siglas en inglés de *National Security Agency*.

<sup>4</sup> Siglas en inglés de *Advanced Encryption Standard*.

distinta de los bits de la clave sumada bit a bit (XOR) con la salida de la iteración anterior. Este entrecruzamiento se conoce como [esquema Feistel](#).

La estructura de *Feistel* asegura que el cifrado y el descifrado sean procesos muy similares. La única diferencia es que las subclaves se aplican en orden inverso cuando desciframos. Esto simplifica enormemente la implementación, en especial sobre hardware, al no haber necesidad de algoritmos distintos para el cifrado y el descifrado.

### ***Ejemplos de algoritmos de cifrado en bloque: 3DES***

Aunque a lo largo de los años el algoritmo DES se ha mostrado muy resistente al criptoanálisis, su principal problema es actualmente la vulnerabilidad a los ataques de fuerza bruta, a causa de la longitud de la clave, de sólo 56 bits. En los años 70 era muy costoso realizar una búsqueda entre las  $2^{56}$  combinaciones posibles, pero la tecnología actual permite romper el algoritmo en un tiempo cada vez más corto. Por este motivo, en 1999 el NIST cambió el algoritmo DES por el “Triple DES” como estándar oficial, mientras no estuviera disponible el nuevo estándar AES. El Triple DES, como su nombre indica, consiste en aplicar el DES tres veces consecutivas. Esto se puede realizar con tres claves ( $k_1$ ,  $k_2$ ,  $k_3$ ), o bien con sólo dos ( $k_1$ ,  $k_2$ , y otra vez  $k_1$ ). La longitud total de la clave con la segunda opción es de 112 bits (dos claves de 56 bits). La primera opción proporciona más seguridad, pero a costa de utilizar una clave total de 168 bits (3 claves de 56 bits), que puede ser un poco más difícil de gestionar e intercambiar. Para conseguir que el sistema sea adaptable al estándar antiguo, en el Triple DES se aplica una secuencia cifrado-descifrado-cifrado (E-D-E) en lugar de tres cifrados:

$$\mathbf{C} = \mathbf{E}(k_3, \mathbf{D}(k_2, \mathbf{E}(k_1, \mathbf{M})))$$

$$\text{o bien: } \mathbf{C} = \mathbf{E}(k_1, \mathbf{D}(k_2, \mathbf{E}(k_1, \mathbf{M})))$$

De este modo, para cifrar un mensaje  $M$ , primero se cifra con  $k_1$ , luego se descifra con  $k_2$  y finalmente se vuelve a cifrar con  $k_3$  (o  $k_1$ ). Nótese que en el caso de usar 2 claves si hacemos que  $k_1=k_2$  tenemos un sistema equivalente al DES simple.

### ***Ejemplos de algoritmos de cifrado en bloque: AES***

La longitud de la clave del algoritmo DES se ha ido convirtiendo en un problema a medida que iban aumentando las capacidades de procesamiento y el algoritmo se hacía cada vez más vulnerable a un ataque por fuerza bruta.

En 1977, a la vista de que el Triple DES no es excesivamente eficiente cuando se implementa en software, el NIST convocó a la comunidad criptográfica a presentar propuestas para un nuevo estándar que sustituyera al DES. De los quince algoritmos candidatos que se aceptaron, se escogieron cinco como finalistas (MARS, RC6, RIJNDAEL, SERPENT y TWOFISH), y en octubre de 2000 se dio a conocer el ganador: el algoritmo Rijndael, propuesto por los criptógrafos belgas Joan Daemen y Vincent Rijmen. En noviembre de 2001 se publicó el documento FIPS 197 donde AES se asumía oficialmente.

El Rijndael puede trabajar en bloques de 128, 192 o 256 bits, y la longitud de la clave también puede ser de 128, 192 o 256 bits. Dependiendo de esta última longitud, el número de iteraciones del algoritmo es 10, 12 ó 14, respectivamente. Cada iteración incluye una sustitución fija byte a byte, una transposición, una transformación consistente en desplazamientos de bits y XORs, y una suma binaria (XOR) con bits obtenidos a partir de la clave.

#### 1.4.4. Cifrado en flujo

Para algunas aplicaciones, tales como el cifrado de conversaciones telefónicas, el cifrado en bloques es inapropiada porque los flujos de datos se producen en tiempo real en pequeños fragmentos. Las muestras de datos pueden ser tan pequeñas como 8 bits o incluso de 1 bit, y sería un desperdicio rellenar el resto de los 64 bits antes de cifrar y transmitirlos.

El funcionamiento de un cifrado en flujo consiste en la combinación de un texto claro **M** con un texto de cifrado **S** que se obtiene a partir la clave simétrica **k** obteniendo un texto cifrado **C**. Para descifrar, sólo se requiere realizar la operación inversa con el texto cifrado **C** y el mismo texto de cifrado **S**.

La operación de combinación que se utiliza normalmente es la suma y como operación inversa la resta. Si el texto está formado por caracteres, el algoritmo sería como un cifrado César en que la clave va cambiando de un carácter a otro. La clave que corresponde viene dada por el texto de cifrado **S** (llamado *keystream* en inglés).

Considerando el texto formado por bits, la suma y la resta son equivalentes. Cuando se aplican bit a bit, ambas son idénticas a la operación lógica “o exclusiva”, denotada con el operador XOR (eXclusive OR). Así pues:

$$\mathbf{C} = \mathbf{M} \text{ XOR } \mathbf{S(k)}$$

$$\mathbf{M} = \mathbf{C} \text{ XOR } \mathbf{S(k)}$$

En los esquemas de cifrado en flujo, el texto claro **M** tiene una longitud variable y el texto de cifrado **S** ha de ser como mínimo igual de largo. No es necesario disponer del mensaje entero antes de empezar a cifrarlo o descifrarlo, ya que se puede implementar el algoritmo para que trabaje con un “flujo de datos” que se va generando a partir de la clave (el texto cifrado). De ahí procede el nombre de este tipo de algoritmos.



Existen varias formas de obtener el texto cifrado **S** en función de la clave **k**:

- Si se escoge una secuencia **k** más corta que el mensaje **M**, una posibilidad sería repetirla cíclicamente tantas veces como sea necesario para ir sumándola al texto en claro. El inconveniente de este método es que se puede romper fácilmente, sobre todo cuanto más corta sea la clave.
- En el otro extremo, se podría tomar directamente **S(k) = k**. Esto quiere decir que la propia clave debe ser tan larga como el mensaje que hay que cifrar. Este es el principio del conocido **cifrado de Vernam**. Si **k** es una secuencia totalmente aleatoria que no se repite cíclicamente, estamos ante un ejemplo de cifrado incondicionalmente seguro. Este método de cifrado se llama en inglés *one-time-pad* (“cuaderno de un solo uso”). Un ejemplo de uso del cifrado de Vernam ocurre a veces entre los portaaviones y los aviones. En este caso, se aprovecha que en un instante dado (antes del despegue) tanto el avión como el portaaviones están en el mismo sitio, con lo cual, intercambiarse un disco duro de 20GB con una secuencia pseudoaleatoria no es ningún problema. Posteriormente cuando el avión despegue puede establecerse una comunicación segura con el portaaviones utilizando un cifrado de Vernam con la clave aleatoria que ambos comparten.
- Lo que en la práctica se utiliza son funciones que generan **secuencias pseudoaleatorias** a partir de una **semilla** (un número que actúa como parámetro del generador), y lo que se intercambia como clave secreta **k** es solamente esta semilla. En cada paso el algoritmo se encontrará en un determinado estado, que vendrá dado por sus variables internas. Dado que las variables serán finitas, habrá un número máximo de posibles estados distintos. Esto significa que al cabo de un cierto período, los datos generados se volverán a repetir. Para que el algoritmo sea seguro, interesa que el período de

repetición sea cuanto más largo mejor (con relación al mensaje que hay que cifrar), con el fin de dificultar el criptoanálisis.

Las características de este tipo de cifrado lo hacen apropiado para entornos en los que se necesita un rendimiento alto y los recursos (capacidad de cálculo, consumo de energía) sean limitados. Por ello se suelen utilizar en comunicaciones móviles: redes sin hilos, telefonía móvil, etc.

Un ejemplo clásico de cifrado en flujo es **RC4** (Ron's Code 4). Fue diseñado por Ronald Rivest en 1987 y publicado en Internet por un remitente anónimo en 1994. Es conocido por ser el algoritmo de cifrado empleado en el sistema de seguridad **WEP** (*Wired Equivalent Privacy*) reconocido en el estándar **IEEE 802.11**. [RC4](#) utiliza claves de 64 bits (40 bits más 24 bits del vector de iniciación IV) o de 128 bits (104 bits más 24 bits del IV).

#### **1.4.5. Cifrado en base a funciones resumen**

A veces los algoritmos de cifrado no sólo se usan para cifrar datos, sino que son utilizados para garantizar la autenticidad de los mismos. Como ejemplo de algoritmo de estas características se puede citar las llamadas **funciones hash**, también conocidas como **funciones de resumen** de mensaje<sup>5</sup>.

En general, podemos decir que una función resumen nos permite obtener una cadena de bits de longitud fija, relativamente corta, a partir de un mensaje de longitud arbitraria:

$$\mathbf{H} = \mathbf{h}(\mathbf{M})$$

Para mensajes **M** iguales, la función **h** debe dar resúmenes **H** iguales. Pero si dos mensajes dan el mismo resumen **H** no deben ser necesariamente iguales. Esto es así porque sólo existe un conjunto limitado de posibles valores **H**, ya que su longitud es fija.

---

<sup>5</sup> *Message Digest*, en inglés.

Para que una función ***h*** se pueda aplicar en sistemas de autenticación, debe cumplir una serie de condiciones que le permitan ser considerada una **función resumen segura**. Entre ellas destacan la **unidireccionalidad y la resistencia a colisiones**.

Para dificultar los ataques contra las funciones de resumen, por un lado los algoritmos tienen que definir una relación compleja entre los bits de entrada y cada bit de salida. Por otro lado, los ataques por fuerza bruta se contrarrestan alargando lo suficiente la longitud del resumen.

Hasta hace poco, el algoritmo de resumen más usado era el **MD5** (*Message Digest 5*). Pero como el resumen que obtiene es de sólo 128 bits, y aparte se han encontrado otras formas de generar colisiones parciales en el algoritmo, actualmente se recomienda utilizar algoritmos más seguros, como el **SHA-1**<sup>6</sup>. El algoritmo **SHA-1**, publicado el 1995 en un estándar del NIST (como revisión de un algoritmo anterior llamado simplemente SHA), obtiene resúmenes de 160 bits. El año 2002 el NIST publicó variantes de este algoritmo que generan resúmenes de 256, 384 y 512 bits.

### 1.5. Criptografía de clave pública o asimétrica

Los **sistemas de cifrado de clave pública** o **sistemas de cifrado asimétricos** se inventaron con el fin de evitar por completo el problema del intercambio de claves de los sistemas de cifrado simétricos.

En un algoritmo criptográfico de clave pública se utilizan claves distintas para el cifrado y el descifrado. Una de ellas, la **clave pública**, se puede obtener fácilmente a partir de la otra, la **clave privada**, pero por el contrario es prácticamente imposible. Los algoritmos de clave pública típicos permiten cifrar con la clave pública ( $k_{pub}$ ) y descifrar con la clave privada ( $k_{pr}$ ):

---

<sup>6</sup> Siglas en inglés de *Secure Hash Algorithm-1*

$$C = e(k_{pub}, M)$$

$$M = d(k_{pr}, C)$$

Pero también puede haber algoritmos que permitan cifrar con la clave privada y descifrar con la pública:

$$C = e(k_{pr}, M)$$

$$M = d(k_{pub}, C)$$

En la práctica, los algoritmos utilizados permiten cifrar y descifrar fácilmente, pero todos ellos **son considerablemente más lentos que los equivalentes con criptografía simétrica**. Por eso, la criptografía de clave pública se suele utilizar solo en los problemas que la criptografía simétrica no puede resolver: el intercambio de claves y la autenticación con no repudio (firmas digitales).

Los mecanismos de **intercambio de claves** permiten que dos partes se pongan de acuerdo en las claves simétricas que utilizarán para comunicarse, sin que un tercero que esté escuchando el diálogo pueda deducir cuáles son estas claves.

La **autenticación** basada en clave pública se puede utilizar si el algoritmo permite utilizar las claves a la inversa: la clave privada para cifrar y la clave pública para descifrar. Si A envía un mensaje cifrado con su clave privada, todo el mundo podrá descifrarlo con la clave pública de A, y al mismo tiempo todo el mundo sabrá que el mensaje sólo lo puede haber generado quien conozca la clave privada asociada (que debería ser A). Ésta es la base de las **firmas digitales**.



### **1.5.1. Ejemplos de algoritmos de clave pública: Diffie-Hellman**

Es un mecanismo que permite que dos partes se pongan de acuerdo de forma segura sobre una clave secreta utilizando un canal inseguro. El algoritmo se basa en la dificultad de calcular logaritmos discretos y se usa generalmente como medio para acordar claves simétricas que serán empleadas para el cifrado de una sesión.

### **1.5.2. Ejemplos de algoritmos de clave pública: RSA**

Es el algoritmo más utilizado en la historia de la criptografía de clave pública. Su nombre procede de las iniciales de quienes lo diseñaron en 1977: Ronald Rivest, Adi Shamir y Leonard Adleman. La clave pública está formada por un número  $n$ , calculado como producto de dos factores primos muy grandes ( $n = p * q$ ) y un exponente  $e$ . La clave privada es otro exponente  $d$  calculado a partir de  $p$ ,  $q$  y  $e$ , de tal forma que el cifrado y el descifrado se puede realizar de la siguiente forma:

$$\text{Cifrado: } \mathbf{C = M^e \text{ mod } n}$$

$$\text{Descifrado: } \mathbf{M = C^d \text{ mod } n}$$

Como se puede ver, la clave pública y la privada son intercambiables: si se usa cualquiera de ellas para cifrar, se deberá utilizar la otra para descifrar. La fortaleza del algoritmo RSA se basa, por un lado, en la dificultad de obtener  $M$  a partir de  $C$  sin conocer  $d$  (problema del logaritmo discreto), y por otro lado, en la dificultad de obtener  $p$  y  $q$  (y, por tanto,  $d$ ) a partir de  $n$  (problema de la factorización de números grandes, que es otro de los problemas considerados difíciles).

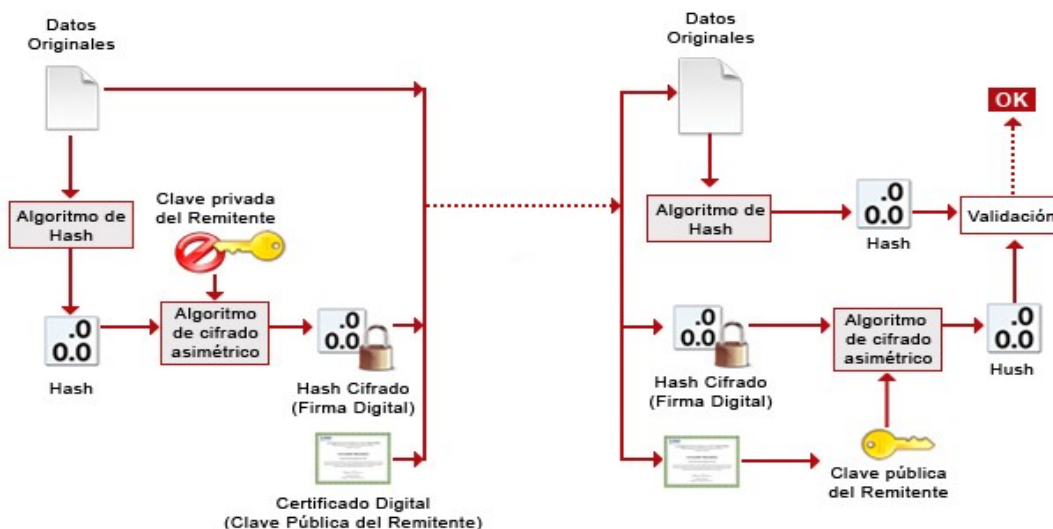
## **2. FIRMA DIGITAL**

Una firma digital es, básicamente, un mensaje cifrado con la clave privada del firmante. Pero, por cuestiones de eficiencia, lo que se cifra no es

directamente el mensaje a firmar, sino solamente su resumen calculado con una función *resumen* segura.

La firma digital está basada en algoritmos criptográficos asimétricos, en los que son necesarias un par de claves para el intercambio de la información: una clave pública y una clave privada. La clave privada está bajo custodia del emisor y solamente es conocida por él. La clave pública es distribuida entre todos los posibles destinatarios de los mensajes o documentos firmados. El proceso para realizar una firma digital se resume a continuación:

- El emisor obtiene un resumen del mensaje a través de una función resumen (*Hash*). La propiedad más importante de ese resumen o *hash* es que dos documentos diferentes siempre deben producir resúmenes diferentes.
- El resumen obtenido se cifra con la clave privada del firmante y se obtiene la firma digital del documento.
- El receptor del mensaje firmado utiliza la clave pública para descifrar la firma, obtiene el resumen del documento recibido y comprueba que es igual que el resumen que le ha llegado cifrado en la firma digital. De esta forma se garantiza que el contenido del mensaje no ha sido manipulado.



**Figura 1: Proceso de creación de una firma digital (fuente: INTECO )**

La firma digital, por si misma, no aporta confidencialidad al mensaje pero es habitual que los mensajes firmados electrónicamente se suelen enviar cifrados con la misma clave privada utilizada para mayor seguridad. La firma digital aporta:

- Identificación del firmante: la firma identifica al firmante de forma única igual que su firma manuscrita.
- Integridad del contenido firmado: es posible verificar que los documentos firmados no hayan sido alterados por terceras partes.
- No repudio del firmante: un documento firmado electrónicamente no puede repudiarse por parte de su firmante.

### 3. INFRAESTRUCTURA DE CLAVE PÚBLICA (PKI)

Como se ha visto hasta ahora, la criptografía de clave pública permite resolver el problema del intercambio de claves, utilizando las claves

públicas de los participantes. Pero se plantea otro problema: si alguien afirma ser A y su clave pública es  $k_{pub}$ ,

*¿cómo podemos saber que realmente  $k_{pub}$  es la clave pública de A?*

Porque es perfectamente posible que un atacante Z genere su par de claves ( $k'_{pr}$ ,  $k'_{pub}$ ) y afirme “yo soy A, y mi clave pública es  $k'_{pub}$ ”.

Una posible solución a este problema es que exista una entidad de confianza que nos asegure que, efectivamente, las claves públicas pertenecen a sus supuestos propietarios. Esta entidad puede firmar un documento que afirme “la clave pública de A es  $k_{pub}$ ”, y publicarlo para que todos los usuarios lo sepan. Este tipo de documento se llama **certificado de clave pública** o **certificado digital**, y es la base de lo que se conoce como **infraestructura de clave pública o PKI**.

Una **PKI** está formada entre otros por los siguientes elementos:

- **Certificados digitales:** Son documentos firmados electrónicamente por las autoridades de certificación que certifican que una clave pública pertenece a un determinado usuario.
- **Autoridades de certificación (AC):** Son entidades de confianza que se encargan de emitir y revocar los certificados digitales.
- **Autoridades de registro (RA):** Son entidades que registran las peticiones que hagan los usuarios para obtener un certificado, comprueban la veracidad y corrección de los datos que aportan los usuarios en dichas peticiones y las envían a una AC para que sean procesadas.
- **Autoridades de validación (VA):** suministran información sobre la vigencia de los certificados electrónicos que, a su vez, hayan sido registrados por una RA y certificados por la AC.

- **Autoridades de sellado de tiempos (TSA):** proporcionan certeza sobre la preexistencia de determinados documentos electrónicos en un momento dado, cuya indicación temporal junto con el hash del documento se firma por la Autoridad de sellado de tiempo.
- **Directorios de certificados:** proporcionan almacenamiento y distribución de certificados y listas de revocación (CRLs).
- **Hardware criptográfico (HSM):** dispositivos criptográficos basados en hardware que generan, almacenan y protegen claves criptográficas y suelen aportar aceleración hardware para operaciones criptográficas
- **Tarjetas criptográficas (TI):** son tarjetas que incluyen un chip con un microprocesador con módulos hardware específicos para realizar operaciones criptográficas.

### 3.1. Certificados Digitales

Un certificado digital es un documento emitido y firmado electrónicamente por una autoridad certificadora en el que certifica la asociación entre una clave pública y un participante.

El certificado garantiza que la clave pública pertenece al participante identificado y que el participante posee la correspondiente clave privada.

Los certificados digitales sólo son útiles si existe una *Autoridad de Certificación (CA)*, de confianza para las dos partes, que los valide

Los certificados digitales proporcionan un **mecanismo criptográfico** para implementar la **autenticación**. También proporcionan un mecanismo seguro y escalable para **distribuir claves públicas** en comunidades con gran número de participantes.

El formato de los certificados X.509 es una recomendación del ITU<sup>7</sup> que se publicó por primera vez en 1988. La revisión actual del estándar fue publicada en 1996 y se conoce con el nombre de X.509 v3. Los elementos que componen un certificado X.509 v3 son:

- **Versión.** Es el número de versión del certificado codificado. Los valores aceptables son 1, 2 y 3.
- **Número de serie del certificado.** Es un entero asignado por la autoridad certificadora. Cada certificado emitido por una CA debe tener un número de serie único.
- **Identificador del algoritmo de firmado.** Especifica el algoritmo empleado para firmar el certificado (ej: *sha1withRSAEncryption*).
- **Nombre del emisor.** identifica la CA que ha firmado y emitido el certificado.
- **Periodo de validez.** Es el periodo de tiempo durante el cual el certificado es válido y la CA está obligada a mantener información sobre el estado del mismo.
- **Nombre del sujeto.** Identifica al sujeto cuya clave pública está certificada en el campo siguiente. El nombre debe ser único para cada entidad certificada por una CA dada, aunque puede emitir más de un certificado con el mismo nombre si es para la misma entidad.
- **Información de clave pública del sujeto.** Almacena la clave pública, sus parámetros y el identificador del algoritmo con el que se emplea la clave.
- **Identificador único del emisor.** Este es un campo opcional que permite reutilizar nombres de emisor.
- **Identificador único del sujeto.** Este es un campo opcional que permite reutilizar nombres de sujeto.
- **Extensiones:** Las extensiones del X.509 v3 proporcionan una manera de asociar información adicional a sujetos, claves públicas, etc.

---

<sup>7</sup>Siglas en inglés de *International Telecommunication Union*.

- **Firma de la AC:** En este campo se almacena la firma digital del certificado por parte de la AC.

Los certificados digitales se diferencian **según la finalidad** para la que son solicitados. Así podemos tener certificados para **personas físicas**, certificados de **servidor**, certificados para la **firma de código**, certificados de **entidad**, etc.

### 3.1.1. Autoridad de Certificación

Una Autoridad de Certificación, es una entidad de confianza, encargada de emitir y revocar los certificados digitales que garantizan de forma unívoca y segura la identidad asociada a una clave pública.

La Autoridad de Certificación, por sí misma o por mediación de una [Autoridad de Registro](#), verifica la identidad del solicitante de un certificado antes de su expedición o, en caso de certificados expedidos con la condición de revocados, elimina la revocación de los certificados al comprobar dicha identidad.

Los certificados son documentos que recogen ciertos datos de su titular y su [clave pública](#) y están [firmados electrónicamente](#) por la Autoridad de Certificación utilizando su clave privada.

La Autoridad de Certificación es un tipo particular de [Prestador de Servicios de Certificación](#) que legitima ante los terceros que confían en sus certificados la relación entre la identidad de un usuario y su clave pública. La confianza de los usuarios en la CA es importante para el funcionamiento del servicio y justifica la filosofía de su empleo, pero no existe un procedimiento normalizado para demostrar que una CA merece dicha confianza.

La autoridad de certificación se encarga de renovar los certificados, proporcionar servicios de backup y archivo de claves de cifrado. También

crea la infraestructura de seguridad para la confianza de los participantes, establece políticas de operación segura y genera información de auditoría.

El mecanismo habitual **de solicitud de un certificado** de servidor web a una CA consiste en que la entidad solicitante, utilizando ciertas funciones del [software](#) de [servidor web](#), completa ciertos datos identificativos (entre los que se incluye el localizador [URL](#) del servidor) y genera una pareja de claves pública/privada. Con esa información el [software](#) de servidor compone un [fichero](#) que contiene una petición CSR<sup>8</sup> en formato *PKCS#10* que contiene la clave pública y que se hace llegar a la CA elegida. Esta, tras verificar por sí o mediante los servicios de una [Autoridad de Registro](#) la información de identificación aportada y la realización del pago, envía el certificado firmado al solicitante, que lo instala en el [servidor web](#) con la misma herramienta con la que generó la petición CSR.

Las CA disponen de sus propios [certificados públicos](#), cuyas claves privadas asociadas son empleadas por las CA para firmar los certificados que emiten. Un certificado de CA estará firmado por otra CA de rango superior estableciéndose así una jerarquía de certificación.

Existen **certificados de CA raíz** que están auto-firmados por la propia CA que los emite y que constituyen el elemento inicial de la jerarquía de certificación.

Una **jerarquía de certificación** consiste en una estructura jerárquica de CAs en la que se parte de una CA auto-firmada, y en cada nivel, existe una o más CAs que pueden firmar certificados de entidad final ([servidor web](#), persona, [aplicación](#) de [software](#)) o bien certificados de otras CA subordinadas plenamente identificadas y cuya [Política de Certificación](#) sea compatible con las CAs de rango superior.

Una de las formas por las que se **establece la confianza en una CA** por parte de un usuario, consiste en la "instalación" en el ordenador del usuario

---

<sup>8</sup> *Certificate Signing Request*



(tercero que confía), del certificado autofirmado de la CA raíz de la jerarquía en la que se desea confiar.

Cuando el modelo de CA incluye una **jerarquía**, es preciso **establecer explícitamente la confianza en los certificados de todas las cadenas de certificación** en las que se confíe. Para ello, se puede localizar sus certificados mediante distintos medios de publicación en internet, pero también es posible que un certificado contenga toda la cadena de certificación necesaria para ser instalado con confianza.

Un **certificado revocado** es un certificado que no es válido aunque se emplee dentro de su período de vigencia. Un certificado revocado tiene la condición de suspendido si su vigencia puede restablecerse en determinadas condiciones.

Es necesario establecer un mecanismo que permita revocar un certificado antes de que este caduque para los casos de sustracción, errores, cambios de derechos, ruptura de la CA, etc.

Para comprobar si un certificado está revocado generalmente se utilizan las **CRL (Certificate Revocation List)**. De este modo, cuando se quiere verificar la firma de un documento, el usuario no sólo ha de verificar el certificado y su validez, sino que también ha de comprobar que el certificado no ha sido revocado consultando para ello la versión más reciente de la CRL .

Con las CRL se opera siguiendo dos modelos:

- **Modelo pull:** el cliente que tiene que hacer la verificación obtiene la CRL de la CA cuando lo necesita.
- **Modelo push:** una vez que la CA actualiza la CRL, la información es enviada a los clientes que necesitan verificar certificados.

Otro método alternativo de comprobación es el *protocolo de estado de certificado en línea* **OCSP**<sup>9</sup>. Este método permite a los clientes desprenderse de la gestión del estado de los certificados y obtener una confirmación online del estado. Para ello la CA debe poner a disposición de todos los usuarios potenciales un servicio seguro online de alta disponibilidad. Este protocolo está definido por el IETF en el RFC 2560.

Los mensajes OCSP se codifican en [ASN.1](#) y habitualmente se transmiten sobre el protocolo [HTTP](#). La naturaleza de las peticiones y respuestas de OCSP hace que a los servidores OCSP se les conozca como "**OCSP responders**". Las CAs delegan la responsabilidad de proporcionar información de revocaciones en los *responders* creando así una arquitectura distribuida. Los **clientes envían una petición de estado** a un *responder* y suspende su aceptación hasta recibir la respuesta. Este modo de funcionamiento evita el uso de CRLs, reduciendo así el ancho de banda consumido, el uso de CPU y se evitan los problemas asociados a la gestión de información sensible que contienen las CRLs.

### 3.1.2. Autoridad de Registro

La Autoridad de Registro **gestiona el registro de usuarios y sus peticiones de certificación/revocación**, así como los certificados respuesta a dichas peticiones. Indica a la CA si debe emitir un certificado. La Autoridad de Registro es la que autoriza la asociación entre una clave pública y el titular de un certificado. Durante el ciclo de vida de un certificado, la Autoridad de Registro, es la que se encarga de las siguientes operaciones:

- Revocación.
- Expiración.

---

<sup>9</sup> Siglas en inglés de *Online Certificate Status Protocol*.

- Renovación (extensión del período de validez del certificado, respetando el plan de claves).
- Reemisión del par de claves del usuario.
- Actualización de datos del certificado.

### 3.1.3. Autoridad de Validación

La Autoridad de Validación **suministra información de forma online acerca del estado de un certificado**. La Autoridad de Validación suele proporcionar dos servicios de validación: el tradicional, permitiendo la descarga de **CRLs** para que el usuario las interprete él mismo, o a través del protocolo **OCSP** (*Online Certification Status Protocol*).

Los usuarios y las aplicaciones que deseen obtener el estado de un certificado, sólo tienen que realizar una petición *OCSP* contra la Autoridad de Validación para obtener dicho estado. La CA actualiza la información de la Autoridad de Validación cada vez que se modifica el estado de un certificado, con lo que, usando *OCSP*, se dispone de información en tiempo real.

### 3.1.4. Autoridad de Sellado de Tiempos

La Autoridad de Sellado de Tiempos (*TSA*) permite **firmar documentos con sellos de tiempo**, de manera que permite obtener una prueba de que un determinado dato existía en una fecha concreta. El sello de tiempo es uno de los servicios más importantes de la firma electrónica. Con el sello de tiempo se puede demostrar que una serie de datos han existido y no han sido alterados desde un instante específico en el tiempo. Este protocolo se describe en el RFC 3161 y está en el registro de estándares de Internet. Una autoridad de sellado de tiempo actúa como tercera parte de confianza testificando la existencia de dichos datos electrónicos en una fecha y hora

concretas. Los pasos que se siguen para generar un sello de tiempo son los siguientes:

- Un usuario quiere obtener un sello de tiempo para un documento electrónico que él posee.
- Un resumen digital (técnicamente un *hash*) se genera para el documento en el ordenador del usuario.
- Este resumen forma la solicitud que se envía a la autoridad de sellado de tiempo (*TSA*).
- La *TSA* genera un sello de tiempo con esta huella, la fecha y hora obtenida de una fuente fiable y la firma electrónica de la *TSA*.
- El sello de tiempo se envía de vuelta al usuario.
- La *TSA* mantiene un registro de los sellos emitidos para su futura verificación.

Las aplicaciones del sellado de tiempo son innumerables, ya que los certificados digitales se emiten con un período de validez determinado y es fundamental por ejemplo, poder verificar que una firma de un documento realizada hace *X* años atrás efectivamente se hizo con un certificado que no estaba revocado en ese instante. Ejemplos de uso: factura electrónica, voto electrónico, protección de la propiedad intelectual, etc.

### **3.1.5. Directorio de Certificados**

Los directorios proporcionan almacenamiento y distribución de certificados y listas de revocación (*CRLs*). Cuando una Autoridad de Certificación emite un certificado o *CRL*, lo envía al Directorio y además, guarda el certificado o *CRL* en su base de datos local. Generalmente se utiliza *LDAP* (*Light-weight Directory Access Protocol*) para acceder a los directorios. El usuario puede obtener certificados de otros usuarios y comprobar el estado de los mismos.

### 3.2. Hardware Criptográfico

Los **Módulos de Seguridad Hardware (HSM)** son dispositivos especializados en realizar labores criptográficas. Proporcionan almacenamiento seguro de claves y/o realización de funciones criptográficas básicas como cifrado, firma, generación de claves, etc. Para ello usan interfaces estándar como *PKCS#11* y *CryptoAPI*. Este tipo de dispositivos aumentan significativamente la seguridad en comparación con los certificados basados en disco por lo siguiente:

- La clave privada y las firmas digitales se generan dentro del HSM.
- La clave privada se almacena cifrada dentro del HSM.

Si se compara el hardware criptográfico con las tecnologías de cifrado basado en software se puede decir que el hardware criptográfico es mucho más rápido a la hora de realizar el proceso. Dependiendo del tipo de hardware *HSM*, los ratios de trabajo oscilan de las 600 a 4000 operaciones de firma RSA/segundo. Además proporcionan seguridad física al no poder modificar los algoritmos de cifrado y limitando el acceso al almacenamiento seguro de claves. Esto permite que estas soluciones puedan ser certificadas por un tercero en jerarquías de certificación.

### 3.3. Tarjetas y chip criptográficos

Una tarjeta inteligente (**smart card**), o tarjeta con [circuito integrado \(TCI\)](#), es cualquier tarjeta de tipo bolsillo con circuitos integrados que permiten la ejecución de cierta lógica programada. Aunque existe un diverso rango de aplicaciones, hay varias categorías de *TCI*: **Las tarjetas de memoria** contienen sólo componentes de memoria no volátil y posiblemente alguna lógica de seguridad. **Las tarjetas microprocesadoras** contienen memoria y tienen capacidad de procesamiento limitada. Las **tarjetas con chip criptográfico** son tarjetas microprocesadas avanzadas en las que

hay módulos hardware para la ejecución de algoritmos usados en cifrado y firmas digitales.

Una **tarjeta inteligente con un chip criptográfico** se puede definir como una llave muy segura, no duplicable e inviolable, que contiene las claves y certificados necesarios para la firma electrónica grabados en la tarjeta y que además está protegida por un PIN secreto y/o biometría. El **chip criptográfico** contiene un microprocesador que realiza las operaciones criptográficas con la clave privada, con la característica adicional de que no es posible el acceso a la clave desde el exterior. Las características principales son las siguientes:

- Doble seguridad: posesión de la tarjeta y PIN de acceso (o mecanismos biométricos).
- Puede ser multipropósito: Tarjeta de identificación gráfica, tarjeta de control de acceso/horario mediante banda magnética o chip de radiofrecuencia, tarjeta monedero, tarjeta generadora de contraseñas de un solo uso (*OTP*).
- Se precisa de un middleware (*CSP*) específico para utilizar la tarjeta, así como de un lector (*USB*, integrado en teclado o *PCMCIA*)
- El número de certificados que se pueden cargar depende del perfil de certificado, de la capacidad del chip y del espacio que se reserve para los certificados.

### **3.4. 3.3. Marco legal y estándares**

#### **Legislación Española**

- **Ley 59/2003**, de 19 de diciembre, de firma electrónica (BOE nº 304, 20/12/2003)

#### **Directiva Europea**

- **Directiva 1999/93/CE del parlamento europeo** y del consejo de 13 de diciembre de 1999 por la que se establece un marco comunitario para la firma electrónica

### **Estándares europeos**

- ETSI TS 102 042: Policy requirements for certification authorities issuing public key certificates
- ETSI TS 102 023: Policy requirements for time-stamping authorities
- ETSI TS 101 862: Qualified Certificate profile
- ETSI TS 101 456: Policy requirements for certification authorities issuing qualified certificates
- CWA 14167-2 Security Req. for Trustworthy Systems Managing Certificates for Electronic Signatures
- CWA 14172 EESSI Conformity Assessment Guidance (Guía para aplicar los estándares de firma electrónica de acuerdo con la iniciativa de estandarización europea)

### **Internet Engineering Task Force (IETF) - Request For Comment**

- RFC 3280: Certificate and Certificate Revocation List (CRL) Profile
- RFC 3739: Qualified Certificates Profile
- RFC 3647: Certificate Policy and Certification Practices Framework (Obsoletes RFC2527)

**PKCS (Public Key Cryptography Standards):** Familia de estándares para los sistemas de criptografía de clave pública definidos por los Laboratorios RSA:

- PKCS#1,#2,#4: RSA Cryptography Standard
- PKCS#3: Diffie-Hellman Key Agreement Standard
- PKCS#5: Password-Based Encryption Standard
- PKCS#6: Extended-Certificate Syntax Standard
- PKCS#7: Cryptographic Message Syntax Standard
- PKCS#8: Private Key Information Syntax Standard
- PKCS#9: Selected Attribute Types



- PKCS#10: Certification Request Standard
- PKCS#11: Cryptographic Token Interface
- PKCS#12: Personal Information Exchange Syntax Standard
- PKCS#13: Elliptic Curve Cryptography Standard



#### **4. REFERENCIAS**

- Universidad de Vigo. Asignatura Seguridad en sistemas de información.  
(<http://ccia.ei.uvigo.es/docencia/SSI/>)
- Centro Criptológico Nacional. (<https://www.ccn.es/>)
- Instituto Nacional de Tecnologías de la Comunicación - DNI Electrónico.  
(<http://cert.inteco.es>)
- Universidad Politécnica de Madrid - Departamento de Matemática Aplicada de la Facultad de Informática.  
(<http://www.dma.fi.upm.es/java/matematicadiscreta/aritmeticamodular/>)
- Universidad Pontificia Comillas (Madrid) - Asignatura de Seguridad Informática. (<http://www.iit.upcomillas.es/seguridad>)

**Autor: Juan Otero Pombo**  
**Ingeniero en Informática en el Concello de Ourense**  
**Colegiado del CPEIG**

(Todos los enlaces fueron verificados en noviembre de 2011)

## **39. VIRUS Y OTRO SOFTWARE MALIGNO. TIPOS. MEDIOS PREVENTIVOS Y REACTIVOS. SISTEMAS ANTIVIRUS Y DE PROTECCIÓN.**

## **TEMA 39: VIRUS Y OTRO SOFTWARE MALIGNO. TIPOS. MEDIOS PREVENTIVOS Y REACTIVOS. SISTEMAS ANTIVIRUS Y DE PROTECCION.**

### **INDICE**

<a href="#"><u>1.1. INTRODUCCIÓN.....</u></a>	<a href="#"><u>3</u></a>
<a href="#"><u>1.2. MALWARE.....</u></a>	<a href="#"><u>4</u></a>
<a href="#"><u>1.3. VIRUS.....</u></a>	<a href="#"><u>7</u></a>
<a href="#"><u>1.3.1. Funcionamiento de los virus.....</u></a>	<a href="#"><u>7</u></a>
<a href="#"><u>1.3.2. Caracterización de los virus.....</u></a>	<a href="#"><u>9</u></a>
<a href="#"><u>1.3.3. Ciclo de vida de los virus.....</u></a>	<a href="#"><u>10</u></a>
<a href="#"><u>1.3.4. Tipos de virus.....</u></a>	<a href="#"><u>13</u></a>
<a href="#"><u>1.4. TROYANOS.....</u></a>	<a href="#"><u>17</u></a>
<a href="#"><u>1.4.1. Tipos de troyanos.....</u></a>	<a href="#"><u>18</u></a>
<a href="#"><u>1.4.2. Modos de infección.....</u></a>	<a href="#"><u>20</u></a>
<a href="#"><u>1.5. GUSANOS.....</u></a>	<a href="#"><u>22</u></a>
<a href="#"><u>1.5.1. Tipos de gusanos.....</u></a>	<a href="#"><u>22</u></a>
<a href="#"><u>2. MEDIOS PREVENTIVOS Y REACTIVOS.....</u></a>	<a href="#"><u>24</u></a>
<a href="#"><u>3. SISTEMAS ANTIVIRUS Y DE PROTECCIÓN.....</u></a>	<a href="#"><u>25</u></a>
<a href="#"><u>3.1. ANTIVIRUS.....</u></a>	<a href="#"><u>25</u></a>
<a href="#"><u>3.2. OTRAS MEDIDAS DE PROTECCIÓN.....</u></a>	<a href="#"><u>29</u></a>
<a href="#"><u>3.2.1. Cortafuegos (firewalls).....</u></a>	<a href="#"><u>29</u></a>

<a href="#"><u>3.2.2. NIDS (Sistemas de detección de intrusos de red).....</u></a>	<a href="#"><u>29</u></a>
<a href="#"><u>3.2.3. HIDS (Sistemas de detección de intrusos de host).....</u></a>	<a href="#"><u>29</u></a>
<a href="#"><u>3.2.4. Sandboxes.....</u></a>	<a href="#"><u>30</u></a>
<a href="#"><u>3.2.5. Honeypot.....</u></a>	<a href="#"><u>30</u></a>
<a href="#"><u>4. REFERENCIAS.....</u></a>	<a href="#"><u>31</u></a>

## VIRUS Y OTRO SOFTWARE MALIGNO. TIPOS.

### **1.1. Introducción**

Los sistemas informáticos se encuentran permanentemente expuestos a la amenaza de los virus informáticos cuyo nombre proviene de la analogía de su comportamiento con los virus biológicos.

Desde que en la década de 1970 apareciesen los primeros virus, han existido siempre amenazas de este tipo que, en algunos casos, han llegado a convertirse en verdaderas infectando millones de ordenadores y generando importantes pérdidas económicas. Ejemplos de estas grandes epidemias fueron las generadas por el Gusano de Morris, Melissa o ILoveYou.

En realidad los virus son un subtipo del software maligno en general, que suele denominarse como *malware*. Los troyanos y gusanos son también tipos de *malware*, que junto a los virus, constituyen los tres tipos principales de software maligno que atacan a los sistemas informáticos, existiendo muchas otras variantes o subtipos de estos: *adware*, *spyware*, *scareware*, etc.

En la actualidad, el tipo de malware más extendido son los troyanos, destinados a instalarse en sistemas de usuarios para robar información confidencial: nombres de usuarios, números de tarjetas de crédito, etc. y enviarlas a los atacantes. Los atacantes se sirven de la red Internet para recibir y usar la información a miles de kilómetros, en otros países donde las leyes no recogen delitos de este tipo se encuentran a salvo. Esto les permite obtener beneficios económicos, ya que la creación de programas maliciosos es un lucrativo negocio.

## 1.2. Malware

El malware<sup>1</sup> es un tipo de software que tiene como objetivo infiltrarse o dañar una computadora sin el consentimiento de su propietario. El término malware es muy utilizado por profesionales de la informática para referirse a una variedad de software hostil, intrusivo o molesto.

Colateralmente, el malware suele perseguir un lucro de modo directo o indirecto por parte del atacante. El nivel de daño que recibe el usuario puede ir desde pequeñas alarmas inofensivas a efectos desastrosos como la pérdida masiva de datos. Internet resulta ser un medio muy apropiado para distribuir el malware de forma que se maximice el número de usuarios afectados.

Podemos establecer una primera clasificación de malware en tres tipos principales perfectamente diferenciados: virus, gusanos y troyanos. A partir de ahí existen multitud de elementos peligrosos que podrían ser catalogados en uno u otro tipo (o en varios a la vez). Así, es común oír hablar de:

- **Adware:** es un software que despliega publicidad de distintos productos o servicios. Estas aplicaciones incluyen código adicional que muestra la publicidad en ventanas emergentes, o a través de una barra que aparece en la pantalla simulando ofrecer distintos servicios útiles para el usuario. Generalmente, agregan ícono gráficos en las barras de herramientas de los navegadores de Internet o en los clientes de correo, la cuales tienen palabras claves predefinidas para que el usuario llegue a sitios con publicidad, sea lo que sea que esté buscando.

---

<sup>1</sup> De las palabras inglesas *malicious software*

- *Spyware*: o software espía es una aplicación que recopila información sobre una persona u organización sin su conocimiento ni consentimiento. El objetivo más común es distribuirlo a empresas publicitarias u otras organizaciones interesadas. Normalmente, este software envía información a sus servidores, en función a los hábitos de navegación del usuario. También, recogen datos acerca de las webs que se navegan y la información que se solicita en esos sitios, así como direcciones IP y URLs que se visitan. Esta información es explotada para propósitos de mercadotecnia, y muchas veces es el origen de otra plaga como el SPAM, ya que pueden encarar publicidad personalizada hacia el usuario afectado. Con esta información, además es posible crear perfiles estadísticos de los hábitos de los internautas. Ambos tipos de software generalmente suelen "disfrazarse" de aplicaciones útiles y que cumplen una función al usuario, además de auto ofrecer su descarga en muchos sitios reconocidos.

Cabe destacar que el atacante no tiene por qué ser un delincuente. Por ejemplo el FBI desarrolló su propia aplicación *spyware*, llamada MagicLantern, usada en investigaciones criminales para obtener información de los sospechosos.

- *Crimeware*: Tipo de programa de computadora diseñado específicamente para cometer crímenes del tipo financiero o similares, intentando pasar desapercibido por la víctima. Por extensión, también hace referencia a aplicaciones web con iguales objetivos.

Un crimeware puede robar datos confidenciales, contraseñas, información bancaria, etc. y también puede servir para robar la identidad o espiar a una persona o empresa.



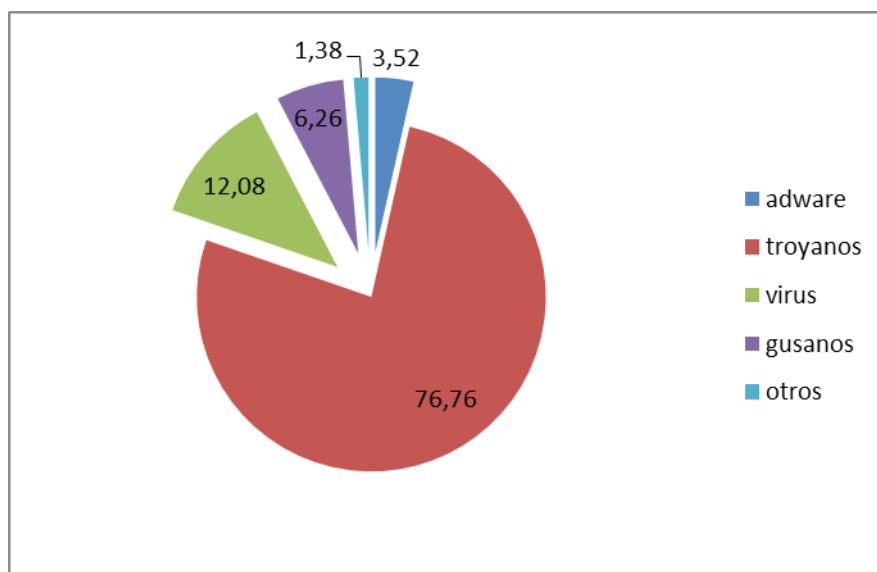
- *Scareware*: término acuñado recientemente, es lo que se conoce como "software de seguridad falso". Normalmente este software toma ventaja de la intención de los usuarios en mantener sus equipos protegidos, y específicamente utiliza técnicas de ingeniería social que se basan principalmente en crear "paranoia" en los mismos, ofreciéndoles una solución definitiva a sus problemas de seguridad. Una vez instalada, este tipo de programas se dedica a robar información como lo haría cualquier troyano bancario o de usurpación de identidad en el equipo de la víctima.
- Un nuevo tipo de *malware* comienza a aparecer con fuerza asociado al crecimiento de las redes sociales. Se trata de aplicaciones maliciosas que se mueven dentro del contexto de redes sociales destinadas a robar datos personales, suplantar identidades, etc. Un ejemplo es el gusano Koobface, que ataca a varias redes sociales muy conocidas, incluyendo Facebook, Twitter, y Myspace.

El siguiente gráfico muestra la distribución de tipos de *malware* detectados en fechas recientes<sup>2</sup>:

---

<sup>2</sup> La mayoría de grandes empresas dedicadas al desarrollo de antivirus ofrecen frecuentemente informes sobre los tipos de malware detectados, los virus más activos, alertas especiales, etc.





**Estadísticas de *Malware* por tipos del tercer trimestre de 2011**  
(fuente: Panda Security)

### 1.3. Virus

Los virus informáticos son porciones de código malicioso que se pueden introducir en los ordenadores y sistemas informáticos de formas muy diversas, produciendo efectos molestos, nocivos e incluso destructivos e irreparables. Además, el efecto inicial que lo caracteriza es una infección del equipo, seguida de una propagación del virus a medida que los archivos donde reside el virus son ejecutados. Es decir, el código del virus se ejecuta solamente cuando se ejecuta el programa o se abre el archivo infectado. Esto es lo que diferencia a los virus de los gusanos (que veremos más adelante): si no se accede al programa o archivo entonces el virus no se ejecutará y por lo tanto no se propagará.

#### 1.3.1. Funcionamiento de los virus

La propia denominación de virus informáticos proviene de la analogía del funcionamiento de estos con los virus biológicos. Un virus biológico no

puede sobrevivir de forma independiente. En realidad no es más que una cadena de ADN protegida por una cobertura externa que se reproduce insertándose en una célula huésped y usando el mecanismo reproductor de esa célula para reproducirse él mismo.

Un virus informático tampoco es independiente ni tiene la capacidad de sobrevivir y reproducirse de modo aislado. Se trata de una secuencia de código maligno que para poder propagarse necesita infectar a un huésped, que en este caso será un programa informático o un documento.

Muchos virus se esconden dentro de archivos de programas aparentemente limpios. Estos virus se conocen como virus de archivo cuyo código se ejecuta cargándose en memoria principal de la máquina junto con el programa que lo contiene. Desde allí, el código del virus puede buscar otros programas en el sistema que puedan ser infectados. Si encuentra uno, el virus inserta su código en ese programa, que una vez infectado también puede ser usado para infectar a más programas iniciando así una cadena infecciosa similar a una epidemia de un virus biológico.

La mayoría de los virus no solo se replican a sí mismos, sino que también realizan otras operaciones que habitualmente son dañinas para sus huéspedes. Así, por ejemplo, un virus puede eliminar ciertos archivos vitales de un sistema, sobrescribir el sector de arranque de un disco duro dejando el disco inhabilitado, mostrar mensajes en la pantalla, emitir ruidos y una gran cantidad de otras acciones de mayor o menor poder destructivo.

En general, los virus están diseñados para ejecutar su código dañino en el momento de ser ejecutados. Sin embargo, hay algunos otros que no atacan en ese momento, sino que están diseñados para esperar a una fecha concreta o a un evento particular. Estos virus permanecen en estado latente en el sistema hasta que actúan.



### 1.3.2. Caracterización de los virus

Existen muchos tipos de virus, pero podemos indicar algunas características comunes a todos ellos:

- **Latencia:** los virus tienen capacidad para permanecer inactivos, en un estado latente, hasta que un evento los active. Este evento puede ser la ejecución de un programa, la lectura del sector de arranque, etc. Una vez activado, el virus puede realizar las tareas para las que estaba programado y replicarse.
- **Tipo de residencia:** los virus más avanzados son capaces de camuflarse para evitar ser detectados y erradicados. Los virus más básicos necesitan permanecer continuamente en la memoria principal, mientras que estos otros virus, más avanzados, son capaces de residir en otros lugares como la memoria secundaria. Pueden llegar incluso a modificar la tabla de asignación de archivos para camuflar su presencia.
- **Forma de infección:** se trata de una característica común de todos los virus. El medio por el que un virus infecta varía enormemente de unos a otros: ejecución de programas, de macros, conexiones mediante determinados protocolos, el simple arranque de una máquina, etc..
- **Composición:** cualquier virus informático cuenta con tres partes destinadas a cumplir con sus objetivos:
  - o **Sistema de reproducción:** encargado de infectar otros sistemas o archivos mediante el aprovechamiento (ilícito) de los recursos de la máquina huésped.

- o **Sistema de ataque:** en caso de existir se trata de una serie de rutinas destinadas a dañar el sistema huésped de un modo u otro. El daño puede ir desde pequeños inconvenientes como mensajes molestos, hasta ataques desastrosos como la pérdida masiva de datos.
- o **Sistema de defensa:** destinado a ocultar la presencia del virus mientras sea posible y a dificultar su eliminación.

### **1.3.3. Ciclo de vida de los virus**

Existen una serie de fases por las que un virus pasa a lo largo de su vida:

- **Creación:** el virus es creado por programadores que escriben su código.
- **Contagio:** El contagio inicial o los contagios posteriores se realizan cuando el programa contaminado está en la memoria para su ejecución. Las vías por las que puede producirse la infección de su sistema son disquetes, memorias flash, redes de ordenadores y cualquier otro medio de transmisión de información. Estos disquetes contaminantes suelen contener programas de fácil y libre circulación y carecen de toda garantía. Es el caso de los programas de dominio público, las copias ilegales de los programas comerciales, juegos, etc.
- **El virus activo:** Cuando se dice que un virus se activa significa que el virus toma el control del sistema, y a la vez que deja funcionar normalmente a los programas que se ejecutan, realiza actividades no deseadas que pueden causar daños a los datos o a los programas.

Lo primero que suele hacer el virus es cargarse en la memoria del ordenador y modificar determinadas variables del sistema que le permiten "hacerse un hueco" e impedir que otro programa lo utilice. A esta acción se le llama "hacerse residente". Así el virus queda a la espera de que se den ciertas condiciones, que varían de unos virus a otros, para replicarse o atacar.

La replicación, que es el mecanismo más característico y para muchos expertos definitorio de la condición de virus, consiste básicamente en la producción por el propio virus de una copia de sí mismo, que se situará en un archivo. El contagio de otros programas suele ser la actividad que más veces realiza el virus, ya que cuanto más deprisa y más discretamente se copie, más posibilidades tendrá de dañar a un mayor número de ordenadores antes de ser descubierto.

- **El ataque:** Mientras que se van copiando en otros programas, los virus comprueban si determinada condición se ha cumplido para atacar, por ejemplo que sea cinco de enero en el caso del conocido virus Barrotes. Es importante tener en cuenta que los virus son diseñados con la intención de no ser descubiertos por el usuario y generalmente, sin programas antivirus, no es descubierto hasta que la tercera fase del ciclo de funcionamiento del virus se produce el daño con la consiguiente pérdida de información.
- **Descubrimiento:** esta fase comienza cuando el virus es detectado, identificado y documentado por primera vez.
- **Asimilación:** las compañías fabricantes de antivirus modifican sus soluciones para conseguir detectar y eliminar las infecciones causadas por el virus.

- **Erradicación:** el uso exhaustivo de antivirus limita las infecciones del virus eliminando su amenaza.

Un virus puede ser catalogado en base a su índice de peligrosidad, que variará dependiendo de la fase de su ciclo de vida en la que se encuentre. El índice de peligrosidad es una medida del impacto que produce su código malicioso y la capacidad de propagación a otros sistemas. Existen varios niveles de peligrosidad que van desde aquellos virus que causan daños leves y están poco extendidos, hasta aquellos otros que causan daños catastróficos y están muy extendidos.

El nivel de daño indica el perjuicio que un virus causa al infectar un sistema informático. Un virus puede ser catalogado como dañino (mensajes o ruidos molestos, apertura de ventanas involuntariamente, etc.) o muy dañino (destrucción o modificación de archivos, formateo de discos duros, envío de información a terceros, generación de gran tráfico en servidores, degradación del rendimiento de los sistemas, apertura de agujeros de seguridad, etc.).

El grado de propagación indica lo extendido que se encuentra el virus. Evidentemente, cuanto más extendido esté un virus, mayores son las probabilidades de estar afectado por él. La propagación de un virus se determina mediante el 'ratio de infección', que mide el porcentaje de ordenadores infectados en relación al total de equipos explorados. El grado de propagación de un virus podría variar entonces desde 'poco extendido' hasta 'epidemia'.

### **1.3.4. Tipos de virus**

En función de qué es exactamente lo que un virus puede infectar, es usual clasificarlos en:

- Virus de sector de arranque maestro<sup>3</sup>: fueron el primer tipo de virus. Se esconde en el código ejecutable del sector de arranque de discos de memoria secundaria o de discos de arranque externos (diskettes, CD-ROMs, etc.). Hasta hace no mucho tiempo atrás, iniciar el ordenador desde un disco de arranque era algo bastante usual, lo que significaba que los virus se distribuían rápidamente.
- Virus de archivo: se caracterizan por unirse a archivos en los que residen y que pueden usar para propagarse entre sistemas. Suelen infectar archivos ejecutables, pero también existen virus capaces de unirse a archivos de código fuente, librerías o módulos de objetos e incluso archivos de datos. El virus *Jerusalem* (también conocido como *Viernes 13*), uno de los más conocidos, pertenece a esta categoría.
- Macrovirus o virus de macro: los virus de macro hacen uso de la capacidad que ciertas aplicaciones, como Word y Excel, tienen para ejecutar internamente ciertos códigos programados llamados macros. El virus se adjunta a dichas macros y se transmite de un documento a otro a través de ellas. El virus causante de una de las mayores epidemias de la historia, *Melissa*, corresponde a este tipo.
- Virus mixtos, bimodales o multiparte: se trata de una combinación de virus de archivo, de macro y de sector de arranque. Realizan infecciones usando varias técnicas para instalarse en cualquiera de las localizaciones posibles. Se consideran muy peligrosos por la gran capacidad de infección que tienen.

---

<sup>3</sup> Conocido habitualmente por su nombre en inglés Master Boot Record (MBR).

- Virus de BIOS: se alojan en la BIOS del ordenador y cada vez que esta se arranca, infectan a los archivos del sistema. El virus *Chernobyl* realizaba una infección de este tipo.

Dentro de los anteriores, siguiendo un criterio más amplio, otros tipos de virus identificables serían:

- Virus de compañía: estos virus no modifican los archivos infectados, sino que crean una copia del archivo original y modifican esta copia. Cuando el archivo original se ejecuta, el virus hace que se pase el control al archivo infectado. Una variante de la infección consiste en renombrar el archivo original, y sustituirlo por otro con el nombre original que contiene directamente el virus.
- Retrovirus: se trata de virus especialmente diseñados para evitar ser detectados e infectar a los programas antivirus.
- Virus de sobreescritura: el código del virus se escribe encima de un fichero ejecutable destruyéndolo. Si el tamaño del virus es menor que el del ejecutable infectado, el resultado final no aumenta de tamaño, dificultando la detección del virus.
- Virus parásitos: los archivos huésped son modificados solo en parte, de modo que no son destruidos y pueden incluso simular que siguen funcionando. En estos casos el virus se aloja en lugares de archivo, normalmente al comienzo o final, donde permite que el archivo huésped siga parcialmente activo.
- Virus mutantes: cuando infectan a un huésped, modifican su propio código para evitar que su 'huella' sea detectada por programas antivirus.



- Virus sin punto de entrada: conocidos como virus EPO (Entry Point Obscuring). Se destacan porque la instrucción que hace pasar el control al virus se inserta en un lugar indeterminado del archivo huésped. Esto hace que el virus no se manifieste hasta que se realiza una acción concreta dentro del ejecutable infectado. Esto les permite permanecer en un estado latente durante mucho tiempo.
- Virus de enlace: se caracterizan porque la infección consiste en la inserción de un enlace a algún otro lugar (por ejemplo el último cluster de un disco duro o un cluster marcado como erróneo) en donde realmente se aloja el código del virus.
- Virus OBJ, LIB y código fuente: en lugar de infectar directamente un ejecutable, infectan librerías o módulos usados por otros ejecutables. Evidentemente, su capacidad para actuar y reproducirse solo aparecerá cuando la librería sea utilizada por otro programa.
- Virus de script: se trata de virus que actúan sobre lenguajes de script habitualmente asociados a páginas web. Actúan incluyéndose en páginas web o modificando los scripts que esas páginas contienen.
- Virus en estado salvaje: son aquellos que se encuentran en circulación en estos momentos y que están en condiciones de infectar.
- Virus de zoológico: son virus que no se encuentran en libertad o que han perdido su capacidad para infectar. Por ejemplo el conocido como 'virus de la pelotita'.
- Generadores de virus: son virus que al mutar generan nuevos virus.
- Virus que crean dependencia: cuando uno de estos virus infecta una máquina, se instala en lugares vitales, de modo que no es posible

eliminarlos sin hacer que la máquina pierda esos elementos vitales y deje de funcionar.

- Bombas de tiempo: se ocultan en los sistemas hasta que llega una fecha concreta o pasa un determinado periodo de tiempo. En ese momento pasan a un estado activo y realizan sus acciones de infección y replicación.

En cuanto al modo en que el virus produce la infección del sistema, podemos enumerar los siguientes:

- Añadidura o empalme: es uno de los modos más básicos y clásicos. El código del virus se añade al final de un archivo huésped (habitualmente un ejecutable) y se modifica la estructura de arranque del mismo, haciendo que el virus se ejecute en primer lugar antes de pasar el control al fichero original. El resultado es un aumento del tamaño inicial del archivo, permitiendo así una fácil detección.
- Inserción: es un modo más avanzado de infección en el que el virus se instala en zonas de código no utilizada o en segmentos de datos para que el tamaño del archivo no varíe.
- Reordenación: se introduce el código del virus en sectores del disco duro que quedan marcados como defectuosos y se distribuyen enlaces a dichos sectores en el código de otros programas ejecutables que quedan así infectados. La ventaja de este método es que al encontrarse el código del virus fuera del archivo, este código puede ser de gran tamaño y por tanto poseer mucha funcionalidad. En cambio, este tipo de virus es muy fácil de eliminar, simplemente sobrescribiendo los sectores defectuosos.

- Polimorfismo: es probablemente el método más avanzado. Se basa en infectar un archivo ejecutable, pero realizando una compactación del código del archivo huésped o del código del propio virus para evitar así un aumento del tamaño que lo delate. En el momento de actuar, el virus descomprime en memoria el código necesario para ejecutarse.
- Sustitución: este método, muy poco sutil, consiste en sustituir directamente el código del programa huésped al completo por el código del virus. De este modo el programa original simplemente desaparece y lo único que se ejecuta es el virus.

#### **1.4. Troyanos**

Hace miles de años, ante la imposibilidad de traspasar las murallas de la ciudad de Troya, los griegos construían un gran caballo de madera en el interior del cual se ocultaban una selección de sus mejores soldados. Colocaron el caballo delante de la ciudad y los troyanos, prendados de su majestuosidad de lo que aparentaba un regalo de los dioses completamente inofensivo introdujeron el caballo en la ciudad.

El tipo de malware denominado troyano debe su nombre a que para su propagación utiliza la misma estrategia que idearon los griegos para entrar en Troya. El concepto básico de un troyano consiste en introducir código malicioso dentro de un sistema que resulte atractivo para la víctima y además parezca seguro, de modo que el conjunto parece inofensivo. Este disfraz podría ser desde un juego descargado de Internet hasta un mensaje de correo electrónico de apariencia inofensiva.

Los Troyanos son códigos maliciosos que intentan mostrarse como algo útil o apetecible para que una víctima lo ejecute. Se caracterizan porque su objetivo es introducirse en el sistema y pasar desapercibidos. Mientras se encuentran en el sistema pueden dedicarse a enviar información (en ese caso suelen conocerse como *spyware*) o a preparar el sistema para un ataque posterior mediante la instalación de *rootkits*<sup>4</sup> o la creación de puertas traseras (*backdoors*). Además, a diferencia de los virus, no tienen la capacidad para replicarse e infectar otros sistemas por sí solos.

Actualmente los troyanos son ampliamente utilizados con el objetivo de obtener datos de una víctima que lo tiene instalado en su sistema sin su conocimiento. Esto pueden hacerlo de muy diversas maneras, que van desde capturar las pulsaciones de teclado (los llamados *keyloggers*) hasta el acceso y manipulación de las carpetas de documentos de un usuario.

#### **1.4.1. Tipos de troyanos**

Los troyanos pueden tener características muy diversas por lo que resulta difícil catalogarlos. Sin embargo, atendiendo a su objetivo, podemos establecer la siguiente clasificación:

- **Troyanos de control remoto:** su objetivo es proporcionar al atacante el control de la máquina donde reside el troyano. Habitualmente el troyano intentará abrir conexiones de red clandestinas, desde las que podría escuchar las órdenes del atacante. Ejemplos de este tipo de troyanos son *Back\_orifice* o *Netbus*

---

<sup>4</sup> Conjunto de herramientas destinadas a permitir que un atacante acceda a los privilegios de administrador del sistema de forma remota.

- **Troyanos que envían datos:** su objetivo es enviar al atacante datos confidenciales tomados de la máquina atacada y de la información que cualquier usuario almacene en ella. Comúnmente persiguen la obtención de usuarios y claves de acceso, número de tarjetas de crédito, cuentas bancarias, etc. El envío de datos puede hacerse de diferentes formas: mediante el envío de un correo electrónico a través de un servidor de correo público, directamente a la página web del atacante mediante un formulario, etc. Un ejemplo de este tipo es el *badtrans.b*, que es capaz de recoger las pulsaciones del teclado y enviarlas vía correo electrónico.
- **Troyanos destructivos:** se comportan en cierto modo como un virus, ya que su objetivo es causar daños mediante la destrucción de información. Pueden llevar a cabo esta tarea inmediatamente tras la infección o actuar como una bomba lógica que se activará cuando se produzca un determinado evento.
- **Troyanos de ataque de denegación de servicio:** el objetivo de estos troyanos es convertir a las víctimas en participantes involuntarios en ataques de denegación de servicio distribuidos (DDoS). La máquina infectada se denomina comúnmente *botnet* o máquina *zombie*. El atacante puede utilizar todas las máquinas infectadas para hacer un ataque coordinado contra otro servidor o incluso contra una cuenta de correo electrónico (cada máquina infectada podría enviar mensajes con remitentes capturados). Un ejemplo de este tipo es el *WinTrinoo*, una herramienta de DDoS muy extendida por su sencillez de uso.
- **Troyanos Proxy:** se trata de troyanos que permiten convertir la máquina infectada en un Proxy a disposición del atacante. Este podrá utilizarlo como punto intermedio para otros ataques, dificultando así que el ataque pueda ser rastreado hasta la máquina original. Es

común encadenar varios saltos entre máquinas Proxy involuntarias para conseguir evitar el rastreo del ataque.

- **Troyanos FTP:** se caracterizan por infectar el sistema a través del puerto del protocolo FTP (el 21) y permitir al atacante usar dicho protocolo libremente contra la máquina infectada. FTP permite la transmisión bidireccional de archivos entre equipos remotos. Es decir, el atacante tendrá capacidad para copiar y borrar archivos a su antojo en la máquina infectada.
- **Deshabilitadores de software de seguridad:** se trata de troyanos avanzados que incluyen herramientas para evitar o incluso eliminar software de protección como antivirus o *firewalls*. Suelen acompañar a virus y gusanos e instalarse cuando se produce la infección. Como en el caso del gusano *Goner*, que incluía un troyano de este tipo.

Como decíamos, existen troyanos de difícil catalogación. La realidad es que constantemente aparecen nuevos troyanos con funciones y objetivos cada vez más extravagantes. Como por ejemplo el *SMSlock.A*, que literalmente secuestra el equipo infectado impidiendo su uso y pide un rescate económico por su recuperación.<sup>5</sup>

#### **1.4.2. Modos de infección**

Las dos formas más frecuentes por las que un troyano puede acceder e instalarse en un equipo huésped son las siguientes:

---

<sup>5</sup> Este tipo de *malware* dedicado a secuestrar recursos y pedir rescate por ellos es denominado también *ransomware*.

- **Mediante adjuntos en correos electrónicos o mensajería instantánea:** el simple hecho de abrir un archivo adjunto a un correo electrónico o enviado a través de una herramienta de mensajería instantánea (como *Win32/SdBot*, que se instalaba a través de *MSN Messenger*) puede permitir a un troyano instalarse en nuestro sistema. Incluso algunos gestores de correo, que no se encuentren configurados apropiadamente, pueden permitir que los adjuntos se ejecuten sin que el usuario lo solicite.
- **Mediante la instalación voluntaria de software:** suele tratarse de software de dudosa procedencia, *shareware*, *freeware*, versiones de prueba, etc. Al descargar de la red este software e instalarlo, el troyano se instalará en un segundo plano sin que la víctima pueda percibir nada. Contra esto, cabe destacar el uso cada vez más habitual de certificados digitales que permiten identificar tanto el servidor al que nos conectamos como el propio software que podemos descargar firmado. De este modo nos aseguraríamos que lo estamos descargando realmente de donde pretendemos y que no ha sido modificado durante la comunicación.

En todo caso, la infección por troyanos suele tener un indispensable componente de ingeniería social. Los ataques suelen usar engaños para conseguir que la víctima abra los adjuntos o llegue incluso a ejecutar su contenido. Pueden simular ser correos de amigos, peticiones solidarias, etc. También suelen aparentar ser programas útiles que se distribuyen gratuitamente para conseguir que la víctima lo descargue e instale. Incluso no es extraño encontrar troyanos detrás de programas antivirus gratuitos.

## **1.5. Gusanos**

Un gusano es un programa que una vez ejecutado se replica sin necesidad de la intervención humana y es capaz de enviar copias de sí mismo a través de redes de comunicaciones, sin la necesidad de que un usuario envíe un correo electrónico infectado ni establezca ninguna comunicación explícita. Se propagará de anfitrión en anfitrión haciendo un uso indebido de servicios desprotegidos: correo electrónico, herramientas de mensajería instantánea, etc. Aunque la propagación en sí no tiene por qué ser dañina, sucede lo mismo que con los virus: es habitual que los gusanos incluyan código malicioso destinado a dañar los sistemas infectados. De hecho, algunas de las infecciones más dañinas y conocidas han sido provocadas por gusanos: *ILoveYou*, *Kournikova*. Algunos gusanos no incluyen código malicioso, pero su ataque consiste en el reenvío de sí mismos hasta conseguir agotar los recursos de la máquina atacada.

Sin embargo, al contrario que los virus, los gusanos son programas completos. No solo en el sentido de que son capaces de enviarse copias de sí mismos a través de Internet, sino porque no necesitan de ningún programa huésped para hacerlo. No necesitan corromper otros programas e insertar su código allí. Su funcionamiento se basa en errores en sistemas operativos, aplicaciones o protocolos que son aprovechadas por los gusanos para ejecutarse.

### **1.5.1. Tipos de gusanos**

Podemos clasificar los gusanos atendiendo al medio que utilizan para su propagación:

- **Gusanos de redes de área local:** se propagan a través de los recursos compartidos de una red local llegando a bloquearla o degradando las medidas de seguridad de la misma. Un ejemplo es el



gusano *Lovgate*, que puede realizar la infección de la red local y también enviarse por correo electrónico a otras máquinas.

- **Gusanos de redes P2P<sup>6</sup>:** usan este tipo de redes y su gran popularidad para incluir en ellas archivos que al ser descargados traen consigo el gusano. El gusano *Redisto.b* utiliza este tipo de mecanismos.
- **Gusanos de correo electrónico:** es probablemente el método más habitual de propagación, y se realiza utilizando ciertos programas clientes de correo. El gusano accede a las direcciones de correo de la agenda de un usuario y se reenvía usando la propia cuenta del usuario. Algunos virus más avanzados pueden contar incluso con su propio servicio SMTP con el que enviar sus copias. *Sircam* o *Nimda* son ejemplos de gusanos de correo electrónico.
- **Gusanos de mensajería instantánea** (IRC, MSN Messenger): otra fuente habitual de entrada para gusanos son las aplicaciones de mensajería instantánea, que además permiten el envío de archivos adjuntos.
- **Gusanos que se propagan directamente por Internet:** los gusanos más avanzados no dependen de ninguna aplicación en particular para propagarse. Su estrategia de infección puede basarse en encontrar puertos abiertos en las máquinas objetivo y conseguir introducirse en la máquina sin que los usuarios se percaten de ello. Otros gusanos infectan los servidores de información, haciendo que la simple petición de una página web haga que el gusano pueda pasar al cliente, como es también el caso de gusano *Nimda* en otro de sus modos de contagio.

---

<sup>6</sup> Peer to Peer

## **2. MEDIOS PREVENTIVOS Y REACTIVOS**

Los medios preventivos contra el *malware* en general, se basa en medidas de índole técnico combinadas con una política de seguridad que promueva buenas prácticas por parte de los administradores y del personal de la organización. Entre todas estas medidas destinadas a la prevención de infecciones y la reacción en caso de que llegase a producirse, podríamos citar:

- Utilización de programas antivirus perfectamente configurados y actualizados. Es la herramienta principal en la lucha contra infecciones. Permiten detectar y en muchos casos eliminar los virus. En el siguiente apartado hablaremos más en profundidad sobre ellos.
- Los sistemas operativos son elementos fundamentales en los sistemas informáticos. Muchos de los métodos de infección se basan en debilidades o vulnerabilidades de los sistemas operativos. Por tanto, deben mantenerse siempre actualizados, especialmente con las actualizaciones específicas de seguridad.
- También todo el resto del software instalado debe mantenerse actualizado. Los administradores deben prestar atención a las noticias sobre nuevas vulnerabilidades de dicho software y proteger el sistema ante ataques que pudieran aprovecharse de las mismas.
- Controlar el software ya instalado en las máquinas y de todo aquel que se vaya a instalar, que una vez más debería ser solamente el necesario para las tareas de la organización. Por supuesto, no debe permitirse la instalación de cualquier software que no sea original. El software pirata es uno de los principales puntos de propagación del *malware*.

- Los servicios activos en el sistema se mantendrán en el mínimo número necesario. Solo aquellos realmente necesarios para los objetivos de la organización deberían permanecer activos. Este hecho debería ser controlado periódicamente para evitar la apertura de conexiones ilegales.
- Mantener copias de seguridad de los datos, de los programas y de los sistemas operativos.
- Gestionar adecuadamente las cuotas de uso de disco y memoria de cada usuario. Esto evitará que si un usuario provoca una infección, no se consuman todos los recursos de la máquina afectada, solo los asignados a dicho usuario.
- Tanto los administradores como los usuarios deben asumir buenas prácticas de prevención: No abrir nunca archivos adjuntos de dudosa procedencia, desactivar las opciones de visualización de imágenes y vista previa de gestores de correo, no aceptar descargas ni instalaciones de software no iniciadas voluntariamente, comprobar los certificados de procedencia del software, etc.

### **3. SISTEMAS ANTIVIRUS Y DE PROTECCIÓN**

#### **3.1. Antivirus**

Los antivirus constituyen la piedra angular sobre la que se basa la mayor parte de la defensa contra virus, troyanos, gusanos y *malware* en general. Su objetivo es el de detectar, bloquear, eliminar y prevenir infecciones provocadas por virus informáticos. La mayoría de las soluciones existentes hoy en día, son capaces también de detectar otros tipos de *malware*: troyanos, gusanos, *spyware*, *rootkits*, etc.

Cuando un antivirus detecta una infección, podrá actuar en dos modos distintos. Si tiene capacidad para ello, podría eliminar la infección sin dañar los recursos afectados. Si no es así, propondrá la puesta en cuarentena de los recursos afectados (habitualmente archivos), que quedarán aislados del resto del sistema y se impedirá que sean ejecutados. Esto último no elimina la infección, pero la bloquea impidiendo que el virus ejecute su código malicioso y evita que el virus pueda propagarse.

Los antivirus pueden detectar las infecciones siguiendo una de estas dos estrategias:

- **Detección de patrones:** cada virus tiene un patrón de identificación, que suele ser una secuencia de código que le identifica. Los antivirus poseen una base de datos de patrones de virus y las comparan con los archivos del sistema para ver si existe alguna infección. Sin embargo, los virus actuales tienen firmas muy pequeñas. Esto dificulta la tarea de los antivirus y pueden provocar falsos positivos, es decir, detecciones que aparentan ser virus y no lo son.

Los antivirus basados en patrones obtienen muy buenos resultados. Detectan un gran número de virus, pero para ello necesitan que sus bases de datos de patrones estén actualizadas. Todo aquel virus que no figure en la base de datos será simplemente indetectable. Además para funcionar de una forma eficiente, requieren el uso de algoritmos de búsqueda optimizados, ya que la estrategia de detección se basa en escanear el contenido de todos los recursos sospechosos.

- **Heurísticas:** usan técnicas de inteligencia artificial para lograr reconocer secuencias de acciones o comportamientos asociados a virus. Se trata de reconocer acciones que usualmente los virus realizan (borrado de ficheros, conexiones a Internet, modificar

archivos ejecutables, etc.) o que no realizan (abrir ventanas, emitir mensajes visibles, etc.).

Las técnicas heurísticas tienen una ventaja fundamental: permiten detectar virus nuevos sin necesidad de actualizaciones. Pero, por otro lado, pueden generar muchos falsos positivos sobre programas que en realidad no son virus.

Existe un variado abanico de soluciones antivirus, disponibles en versiones propietarias y también en versiones ofrecidas gratuitamente o como software libre. Entre los primeros podemos encontrar soluciones muy conocidas desarrolladas por grandes empresas: Norton, Symantec, PandaSoftware, McAfee, Kaspersky, etc. Entre los segundos podemos citar AVG, Avast!, ClamWin, etc.

A la hora de seleccionar el antivirus más apropiado para nuestros sistemas, deberíamos tener en cuenta las siguientes características:

- Frecuencia de actualización: cuanto más actualizado esté nuestro antivirus, más preparados estaremos para luchar contra las infecciones
- Protección en tiempo real: es conveniente que el agente antivirus resida en la memoria principal y realice un escaneado continuo en busca de posibles infecciones. Por supuesto, esto tiene un coste en recursos para la máquina que debe ser tenido en cuenta antes de optar por una solución de este tipo.
- Capacidad de centralización: algunos antivirus permiten ser instalados en varias máquinas, pero ser controlados y gestionados desde una única máquina. Esto facilita la tarea de los administradores.

- Programación de tareas: es muy interesante que los escaneos, que pueden consumir bastante tiempo y recursos de máquina, puedan ser programados para que se ejecuten en horas de poca actividad (durante la noche, por ejemplo).
- Protección del correo: hoy en día, uno de los puntos de entrada más comunes para los virus es a través de los correos electrónicos y sus archivos adjuntos. Es conveniente que nuestro antivirus sea capaz de analizar automáticamente esos archivos adjuntos.
- Generación de informes: es muy útil para los responsables de la seguridad que el antivirus sea capaz de generar informes con sus resultados y las acciones que ha llevado a cabo.

Una vez que seleccionemos un antivirus, es conveniente tener claro cómo y dónde utilizarlo. La norma base es ‘un equipo, un antivirus’. Es decir, todos los equipos del sistema deberían tener su propio antivirus instalado. Sin embargo, en algunos equipos concretos, el antivirus puede ofrecer una protección más eficiente al resto de equipos. Por ejemplo, en equipos que actúan como Proxy de conexión con redes externas. Un antivirus bien configurado y escaneando en tiempo real se protegerá a sí mismo de virus, pero también evitará que gran parte del *malware* pase a los equipos de la red interna y por tanto a los usuarios finales.

En el entorno de los antivirus, el European Institute for Computer Antivirus Research (EICAR) ha desarrollado una prueba para validar la operatividad de los antivirus. Su objetivo es comprobar que un antivirus realmente funciona sin poner en peligro una máquina con virus reales. La prueba **EICAR**, que así se conoce, consiste en un inofensivo archivo de texto que debe ser guardado con extensión de archivo ejecutable. Una vez hecho, todo antivirus debería detectarlo como un virus a eliminar.

### **3.2. Otras medidas de protección**

Para proteger nuestros sistemas contra infecciones sería conveniente disponer de algunos otros elementos:

#### **3.2.1. Cortafuegos (*firewalls*)**

Los gusanos se propagan por la red conectándose a servicios con agujeros de seguridad, alojados en diferentes sistemas a lo largo de la red. Además de asegurarse que estos servicios vulnerables no se estén ejecutando, el siguiente paso que debe seguir el administrador, es verificar que el *firewall* no permita conexiones a estos servicios. Muchos *firewalls* modernos son capaces de filtrar en el tráfico de la red aquellos paquetes en los que se detecten ciertas firmas asociadas a virus o gusanos.

#### **3.2.2. NIDS (Sistemas de detección de intrusos de red)**

Los sistemas de detección de intrusiones de red son similares a los antivirus pero aplicados al tráfico de la red. Buscan en el tráfico de red firmas o patrones de comportamiento relacionados con virus o gusanos. Son capaces de alertar al usuario atacado o de detener el tráfico de red que intenta distribuir el *malware*.

#### **3.2.3. HIDS (Sistemas de detección de intrusos de *host*)**

Los sistemas de detección de intrusión de *host*, como por ejemplo las herramientas de software libre Tripwire y Aide, son capaces de detectar cambios realizados sobre archivos alojados en un servidor. Se basan en la hipótesis de que un archivo ejecutable, una vez compilado, no necesita ser modificado. Entonces, mediante el control de sus características, tales

como tamaño, fecha de creación y control de integridad, pueden detectar inmediatamente si ha ocurrido algo irregular que apunte a una infección.

#### **3.2.4. Sandboxes**

El concepto de *sandbox* se basa en que una aplicación o programa tiene su propio entorno para ejecutarse y no puede afectar al resto del sistema. Esto quiere decir que los recursos y los privilegios que la aplicación tiene mientras es ejecutada, son limitados. La ventaja de los *sandboxes* es que restringen el daño que un *malware* puede ocasionar al sistema infectado simplemente restringiendo los accesos de los que el *malware* dispone.

Otra opción en auge es la virtualización, que consiste en crear una máquina virtual completa mediante productos como VMWare. Esto aísla a la máquina virtual del sistema anfitrión ‘real’ limitando el acceso al mismo según lo haya configurado el administrador.

#### **3.2.5. Honeypot**

Se denomina *honeypot* a un sistema especialmente preparado para ser o parecer vulnerable, de modo que sea fácil de infectar por *malware*. Por supuesto, este sistema no contiene información ni programas valiosos (y muchas veces se trata de un sistema virtualizado). Sin embargo, este sistema está estrictamente monitorizado, de modo que el administrador puede obtener información sobre las amenazas que se ciernen sobre el sistema real con antelación. Esto permite al administrador gestionar las medidas de seguridad del sistema real para protegerse contra nuevos virus o ataques.



#### **4. REFERENCIAS**

- RFC 4949 Internet Security Glossary, Version 2

<http://tools.ietf.org/html/rfc4949>

- Observatorio de seguridad del Instituto Nacional de Tecnologías de la Comunicación

<http://www.inteco.es/Seguridad/Observatorio/>

- Web y enciclopedia del *malware* de PandaSoftware

<http://www.pandasecurity.com>

- Virus list, noticias y datos sobre *malware* de Kaspersky Labs.

<http://www.viruslist.com/sp/>

- Curso de Extensión Universitaria "Ferramentas de seguridade en GNU/Linux" (terceira edición) - Escola Superior de Enxeñaría Informática da Universidade de Vigo - (<http://ccia.ei.uvigo.es/curso2010/index.html>)

**Autor: Juan Otero Pombo**

**Ingeniero en Informática en el Concello de Ourense**

**Colegiado del CPEIG**

# **TELECOMUNICACIONES.**

**40. REDES LAN, MAN Y WAN.  
ESTRUCTURA DE REDES:  
TRONCAL, DISTRIBUCIÓN  
ACCESO. REDES PÚBLICAS DE  
TRANSMISIÓN DE DATOS.  
MODELO OSI. PROTOCOLOS DE  
RED. TCP/IP. ELEMENTOS DE  
INTERCONEXIÓN DE REDES.  
CONCENTRADORES,  
CONMUTADORES,  
REPETIDORES, PUENTES,  
ENCAMINADORES,  
PASARELAS. GESTIÓN DE  
REDES. CONFIGURACIÓN DE  
REDES.**

**Tema 40. Redes LAN, MAN y WAN. Estructura de redes: troncal, distribución acceso. Redes públicas de transmisión de datos. Modelo OSI. Protocolos de red. TCP/IP. Elementos de interconexión de red: concentradores, conmutadores, puentes, encaminadores, pasarelas. Gestión de redes. Configuración de redes**

**INDICE**

40.1 Redes LAN, MAN y WAN

40.1.1 PAN

40.1.2 LAN

40.1.3 MAN

40.1.4 WAN

40.2 Estructura de redes: troncal, distribución y acceso

40.2.1 Troncal

40.2.2 Distribución

40.2.3 Acceso

40.3 Redes públicas de transmisión de datos

40.3.1 Conceptos generales

40.4 Modelo OSI

40.4.1 Introducción

40.4.2 Conceptos generales

40.4.3 Transmisión entre entidades del mismo nivel (pares)

40.4.4 Conexiones

40.4.5 Capas del modelo OSI

40.4.6 Críticas al modelo OSI

40.5 Protocolos de red

40.5.1 Redes de conmutación de circuitos

40.5.2 Redes de conmutación de paquetes

40.6 TCP/IP

40.6.1 Enderezamiento y sistemas de nombres de dominio

40.6.2 Protocolos IP, TCP

40.7 Elementos de interconexión de red

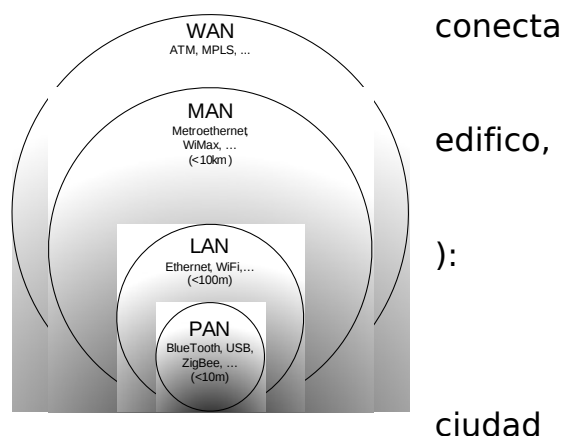
## 40.8 Gestión de redes. Configuración de redes

## 40.9 Bibliografía

### 40.1 REDES LAN, MAN Y WAN

Una de las formas clásicas de clasificar las redes es por su extensión física (o alcance) de la que se obtienen los siguientes tipos:

- Personal Area Network (PAN): red que conecta elementos próximos a una persona.
- Local Area Network (LAN): elementos en una área geográfica limitada, como son una casa, un etc.
- Metropolitan Area Network (MAN) algunas veces referida como Medium Area Network conecta elementos a lo largo de una o espacios de similar tamaño.
- Wide Area Network (WAN): son redes de gran extensión cubriendo ciudades, una provincia o incluso varios países.

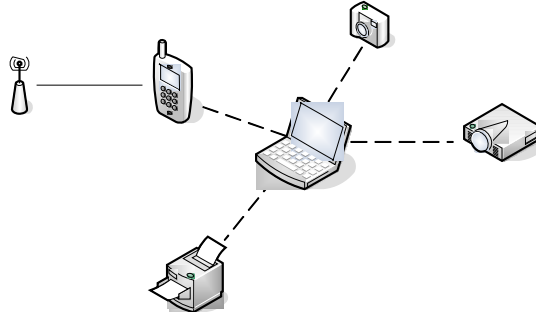


#### 40.1.1 PAN

Una PAN es una red usada para la comunicación de ordenadores y diferentes dispositivos informáticos próximos a una persona. Algunos ejemplos de estos dispositivos usados en una PAN son PCs, impresoras, teléfonos, PDAs o consolas de videojuegos.

La necesidad de estas redes es doble, por un lado conectar dispositivos de uso personal próximos como el teléfono móvil con un manos libres o con dispositivos de recogida de datos médicos. Y por otro permitir la movilidad de las personas aprovechando el uso y conectividad de estos dispositivos, siguiendo el ejemplo anterior, el móvil puede cambiar de conectarse a un manos libres en el coche a conectarse a otro en la casa.

Una PAN puede incluir dispositivos conectados por cable y dispositivos sin hilos alcanzando un máximo de 10 metros de distancia.

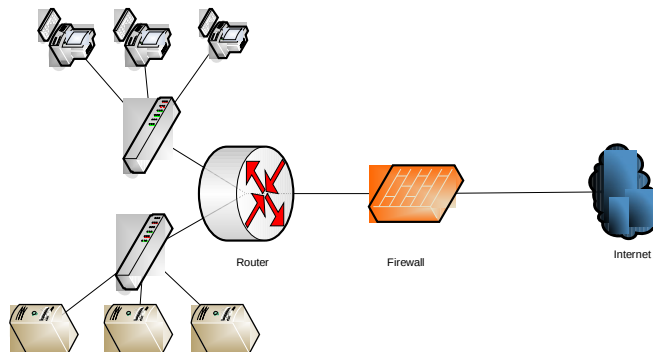


Las típicas tecnologías en las que se basan las PAN son USB y FireWire para las cableadas y Bluetooth y ZigBee para las sin hilos.

#### **40.1.2 LAN**

Una LAN conecta ordenadores y otros dispositivos en un espacio limitado como puede ser una casa, un edificio, una oficina o un conjunto de edificios próximos entre sí.

Normalmente la distancia que abarca una LAN no supera los 100 metros.



El ejemplo más común de LAN se da en el ámbito doméstico y de las pequeñas empresas donde varios equipos están conectados a un concentrador (switch), posiblemente varios servidores están conectados a otro y esos concentradores están conectados a un encaminador (router) para la conexión a Internet.

Las tecnologías dominantes usadas en las LAN son Ethernet (hoy en día gigabit ethernet) y WiFi (habitualmente 802.11g) aunque existen muchas otras tecnologías que empiezan (o continúan) a emplearse, como pueden ser las basadas en PLC, como por ejemplo HomePlug.

#### **40.1.3 MAN**

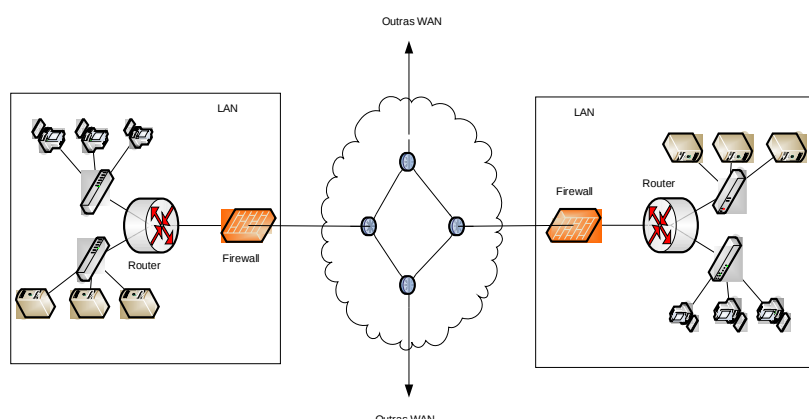
Una MAN es una red optimizada para un área geográfica mayor que una LAN, que va desde varios bloques de edificios incluso una ciudad. Una MAN puede ser propiedad de una organización pero normalmente es usada por muchos individuos y organizaciones distintas. Su utilidad típica es proporcionar conectividad entre varias LAN.

La distancia típica que cubre una MAN es de 10 km.

Normalmente estas redes están basadas en tecnologías como MetroEthernet, en redes cableadas, o Wimax, en redes sin hilos.

#### **40.1.4 WAN**

Las WAN son redes de ordenadores que cubren grandes áreas y suelen enlazar varias ciudades, provincias o países.



De forma análoga a las MAN la función típica de las WAN es conectar varias redes de menor extensión como varias MAN o varias LAN además de conectarse con otras redes WAN.

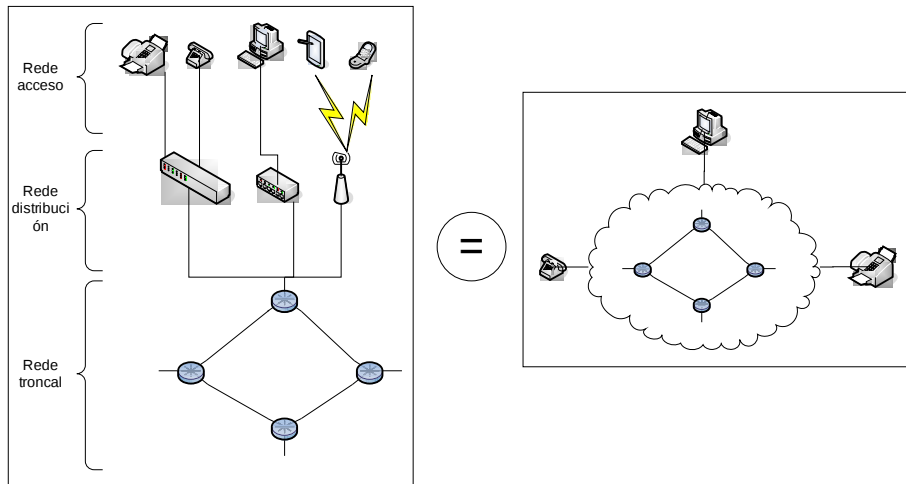
En contraste con los otros tipos de redes, las WAN no están limitadas a un tamaño máximo.

ATM o MPLS son algunas de las tecnologías usadas para desplegar redes WAN.

### **40.2 ESTRUCTURA DE REDES: TRONCAL, DISTRIBUCIÓN Y ACCESO**

Las redes (ya sea la red de un operador o de una compañía) se estructuran de forma jerárquica. Esta estructura les aporta modularidad, permite

aumentar los elementos de un nivel para expandir las redes sin afectar el resto de la misma (escalabilidad) y facilita la identificación y resolución de problemas.



Normalmente esta jerarquía se divide en 3 niveles (aunque, dependiendo de la red concreta, puede haber más como, por ejemplo, en las redes de cable):

- Troncal (en inglés *backbone* o *core*): es la columna vertebral de la red conectando los distintos elementos del nivel de distribución.
- Distribución: conecta varios elementos del nivel inferior con el nivel superior y suele estar limitada a una de las zonas físicas (por ejemplo una ciudad) en las que la red está presente.
- Acceso: los elementos del nivel de acceso permite conectar los equipos finales.

#### **40.2.1 TRONCAL**

A red troncal es el nivel jerárquico más alto dentro de la división en niveles y está formada por una parte de la infraestructura de la red de ordenadores que conecta varias partes de la misma (subredes).

Normalmente la capacidad de transferencia de datos de la red troncal es mayor que la de las subredes que conecta y posee caminos redundantes (los operadores suelen usar varios anillos y en las redes corporativas suele haber varias conexiones).



De tratarse de una red corporativa el acceso a Internet suele encontrarse en la red troncal.

Un ejemplo de este tipo puede ser la red que conecta las distintas ciudades donde da servicio un proveedor de internet.

#### **40.2.2 DISTRIBUCIÓN**

En el nivel de distribución se agregan los datos provenientes de los elementos del nivel de acceso para ser enviados al nivel superior.

Los elementos de este nivel también suelen estar redundados pero en menor medida que en el nivel superior.

De tratarse de una red corporativa es en este nivel donde se establecen las VLANs para cada departamento / división y donde se limitan los dominios de broadcast.

Siguiendo el ejemplo de antes estas podrían ser las redes que conectan los distintos nodos dentro de una ciudad.

#### **40.2.3 ACCESO**

El nivel de acceso es el nivel de la red donde se conectan los equipos finales (ordenadores, teléfonos,...).

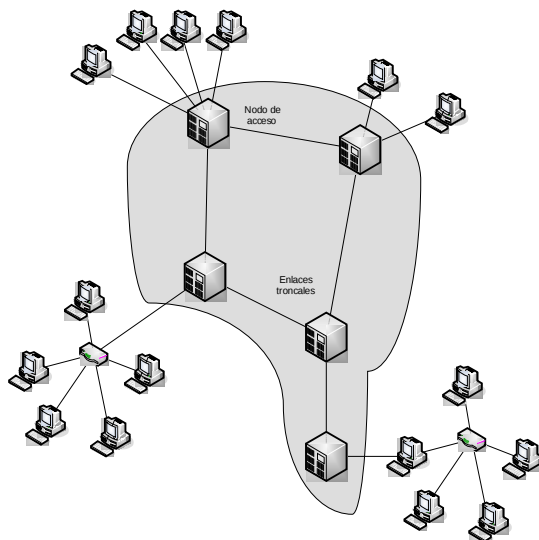
En el ejemplo anterior esta sería la red que conecta los distintos usuarios a un nodo.

### **40.3 REDES PÚBLICAS DE TRANSMISIÓN DE DATOS**

Desde el momento en que nació la necesidad de conectar dos localizaciones pasando por el dominio público, nació la necesidad de las redes públicas. Una red pública es aquella en la que cualquier entidad puede conectarse (normalmente bajo pago de una cuota) para comunicarse con otra entidad conectada a la misma red pero en distinta localización geográfica.

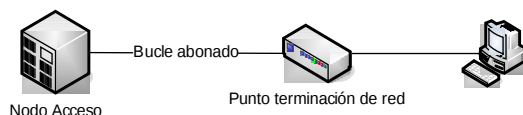
Esta disponibilidad de conectar cualesquiera dos entidades conduce a que haya varias entidades (individuos, compañías, gobiernos,...) conectadas a esta red, como oposición a una red personal donde sólo hay una entidad aprovechando los recursos de la red. Esto da como resultado que en muchas ocasiones las entidades quieran usar la red pública como una red personal (ya sea por simplicidad en la configuración, seguridad u otras

razones) dando lugar a las Redes Personales Virtuales (VPN por sus siglas en inglés).



#### 40.3.1 CONCEPTOS GENERALES

El bucle de abonado (bucle local o lazo local) es el cableado que se extiende desde los nodos de acceso (centrales de teléfonos,...) hasta el domicilio o local del usuario.

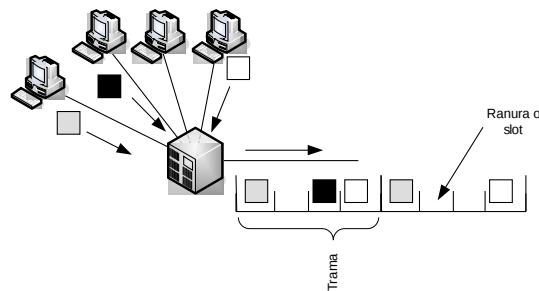


La conmutación es la conexión que realizan los diferentes nodos para lograr un camino apropiado para conectar dos usuarios de una red de telecomunicaciones. La conmutación permite la descongestión entre los usuarios de la red reduciendo el tráfico y aumentando el ancho de banda (comparándola con los sistemas basados en bus, por ejemplo).

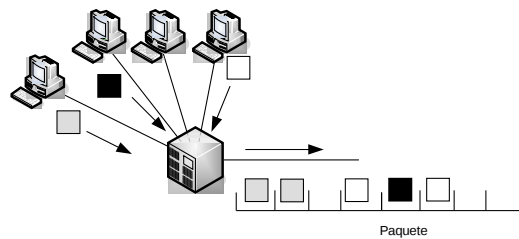
La multiplexación es la combinación de dos o más canales de información en un sólo medio de transmisión usando un dispositivo llamado multiplexor. El proceso conocido como demultiplexación. Existen muchas estrategias de multiplexación según el protocolo de comunicación empleado se pueden combinar para alcanzar el uso más eficiente; las más utilizadas son:

- La multiplexación por división de tiempo el TDM (Time division multiplexing). Dentro de esta estrategia podemos encontrar:

a. Multiplexación estática o síncrona.



b. Multiplexación estadística o asíncrona.



- La multiplexación por división de frecuencia el FDM (Frequency-division multiplexing) y su equivalente para medios ópticos, por división de longitud de onda o WDM (de Wavelength).
- La multiplexación por división en código o CDM (Code division multiplexing).

## **40.4 MODELO OSI**

### **40.4.1 INTRODUCCIÓN**

Inicialmente, los computadores eran elementos aislados que almacenaban en sus propios ficheros y precisaban la conexión de sus propios periféricos. La independencia era tal que, si se necesitaba imprimir un documento alojado en una máquina que no disponía de impresora, era necesario copiar el fichero en un disquete y llevarlo hasta un equipo con una impresora, conectarla e imprimirlo en este. Para evitar esto, la solución era instalar una impresora en el computador inicial, con la consiguiente duplicación de recursos y dispositivos.

Con instalaciones informáticas así la configuración y gestión de todos los ordenadores y periféricos a ellos conectados suponía un coste y una tarea

muy grande, llegando a ser poco práctica cuando el número de computadores fue creciendo en las distintas empresas.

Por esta razón apareció la necesidad de conectar los diferentes ordenadores entre sí e implantar métodos de comunicación y transferencia de datos entre ellos. Nace el concepto de “redes de ordenadores” y “trabajo en red”.

A mediados de los 70 diversos fabricantes desarrollan sus propios sistemas de redes locales. Sin embargo, la comunicación entre ordenadores pertenecientes a redes distintas de distintos fabricantes era imposible, debido a que los sistemas de comunicación de cada red eran propietarios. Es decir, estaban desarrollados con hardware y software propios y usaban protocolos y arquitecturas diferentes a los de otros fabricantes.

Las empresas se dieron cuenta de la necesidad de abandonar los sistemas propietarios y definir una arquitectura de red con un modelo común que permitiera conectar varias redes sin problemas.

En 1977, la Organización Internacional de Normas (ISO, International Standard Organization), integrada por industrias representativas del sector, creó un subcomité para el desarrollo de estándares de comunicación de datos que permitieran la interoperabilidad entre productos de diferentes fabricantes. Tras varias investigaciones sobre los modelos de red, elaboraron el modelo de referencia OSI (Open Systems Interconnection), en 1984.

#### **40.4.2 CONCEPTOS GENERALES**

El modelo de referencia para la Interconexión de Sistemas Abiertos se caracteriza por:

- Se ocupa de la conexión de sistemas abiertos, es decir, sistemas que permiten la comunicación con otros sistemas.
- Consta de siete capas. Por capa se entiende una (o varias) entidad(s) que realizan por sí mismas una función específica. Las entidades del mismo nivel se denominan entidades pares.



- Representa el primer paso a la estandarización internacional de los protocolos que se usan en las diversas capas.
- No es una arquitectura de red en sí, ya que no especifica los servicios y protocolos exactos que se tienen que usar en cada capa, si no que sólo define lo que debe hacer cada capa.

En el modelo OSI existen tres conceptos fundamentales:

- Servicio: Capacidad de comportamiento de una capa. Cada capa presta algunos servicios a las entidades que se encuentran sobre ella, que acceden a los mismos a través de los puntos de acceso al servicio (SAP), intercambiando primitivas de servicio.
- Interfaz: Indica a los procesos de la capa superior cómo acceder a ella, especificando cuáles son los parámetros y qué resultados esperar. La interfaz entre dos capas en una máquina no tiene por qué ser igual a la correspondiente en otra máquina.
- Protocolo: Conjunto de reglas que determinan el comportamiento de comunicación horizontal entre entidades pares. Se pueden cambiar los protocolos de una capa sin afectar a las demás.

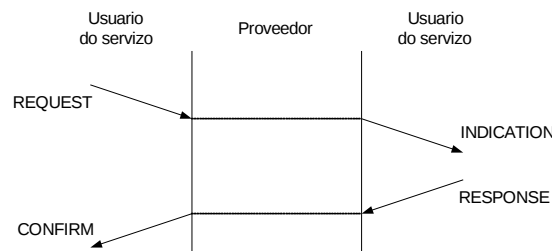
#### **40.4.3 TRANSMISIÓN ENTRE ENTIDADES DEL MISMO NIVEL (PARES)**

Las entidades pares residentes en el nivel N+1 se comunican entre sí a través del nivel N, mediante el uso de primitivas de servicio. Sin embargo, existe una comunicación lógica horizontal entre entidades pares. Las reglas que regulan esta comunicación vienen reflejadas en el protocolo de pares.

Por lo tanto, en la especificación de cada capa existen dos documentos:

- Especificación del servicio, que informa sobre las primitivas existentes. En la descripción de las primitivas se indica cuántos parámetros puede o debe haber y qué información contienen, pero no se especifica cómo ni con qué formato tienen que ser “pasados”. Esto es un asunto local y definir esto equivale a definir la Interfaz. Existen cuatro tipos de primitivas.
  - o De petición (REQUEST). Empleada para invocar un servicio y pasarle los parámetros necesarios para su ejecución.

- o De indicación (INDICATION). Usada para indicar que un procedimiento fue invocado por el usuario par del servicio en la conexión y pasar los parámetros asociados o para indicar al usuario del servicio el inicio de una acción por parte del proveedor.
- o De respuesta (RESPONSE). Empleada por el usuario del servicio para reconocer o completar algún procedimiento previamente iniciado por una indicación del proveedor.
- o De confirmación (CONFIRM). Usada por el proveedor del servicio para reconocer o completar algún procedimiento previamente iniciado por una petición del usuario.



- Especificación del protocolo, que describe las PDUs (Protocol Data Units) y las reglas que determinan su intercambio entre unidades pares. Existen dos clases de PDUs:
  - o De datos, que contiene los datos del usuario final (en el caso de la capa de aplicación) o la PDU del nivel ya superior.
  - o De control, que sirve para gobernar el comportamiento completo del protocolo en sus funciones de establecimiento y ruptura de la conexión, control de flujo, control de errores, etc.No contienen información alguna procedente del nivel N+1.

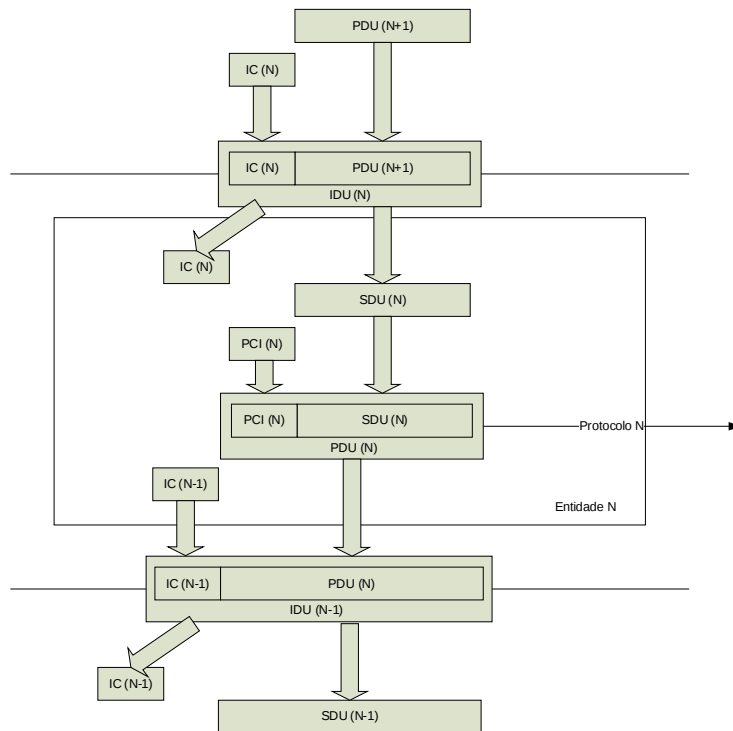
En la siguiente figura se ilustra la nomenclatura utilizada por ISO en la pila que envía la información.

- PDU (N) -> Unidad de Datos del Protocolo del nivel N. Contiene información de control del protocolo y, posiblemente, datos de

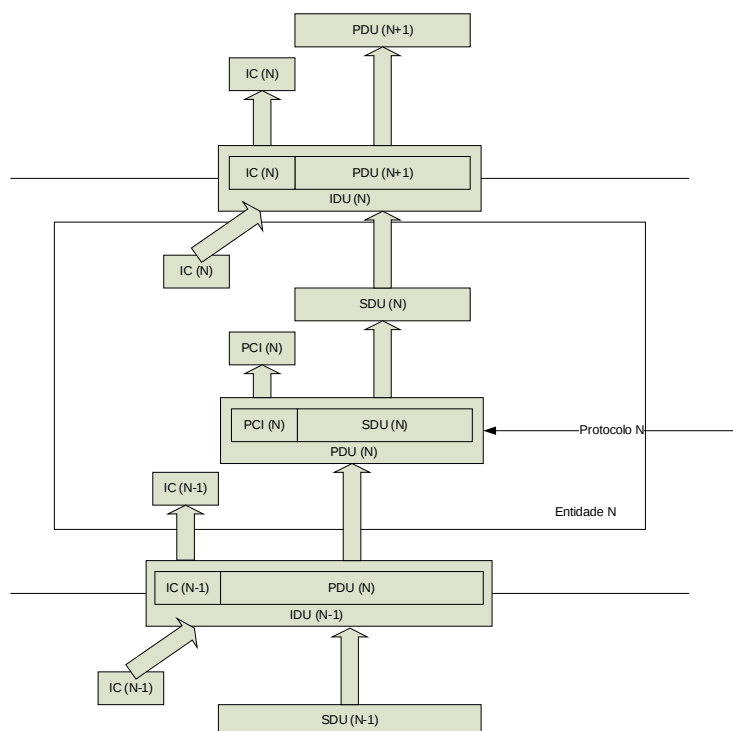


usuario. Debe estar puntual y perfectamente definida (sintáctica y semánticamente).

- ICI (N) -> Información de Control de la Interfaz del nivel N. Es transferida entre una entidad N+1 y una entidad N para coordinar el funcionamiento local conjunto. Su sintaxis y semántica son un asunto local cuando actúa como información complementaria en la transferencia de una PDU.
- IDU (N) -> Unidad de Datos de la Interfaz del nivel N. Es transferida a través del punto de acceso al servicio N. Contiene la ICI más la totalidad (o parte) de la información de la PDU (N+1). La estructura de las informaciones de la IDU es un asunto local.
- SDU (N) -> Unidad de Datos del Servicio del nivel N. Representa la información entregada por el nivel inmediatamente superior.
- PCI (N) -> Información de Control del Protocolo del nivel N. Información generada por la entidad N para coordinar el funcionamiento conjunto con otra u otras entidades del nivel N con las que está intercambiando información “horizontal”.
- UD (N) -> Datos del Usuario. Datos transferidos entre entidades del nivel N en nombre de las entidades del nivel N+1.



En la pila que recibe la información, la relación existente entre las unidades de datos residentes en las entidades del nivel N-1, N y N+1 es similar a la representada más arriba, sólo que en lugar de añadir elementos lo que se produce son reducciones y las flechas son ascendentes, como muestra la figura siguiente.





Cuando las unidades de datos de los niveles limítrofes no tienen tamaños compatibles, se recurre a alguna de las siguientes funciones:

- Segmentación de la SDU. Cuando la SDU es demasiado grande, se reparte en más de una PDU. La función simétrica en el extremo receptor es el reensamblado, que consiste en la identificación de varias PDUs con una sola SDU. La PCI incluye, en este caso, información adicional para posibilitar el reensamblado.
- Empaquetado de la SDU. Cuando el tamaño de la SDU es más pequeño que el de la PDU, puede ser conveniente o necesario agrupar varias SDUs en una sola PDU. El empaquetado es el caso contrario a la segmentación de la SDU. La función inversa del empaquetado es el desempaquetado, que consiste en descomponer una PDU en varias SDUs. El caso de segmentación se da con más frecuencia que el de empaquetado.
- Segmentación de la PDU. Si la PDU es muy grande, puede ser necesario repartirla en más de una IDU del nivel inferior. Por eso, en la definición de IDU se dice que contiene la ICI más la totalidad (o parte) de la PDU. También en este caso deben existir informaciones adicionales que posibiliten el reensamblado en el extremo receptor.
- Concatenación de PDU. Si el tamaño de la SDU del nivel inferior, y como resultado, de la IDU del nivel inferior, es mayor que el de la PDU, puede convenir agrupar varias PDUs sobre una sola SDU. La función inversa a esta, que se realiza en el extremo receptor, es la separación. La concatenación-separación es el caso contrario de la segmentación-reensamblado de la PDU, siendo este último el más frecuente.

#### **40.4.4 CONEXIONES**

El modelo de referencia OSI está orientado a conexión. Esto significa que, en todos los niveles, es necesario que se establezca previamente una conexión para que pueda existir intercambio de datos. Sin embargo,

existen protocolos que no requieren esta condición, son los no orientados a conexión (connectionless).

En las comunicaciones orientadas a conexión se pierde tiempo y recursos en establecer y liberar la conexión entre dos nodos, pero se garantiza que el nodo remoto está escuchando. Por el contrario, en las comunicaciones no orientadas a conexión se ahorra tiempo y recursos, pero a costa de no saber si el otro extremo está a la escucha.

A nivel N-1 se establece una asociación, una conexión N-1, para que dos entidades del nivel N puedan comunicarse. La conexión N-1 es un servicio ofrecido por el nivel N-1, a través del cual circulan unidades de información del nivel N.

El Punto de Acceso al Servicio (SAP) del nivel N identifica la dirección del nivel N a la que se conectan las entidades del nivel N+1. La relación entre direcciones de dos niveles consecutivos puede ser 1 a 1 (1 dirección del nivel N por cada dirección del N +1), N a 1 (Varias direcciones del nivel N por cada dirección del N +1) (no confundir con la multiplexación, que se explica más adelante) o 1 a N (1 dirección del nivel N para varias direcciones del N +1).

#### **40.4.4.1 ESTABLECIMIENTO DE CONEXIONES**

Para que dos entidades N establezcan una conexión, es necesario:

- Disponer de una conexión N-1 por debajo. Es necesario establecer previamente la conexión N-1 antes de intentar establecer la conexión N, descendiendo hasta que se encuentra una disponible a nivel físico (el nivel más bajo). Sin embargo, en los niveles altos, se aprovecha la circunstancia de establecimiento de la conexión N para establecer, al mismo tiempo, la conexión N+1.
- Estar ambas entidades conformes con el establecimiento.

Una vez establecida la conexión, es como si la entidad dispusiera de un “tubo” a través del cual pudiera enviar datos a su entidad de comunicación correspondiente.

#### **40.4.4.2 LIBERACIÓN DE CONEXIONES**

La liberación de una conexión N es iniciada, normalmente, por una de las entidades N+1 que la está usando. Sin embargo, esta ruptura puede ser también iniciada por una de las entidades N que le dan soporte.

Al contrario que ocurre en el establecimiento, la liberación de una conexión N-1 no implica la liberación de la conexión N. Esto es así para permitir que, si la conexión N-1 rompe por dificultades de la comunicación, pueda intentarse restablecerla o sustituirla por otra.

La liberación de una conexión puede ser:

- Abrupta. Se libera de inmediato y los datos pendientes de envío se pierden.
- Suave o diferida. Antes de romper la conexión, se espera a no tener datos pendientes.

#### **40.4.4.3 MULTIPLEXACIÓN Y DIVISIÓN**

La multiplexación es la función que permite utilizar una sola conexión N-1 para soportar varias conexiones del nivel N. Todos los “tubos” de conexión N viajan por dentro del “tubo” de conexión N-1. Varias comunicaciones entre entidades pares de nivel N se realizan apoyadas en una sola conexión del nivel N-1. Es decir, las distintas entidades usan un solo punto de acceso al servicio N-1. La función inversa realizada en el receptor se denomina demultiplexación. No debe confundirse el concepto de multiplexación con el de concatenación, ya explicado.

La división es la función que permite la utilización de más de una conexión N-1 por una sola conexión de nivel N. Con ello, el flujo de datos que soporta puede ser mayor. El flujo de datos del “tubo” correspondiente a la conexión N se reparte entre todos los “tubos” de conexiones N-1. En el extremo receptor, la función inversa se denomina recombinación y debe ser capaz de recuperar el orden en que las PDUs fueron generadas por el extremo emisor. No debe confundirse el concepto de división con el de segmentación, ya explicado.

#### **40.4.4.4 TRANSMISIÓN DE DATOS**

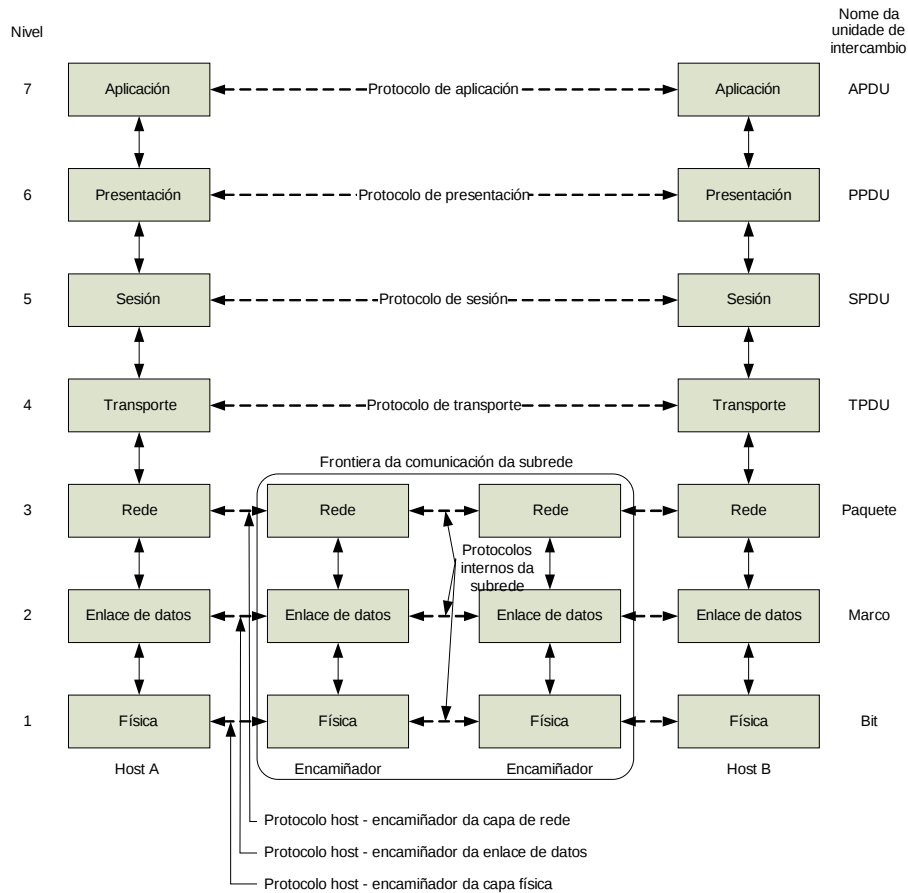
Una capa de una máquina no puede transferir los datos de forma directa a su capa par en otra máquina, si no que necesita de los servicios de todas las capas que se encuentran por debajo de ella en la jerarquía de capas, pasándose la información hacia abajo hasta llegar al nivel físico, donde se transmiten a la máquina receptora.

Cada capa utiliza el encapsulamiento para colocar la PDU de la capa superior en su campo de datos y agregar cualquier encabezado e información final que la capa necesite para realizar su función. De esta forma, a medida que los datos se desplazan hacia abajo a través de las capas del modelo OSI, el tamaño del mensaje va creciendo. A nivel 3, la PDU se llama paquete e incluye las direcciones lógicas origen y destino. A nivel 2, la trama incluye las direcciones físicas. Y, finalmente, la capa física codifica los datos de la trama de enlace de datos en un patrón de unos y ceros para su transmisión a través del medio.

En la máquina receptora se realiza el proceso inverso, retirando los distintos encabezados, uno por uno, conforme el mensaje se propaga hacia arriba por las capas.

#### **40.4.5 CAPAS DEL MODELO OSI**

Como se dijo en el punto anterior, el modelo de referencia OSI se dividió en siete niveles o capas, para poder simplificar la implementación de la arquitectura necesaria.



Los principios que se aplicaron para llegar a estas siete capas son los siguientes:

- Se debe crear una nueva capa siempre que se precise un grado diferente de abstracción.
- A cada capa se le asigna una función bien definida o un conjunto de funciones relacionadas entre sí, tratando de resolver en cada capa un problema distinto.
- La funcionalidad de cada capa se debe elegir habida cuenta la posibilidad de definir protocolos normalizados a nivel internacional.
- La frontera de las capas será tal que se minimice el flujo de información a través de la interfaz existente entre ambas.
- El número de capas debe ser lo suficientemente grande para no reunir en un mismo nivel funcionalidades distintas y suficientemente pequeño para que la arquitectura resultante sea manejable.

#### **40.4.5.1 CAPA FÍSICA (NIVEL 1)**

La capa física está relacionada con la transmisión de bits por un canal de comunicación, de forma que sólo reconoce bits individuales, sin estructura alguna. Es decir, la PDU del nivel físico se corresponde con un bit o, dicho de otro modo, cada bit se considera una unidad de datos.

Las consideraciones de diseño tienen que ver con las interfaces mecánica, eléctrica y de procedimiento, así como con el medio de transmisión que está bajo la capa física, asegurando que cuando un lado envíe un bit con el valor “1”, se reciba en el otro extremo el valor “1”, no como el valor “0”.

La capa física proporciona sus servicios a la capa de enlace de datos. Sus principales funciones son:

- Definición de características materiales (componentes y conectores mecánicos) y eléctricas (niveles de tensión, tipo de señal) que se van a utilizar en la transmisión de los datos por el medio físico.
- Definición de las características funcionales de la interfaz en cuanto al establecimiento, mantenimiento y liberación del enlace físico.
- Definición de reglas de procedimiento, es decir, la secuencia de eventos para transmitir.
- Transmisión de flujos de bits a través del medio.
- Manejo de voltajes y pulsos eléctricos para representar 1's o 0's.
- Especificación de cables, polos en un enchufe, componentes de la interfaz con el medio, etc.
- Especificación del medio físico de transmisión (coaxial, fibra óptica, par trenzado, etc.)
- Garantizar la conexión física, pero no la fiabilidad de la misma. Es decir, no se realiza ningún control de errores en este nivel. Eso corresponde al nivel superior.

#### **40.4.5.2 CAPA DE ENLACE DE DATOS (NIVEL 2)**

Puesto que la capa física sólo acepta y transmite una corriente de bits sin preocuparse por su significado o estructura, corresponde al nivel de enlace tomar el medio de transmisión en bruto y transformarlo en una línea que parezca estar libre de errores a los ojos de la capa de red.

La capa de enlace de datos puede ofrecer a la capa de red varias clases de servicio con diferentes calidades.

Algunas de las funciones más importantes de la capa de enlace son:

- Establecimiento de los medios necesarios para la comunicación fiable y eficiente entre dos máquinas de la red.
- Estructuración de los datos en un formato predefinido, denominado trama, que suele ser de unos cientos de bytes, añadiendo una secuencia especial de bits al principio y al final de la misma.
- Sincronización en el envío de tramas.
- Detección y control de errores provenientes del medio físico mediante el uso de bits de paridad, CRC (Códigos de Redundancia Cíclica) y envío de acuses de recibo por parte del receptor que debe procesar el emisor.
- Utilización de números de secuencia en las tramas para evitar pérdidas y duplicidades.
- Utilización de la técnica de “piggybacking”, consistente en el envío de acuses de recibo dentro de tramas de datos.
- Resolución de los problemas provocados por las tramas dañadas, perdidas o duplicadas.
- Control de la congestión de la red.
- Mecanismos de regulación de tráfico o control de flujo, para evitar que un transmisor veloz sature de datos a un receptor lento.
- Control del acceso al canal compartido en las redes de difusión.

#### **40.4.5.3 CAPA DE RED (NIVEL 3)**

La capa de red es una capa compleja que ofrece sus servicios a la capa de transporte. Responsable de la conmutación y encaminado de la información, sus funciones se pueden resumir de la siguiente forma:

- Conocimiento de la topología de la red, es decir, de la forma en que están interconectados los nodos, con objeto de determinar la mejor ruta para la comunicación entre máquinas que puedan estar situadas en redes geográficamente distintas.



- División de los mensajes de la capa de transporte en unidades más complejas, llamadas paquetes (NPDUs), y asignación de direcciones lógicas a los mismos.
- Ensamblado de paquetes en el host destino.
- Establecimiento, mantenimiento y liberación de las conexiones de red entre sistemas.
- Determinación del camino de los paquetes desde la fuente hasta el destino a través de dispositivos intermedios (routers):
  - o Las rutas pueden basarse en tablas estáticas.
  - o Las rutas se pueden determinar al inicio de cada conversación.
  - o Las rutas pueden ser dinámicas, determinándose con cada paquete en función de la carga de la red.
- Envío de paquetes de nodo a nodo usando un circuito virtual (orientado a conexión) o datagramas (no orientado a conexión).
- Control de la congestión.
- Control de flujo.
- Control de errores.
- Reencaminamiento de paquetes en caso de caída de un enlace.
- Con frecuencia, funciones de contabilidad, para determinar cuántos paquetes, caracteres o bits envía cada cliente y producir información de facturación.

Esta capa sólo es necesaria en las redes de conmutación o redes interconectadas. En redes punto a punto o de difusión existe un canal directo entre los dos equipos, por lo que el nivel 2 proporciona directamente conexión fiable entre los dos equipos.

#### **40.4.5.4 CAPA DE TRANSPORTE (NIVEL 4)**

Se trata de una verdadera capa extremo a extremo, desde el origen hasta el destino. La comunicación en los niveles inferiores es entre máquinas adyacentes.



La capa de transporte proporciona sus servicios a la capa de sesión, efectuando la transferencia de datos de manera transparente entre dos entidades de sesión.

El nivel 4 tiene la interfaz más sencilla de todo el modelo OSI, siendo la que tiene menos primitivas. No tiene primitivas de confirmación, pues se considera a todos los efectos que es un nivel fiable.

Su función más importante es la aceptación de datos de la capa de sesión, división en unidades más pequeñas, si es preciso, denominadas segmentos, y envío de esta información a la capa de red, asegurando que todas las partes lleguen correctamente al otro extremo de forma eficiente, donde son re ensambladas.

Otras funcionalidades son:

- Establecimiento, mantenimiento y terminación adecuados de los circuitos virtuales (conexiones que se establecen dentro de una red). Cuando se inicia la conexión se determina una ruta desde la fuente hasta el destino, ruta que es usada para todo el tráfico de datos posterior.
- Determinación, en el momento del establecimiento de la sesión, del tipo de clase de servicio de transporte que se proporcionará a la capa de sesión:
  - o Canal punto a punto libre de errores, que entrega los mensajes o bytes en la orden en que se envían.
  - o Mensajes aislados sin garantía respecto a la orden de entrega.
  - o Difusión de mensajes a múltiples destinos.
- Control de flujo, que desempeña un papel clave en esta capa. El control de flujo entre nodos es distinto del control de flujo entre encaminadores, que tiene lugar en la capa de red. Los datos pueden ser normales o urgentes. Estos últimos saltan los mecanismos de control de flujo.
- Detección y recuperación de errores de transporte.
- Control de la congestión.

- Numeración de los segmentos para prevenir pérdidas y doble procesamiento de transmisiones.
- Garantía de recepción de todos los datos y en el orden adecuado, sin pérdidas ni duplicados.
- Asignación de una dirección única de transporte a cada usuario.
- Aislamiento de las capas superiores de los cambios inevitables de la tecnología del hardware.
- Contabilidad a través de la red.

El normal es que la capa de nivel 4 cree una conexión de red distinta para cada conexión de transporte que requiere la capa de sesión. Sin embargo, es posible crear múltiples conexiones de red, dividiendo los datos entre ellas para aumentar el volumen, si se requiere un volumen de transmisión alto. De igual forma, si resulta costoso mantener una conexión de red, el nivel 4 puede multiplexar varias conexiones de transporte en la misma conexión de red para reducir el coste. En la cabecera que añade este nivel se envía la información que identifica a qué conexión pertenece cada mensaje. En cualquier caso, la capa de transporte debe hacer esto de forma transparente a la capa de sesión.

#### **40.4.5.5 CAPA DE SESIÓN (NIVEL 5)**

Esta capa proporciona sus servicios a la capa de presentación, facilitando el medio necesario para que las entidades de presentación de dos máquinas diferentes organicen y sincronicen su diálogo y procedan al intercambio de datos, mediante el establecimiento de sesiones.

Por tanto, la función principal de la capa de sesión es el establecimiento, administración y finalización ordenada de sesiones entre dos máquinas.

Una sesión permite el transporte común de datos, como efectuar un login en un sistema remoto o transferir un fichero entre dos nodos, pero también proporciona servicios mejorados, útiles en algunas aplicaciones, como los que se detallan a continuación.

- Manejo del control del diálogo (quién habla, cuándo, cuánto tiempo, half duplex o full duplex). Las sesiones pueden permitir que el tráfico



vaya en una única dirección, comunicaciones bidireccionales alternadas (half duplex), o en ambas direcciones al mismo tiempo, comunicaciones bidireccionales simultáneas (full duplex). En las comunicaciones half duplex, la capa de sesión ayuda a llevar el control de los turnos, mediante el manejo de fichas, también llamadas testigos o tokens. Sólo el lado que posea la ficha puede efectuar la operación.

- Sincronización del diálogo, mediante la inserción de puntos de verificación en la corriente de datos (APDU), de modo que si se produce una interrupción sólo es necesario repetir la transferencia de los datos después del último punto de verificación. La decisión de dónde colocar los puntos de sincronización es competencia directa del nivel de aplicación. Los puntos de sincronización pueden ser de dos tipos.
  - o Mayor. Necesita confirmación del otro extremo para seguir con la transferencia del siguiente bloque.
  - o Menor. Se intercalan entre dos puntos de sincronización mayores. No necesitan confirmación. Al confirmarse un punto de sincronización mayor, se dan por confirmados los puntos menores intermedios.

El bloque entre el primero y el último punto de sincronización mayor se llama actividad. Cuando se establece una conexión de sesión, automáticamente se abre una actividad para poder trabajar. Sólo un tipo de datos concreto puede enviarse además de una actividad, los datos de capacidades (CD), que son datos de control. Las actividades se dividen en unidades de diálogo, que es el contenido entre dos puntos de sincronización mayor consecutivos.

En esta capa la referencia a los dispositivos es por el nombre y no por la dirección. Además, es aquí donde se definen las API's (Application Program Interface).

El protocolo de nivel de sesión es orientado a la aplicación, ya que sus funcionalidades se adaptan a las necesidades de la aplicación.

Las unidades de datos del nivel de sesión, SPDUs, que regulan el diálogo, fluyen horizontalmente a través del nivel 5, pero son puestas en circulación por iniciativa de los correspondientes procesos de aplicación que residen en el nivel 7. Es decir, la capa de sesión no es un nivel autónomo que tenga capacidad para tomar decisiones sobre quién habla y quién escucha. Estas decisiones están reservadas a las entidades de la capa de aplicación. El nivel 5 sólo proporciona los mecanismos para que las entidades de aplicación puedan regular el diálogo entre sí.

En el párrafo anterior se habla como si la capa de aplicación residiera directamente encima de la de sesión. Esto no es así. Como se indica en la enumeración de capa, la capa de sesión ofrece sus servicios a la capa de presentación. Lo que ocurre es que el protocolo de nivel 6 no es un protocolo “normal”. De hecho, la mayor parte de las primitivas que comunican la capa de presentación con el nivel 7 son translación exacta de las correspondientes primitivas entre el nivel 6 y la capa de sesión.

#### **40.4.5.6 CAPA DE PRESENTACIÓN (NIVEL 6)**

A diferencia de las capas inferiores, las explicadas hasta ahora, que se ocupan sólo del movimiento fiable de bits de parte a parte, la capa de presentación se encarga de la sintaxis y la semántica de la información que se transmite. Además, aísla de dichas capas inferiores el formato de los datos de las aplicaciones específicas.

Las estructuras de datos a intercambiar se tienen que definir de forma abstracta, mediante la codificación de estos datos de una manera estándar acordada, haciendo posible así la comunicación entre computadoras con representaciones locales diferentes. La capa de presentación maneja estas estructuras de datos abstractas y las convierte de la representación de la computadora a la representación estándar de la red y viceversa.

Además de esta funcionalidad, la capa de presentación ofrece a la capa de aplicación los servicios de:

- Garantía de que la información que envía la capa de aplicación de un sistema pueda ser entendida y utilizada por la capa de aplicación de otro sistema.
- Acuerdo y negociación de la sintaxis de transferencia en la fase de establecimiento de la conexión. La sintaxis escogida puede ser cambiada durante el tiempo que dure la conexión.
- Definición del código a utilizar para representar una cadena de caracteres (ASCII, EBCDIC, etc.)
- Interpretación de los formatos de números...
- Compresión de los datos, si es necesario.
- Aplicación de procesos criptográficos, si así se requiere. Es el nivel clave para el sistema de seguridad del modelo OSI.
- Formateo de la información para su visualización o impresión.

#### **40.4.5.7 CAPA DE APLICACIÓN (NIVEL 7)**

Es la capa del modelo OSI más próxima al usuario. Difiere de las demás capas en que no proporciona servicios a ninguna otra capa OSI, si no a las aplicaciones que se encuentran fuera del modelo. Todas las capas anteriores sirven de mera infraestructura de telecomunicación, es decir, mantienen en buen estado el camino para que fluyan los datos. Es la capa de aplicación la que hace posible que una red se pueda usar, a pesar de estar abstraída de todas las restantes funciones necesarias para el establecimiento de la comunicación.

Las aplicaciones más importantes que hacen uso de esta capa, para que los procesos de las aplicaciones accedan al entorno OSI son, entre otras:

- Correo electrónico. Primera aplicación que se normalizó en OSI.
- Terminal virtual de red abstracta, que diferentes editores y programas puedan manejar.
- Transferencia de archivos.
- Carga remota de trabajos.
- Servicios de directorio.

- Login remoto (rlogin, telnet).
- Acceso a bases de datos.
- Sistemas operativos de red.
- Aplicaciones Cliente/Servidor...

Por supuesto, en el nivel 7 también hay cabida para aplicaciones “particulares” diseñadas por y para un núcleo reducido de usuarios, pero carecen de demasiado interés.

Las PDUs de la capa de aplicación, APDUs, son de formato muy flexible y variable. Entre dos APDUs pueden encontrarse diferencias sustanciales en cuanto a su tamaño, número de campos presentes, etc., que dependen de las necesidades de cada momento.

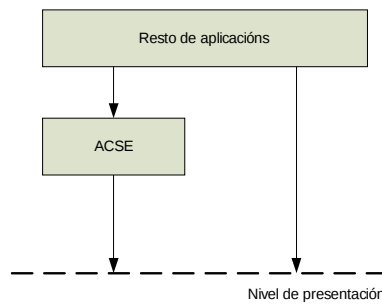
A cada una de las partes de una aplicación que se encarga de una tarea específica se la denomina Elemento de Servicio de Aplicación (ASE). El conjunto de todos los ASEs que forman una aplicación concreta y la relación entre ellos forman el contexto de aplicación.

Hay ASEs válidos para varias aplicaciones:

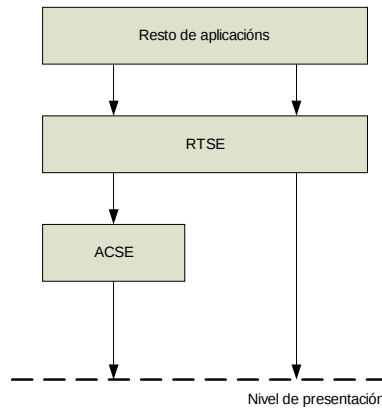
- ACSE (Association Control Service Element). Establecimiento, manejo y liberación ordenada o abrupta de conexiones. Lo utilizan todas las aplicaciones.
- RTSE (Reliable Transfer Service Element). Garantiza la fiabilidad en la transferencia de datos, solucionando los problemas que se habían producido del nivel 4 hacia arriba. Es responsable de manejar todas las funciones de nivel 5. Lo utilizan algunas aplicaciones, no todas.
- ROSE (Remote Operation Service Element). Facilita el trabajo de petición de operaciones remotas y devolución de los resultados.

Las aplicaciones se componen de una mezcla de elementos específicos y comunes. Las siguientes figuras ilustran las relaciones entre ASEs comunes.

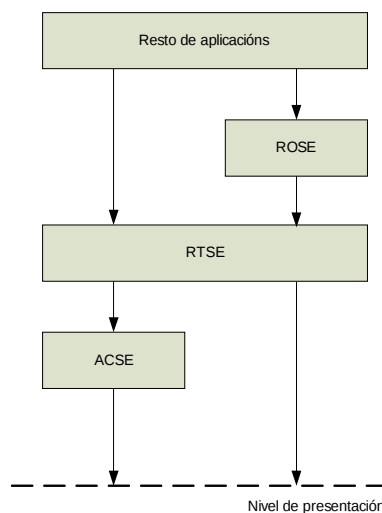
- Para aplicaciones que no manejan RTSE ni ROSE. Tienen que gestionar por sí mismas los puntos de sincronización, los token, etc.



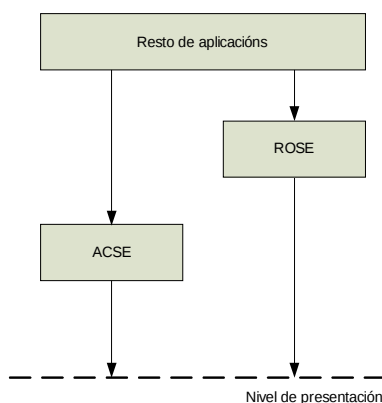
- RTSE es el que se encarga de las RTSE actividades, los puntos de sincronización, etc. En este caso, la aplicación no maneja directamente el nivel 5, si no que lo hace a través de RTSE.



- Para las aplicaciones que hacen uso de los tres ASEs comunes.



- En este caso, ROSE trabaja directamente sobre el nivel 5.



#### 40.4.6 CRÍTICAS AL MODELO OSI

La verdadera razón de que el modelo OSI tenga siete capas es que, en el momento de diseño, IBM tenía un protocolo patentado de siete capas, llamado SNA (System Network Architecture, Arquitectura de Red de Sistemas) y, en esa época, IBM dominaba la industria de la computación. Por otro lado, el proceso de estandarización fue demasiado largo. Cuando aún se trabajaba en la definición de OSI, ya existían implementaciones completas y gratuitas de TCP/IP y aplicaciones como e-mail, telnet, ftp, etc. Algunos de los problemas o fallos que se detectaron en el modelo de referencia OSI son:

- Aunque el modelo OSI, junto con sus definiciones y protocolos de servicios, es muy completo; hay que reconocer que los estándares son difíciles de implementar e ineficientes en su operación. Las implementaciones iniciales fueron enormes, inmanejables y lentas.
- OSI se desarrolló antes de que se hubiesen inventado los protocolos. Así que los diseñadores no supieron bien qué funcionalidad poner en cada capa.
- La capa de sesión tiene poco uso en la mayor parte de las aplicaciones.
- La capa de presentación está prácticamente vacía.
- Por el contrario, las capas de red y de enlace de datos están muy llenas, hasta tal punto que llegaron a dividirse en múltiples subcapas, cada una con funciones distintas.





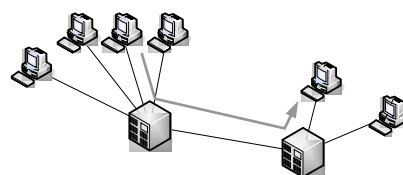
- Algunas funciones, como el direccionamiento, el control de flujo y el control de errores, reaparecen una y otra vez en cada capa.
- Omisión de la administración de la red en el modelo.
- Aunque en el presente documento se situó en la capa de presentación la función de cifrado y seguridad de los datos, inicialmente se dejó fuera del modelo por falta de acuerdo sobre en qué capa colocarlo.
- En la capa de red se ofrece servicio orientado a conexión y no orientado a conexión. Sin embargo, en la capa de transporte, donde el servicio es visible a los usuarios, sólo se ofrece comunicación orientada a conexión.
- El modelo está dominado por una mentalidad de comunicaciones. Las computadoras son diferentes de los teléfonos. Muchas de las decisiones tomadas son inapropiadas para la forma de trabajar de las computadoras y el software. El modelo de un sistema controlado por interrupciones no se ajusta conceptualmente a las ideas modernas de la programación estructurada.

## **40.5 PROTOCOLOS DE REDE**

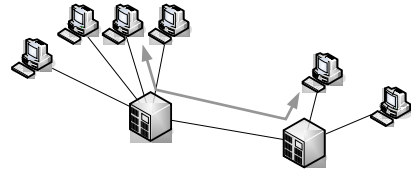
### **40.5.1 REDES DE CONMUTACIÓN DE CIRCUITOS**

La conmutación de circuitos es un tipo de conexión que realizan los diferentes nodos de una red para lograr un camino apropiado para conectar dos usuarios. En este tipo de conmutación se establece un canal de comunicaciones dedicado entre las dos estaciones. Se reservan recursos de transmisión y de conmutación de la red para su uso exclusivo en el circuito durante la conexión.

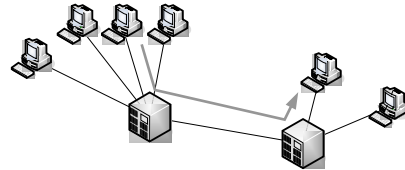
En este tipo de redes la comunicación tiene tres fases:



### 1. Establecimiento del circuito



### 2. Transmisión de los datos



### 3. Liberación del circuito

Este tipo de redes usa TDM, lo que permite disponer de un retardo fijo y predecible. Desde el punto de vista del usuario el enlace es punto a punto. Un ejemplo de este tipo de red es la Red Telefónica Conmutada (RTC).

## **40.5.2 REDES DE CONMUTACIÓN DE PAQUETES**

La conmutación de paquetes es el sistema más usado para el envío de datos en una red de ordenadores. Un paquete es un grupo de información que consta de dos partes: los datos propiamente dichos, y una información de control, que especifica la ruta a seguir a lo largo de la red hasta el destino del paquete. Existe un límite superior para el tamaño de los paquetes; en caso de superarlo es necesario dividir el paquete en otros más pequeños. También puede existir un límite inferior para el tamaño del paquete dependiendo de la tecnología de transmisión usada.

Existen dos técnicas para la transmisión de paquetes en las redes de conmutación de paquetes:

- La basada en circuitos virtuales: Muy similar a la conmutación de circuitos, la diferencia radica en que con los circuitos virtuales la ruta no es dedicada, si no que un único enlace entre dos nodos se puede compartir dinámicamente en el tiempo por varios paquetes (TDM asíncrono). Requiere las mismas 3 fases que la conmutación de circuitos (Establecimiento del circuito, transmisión de datos y liberación del circuito).

- La basada en datagramas: No debemos establecer el circuito de forma previa a la transferencia de información. Cada paquete debe llevar la dirección de destino y se trata de forma individualizada, sin establecer ningún vínculo con los demás paquetes que llevan datos de A a B, sean o no de la misma aplicación.

Los usuarios comparten los medios de transmisión por TDM estadístico.

Los retardos son ahora variables, dependientes de la carga instantánea en la red.

Cuando se establece el circuito virtual o cada vez que se transmite un datagrama conmutador debe seleccionar por qué enlace encamina los datos usando un algoritmo de encaminamiento. Esta decisión tiene que ser tomada por cada nodo de la red implicado. Esta decisión debe tomarse minimizando el coste (tiempo, recursos,...) y, con este fin, cada nodo construye una tabla (usando el mencionado algoritmo), llamada tabla de encaminamiento que indica por qué enlace debe transmitir los datos para llegar el destino.

En el caso de los circuitos virtuales (además de la tabla de encaminamiento que se usará para seleccionar la ruta del circuito virtual), el nodo debe construir una tabla con los circuitos virtuales, donde se asigna el identificador de un circuito virtual de un enlace (entrada) a otro enlace (salida).

Un ejemplo de red que usa circuitos virtuales es X.25 y uno de conmutación de paquetes es IP.

## **40.6 TCP/IP**

El alumbramiento del modelo TCP/IP se remonta a la red ARPANET. Esta era una red de investigación controlada por el Departamento de Defensa de EE.UU. Poco a poco fueron conectándose instituciones, mediante el uso de líneas de la red telefónica. La necesidad de buscar una arquitectura de referencia nueva surgió cuando empezaron a añadirse redes de satélite y radio con los consiguientes problemas de interactuar con los protocolos

existentes. Uno de los principales objetivos de esta nueva arquitectura fue la capacidad de conexión de múltiples redes entre sí desembocando en lo que hoy conocemos como modelo TCP/IP.

TCP/IP tiene una mayor aplicación que el modelo OSI, ya que se desarrolló antes y se implantó TCP/IP mientras se esperaba al protocolo OSI. Además, como todas las especificaciones asociadas a los protocolos TCP/IP son de dominio público, y por lo tanto no hay que pagar nada para usarlos, fueron utilizados extensivamente por entidades comerciales y públicas para crear entornos de redes abiertos.

Las ventajas de TCP/IP son:

- Agrupa redes, creando una red mayor llamada Internet.
- Es independiente del hardware de los nodos, del sistema operativo y de la tecnología del medio y del enlace.
- Ofrece capacidad de encaminamiento adaptativo, transparente al usuario.
- Es el software de red más disponible universalmente.

Las diferencias con el modelo OSI son:

- Mientras que en OSI la distinción entre los conceptos de servicio, interfaz y protocolo es clara, en TCP/IP no existía esta distinción inicialmente. Posteriormente, se intentó ajustar esto para acercarse más a OSI.
- OSI se desarrolló antes de que se hubiesen definido los protocolos, mientras que TCP/IP fue, en realidad, el resultado de los protocolos existentes.
- Una diferencia clara es que OSI cuenta con siete capas bien definidas y TCP/IP sólo tiene 4.
- El modelo OSI considera los dos tipos de comunicación, orientada y no orientada a conexión, en la capa de red. Sin embargo, en la capa de transporte ofrece sólo orientada a conexión. Por otro lado, el modelo TCP/IP en la capa de red sólo soporta comunicaciones no orientadas a conexión pero considera ambos modos en la capa de transporte.

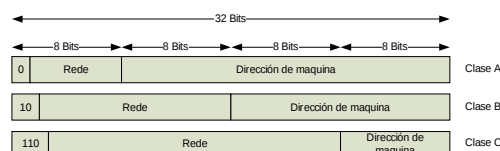
#### **40.6.1 ENDEREZAMIENTO Y SISTEMAS DE NOMBRES DE DOMINIO**

#### 40.6.1.1 ENDEREZAMIENTO IP

Cada nodo de la red tiene una dirección IP única, formada por el número de la red y el número del nodo. La asignación de direcciones IP está regulado por la ICANN (Internet Corporation for Assigned Names and Numbers) para evitar que una misma dirección sea usada por varias maquinas.

Una dirección IP en su versión 4 consta de 32 bits de longitud y generalmente se escribe como concatenación de 4 bytes en formato decimal separados por puntos.

Tradicionalmente las direcciones se agrupan en clases que podemos ver en la siguiente imagen.



Además históricamente existen las clases D (para multicast) y E (originalmente reservada para uso futuro).

Hay determinadas direcciones reservadas, estas son:

- 0.0.0.0 Esta IP se usa por las estaciones cuando aún no tienen una IP asignada.
- La dirección de la red se representa por una dirección IP donde la primera parte es la parte de red y el resto de bits están a 0.
- 127.X.X.X Red loopback (acceso a la propia maquina desde la propia máquina) siendo la dirección predilecta para este fin 127.0.0.1 aunque todas funcionan de la misma forma.
- 255.255.255.255 Es la dirección de broadcast (normalmente no retransmitido por los encaminadores). De forma equivalente la dirección de broadcast para la red y de la dirección de la red con el resto de bits a 1.

Cuando el número de red va todo a ceros se asume que esa dirección se refiere a la red actual.

No todas las direcciones son únicas en la red (estas direcciones son conocidas como IPs públicas) si no que existe en una serie de rangos reservados para su uso en registro:

- Para la clase A: de 10.0.0.0 a 10.255.255.255 (8 bits red, 24 bits estación).
- Para la clase B: 172.16.0.0 a 172.31.255.255 (16 bits red, 16 bits estación). 16 redes clase B contiguas, uso en universidades y grandes compañías.
- Para la clase C: 192.168.0.0 a 192.168.255.255 (24 bits red, 8 bits estación). 256 redes clase C contiguas, uso de compañías medias y pequeñas.

#### **40.6.1.1.1 SUBREDES**

Todos los ordenadores de una red deben tener el mismo número de red. Esta característica puede llegar a ser un problema a medida que crecen las redes. La solución a este problema es la división de una red en varias partes o subredes. Desde el punto de vista del mundo exterior las subredes no son visibles, si no que se ve la red como un todo.

La parte de la dirección que define el número de red permanece igual una vez dividida, pero el número de estación se divide en número de subred y número de estación.

#### **40.6.1.1.2 MÁSCARAS DE RED**

La herramienta que permite obtener la dirección de red de una dirección IP dada es la máscara de red. Es una especie de dirección IP especial que, en binario, tiene todos los bits que definen la red puestos a 1 y los bits correspondientes al host puestos a 0. Así, las máscaras de red de los diferentes tipos de redes principales son:

- Red de clase A: Máscara de red = 255.0.0.0
- Red de clase B: Máscara de red = 255.255.0.0
- Red de clase C: Máscara de red = 255.255.255.0

La máscara de red posee la propiedad de que cuando se combina, mediante una operación AND lógica, con la dirección IP de un host se obtiene la dirección propia de la red en la que se encuentra el mismo. En las redes donde existen definidas subredes se aplica el concepto de máscara de subred, que es el resultado de poner a 1 todos los bits que representan red o subred y a 0 todos los bits que representan una estación.

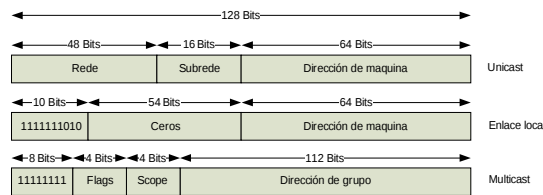
#### **40.6.1.1.3 DIRECCIONES IPV6**

A diferencia de IPv4, que utiliza una dirección IP de 32 bits, las direcciones IPv6 están compuestas de 128 bits, ampliando enormemente la capacidad de direcciones del protocolo IP.

De similar forma que las direcciones IPv4, las direcciones IPv6 se agrupan en 8 números de 4 dígitos hexadecimales separados entre sí por dos puntos (:). Cuando uno de estos números es todo ceros, se puede omitir en la escritura de la dirección.

Existen 3 tipos de direcciones:

- Unicast que identifica un único interfaz de red. El protocolo IP entrega los paquetes enviados a una dirección unicast a la interfaz específica.
- Anycast que es asignada a un grupo de interfaces, normalmente de nodos diferentes. Un paquete enviado a una dirección anycast se entrega sólo a uno de los miembros, normalmente el host con menos coste, según la definición de métrica del protocolo de encaminamiento. Las direcciones anycast no se identifican fácilmente pues tienen el mismo formato que las unicast, diferenciándose sólo por estar presentes en varios puntos de la red. Casi cualquier dirección unicast puede utilizarse como dirección anycast.
- Multicast que también es usada por múltiples hosts, que consiguen la dirección multicast participando del protocolo de multidifusión (multicast) entre los routers de red. Un paquete enviado a una dirección multicast es entregado a todos los interfaces que se unieron al grupo multicast correspondiente.



#### 40.6.1.2 DNS

El sistema de nombres de dominio (DNS, por sus siglas en inglés: Domain Name System) es un sistema de nombres jerárquico y distribuido para equipos, servicios o cualquier tipo de recurso conectado a Internet. Su función más importante es traducir los nombres comprensibles por los usuarios a direcciones IP numéricas con el propósito de localizarlos en la red.

El espacio de nombres está organizado en árbol con cada nodo u hoja en el árbol conteniendo cero o más registros de recursos (siendo estos los que almacenan la información correspondiente al nombre de dominio).

Una zona DNS consiste en uno o varios dominios y subdominios dependiendo de la autoridad delegada en ese gestor. Esta autoridad puede dividirse creando mas zonas (que normalmente se ocuparán de subdominios) cuya autoridad será asumida por otra entidad, perdiendo la original la autoridad sobre estas nuevas zonas.

#### 40.6.2 PROTOCOLOS IP, TCP

##### 40.6.2.1 CAPAS DEL MODELO TCP/IP

La arquitectura del modelo TCP/IP consta de 4 capas:

- Aplicación. Proporciona comunicación entre procesos o aplicaciones en ordenadores distintos. Contiene todos los protocolos de alto nivel.
- Transporte. Al igual que la capa de transporte de OSI, permite que se pueda establecer una comunicación entre las entidades pares de los nodos origen y destino. Proporciona, por tanto, transferencia de datos extremo a extremo, asegurando que los datos llegan en el mismo orden en que fueron enviados y sin errores. Esta capa también puede incluir mecanismos de seguridad. Se puede resumir la funcionalidad de la capa de transporte como calidad de servicio. En este nivel se definen dos protocolos importantes, que se explican en detalle más adelante:





- o TCP: Protocolo orientado a conexión que proporciona entrega fiable de mensajes entre máquinas. Realiza control de flujo para que un emisor rápido no pueda saturar a un receptor lento.
- o UDP: Protocolo sin conexión y no fiable. Se utiliza en aplicaciones donde la entrega rápida es más importante que la entrega precisa, como transmisión de voz y video, y en aplicaciones de consulta de petición y respuesta.
- Red. Esta capa es el eje de la arquitectura y permite que los nodos inyecten paquetes en cualquier red y los hagan viajar de forma independiente a su destino, sin importar si está en la misma red, o si hay otras redes entre ellas. Su misión principal, por tanto, es el encaminamiento de los paquetes, pero sin garantía de que lleguen al extremo final ni de que lo hagan en el mismo orden en el que se enviaron. Si se desea una entrega ordenada, las capas superiores deben reordenar los paquetes. En este nivel se define un formato de paquete y el protocolo IP, que se detalla seguidamente.
- Capa del nodo a la red. El modelo TCP/IP no dice mucho de lo que sucede bajo la capa de red, existe un gran vacío. Esta abstracción de la topología de red pone en relieve la capacidad de la capa de red de soportar cualquier tipo de red por debajo. Lo que está claro es que debe permitir que un nodo se conecte a la red para que pueda enviar por ella paquetes IP. El protocolo que regula esta conexión puede variar de un nodo a otro.

#### **40.6.2.2 PROTOCOLO IP**

IP proporciona un servicio de distribución de paquetes caracterizado por:

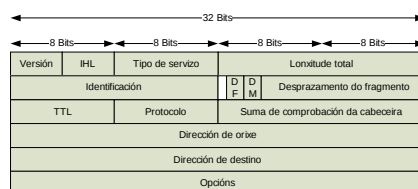
- Transmisión de datos en datagramas (paquetes IP).
- No es orientado a conexión, por lo que los paquetes son tratados de forma independiente y cada uno puede seguir una trayectoria diferente en su viaje hacia el host destino.

- No es fiable, por lo que no garantiza la entrega de los paquetes, ni la entrega en secuencia, ni la entrega única. Esto es responsabilidad del protocolo TCP de la capa superior.
- No implementa control de errores ni control de congestión.
- Puede fragmentar los paquetes si es necesario.
- Direcciona los paquetes empleando direcciones lógicas IP de 32 bits (en IPv4).
- Verifica la integridad del paquete en sí, no de los datos que contiene.

#### 40.6.2.2.1 DATAGRAMA IP

Los datos proporcionados por la capa de transporte son divididos en datagramas y transmitidos a través de la capa de red. A lo largo del camino pueden ser fragmentados en unidades más pequeñas para atravesar una red o subred cuya MTU (Unidad de Transferencia Máxima) sea más pequeña que el paquete. En la máquina destino, estas unidades son reensambladas para volver a tener el datagrama original que es entregado a la capa de transporte.

Un datagrama IP está formado por un cuerpo y una cabecera. El cuerpo se corresponde con el segmento TCP/UDP de la capa de transporte. La cabecera tiene una parte fija de 20 bytes y una parte opcional de longitud variable. En la siguiente figura puede verse la estructura de dicha cabecera.



#### 40.6.2.2 ICMP PROTOCOLO DE CONTROL DE MENSAJES DE INTERNET

Cuando ocurre algún suceso, ICMP (Internet Control Message Protocol) es el protocolo encargado de informar del mismo. No toma ninguna decisión al

respecto, esto es tarea de las capas superiores. Los mensajes de ICMP se encapsulan dentro del campo de datos de los paquetes IP.

El mensaje ICMP tiene tres campos fijos y a continuación el cuerpo del mensaje que varía en función del tipo. Los campos obligatorios son:

- Tipo (8 bits). Se utiliza para distinguir los tipos de mensajes ICMP, descritos más abajo, y determinar su formato.
- Código (8 bits). En algunos mensajes ICMP se utiliza este campo para distinguir distintos subtipos dentro de un tipo de mensaje, es decir, para ofrecer una descripción concreta del error que se produjo.
- Checksum (16 bits). Código de protección contra errores de transmisión.

Existen diversos tipos de mensajes ICMP. Por un lado, están los mensajes informativos. Todos ellos contienen, además de los 3 campos fijos, un identificador de 16 bits y un número de secuencia, también de 16 bits. El resto son mensajes de error. Todos ellos contienen, además de los 3 campos fijos, el encabezado y los 8 primeros bytes del datagrama que ocasionó el error.

#### **40.6.2.3 PROTOCOLO TCP**

TCP (Transmisión Control Protocol) se diseñó específicamente para enviar una secuencia de bytes fiable a través de una red no fiable. Sus características principales son:

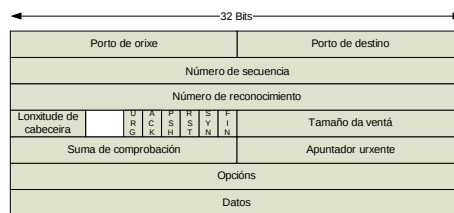
- Es un protocolo orientado la conexión. TCP establece una conexión entre un socket de la máquina transmisora y un socket de la máquina receptora. Un socket es un punto terminal al que se le asigna un número de socket formado por la dirección IP del host y un número de 16 bits local a ese host, llamado puerto. Una vez establecida la conexión se pueden transferir datos entre el origen y el destino. Aunque cada paquete enviado desde el host origen puede viajar por un camino o ruta diferente hasta llegar al host destino, por medio del protocolo IP, TCP consigue que parezca que existe un único circuito de comunicación entre ambos hosts.



- Es un protocolo fiable.
- Es un protocolo de flujo no estructurado, con posibilidad de enviar información de control junto a los datos.
- Es un protocolo con transferencia de memoria intermedia. Con el objeto de minimizar el tráfico de red y conseguir una transferencia eficiente, se van almacenando los datos del flujo de transmisión hasta completar un paquete lo suficientemente largo como para ser enviado. En el destino, se almacenan los datos recibidos hasta completar una secuencia completa y correcta para pasarla al proceso de aplicación destino.
- Usa conexiones full-dúplex, es decir, el tráfico puede ir en ambos sentidos al mismo tiempo.

#### **40.6.2.3.1 SEGMENTO TCP**

Cada segmento comienza con una cabecera de formato fijo de 20 bytes. Esta puede ir seguida de opciones de cabecera. Tras las opciones, si las hay, se encuentran los datos. También puede haber segmentos sin datos, usados normalmente para acuses de recibo y mensajes de control. El formato de la cabecera TCP es la siguiente:



#### **40.6.2.3.2 CONEXIONES TCP**

Para establecer una conexión uno de los lados espera pasivamente una conexión entrante y el otro ejecuta una primitiva de conexión, especificando la dirección y el puerto IP con el que se desea conectar, el tamaño máximo de segmento TCP que está dispuesto a aceptar y, opcionalmente, algunos datos de usuario. Esta primitiva genera un segmento TCP con el bit SYN a 1 y el bit ACK a 0. Al llegar el segmento al destino la entidad TCP revisa si hay algún proceso escuchando en el puerto

indicado en el campo de puerto de destino. Si no lo hay, envía una contestación con el bit RST a 1 para rechazar la conexión. En caso contrario, el proceso recibe el segmento TCP entrante y puede aceptar o rechazar la conexión. Si la acepta, se envía de vuelta un segmento de acuse de recibo.

Cuando se cae un host, por seguridad, no puede reiniciarse durante el tiempo máximo de paquete (120 seg.) para asegurar que no haya paquetes de conexiones previas vagando por Internet.

Para liberar una conexión, cualquiera de las partes puede enviar un segmento TCP con el bit FIN activado, indicando que no tiene más datos que transmitir. Al reconocerse el FIN, ese sentido apagará. Sin embargo, el flujo de datos en el otro sentido puede continuar. Cuando ambos sentidos se apagan se libera la conexión. En algunas implementaciones de TCP existe un temporizador de seguir con vida (keepalive timer). Cuando una conexión está ociosa durante demasiado tiempo este contador puede agotarse. Si esto ocurre, un lado de la conexión comprueba si el otro aún responde. Si no se recibe respuesta se termina la conexión.

#### **40.6.2.3.3 DETECCIÓN DE ERRORES**

Las técnicas más efectivas y utilizadas son las siguientes:

- Detección de errores. Consiste en añadir uno o más bits de información a cada segmento de forma que indiquen claramente si se alteró alguno de los bits del mismo en el camino desde emisor al receptor (Pardidad, CheckSum, CRC,...).
- Confirmaciones positivas. El receptor devuelve un acuse de recibo positivo por cada uno de los segmentos recibidos correctamente. Se usa para detectar y solicitar el reenvío de segmentos perdidos. Esta es la técnica que se utiliza en el sistema de parada y espera.
- Expiración de intervalos de tiempo. El emisor inicia un contador de tiempo tras enviar un segmento (el temporizador de retransmisión). Si este contador se agotara sin que se reciba un ACK positivo el emisor vuelve a transmitir el mismo segmento.

- Confirmación negativa y transmisión. El receptor sólo confirma los segmentos recibidos erróneamente para que el emisor los vuelva a enviar. Por tanto, se utiliza para solicitar el reenvío de segmentos dañados.

#### **40.6.2.3.4 CONTROL DE FLUJO**

El control de flujo más simple es lo que se lleva a cabo mediante el sistema de parada y espera. El transmisor guarda un registro de cada segmento que envía, esperando un ACK antes de enviar el siguiente. También arranca un temporizador cuando envía el segmento. Si el temporizador expira antes de recibir el acuse de recibo, retransmite el segmento y reinicia el temporizador. Este mecanismo es el más barato y el más usado cuando se transmiten tramas muy grandes pero es ineficiente ya que está el canal de transmisión desaprovechado la mayor parte del tiempo.

El control de flujo mediante ventana deslizante permite que el transmisor envíe varios segmentos sin esperar los ACK correspondientes. En este sistema el emisor y el receptor se ponen de acuerdo en el número de segmentos sin procesar que puede guardar este último, dependiendo del tamaño de sus buffers. También se ponen de acuerdo en el número de bits a utilizar para numerar cada segmento. Cuando la ventana tiene un tamaño cero el emisor no puede enviar más segmentos, salvo en dos casos excepcionales: cuando se trata de datos urgentes y cuando el emisor envía un segmento de 1 byte para provocar que el receptor genere un nuevo acuse de recibo con un nuevo tamaño de ventana, evitando así uno bloqueo indefinido de la conexión.

Una variedad mejorada del sistema de ventana deslizante es el sistema de control de flujo con adelante-atrás-N, en el que cuando la estación destino encuentra un segmento erróneo devuelve un ACK negativo, rechazando todos los que le lleguen hasta que no reciba otra vez el segmento incorrecto en buenas condiciones. El emisor, al recibir el ACK negativo, sabe que tiene que volver a transmitir ese segmento y todos los siguientes.

Por último, existe otro sistema denominado sistema de control con rechazo selectivo, que se basa en que los únicos segmentos que se vuelven a retransmitir son aquellos rechazados por el receptor o aquellos cuyo temporizador expira sin confirmación. Este método es más eficiente que los anteriores pero precisa que el receptor disponga de un buffer intermedio de gran capacidad en el que guardar todos los segmentos recibidos tras el rechazo de un dado hasta recibir de nuevo el segmento.

#### **40.6.2.3.5 CONTROL DE CONGESTIÓN**

Cuando la carga ofrecida a la red es mayor que la que puede gestionar se produce congestión. Todos los algoritmos TCP suponen que las terminaciones de temporización son causadas por congestiones y las revisan en busca de problemas.

Cada transmisor mantiene dos ventanas diferentes:

- Ventana negociada con el receptor al establecerse la conexión, cuyo tamaño está basado en el tamaño del buffer de memoria de destino. Esto permite que el transmisor no envíe más datos de los que el receptor puede almacenar evitando así que lo sature.
- Ventana de congestión, determinada por el tamaño de los datos que se pueden enviar sin que se produzca timeout.

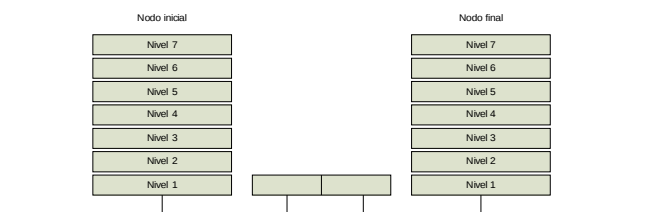
El transmisor sólo puede mandar un número de segmentos limitado por el tamaño de la ventana más pequeña. Al establecerse una conexión, el transmisor asigna a la ventana de congestión el tamaño de segmento máximo usado por la conexión. Cada envío reconocido con éxito duplica la ventana de congestión. Este algoritmo se llama arranque lento (slow start) y permite que el tamaño de la ventana de congestión crezca exponencialmente hasta que se produzca una terminación de temporización (timeout) o se alcance el tamaño de la ventana receptora. Este crecimiento exponencial puede producir saturación. Para evitarlo, se introduce otro parámetro, denominado umbral, que toma como valor inicial 64 KBytes. Cuando se produce el timeout se cambia el valor del umbral a la mitad del tamaño de la ventana de congestión, se establece el valor de la

ventana al del tamaño de un segmento máximo y se inicializa otra vez el proceso de arranque lento. Ahora, cuando el tamaño de la ventana llega al del umbral esta crece solamente en saltos de un segmento máximo, es decir, con un progreso lineal hasta que se produzca una nueva terminación de temporización.

## 40.7 ELEMENTOS DE INTERCONEXIÓN DE RED

### 40.7.1 REPETIDORES

El repetidor es un elemento que permite la conexión de dos tramos de red y que tiene como función principal regenerar la señal para permitir alcanzar distancias mayores. Normalmente el repetidor recibe la señal desde uno de los segmentos, lo amplifica y lo emite en el otro segmento.

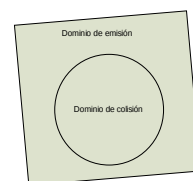


Sus características principales son:

- Es la forma más simple y barata de conectar segmentos de red.
- Se utilizan para superar limitaciones de distancia.
- Sólo valen para conectar topologías de red compatibles.
- No aíslan tráfico ni segmentan la red.

### 40.7.2 CONCENTRADORES

Un concentrador es un dispositivo que funciona como centro de cableado para una red con topología en estrella. Su función consiste en que el tráfico que llega a cualquiera de sus puertos se propague a través de los demás puertos. Esto crea un medio de red compartido y reúne a las computadoras conectadas a la red en un único dominio de colisión y de difusión, de la misma forma que si estuviesen conectadas a uno único cable.





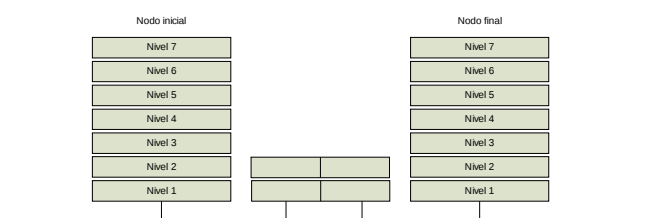
Como implicación directa de esto último tenemos que la velocidad de transmisión entre todos los nodos conectados a un concentrador es la misma que entre dos máquinas conectadas por un cable.

Los concentradores se pueden apilar o interconectar entre ellos, funcionando como un único concentrador con más puertos, con las mismas ventajas y limitaciones.

### 40.7.3 PUENTES

Un puente es un dispositivo utilizado para conectar segmentos de redes. Opera en el nivel de enlace de datos y es selectivo respecto a los paquetes que pasan a través de él. Frente a los repetidores que trabajan solo con señales, los puentes trabajan con tramas.

Un puente no transmite datos a los segmentos conectados hasta que llega toda la trama. Por este motivo, dos sistemas que se encuentran en segmentos separados por un puente pueden transmitir a la vez sin que se produzca una colisión. Un puente conecta segmentos de red de tal forma que mantiene en el mismo dominio de difusión pero en distintos dominios de colisión.



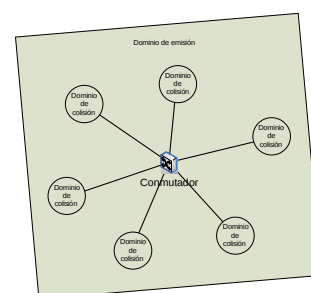
Sus características principales son:

- Pueden aislar el tráfico basándose en la dirección MAC.
- Al igual que los repetidores, los puentes no son direccionables en la red (transparentes para niveles superiores).
- Sólo operan en el nivel MAC de enlace (conectan segmentos de la misma red).
- Extienden la topología de la red (i.e. Anillo-bus)
- Aíslan errores MAC (i.e. Tramas demasiado largas)

- Existen dos tipos:
  - o Transparentes: Conectan topologías de red compatibles (i.e. 10BaseT-10Base2) y no modifican ninguna parte de la trama.
  - o De traslación: Conectan diferentes topologías de red (i.e. 10BaseT-Token Ring) adaptando la trama al protocolo MAC destino.

#### **40.7.4 CONMUTADORES**

Un conmutador opera en el nivel de enlace de datos y es en esencia un puente multipuerto en el que cada uno de los puertos es un segmento de red independiente. Un conmutador recibe tráfico por sus puertos y al contrario que un concentrador, el cual reenvía el tráfico a través de todos los demás puertos, sólo lo reenvía por el puerto necesario para alcanzar su destino.



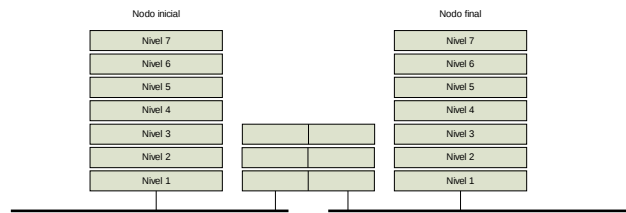
Un sistema conectado a un conmutador posee el equivalente a una conexión dedicada con cada uno de los sistemas restantes conectados al conmutador y con todo el ancho de banda.

Sus principales características son:

- Diseñados para solucionar problemas de rendimiento de LAN (escaseo de ancho de banda, cuellos de botella en la red).
- Alto rendimiento en el envío de paquetes y baja latencia.
- Segmentan un dominio de colisión en otros más pequeños.
- Reduce o casi elimina la contienda por el acceso al medio.

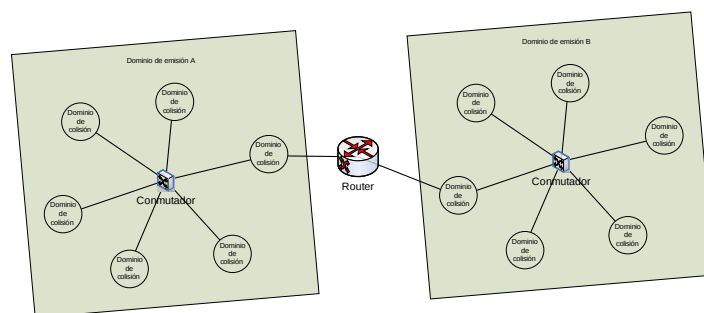
#### **40.7.5 ENCAMINADORES**

La labor de un encaminador es la de conectar dos redes completamente independientes en el nivel de red. Los encaminadores son más selectivos que los puentes en el tráfico que pasa entre las redes y son capaces de seleccionar de forma inteligente a ruta más eficiente hacia un destino específico.



Las funciones básicas de un encaminador son:

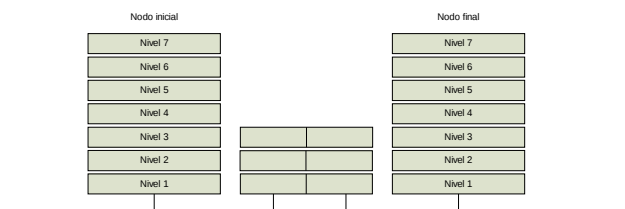
- Segmentar la red en dominios individuales de envío (redes aisladas a nivel MAC)
- Proporcionan envío inteligente de paquetes: analizan el tráfico y para cada paquete seleccionan la red que proporciona la mejor ruta hacia el destino. Un paquete puede pasar por varios encaminadores en su camino hacia el destino, cada uno de ellos se conoce como salto. El objetivo suele ser que llegue con el menor número de saltos. Para ello utilizan las llamadas tablas de encaminamiento.
- Proporcionan acceso a las WAN de manera eficiente.
- Soportan caminos redundantes (tolerancia a fallos).
- Proporcionan seguridad / firewall: analizan todo paquete que llega de una de las redes a la que está conectado. Si la dirección de origen y de destino pertenecen a la misma red lo descartan, si no lo reenvían a su destino a través de otra red.



#### **40.7.6 PASARELAS**

Una pasarela conecta dos redes distintas que usan protocolos y arquitecturas distintos a todos los niveles. Su función es traducir el protocolo de una red en el protocolo de la otra, pero también pueden conectar redes que usen el mismo protocolo. En este último caso entran a las que traducen IP la IP, como por ejemplo haciendo NAT (Network Address

Translation, que convierte una dirección IP de una red –normalmente una IP personal de una LAN- en otra dirección IP –normalmente en una IP pública- siendo capaz de invertir el proceso).



## **40.8 GESTIÓN DE REDES**

### **40.8.1 MODELO OSI DE GESTIÓN DE RED**

ISO, siguiendo las directrices del grupo OSI, definió el modelo de gestión de red como la forma más importante para entender las funciones principales de los sistemas de gestión de red.

El modelo OSI de gestión de red categoriza las funciones en 5 áreas que a veces se denominan modelo FCAPS (Fault, Configuration, Accounting, Performance y Security):

- **Falla (Fault):** Siendo el objetivo de esta área detectar, aislar, corregir y registrar las fallas que se produzcan en la red.
- **Configuración (Configuration):** Los objetivos de esta área son recoger/fijar/hacer seguimiento de la configuración de los dispositivos. La gestión de la configuración se ocupa de monitorizar la información de configuración del sistema y cualquier cambio que se produzcan en la misma. La importancia de esta área viene dada por que muchos incidentes en la red son el resultado de cambio sin archivos de configuración, actualización de versiones, etc. Una adecuada gestión de la configuración obliga a registrar todos los cambios en la configuración software y hardware.
- **Contabilidad (Accounting):** Siendo el objetivo principal recoger estadísticas. La gestión de la contabilidad se preocupa por mantener la

información referida a la utilización de la red, de forma que se pueda facturar a usuarios individuales, departamentos, etc.

- **Rendimiento (Performance):** El objetivo de esta área es doble: por un lado preparar la red para el futuro y por otro medir la eficiencia actual de la misma asegurándose que está dentro de los niveles aceptables. La gestión del rendimiento se preocupa de recoger regularmente la información de rendimiento de la red como son los tiempos de respuesta, ratios de pérdida de paquetes, utilización de enlaces de datos, etc.
- **Seguridad (Security):** El objetivo de la gestión de la seguridad es controlar el acceso a los recursos de la red. Esta área no solo se preocupa de que la red sea segura y no de recabar y analizar la información referida a la seguridad. A las funciones típicas dentro de esta área son la autenticación, autorización, auditoría, de forma que los usuarios (tanto internos como externos) tenga el acceso adecuado a los recursos de la red.

#### **40.8.2 SNMP**

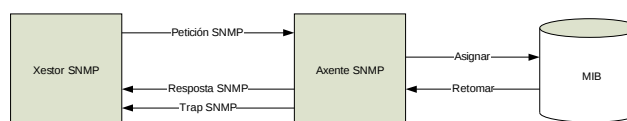
SNMP (Simple Network Management Protocol) es el protocolo definido por los comités técnicos de Internet para ser utilizado como herramienta de administración de los distintos dispositivos en cualquier red. El funcionamiento de SNMP es sencillo, como su propio nombre indica, aunque su implementación puede llegar a ser tremendamente compleja. SNMP utiliza la capa de transporte de TCP/IP mediante el envío de datagramas UDP (los agentes escuchan en el puerto 161 y las estaciones gestoras en el 162). Sin embargo, el hecho de usar UDP hace que el protocolo no sea fiable (en UDP no se garantiza la recepción de los paquetes enviados, como en TCP).

El protocolo SNMP está definido en un gran número de RFCs (Request For Comments), entre ellos el RFC 1157, 1215 (que definen la versión 1), del 1441 al 1452 (que definen la versión 2), del 2271 al 2275 y del 2570 al 2575 (para SNMP v3).

#### 40.8.2.1 FUNCIONAMIENTO DE SNMP

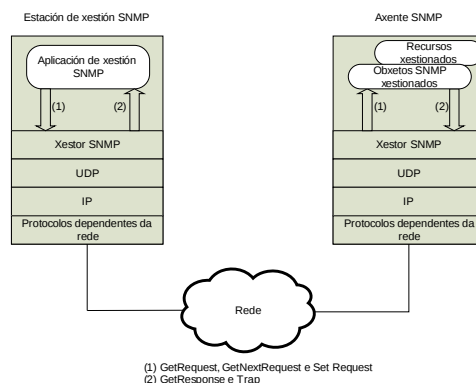
Cada agente (se puede ver a un agente como una máquina en la que queremos monitorizar alguno de sus estados) ofrece una determinada serie de variables, que pueden ser leídas o modificadas. Además, un agente puede enviar “alarmas” (Traps) a otros agentes para avisar de eventos que tienen lugar. Lo normal es que el agente encargado de recibir los eventos se denomine “gestor” (podemos ver a este cómo a la máquina que monitoriza el estado de toda la red). De forma muy resumida podemos ver las capacidades expuestas:

- GET : La estación gestora extrae (lee) el valor de un objeto del agente
- SET : La estación gestora fija (escribe) el valor de un objeto del agente
- TRAP : Permite a un agente notificar a la estación gestora eventos significativos



Las variables ofrecidas para la consulta en los agentes SNMP se definen a través de una MIB (Management Information Base, Base de Información de Gestión). La MIB es una forma de determinar la información que ofrece un dispositivo SNMP y la forma en que se representa. La versión de la MIB actual es MIB-II y está definida en el RFC 1213, aunque hay múltiples extensiones definidas en otros RFCs. La MIB está descrita en ASN.1 para facilitar su transporte transparente por la capa de red.

Cada agente SNMP ofrece información dentro de una MIB, tanto de la estándar (definida en los distintos RFCs) como de aquellas extensiones que desee proporcionar cada uno de los fabricantes.



ASN.1 (Abstract Syntax Notation One) es un estándar de nota que describe la representación, la transmisión, la codificación y decodificación de estructuras de datos. Proporciona un conjunto de reglas formales para describir la estructura de objetos que son independientes de las técnicas de codificación de una máquina, aportando una nota formal que elimina las ambigüedades.

#### **40.8.2.2 ESPECIFICACIONES TÉCNICAS SNMP MÍNIMAS REQUERIDAS**

Existen diversas RFCs que definen SNMP. Por ello es importante establecer unos requisitos o especificaciones mínimas. Estas especificaciones mínimas son:

- Versión del protocolo SNMPv2c (Community-based SNMPv2 - RFC 1901)  
Utiliza el mismo modelo que la primera versión del protocolo SNMP, y como tal no incluye mecanismos de seguridad. Los únicos avances introducidos en esta versión consisten en una mayor flexibilidad de los mecanismos de control de acceso, ya que se permite la definición de políticas de acceso consistentes en asociar un nombre de comunidad con un perfil de comunidad formado por una vista MIB y unos derechos de acceso a dicha vista (solo lectura o lectura y escritura).
- La MIB deberá ser compatible con el formato ASN.1. Las implementaciones de otros estándares de la MIB son opcionales. ASN.1 está diseñado para definir información estructurada (mensajes) de tal forma que sea independiente de la máquina utilizada. Para hacer esto ASN.1 define tipos de datos básicos, como enteros y cadenas de texto, y permite construir nuevos tipos de datos a partir de los ya definidos.

También utiliza palabras especiales (keywords) para definir sus procedimientos, definir nuevos tipos, asignar valores, definir macros y módulos.

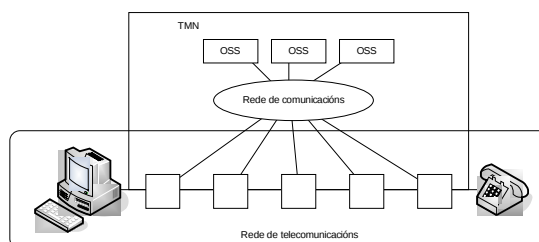
- El acceso múltiple deberá ser permitido, existiendo 3 niveles de acceso con sus correspondientes “login” y “password”.
- El requerimiento mínimo respecto a la seguridad, es la generación de “Traps” en el caso de una autenticación fallida. La información relevante del infractor que deberá ser enviada en el TRAP, será la dirección IP.

#### **40.8.3 TMN**

TMN (Telecommunications Management Network) define un marco de trabajo para alcanzar la interoperabilidad y comunicación entre redes de comunicaciones y sistemas operativos heterogéneos. TMN fue desarrollado por ITU como una infraestructura para soportar la gestión y despliegue de servicios dinámicos de telecomunicaciones.

TMN proporciona un framework flexible, escalable, fiable, barato y fácil de mejorar. TMN permite crear redes más capaces y eficientes definiendo una forma estándar de realizar las tareas de gestión de la red y comunicaciones. El procesamiento en TMN puede ser distribuido para mejorar la escalabilidad.

Una red de telecomunicación está compuesta de elementos de conmutación, circuitos, terminales, etc. En terminología TMN todos estos elementos son Elementos de la Red (NE Network Elements). TMN permite la comunicación entre los NEs y OSS (Operations Support Systems).



TMN usa principios de orientación a objetos e interfaces estándar para definir las comunicaciones entre diferentes entidades en la red. La interfaz de gestión estándar se llama Q3. TMN se basa en estándares OSI como



CMIP (Common Management Information Protocol), GDMO (Guideline for definition of management objects), ASN.1 y el modelo de referencia OSI. Las funciones de gestión se realizan a través de operaciones compuestas de primitivas CMIS (Common Management Information Service).

La información de gestión de la red, así como las reglas por las que la información se presenta y gestiona, se llaman MIB (Management Information Database). Los procesos que gestionan la información se llaman entidades que pueden ser de dos tipos: gestor o agente.

TMN describe las redes de telecomunicación desde distintos puntos de vista:

- Modelo funcional: representado por bloques que aportan una visión general de las funciones y características de TMN.
  - o OS (Operation System): realiza funciones de operación del sistema incluyendo monitorización y control de las funciones de gestión de telecomunicaciones.
  - o MD (Mediation Device): Realiza funciones de mediación entre los interfaces locales TMN y el modelo de información de los OS.
  - o QA (Q-Adapters): Permite a TMN gestionar NEs que no tienen interfaces TMN.
  - o NE (Network Entity): Contiene información gestionable que es monitorizada y controlada por los OS.
  - o WS (Workstations): Traducen la información entre formato TMN y un formato comprensible por el usuario.
  - o DCN (Data Communication Network): Representa la red de comunicaciones cubriendo los niveles 1 a 3 de OSI.
- Conjunto de interfaces:
  - o Q: Los interfaces Q existen entre dos bloques funcionales TMN que pertenecen al mismo dominio.
    - Qx: existe entre los NE y los MD, QA y MD y entre los MD y otro MD.



- Q3: es la interfaz del OS y existe entre los NEs y OS, QA y OS, MD y OS y entre OS y otro OS.
- o F: Las interfaces F existen entre los WS y OS y entre los WS y MD.
- o X: Estas interfaces existen entre dos OS TMN de diferentes dominios o entre un OS TMN y otro OS en una red no TMN.
- Modelo lógico o de negocio: Este modelo está basado en capas de distintos niveles jerárquicos:
  - o BML (Business Management Layer): Planificación de alto nivel, presupuestos, BLAs (Business Level Agreements), etc.
  - o SML (Service Management Layer): Usa la información presentada por la capa NML para gestionar los servicios contratados por clientes actuales o potenciales. También es el punto clave de contacto con proveedores de servicio y otras entidades administrativas.
  - o NML (Network Management Layer): La NML tiene visibilidad de toda la red basada en la información de los OSs de la capa EML. NML permite gestionar los NEs de forma individual o como un grupo.
  - o EML (Element Management Layer): Gestiona cada elemento de la red conteniendo OSs, cada uno de los cuales gestiona ciertos NEs. También contiene a los MDs.
  - o NEL (Network Element Layer): Representa la información gestionable por TMN en un NE. OS QA y los NE están localizados en esta capa.

## 40.9 BIBLIOGRAFÍA

- Andrew S. Tanenbaum. Redes de computadoras. PRENTICE HALL, 1997

- ISO 7498:1984 - Information processing systems - Open Systems Interconnection
- Groth, David; Toby Skandier (2005). Network Study Guide

**Autor:** Matías Villanueva Sampayo

Director de Informática Asociación Provincial de Pensionistas y Jubilados de A Coruña.

Colegiado del CPEIG

## **41. TECNOLOGÍAS DE ACCESO: REDES TELEFÓNICAS (RDSI, XDSL), REDES DE TELEFONÍA MÓVIL, CABLE, PLC, REDES RADIO (LMDS, WIMAX), SATÉLITE, LÍNEAS PUNTO A PUNTO, METROETHERNET.**

**Tema 41. Tecnologías de acceso: redes telefónicas (RDSI, xDSL), redes de telefonía móvil, cable, PLC, redes radio (LMDS, Wimax), satélite, líneas punto a punto, MetroEthernet.**

**41.1 Redes telefónicas (RDSI, xDSL)**

**41.1.1 RDSI**

**41.1.1.1 Ventajas de RDSI**

**41.1.1.2 RDSI de banda estrecha**

**41.1.1.3 RDSI de banda ancha**

**41.1.1.3.1 Servicios RDSI-BA**

**41.1.2 XDSL**

**41.1.2.1 ADSL**

**41.1.2.1.1 Arquitectura ADSL**

**41.1.2.1.2 Nivel físico**

**41.1.2.2 HDSL**

**41.1.2.3 SDSL**

**41.1.2.4 VDSL**

**41.2 Redes de telefonía móvil**

**41.2.1 GSM**

**41.2.1.1 Arquitectura de la red GSM**

**41.2.2 GPRS, HSCSD**

**41.2.2.1 GPRS**

**41.2.2.1.1 Arquitectura de GPRS**

**41.2.2.1.2 EDGE o E-GPRS**

**41.2.2.2 HSCSD**

**41.2.3 Sistemas de tercera generación: UMTS**

**41.2.3.1 HSPA**

**41.3 Cable**

**41.4 PLC**

**41.5 Redes radio (LMDS, Wimax)**

**41.5.1 LMDS**

**41.5.2 WIMAX**

#### 41.6 Satélite

#### 41.7 Líneas punto a punto

##### 41.7.1 X.25

##### 41.7.2 FrameRelay

##### 41.7.3 MetroEthernet

#### 41.8 MetroEthernet

##### 41.8.1 MAN basada en Ethernet

##### 41.8.2 MAN basada en SDH

##### 41.8.3 MAN basada en MPLS

#### 41.9 Bibliografía

### **41.1 REDES TELEFÓNICAS (RDSI, XDSL)**

Originalmente la única red pública disponible era la Red Telefónica Conmutada (RTC) que estaba compuesta por elementos analógicos siendo su principal objetivo el transporte de la voz que se transmitía por líneas modulada como una forma de onda analógica.

Para poder transmitir datos sobre esta red se necesitaba convertir la señal digital en una señal analógica en origen y volver a convertirla en digital en destino. Dado que las líneas de voz se pensaron solo para transmitir voz usaban conmutación de circuitos. Además, debido a que no fueron diseñadas para garantizar la transmisión sin pérdida, eran los protocolos de transmisión de datos los que debían garantizar la corrección de errores, reconexión, etc.

Posteriormente, para solucionar el problema de la pérdida de calidad del sonido en las llamadas a larga distancia aparecieron las centrales digitales, menos propensas a fallos, y permitieron controlar más líneas de usuario y realizar las conexiones mucho más rápido. De esta forma, una comunicación por una línea telefónica convencional se realiza de forma analógica en el bucle de abonado, pero de forma digital hasta llegar a la central donde está conectado el abonado destino. La RDSI (Red Digital de Servicios Integrados) suponen el último avance: la comunicación digital entre el abonado y la central telefónica.

#### **41.1.1.1 RDSI**

RDSI es una red desarrollada a partir de la red telefónica que proporciona una conexión digital extremo a extremo y que soporta una gran variedad de servicios.

Se denomina “Digital” porque se basa en técnicas digitales, garantizando la integridad de la información y la transmisión de la misma libre de degradaciones o perturbaciones externas; y es “de Servicios Integrados” porque utiliza la misma infraestructura para muchos servicios que tradicionalmente requerían interfaces distintas (télex, voz, conmutación de circuitos, conmutación de paquetes, etc.).

Las características más importantes son:

- Su arquitectura está estratificada en niveles: físico, enlace y red.
- Proporciona conexiones de 64Kbps.
- La señalización va por un canal diferente a la información propiamente dicha. En ciertas ocasiones se utiliza el canal de señalización para enviar información aunque a una velocidad más baja.
- Soporta una gran variedad de aplicaciones, independientemente de si están basadas en conmutación de circuitos o de paquetes.

##### **41.1.1.1.1 VENTAJAS DE RDSI**

Entre las ventajas que ofrece RDSI se pueden destacar:

- Velocidad. Ofrece múltiples canales digitales que pueden operar simultáneamente a través de la misma conexión telefónica entre central y el usuario. Usando un protocolo de agregación de canales se puede alcanzar una velocidad de datos sin comprimir de unos 128 Kbps, en el servicio de acceso básico. Este esquema permite una transferencia de datos a una velocidad mucho mayor que la línea telefónica. Además, el tiempo necesario para establecer una comunicación en RDSI es aproximadamente la mitad del tiempo empleado con una línea analógica.

- Conexión de múltiples dispositivos. Es posible combinar diferentes fuentes de datos digitales y hacer que la información llegue al destino correcto. Como la línea es digital, es fácil controlar el ruido y las interferencias producidas al combinar las señales.
- Señalización. En una conexión RDSI la llamada se establece enviando un paquete de datos especial a través de un canal independiente de los canales para datos. Permite establecer la llamada en un par de segundos.
- Servicios. La RDSI no se limita a ofrecer comunicaciones de voz. Ofrece otros muchos servicios como transmisión de datos informáticos (servicios portadores), télex, facsímil, videoconferencia (usando, por ejemplo, H.320), conexión a Internet y opciones como llamada en espera, identidad del origen, etc.

En función del ancho de banda se distinguen entre RDSI de “banda estrecha” que permite velocidades de 64Kbps o agrupaciones de esta velocidad hasta 1984Kbps y RDSI de banda ancha donde la velocidad mínima a la que se trabaja es 2Mbps, pudiendo llegar hasta los 100Mbps.

#### **41.1.1.2 RDSI DE BANDA ESTRECHA**

La RDSI dispone de tres tipos de canales:

- Canal B. Los canales tipo B transmiten información en modo circuito o en modo paquete a 64Kbps y se emplean para transportar cualquier tipo de información de usuario, bien sea voz o datos.
- Canal D. Se utiliza principalmente para enviar información de control, como es el caso de los datos necesarios para establecer una llamada o para liberarla. Estos canales trabajan a 16Kbps o 64kbps según el tipo de servicio contratado.
- Canales H. Combinando varios canales B se obtienen canales tipo H, que también son canales para transportar solo información de usuario pero a velocidades mucho mayores. Hay varios tipos de canales H:
  - Canales H0, que trabajan a 384Kbps (6 canales B).



- Canales H10, que trabajan a 1472Kbps (23 canales B).
- Canales H11, que trabajan a 1536Kbps (24 canales B).
- Canales H12, que trabajan a 1920Kbps (30 canales B).

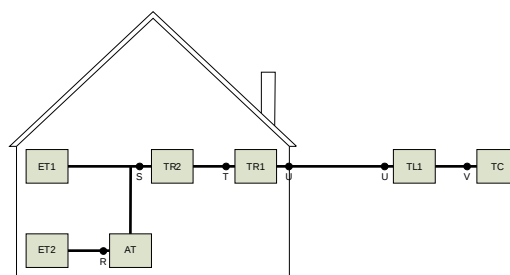
Un usuario puede contratar dos tipos de servicio diferentes con el proveedor telefónico según sus necesidades:

- Acceso básico o BRI (Basic Rate Interface). Proporciona dos canales B y un canal D.
- Acceso primario o PRI (Primary Rate Interface). En Europa el PRI consta de 30 canales B y un canal D. En este caso, los canales B también pueden estar agrupados como 5 canales H0 o un canal H12.

La RDSI ofrece la capacidad de agregar canales para realizar conexiones a mayor velocidad.

En un acceso básico una llamada a 128Kbps son en realidad dos llamadas diferentes a 64Kbps cada una, existiendo un protocolo por encima que permite ver esa llamada como una sola. Muchos fabricantes de hardware para RDSI permiten la agregación de canales utilizando protocolos propios. Para garantizar la compatibilidad entre equipos de diversos fabricantes es conveniente que el hardware soporte el protocolo MPPP (Multilink Point to Point Protocol).

La configuración de referencia se define por agrupaciones funcionales y puntos de referencia o interfaces, como se muestra en la figura:



Las agrupaciones funcionales son:

- TC (Terminación de Central). Situada en la central de conmutación, se encarga del mantenimiento del Acceso del Usuario.
- TL (Terminación de Línea). Situada en la central, se encarga de los aspectos de transmisión.

- TR1 (Terminación de Red nº 1). Dispositivo frontera que separa las instalaciones de usuario de las de la red y convierte los dos hilos de la interfaz U en los cuatro hilos empleados en una interfaz T o S/T. Siempre lo proporciona el proveedor del servicio. En general, realiza funciones del nivel físico.
- TR2 (Terminación de Red nº 2). Convierte la interfaz T en una interfaz S. Hace referencia a una centralita o PABX (Private Automatic Branch Exchange). En el acceso básico el TR2 no existe, con lo que el punto de referencia S y el T coinciden, pasándose a llamar este punto S/T.
- ET1 (Equipo Terminal nº 1). Es un terminal específico para RDSI, preparado para la señalización en modo paquete y gestión de canales de información.
- AT (Adaptador de Terminal). Se trata de un equipo RDSI que tiene la capacidad de adaptar interfaces. Convierte las señales de otros equipos no RDSI a señales adecuadas a la interfaz correspondiente, S y/o T.
- ET2 (Equipo Terminal nº 2). Equipos no RDSI que pueden conectarse mediante un AT al bus RDSI.

Los Puntos de Referencia o interfaces son:

- V. Representa la separación entre las funciones de conmutación y transmisión en la central.
- U. En un acceso básico está formado por la línea típica de un par trenzado de hilos procedente de la red telefónica. En un acceso primario está formado por una línea de cable coaxial o fibra óptica que se suele conectar directamente a una central local de distribución o PABX que actúa como TR2.
- T. Representa la separación entre la transmisión de línea y la transmisión en el domicilio del cliente. Consta de cuatro hilos, dos para recibir y dos para enviar datos, permitiendo también una conexión full dúplex.

- S. Representa la interfaz de conexión física de los equipos terminales RDSI y define la estructura de la trama, la gestión del Canal D, la sincronización y las características de transmisión.
- R. Representa una interfaz no normalizada en RDSI.

La RDSI se estructura en tres capas: física, de enlace y red:

- Física. Las funciones más destacadas de este nivel son la codificación de datos digitales para la transmisión a través de la interfaz correspondiente, transmisión full dúplex, formación de la trama, activación y desactivación del circuito físico, etc.
- Enlace. Este nivel emplea principalmente el protocolo LAP-D (Link Access Protocol o protocolo de acceso al enlace). Proporciona al nivel superior un servicio orientado a conexión con transferencia de información confirmada, servicio sin conexión con transferencia de información no confirmada y servicios de administración, que permiten identificar los equipos específicos dentro del bus S/T asociado a una conexión RDSI.
- Red. En esta capa, la Recomendación Q.931 especifica los procedimientos para establecer, mantener y liberar las conexiones en la interfaz usuario-red. Se manejan distintos tipos de mensajes: de establecimiento de llamada, durante la fase de transmisión de información, de liberación de la llamada y otros.

#### **41.1.1.3 RDSI DE BANDA ANCHA**

La RDSI de banda ancha representa un término medio entre la conmutación de circuitos pura y la conmutación de paquetes pura. El servicio ofrecido está orientado a conexión, pero internamente se implementa con conmutación de paquetes. La RDSI-BA está basada en la tecnología ATM. Las razones que llevaron a elegir esta tecnología fueron, entre otras, que permite manejar tanto tráfico de velocidad constante como variable y que, a velocidades altas, la conmutación digital de celdas es más fácil que las técnicas tradicionales de multiplexación. Las velocidades conseguibles por de RDSI-BA dependen de las tecnologías

concretas usadas en la red y van desde un mínimo de 2 Mbps hasta los 155 o 622Mbps.

En ATM, el flujo de información se organiza en bloques de tamaño fijo y pequeño (53 bytes), llamados celdas. No se garantiza la entrega de todas las celdas, pero las que llegan lo hacen en orden. El modelo ATM también se divide en capas:

- Capa física. Tiene que ver con el medio físico. Se divide en dos subcapas: PDM (Physical Medium Dependent) y TC (Transmisión Convergence).
- Capa ATM. Tiene que ver con las celdas y su transporte.
- Capa de adaptación de ATM (AAL, ATM Adaptation Layer). Permite a los usuarios enviar paquetes mayores que una celda. Se divide en dos subcapas: SAR (Segmentation And Reassembly) y CS (Convergence Sublayer).

ATM se tratará en otro tema más en detalle.

#### **41.1.1.3.1 SERVICIOS RDSI-BA**

En la RDSI de Banda Ancha se pueden incorporar distintos tipos de servicios que podemos clasificar en servicios interactivos y servicios de distribución.

Dentro de los servicios interactivos podemos encontrar:

- Servicios conversacionales, como pueden ser:
  - o Videoconferencia
  - o Videovigilancia
  - o Fax de alta velocidad
  - o Transferencia de documentos
- Servicios de mensajería:
  - o Video-mail
  - o Correo con contenido multimedia
- Servicios de consulta:
  - o Videotex
  - o Recuperación de datos, documentos, etc.

Ejemplos de servicios de distribución podrían ser:

- Servicios sin control de presentación:
  - o Televisión
  - o Televisión a la carta
  - o Distribución de documentos
  - o Video bajo demanda
  - o ...
- Servicios con control de presentación:
  - o Video

#### **41.1.2 XDSL**

Una Digital Subscriber Line (DSL) es el nombre que identifica a todos los estándares digitales sobre bucle de abonado, un ejemplo es la RDSI. Por su parte xDSL identifica un conjunto de estándares para bucle de abonado sobre hilo de cobre como son (entre otras):

- ADSL (Asymmetrical Digital Subscriber Line)
- SDSL (Symmetrical Digital Subscriber Line)
- HDSL (High data rate Digital Subscriber Line)
- VDSL (Very high rate Digital Subscriber Line)

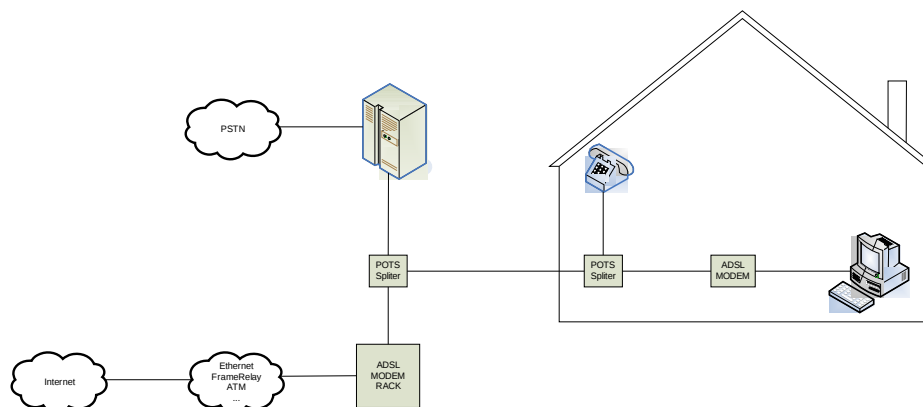
##### **41.1.2.1 ADSL**

Proporciona servicios digitales de alta velocidad sobre redes de pares de cobre existentes. Permitiendo trabajar sin interferir con los tradicionales servicios de voz analógica (POTS Plain Old Telephone Service).

Utiliza técnicas eficientes de codificación de línea como QAM. Y Soporta nuevos servicios sobre un par trenzado simple, como el acceso a Internet de alta velocidad.

Su ancho de banda asimétrico (64-640 kbit/s upstream, 500 kbit/s - 8 Mbit/s downstream) la hace atractiva para la mayoría de las aplicaciones cliente/servidor como el acceso a Web, acceso a LAN remotas, donde normalmente el cliente recibe mucha más información del servidor de la que genera.

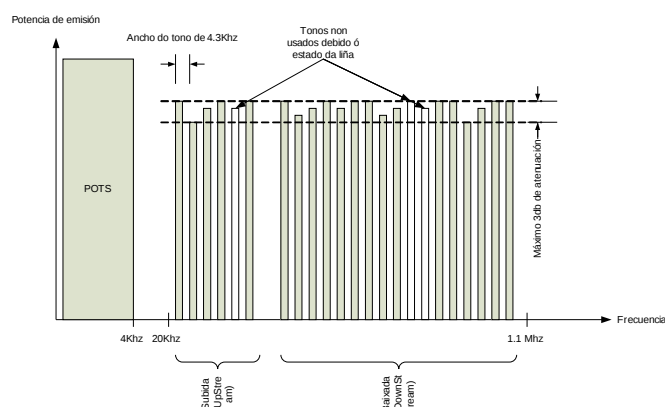
##### **41.1.2.1.1 ARQUITECTURA ADSL**



La arquitectura de ADSL hace uso de filtros tanto en el domicilio del abonado como en la central para separar la señal del teléfono y de la conexión de datos, estos filtros se llaman splitters. En el domicilio del abonado el teléfono se conectará directamente a la salida correspondiente del splitter mientras que los equipos de transmisión de datos necesitarán un MODEM ADSL. En la central la salida del splitter correspondiente se conecta a la PSTN (Public Switching Telephone Network o RTC) mientras que la salida de datos se conecta a uno de los modems de la central que está conectado a la red de distribución que su vez está conectada a Internet.

#### **41.1.2.1.2 NIVEL FÍSICO**

El canal se divide en tres bandas diferentes, se utiliza DMT (Discrete Multi-Tone) que permite utilizar diferentes portadoras (codificadas en QAM) en distintas frecuencias:



DMT utiliza la codificación QAM para conseguir codificar más bits en frecuencias

donde la señal presenta menos interferencias. De esta forma se modulan un número variable de bits en cada una de esas portadoras, dependiendo este número de las características del cable de pares, del espectro de frecuencias y de las interferencias en la señal. De este modo, los rateos de velocidad pueden ser optimizados haciendo posible el uso del mismo modem sobre bucles locales con diferentes características.

La velocidad de bajada depende de un buen número de factores, entre ellos:

- Longitud de la línea de cobre.
- Sección del cable.
- Presencia de bobinas de carga, por atenuación en líneas analógicas.
- Interferencias por paradiafonía.

El alcance depende de la velocidad y va desde 2,7 Km (la máxima velocidad en el peor caso) hasta 5,5km (la velocidad más baja en el mejor caso).

Actualmente este estándar evolucionó existiendo ADSL2 y ADSL2+ que proporcionan mayor velocidad (12 Mb en el caso de ADSL2 y 24Mb en el caso de ADSL2+) simplemente usando más espectro de frecuencias.

#### **41.1.2.2 HDSL**

HDSL es simplemente una forma mejor de transmitir circuitos T1 o E1 (32 canales de 64Kbs) sobre líneas de pares de cobre. Necesita un menor ancho de banda para transmitir estas líneas y no necesita utilizar repetidores.

Utilizando avanzadas técnicas de modulación, HDSL transmite 1,544 Mbps o 2,048 Mbps utilizando rangos de frecuencia entre 80 kHz y 240 kHz, bastante menos con los 1,5 MHz necesarios para las E1/T1 tradicionales. Sobre un cable con un calibre 24 AWG (0,5 mm) la distancia que se puede alcanzar es de aproximadamente 3,7 Km, aunque puede llegar los 4,5 Km, siempre sobre dos pares de cobre.

Este tipo de tecnología se utiliza para conexiones entre PBX, conexiones entre estaciones de antenas celulares, circuitos digitales, servidores de Internet y Redes de Datos Privadas. Pero no está falta de problemas, siendo los más destacables:

- La existencia de gran cantidad de implementaciones propietarias de HDSL pues el estándar sólo contempla las características básicas.
- Los beneficios técnicos son transparentes a los usuarios, pues ellos siguen percibiendo una T1/E1, aún se podrían alcanzar mejores rendimientos en términos de velocidad y de costes.
- Para distancias mayores de 3,7 Km se necesitan repetidores.
- La necesidad de utilizar múltiples pares de hilos, reduce la disponibilidad del servicio T1/E1 en un área determinada entre un 50 y 66 por ciento. Además, muchas veces existen problemas para encontrar 2 o 3 pares libres dentro de la misma ruta.

Al igual que con ADSL, HDSL evolucionó tratando de eliminar los problemas de la primera generación a HDSL2 que simplemente usa menos hilos.

#### **41.1.2.3 SDSL**

SDSL (Symmetrical Digital Subscriber Line) puede referirse a:

- En un sentido amplio a cualquier tecnología DSL simétrica
- O a un estándar concreto que proporciona servicios T1/E1 (el caso que nos ocupa)

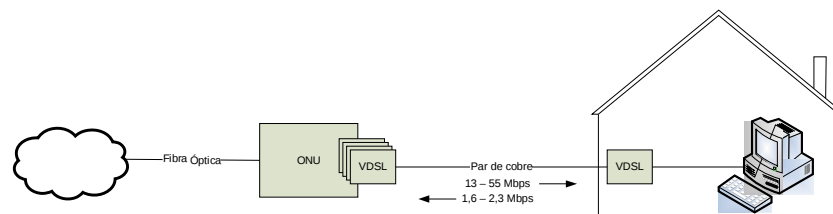
SDSL permite la transmisión de señales T1 o E1 sobre un único par de cobre. Esto supone una gran ventaja sobre HDSL puesto que se pueden usar líneas individuales extendiéndose el servicio a domicilios particulares. La distancia máxima de SDSL es de 3 Km, distancia a la que un ADSL podría operar a 6 Mbps.

#### **41.1.2.4 VDSL**

Es una de las más recientes tecnologías de la familia xDSL, confía en que el tendido de cobre será corto ya que las operadoras están instalando cada vez más tramos de F.O. (Fibra Óptica). Puesto que las operadoras quieren ofrecer nuevos servicios que requieran la combinación de voz, video y



audio, lo cual, sólo es posible con transportes como el ofrecido por ATM. VDSL está preparado para actuar como capa física que dé soporte a redes ATM. Para lograrlo, VDSL incluye una unidad de red óptica (ONU) que se encarga de convertir y concentrar señales VDSL sobre una red de fibra.



#### Problemas de VDSL:

- Es muy sensible a las interferencias de radio. A las frecuencias a las que trabaja, un bucle de abonado se comporta como una antena receptora para este tipo de señales.
- VDSL fue diseñada para trabajar sobre redes ATM.
- ADSL tiene un coste mucho menor, pues los filtros pueden instalarse en la propia central local. Con VDSL se necesitan ONU residenciales que deben ser instalados y mantenidos por la operadora.

El igual que con las tecnologías xDSL anteriores, VDSL tiene su evolución en VDSL2 capaz de proporcionar 250Mbps de bajada.

## **41.2 REDES DE TELEFONÍA MÓVIL**

### **41.2.1 GSM**

La tecnología GSM pertenece a los sistemas de segunda generación, que al igual que otros como CDMA, TDMA, NADC o PDC se caracterizan porque son digitales. GSM se implantó en Europa y en otros países del resto del mundo, mientras que TDMA y CDMA se implantó en EEUU, y PDC en Japón. GSM utiliza multiplexación por división del tiempo (TDM), lo que posibilita que en cada frecuencia se puedan transmitir varias conversaciones; el tiempo de transmisión se divide en pequeños intervalos de tiempo, cada uno de los cuales puede ser utilizado por una transmisión distinta. Además, una misma conversación se lleva a cabo en intervalos de distintas frecuencias, con lo que no se puede asociar una llamada a una frecuencia. Esto tiene la ventaja de que si una de las frecuencias se ve afectada por

una interferencia, una conversación que utilice esta frecuencia sólo observará problemas en los intervalos pertenecientes a dicha frecuencia. Esto se denomina TDMA.

El sistema GSM tiene asignadas dos bandas de frecuencias, 900 Mhz (llamado GSM-900) y 1800 Mhz (llamado DCS-1800). En cada una de las bandas, las frecuencias más bajas se usan para el enlace ascendente y las más altas para el descendente. Así, en la banda de los 900 Mhz, las frecuencias comprendidas entre los 890 y 915 Mhz comprenden 125 portadoras, cada una de ellas con un ancho de banda de 200 Khz para transmisiones móvil-estación base. De la misma manera, la banda comprendida entre los 935 y los 960 Mhz se subdivide en otras 125 portadoras de 200 Khz para transmisiones estación base-móvil.

Además, existe también una tercera banda de frecuencia en los 1900 Mhz, en los que GSM opera en algunos países como EEUU.

Cada portadora se subdivide en 8 canales lógicos o slots, cada uno de ellos tiene su uso particular: señalización, control de la comunicación o tramas de información.

En GSM, la señalización se realiza por el canal común, según el protocolo SS7.

GSM es un sistema basado en conmutación de circuitos y por tanto es un servicio orientado a conexión, es decir, en toda comunicación habrá 3 etapas diferenciadas: establecimiento, comunicación y liberación. En GSM, lo que se tarifica justo es el establecimiento de un canal.

La tasa de transmisión que se alcanza con GSM es de 9'6 Kbps.

#### **41.2.1.1 ARQUITECTURA DE LA RED GSM**

La arquitectura GSM se basa en tres subsistemas diferenciados:

- Subsistema estación base (BSS): Agrupa las máquinas específicas a los aspectos de radio y celulares del GSM. El BSS está en contacto directo con las estaciones móviles a través de la interfaz radio. El BSS incluye dos tipos de elementos: la Estación de Base (BTS, Base

Transceiver Station) y el Controlador de Estaciones de Base (BSC, Base Station Controller).

- Subsistema conmutación (NSS): Incluye las funciones básicas de conmutación del GSM, así como las bases de datos necesarias para los datos de usuario y la gestión de la movilidad. La función principal del NSS es gestionar las comunicaciones entre los usuarios GSM y los usuarios de otras redes de telecomunicación. Dentro del NSS, la función básica de conmutación se realiza en la MSC (Mobile services Switching Centre), cuya misión principal es coordinar el establecimiento de llamadas desde y hasta usuarios GSM.
- Subsistema operación y mantenimiento (OMS): Controla y monitoriza el estado general de la red. Se componen de los siguientes elementos:
  - o HLR: Hombre Location Registry. Existe un solo HLR por compañía. En esta base de datos se guarda la información estática del abonado (servicios contratados, etc.) y dinámica (dónde se localiza el móvil en un determinado momento)
  - o EIR: base de datos utilizada para comprobar la pertenencia del dispositivo móvil a la red del operador, mediante el chequeo del número IMEI, que es un código que identifica unívocamente el móvil, una especie de número de serie del aparato.
  - o AUC: centro de autenticación de usuarios
  - o OMC: Centro de gestión de la red (mantenimiento)

#### **41.2.2 GPRS, HSCSD**

GPRS y HSCSD pertenecen a los denominados sistemas de la generación 2'5, que harán de puente entre los de segunda y tercera generación (UMTS, que se verá más adelante).

##### **41.2.2.1 GPRS**

GPRS significa "General Packet Radio Service". Como su nombre indica, se trata de un servicio portador basado en la conmutación de paquetes, que

se realiza utilizando la red GSM actual, aunque necesita terminales que la soporten.

Las principales características de GPRS son las siguientes:

- Velocidad: la velocidad máxima teórica es de 171'2 Kbps, aunque se habla de una velocidad de conexión máxima en la práctica de 115 Kbps.
- Inmediatez: GPRS facilita las conexiones instantáneas tan pronto como se necesita enviar o recibir información; se puede decir que con GPRS estamos siempre conectados.
- Noticias y mejores aplicaciones: gracias a la mayor velocidad de GPRS.
- Tarificación: GPRS se tarifica por volumen de datos intercambiado, calidad de servicio y tipo de servicio; en GSM por el contrario se tarifica por duración de la llamada.

#### **41.2.2.1.1 ARQUITECTURA DE GPRS**

Como dijimos anteriormente, el sistema GPRS se despliega sobre la red GSM existente, aunque requiere de algunos elementos nuevos como:

- El nodo GGSN (Gateway GPRS Support Node): este nodo es la interfaz con las redes de datos externas, como X.25 y las redes IP
- El nodo SGSN (Serving GPRS Support Node): nodo de conmutación de paquetes, al mismo nivel que las centrales convencionales de GSM (las MSC)
- Estructura principal o red troncal GPRS (backbone)

#### **41.2.2.1.2 EDGE O E-GPRS**

EDGE responde a las siglas de Evolved (o Enhanced) Data rates for GSM Evolution. No es en sí una nueva arquitectura, sino un avance de la modulación del canal, es decir, es una versión mejorada de GPRS.

Con estas técnicas se consigue hasta 384 Kbps, combinando hasta 8 slots.

#### **41.2.2.2 HSCSD**

HSCSD (High Speed Circuit Switched Data) es una especificación homologada por el ETSI (European Telecommunication Standard Institute),

que supone una evolución de GSM, ya que aprovecha dicha infraestructura. Se trata de un servicio multi-slot para la transmisión de datos a alta velocidad mediante circuitos conmutados.

El HSCSD aporta un esquema de codificación mejorado que permite 14'4 Kbps, frente a los 9'6 Kbps de GSM, lo cual posibilita velocidades de transmisión de datos de hasta 57'6 Kbps combinando hasta 4 canales GSM. HSCSD fue desarrollado en paralelo con GPRS pero son servicios de alta velocidad totalmente diferentes. Como su propio nombre indica HSCSD utiliza la conmutación de circuitos a diferencia de GPRS que utiliza la conmutación de paquetes.

La ventaja de HSCSD sobre GPRS es la calidad de servicio garantizada, proporcionada por el canal de comunicación dedicado. Esto, sin embargo, la hace menos eficiente para la transmisión de datos pues la conexión tiene que mantenerse incluso en los momentos en los que no existe transmisión de datos. GPRS hace un uso más eficiente del ancho de banda y permite hacer la tarificación en función de la cantidad de contenido que se recibe y no solo en función del tiempo de conexión.

#### **41.2.3 SISTEMAS DE TERCERA GENERACIÓN: UMTS**

UMTS es el tercer escalón en la historia de la telefonía móvil, después de la analógica y la digital. UMTS son la siglas de Universal Mobile Telecommunication System o Sistema Universal de Comunicaciones Móviles. UMTS es un miembro de la familia global IMT-2000 del sistema de comunicaciones móviles de “tercera generación” del UIT (Unión Internacional de Telecomunicaciones). Este sistema es revolucionario ya que por primera vez se trata de un estándar universal.

El funcionamiento también es novedoso, ya que el usuario paga según la cantidad de información que se descargue de la red, y no ya por el tiempo de uso del servicio. De esta forma podremos estar constantemente conectados a la red, lo que permite, por ejemplo, acceder al correo electrónico de forma instantánea.

Esta tecnología permite que los teléfonos transmitan y reciban datos con una velocidad 200 veces superior a la de GSM. La máxima velocidad del

UMTS es 2 Mbits. Este tope puede alcanzarse solamente si la red está al máximo nivel, el usuario está parado y sin móviles a su alrededor.

UMTS también plantea importantes innovaciones con respecto a la arquitectura de red. UMTS R'99 definió una arquitectura que da cabida a redes de acceso GSM y a la red de acceso UMTS (UTRAN), y proponen una red central (CN, Core Network) diseñada como una evolución de la red GSM/GPRS para facilitar la migración de redes GSM/GPRS a UMTS.

UMTS ofrece una nueva interfaz radio denominada UTRA (UMTS Terrestrial Radio Access). Dicha interfaz está basada en tecnología CDMA (Code Division Multiple Access) permitiendo aumentar considerablemente la velocidad de transferencia de datos, y soporta dos modos de operación el FDD (Frequency Division Duplex) y el TDD (Time Division Duplex). FDD está basada en un esquema de Secuencia Directa CDMA y soporta una velocidad de hasta 384 Kbit/s. El TDD está basado en la multiplexación en tiempo y en código, se diseñó y se optimizó para ser usado en zonas con alta densidad de tráfico, y soporta una velocidad de hasta 2 Mbit/s.

Entre las cosas que nos ofrece UMTS, destacamos su facilidad de uso y bajos costes, nuevos y mejores servicios, acceso rápido, transmisión de paquetes de datos y velocidad de transferencia de datos a pedido, entorno de servicios amigable y consistente, movilidad de cobertura y servicios UMTS disponibles globalmente por satélite.

Su velocidad sumada al soporte inherente del Protocolo de Internet (IP), se combinan para prestar servicios multimedia interactivos y nuevas aplicaciones de banda ancha, tales como servicios de video telefonía y videoconferencia.

#### **41.2.3.1 HSPA**

High-Speed Downlink Packet Access (HSDPA) es un protocolo mejorado de la tercera generación que pertenece a la familia High-Speed Packet Access (HSPA) también conocida como 3.5G, 3G+ o turbo 3G. HSDPA permite a las redes UMTS tener un ancho de banda de descarga mayor que actualmente puede ser de 1,8, 3,6, 7,2 y 14,4 Mbps. Ya está disponible también HSPA+

que entrega hasta 84Mbps gracias al uso de varias antenas (MIMO Multiple Input Multiple Output).

High-Speed Uplink Packet Access (HSUPA) es otro de los protocolos HSPA que permite una velocidad de subida de 5,76Mbps. El nombre HSUPA fue acuñado por Nokia, el nombre oficial es Enhanced Uplink (EUL).

### **41.3 CABLE**

Las redes de cable actuales presentan las siguientes características: servicios integrados, alta capacidad, y redundancia. Los servicios que ofrece una red de cable moderna incluyen los siguientes: amplia oferta de canales de TV (terrestres, vía satélite, y de producción propia), video la la carta, PPV, datos e Internet (mediante módem cable), telefonía (básica y RDSI, con opción de acceso a Internet), alquiler de líneas y fibras.

La transmisión de la señal hasta el abonado se lleva a cabo mediante el canal denominado descendente o directo (de 86 a 862 MHz), mientras que las que parten del abonado se realizan a través del canal ascendente o de retorno (de 5 a 65 MHz).

La topología de una red de cable basada en tecnología HFC (Hybrid fibre-coaxial):

- Red troncal primaria: a nivel físico anillos redundantes de fibra óptica, a nivel lógico topología de estrella. Estos anillos comunican la cabecera con los nodos primarios. El respaldo es activo.
- Red secundaria o de distribución: conecta un nodo primario con varios nodos secundarios a través de anillos con arquitectura en estrella formando lóbulos que abarcan 12.000 hogares. Cada lóbulo interconecta 5 o 6 nodos secundarios. Hay redundancia en ruta y equipos. El servicio de telefonía a veces no se proporciona mediante la red HFC (es decir, no presenta telefonía integrada), sino que hace uso de una red paralela de tipo SDH (Synchronous Digital Hierarchy), hablándose entonces de telefonía superpuesta.
- Red terciaria o de dispersión: conecta cada nodo secundario con cada uno de los cuatro nodos ópticos terminales que dependen de él. Cada

nodo óptico terminal cubre un área de 500 hogares. La red de dispersión presenta una disposición en estrella sin redundancia en ruta. En el nodo secundario se realiza la interconexión de las fibras provenientes del nodo primario con las fibras que van hasta los nodos terminales.

- Red de distribución de coaxial: distribuye las señales desde el nodo óptico terminal hasta cada punto de derivación en los edificios a los que da servicio. La distribución se realiza con estructura en árbol, de forma que cada nodo óptico terminal da lugar a 4 ramas de unos 125 hogares aproximadamente. Los nodos ópticos terminales se ubican físicamente en armarios de exterior. El nodo óptico terminal realiza la conversión óptico-eléctrica de las señales transportadas en sentido descendente. De él va a los amplificadores que atacan las cuatro ramas de coaxial que parten del nodo óptico. Cada rama de coaxial alimenta (si es necesario, mediante amplificadores) a una red de derivadores, cuyas salidas están conectadas a las atacadas individuales de abonado. Para el camino de retorno se utiliza la misma infraestructura de red, equipando adecuadamente a los amplificadores.
- Red de atacada de abonado: conecta la red de distribución de coaxial con el punto de terminación de red. Existen dos arquitecturas:
  - o Estrella: un mismo derivador da servicio a todas las viviendas de las diferentes plantas de un edificio.
  - o Árbol: se coloca un derivador en cada planta, del que parten los coaxiales que dan servicio a los abonados de esa planta.La red de atacada de abonado se puede dividir en dos partes: cableado de edificio o verticales, y cableado de vivienda.

Las redes HFC tienen los siguientes puntos singulares:

- Cabecera: está equipada para la prestación del servicio de difusión de televisión. Se puede descomponer en cuatro bloques:



- o Sistemas de recepción y transmisión analógica: compuesto por antenas de recepción, equipos de recepción, equipos para banda base, etapa de codificación, y etapa de modulación y salida.
  - o Sistemas de recepción y transmisión analógica de reserva: antenas de recepción, equipos de recepción, y etapa de modulación.
  - o Sistemas de monitorización.
  - o Sistemas de transmisión óptica.
- **Nodo primario:** recibe la señal de la red troncal primaria procedente de la cabecera de red. El nodo primario presenta dos módulos independientes:
  - o El módulo del camino descendente.
  - o El módulo del camino ascendente.
- **Nodo secundario:** encaminan las señales procedentes del nodo primario (mediante la red troncal secundaria) hasta los nodos ópticos terminales (a través de la red terciaria). Se ubican físicamente en una arqueta, habitualmente junto a uno de sus nodos ópticos terminales.
- **Nodo óptico terminal:** dan servicio a áreas de aproximadamente 500 hogares. Se ubican en armarios de exterior. Se pueden descomponer en dos grandes bloques:
  - o Canal descendente.
  - o Canal ascendente.
- **Terminal direccionable de abonado:** permite al cliente acceder a los servicios TV de la red y es instalado en el propio domicilio del abonado. Descodifica los canales correspondientes al servicio contratado por el abonado y permite al cliente interactuar con el sistema.
- **Subredes de telefonía y datos:**
  - o Subred de datos:
    - Servicios ofrecidos: portadores (alquiler de circuitos digitales), de transmisión de datos (se basan en conmutación de circuitos y conmutación de paquetes o celdas), de acceso a redes

(acceso a Internet y a otros proveedores de contenido multimedia), y de valor añadido.

- Estructura: los equipos que conectan la red de datos con Internet están situados en la cabecera. El nodo primario que reside junto a ésta es el encargado del funcionamiento de los módems cable, a través de los cuales el abonado tiene acceso a la red. Sus elementos son:
  - Router: encamina el tráfico IP entre la red de datos e Internet.
  - Servidor Proxy: actúa a modo de caché.
  - Firewall (cortafuegos): protege la red de datos de ataques externos.
  - Servidores: se encargan de dar diversos servicios: WWW, FTP, IRC, e-mail, DNS, etc.
  - Conmutador ATM multiservicio: permite la interconexión de equipos de diferentes tecnologías.
  - Conmutador LAN: conecta los servidores con el conmutador ATM multiservicio.
  - Conmutador ATM de acceso: como el conmutador ATM multiservicio, pero de menor capacidad.
  - Cabecera de módems cable: la cabecera de módems y los módems cable, componen la red de acceso a datos integrada en HFC.
  - Módems cable: se sitúa en el domicilio del abonado, y permite acceso a la red de datos mediante HFC.
- o Red de telefonía:
  - Servicios ofrecidos: telefonía analógica tradicional, acceso digital RDSI básico, acceso digital RDSI primario.
  - Estructura: La red soporta tanto telefonía integrada como superpuesta.

- Centro de conmutación: canaliza todo el tráfico de llamadas.
- Red de acceso mediante telefonía integrada: aprovecha la red HFC de distribución de TV y datos para llegar hasta el cliente. Para realizar la interfaz con la red HFC, son necesarios dos equipos específicos (HDT y MDU).
- Red de acceso mediante telefonía superpuesta: no usa la red HFC. Desde la cabecera se distribuye la señal mediante fibra óptica hasta los nodos primarios, y desde ellos hasta los lóbulos de 12.000 hogares de la red SDH.
- Cableado de viviendas.

#### **41.4 PLC**

A tecnología Power Line Communications, "PLC", posibilita la transmisión de voz y datos a través de los cables eléctricos, convirtiendo cualquier enchufe de la casa en conexión potencial a todos los servicios de telecomunicaciones. El cliente sólo necesitará conectar un pequeño módem para acceder a Internet, telefonía y datos al mismo tiempo y a la alta velocidad (banda ancha).

La red eléctrica transporta electricidad a una frecuencia de 50 Hz. En PLC se añaden frecuencias en la banda que va desde 1,6MHz hasta 30MHz para el transporte de los datos. Unos filtros instalados en el transformador de baja tensión separan las frecuencias altas de datos, de la frecuencia de 50Hz de la electricidad.

Power Line Communications emplea una red conocida como High Frequency Conditioned Power Network (HFPCN) para transmitir simultáneamente energía e información. Una serie de unidades acondicionadoras son las que se encargan del filtrado y separación de ambas señales.

En la actualidad no existen estándares tecnológicos para el PLC de acceso. Este es uno de los principales problemas de esta tecnología, al no permitir la interoperabilidad entre los equipos suministrados por los distintos fabricantes. Tampoco existía una regulación en cuanto a la utilización de

frecuencias hasta 2005. Garantizando ahora la coexistencia de sistemas domésticos (como HomePlug) y las tecnologías de acceso.

Es posible que el precio de la tecnología PLC sea bastante inferior al de los actuales ADSL y Cable en el mismo rango de velocidades, lo que la convierte en una tecnología interesante.

Los servicios típicos de telecomunicaciones que podrían ser proporcionados son:

- Telefonía
- Acceso rápido a Internet
- Video bajo demanda

Ventajas de PLC:

- Como la PLC se posicionó como un servicio de tipo IP utilizará routers de paquetes en vez de los de conmutación de circuitos típicos de los suministradores de telecomunicaciones tradicionales, manteniendo así los costes de los equipos de IT bajos.
- Las compañías eléctricas podrían pues comercializar un servicio básico de conexión a Internet con una suscripción mensual de tarifa plana.
- Esta tecnología se pone virtualmente al alcance de cualquiera.
- Ya existen varias tecnologías que transforman los cables eléctricos existentes en un cableado LAN (Local Area Network)
- PLC podría facilitar a las compañías eléctricas la oportunidad de ofrecer servicios de valor añadido.

Inconvenientes de PLC:

- El número máximo de hogares por transformador. Como las señales de datos de Power Line no pueden sobrevivir a su paso por un transformador, solo se utilizan en la última milla. El modelo europeo de red eléctrica suele colocar un transformador cada 150 hogares aproximadamente.
- Por lo que es necesario que todos los transformadores vengan dotados de servidores de estación base PowerLine.

- Cualquier línea conductora es, por definición, una antena por lo que la instalación eléctrica de una casa actúa como tal, y es muy sensible a las interferencias que se produzcan en las frecuencias de transmisión de datos, alrededor de los 30 MHz.

## **41.5 REDES RADIO (LMDS, WIMAX),**

### **41.5.1 LMDS**

LDMS nace en el contexto de las emergentes tecnologías sin hilos y el creciente interés en IP como una alternativa para proporcionar servicios multimedia al usuario final, junto con el aumento de la demanda de nuevos servicios de telecomunicación orientados a voz y datos (acceso rápido a Internet, etc.)

LDMS (Local Multipoint Distribution Service) es una tecnología de acceso sin hilos de banda ancha o bucle de abonado sin cable. Los sistemas LDMS utilizan ondas radioeléctricas de alta frecuencia para ofrecer servicios multimedia y de difusión a usuarios finales en distancias similares a las alcanzadas con las tecnologías de cable.

Entre las ventajas que ofrece LDMS podemos señalar:

- Rápido despliegue, comparado con las tecnologías de cable: LDMS permite instalar redes rápidamente ya que, por ejemplo, el emplazamiento de las antenas es muy sencillo dado el pequeño tamaño de estas.
- Posibilidad de integrar distintos tipos de tráfico (voz, video, datos,...)
- Alta velocidad de acceso a Internet
- Flexibilidad y modularidad

Como otras características de LDMS, podemos señalar que requiere LoS (Line of Sight), es decir, visión directa entre los dos puntos que se comunican. Las velocidades de acceso que se alcanzan se encuentran en el entorno de los 512 Kbps - 2 Mbps.

El servicio LDMS prestará en dos bandas:

- Banda S: esta banda trabaja en los 3,5 GHz. Es la que se utiliza para el despliegue del bucle de abonado. Tiene un alcance de alrededor de 15 Km y posee un ancho de banda de 20 Mhz.
- Banda K: trabaja en los 26 GHz. Es la banda utilizada para el acceso de banda ancha. Dispone de un alcance menor que la banda S (alrededor de los 3 Km), pero un mayor ancho de banda (unos 56 Mhz).

La comunicación en LDMS se establece mediante radiodifusión punto-multipunto: las señales viajan desde o hasta una estación central, hasta o desde los diferentes puntos de recepción distribuidos en la zona de cobertura.

LDMS utiliza modulación QPSK (Cuadratura Phase Shift Keying), que permite reducir las interferencias y aumentar la reutilización del espectro, alcanzando un ancho de banda cercano a 1 Gbps.

#### **41.5.2 WIMAX**

WiMAX (Worldwide Interoperability for Microwave Access) es un protocolo de telecomunicaciones que proporciona acceso a Internet en puntos fijos o móviles.

La actual revisión de WiMAX permite hasta 40Mbps, con la nueva versión (IEEE 802.16m) se esperan velocidades de hasta 1 Gbps.

El nombre WiMAX fue creado por el WiMAX Forum que fue fundado en junio de 2001 para fomentar la interoperabilidad del estándar. Este foro describe WiMAX como una tecnología basada en estándares permitiendo el acceso de banda ancha de última milla como alternativa al cable y a las xDSL.

El estándar 802.16 Broadband Wireless Access (BWA) define:

- Capa 1 - Capa física: La versión original de IEEE 802.16 especifica una capa física que opera el rango entre los 10 y los 66GHz. 802.16a (actualización de 2004) añadió la posibilidad de operar entre 2 y 11 GHz. En 2005 la versión 802.16e-2005 vio la luz usando SOFDMA (Scalable Orthogonal Frequency-Division Multiple Access) en vez de la original OFDM (Orthogonal Frequency División Multiplexing). 802.16e también define el uso de varias antenas con MIMO.

- Capa 2 - Capa de acceso al medio (MAC): Usa un algoritmo de planificación para lo cual la estación del abonado necesita competir solo una vez para conectarse a la red. La ventana de tiempo puede alargarse o contraerse, pero permanece asignada al abonado. El algoritmo es estable en situaciones de sobrecarga y gran número de abonados, permitiendo a la estación base el control de la calidad del servicio (QoS).
- Morbilidad
- Características opcionales y obligatorias del enlace de radio

#### **41.6 SATÉLITE**

El acceso a Internet por satélite puede obtenerse en cualquier parte del mundo usando satélites LEO (Low Earth Orbit) aportando una relativamente baja latencia pero baja velocidad, o satélites geoestacionarios aportando mayor velocidad pero también mayor latencia y no pudiendo llegar a ciertas partes de los polos. Las desventajas de este sistema no se quedan ahí (alta latencia) sino que también incluyen problemas de cobertura cuando llueve, además requiere línea directa de visión al satélite (en un terreno escarpado esto puede ser un problema).

El equipo del cliente para una comunicación de satélite requiere la instalación de una antena parabólica de un tamaño dependiente de la tecnología concreta, del satélite y del modo en que se use (comunicación uni o bidireccional,...) además de un MÓDEM específico.

Existen distintas técnicas usadas para compartir cada portadora (TDMA, SCPC,...) aportando velocidades de hasta 40Mbps de descarga.

Los típicos modos de comunicación son:

- Comunicación bidireccional por satélite. En este caso tanto la subida como la bajada se produce usando al satélite, requiriendo una muy precisa orientación de la antena.
- Comunicación unidirección mas conexión terrestre. En este caso la antena sólo recibe la señal del satélite, y el envío de datos se produce por una línea terrestre (RTC, GSM, GPRS,...). Tiene la ventaja sobre el

modelo anterior de que la antena no necesita tener una orientación tan precisa y que, por lo tanto, es mejor para instalaciones móviles.

- Comunicación unidireccional / multicast, sin retorno. Dentro de esta modalidad tenemos los servicios de multicast por IP que no requieren retorno (además de la transmisión de audio y video).

#### **41.7 LÍNEAS PUNTO A PUNTO**

Una línea punto a punto o línea personal (como contraposición a una VPN) es una línea física entre dos ubicaciones de forma transparente, de modo que una línea punto a punto desde una ubicación remota funciona de la misma manera que se estuviera conectada en la ubicación de destino. Antes, cuando se contrataba una línea punto a punto se especificaban las características y velocidad de la misma usando múltiplos del DS0, desde un DS0 (64Kbps) hasta un T1/E1 (1,5Mbps) o DS-3 (672 canales de 64Kbps). Los operadores usaban conmutación de circuitos o circuitos virtuales con tecnologías como X.25.

En la actualidad estas líneas se suelen crear como una VPN dentro de las redes del operador, por ejemplo usando MetroEthernet y MPLS.

##### **41.7.1 X.25**

X.25 es un estándar para el acceso a redes públicas de conmutación de paquetes. No especifica cómo está implementada la red internamente aunque el protocolo interno suele ser parecido a X.25. Implementa un servicio de circuito virtual externo.

El servicio que ofrece es orientado a conexión, fiable, en el sentido de que no duplica, ni pierde ni desordena, y ofrece multiplexación, esto es, a través de un único interfaz se mantienen abiertas distintas comunicaciones.

Los elementos usados por X.25 se denominan:

- DTE (Data Terminal Equipment): Es el equipo final de usuario (PC con placa X.25, por ejemplo).
- DCE (Data Circuit Terminating Equipment): Podemos interpretarlo como un nodo local. A nivel de enlace (LAPB) las conexiones se establecen



DTE-DCE. Con el nivel de red ampliamos las comunicaciones mas allá del DCE, que hace de interconexión.

X.25 define 3 niveles:

- Nivel Físico:

Existen dos posibilidades:

- X.21: Se utiliza para el acceso a redes de conmutación digital. (Similares a las de telefonía digital.)
- X.21bis: Se emplea para el acceso a través de un enlace punto a punto. (Similar a RS-232 en modo síncrono.)

En cuanto a las características mecánicas, se usan conectores Canon de 15 pines o de 25 pines.

Las velocidades van entre los 64kbps y los 2Mbps, velocidades que pueden parecer bajas y, de hecho, lo son. X.25 presenta un problema de baja eficiencia por la exagerada protección contra errores que implementa y que con las redes actuales no tiene sentido.

- Nivel de Enlace (LAP-B):

En X.25, este nivel queda implementado con el protocolo LAP-B (Link Access Procedure - B) que es un protocolo de enlace con rechazo simple y en el cual las tramas de información pueden ser utilizadas como tramas de control.

- Nivel de Paquete (PLP):

Este nivel está especificado por el PLP (Packet Layer Protocol) que es un protocolo de acceso a nivel de red y que proporciona servicios al nivel superior.

Permite establecer circuitos virtuales (CV): Que podríamos definir como la asociación lógica entre usuarios para comunicarse entre ellos. Existen dos tipos de CV:

- Conmutados (CVC): Hay que realizar un diálogo previo a la transmisión con el nodo local para establecerlos.
- Permanentes (CVP): Están establecidos de antemano (por contrato), así que no hace falta fase de establecimiento. Son muy útiles si

se transmite mucho y con mucha frecuencia hacia un mismo destino.

#### **41.7.2 FRAMERELAY**

Es posible usar FrameRelay como tecnología de soporte para conseguir redes punto a punto. Esta tecnología se explica en otro tema.

#### **41.7.3 METROETHERNET**

Es posible usar MetroEthernet como tecnología de soporte para conseguir redes punto a punto. Esta tecnología se explicará más adelante en este mismo tema.

#### **41.8 METROETHERNET**

MetroEthernet es una red tipo MAN (está diseñada para cubrir un área metropolitana) basada en el muy conocido estándar Ethernet. El uso habitual de MetroEthernet es servir de red de acceso para conectarse a Internet o servir de red de interconexión de varias oficinas de una compañía.

Habitualmente el proveedor de una conexión MetroEthernet proporcionará la misma por fibra óptica terminando en un equipo que normalmente tiene capacidades no solo de nivel 2 sino de nivel 3 (red).

Esta tecnología puede desplegarse de varias formas atendiendo a la base usada para MAN que la soporta:

- Ethernet pura, toda la red está basada en Ethernet, sin ninguna otra tecnología de soporte. Esta opción aporta una gran simplicidad y bajo coste, pero tiene graves problemas de fiabilidad y de escalabilidad por lo que está limitada a pequeños despliegues.
- MetroEthernet basadas en redes SDH ya existentes, con las limitaciones que impone SDH en el manejo del ancho de banda.
- Y, por último, MetroEthernet basadas en MPLS. Esta es la opción más cara pero la más escalable y fiable siendo la típica a desplegar por operadores (de no tener una red SDH existente).

##### **41.8.1 MAN BASADA EN ETHERNET**

Un despliegue basado solo en Ethernet hace solo uso de switches de nivel 2. Esto permite un diseño y configuración muy simples a un bajo coste. Este tipo de despliegue solo fue posible tras la incorporación de las VLAN (Virtual LAN) aportando la posibilidad de “punto a punto” y “multipunto a multipunto” combinadas con VLAN Stacking (también conocida como VLAN Tunneling) y VLAN Translation, ya que previamente no era posible aislar el tráfico de cada usuario (dada la naturaleza de Ethernet) para formar circuitos.

VLAN Stacking permite el uso de varias LANs virtuales sobre el mismo circuito de la red troncal gracias al uso de dos identificadores: uno para red troncal y otro para la red Ethernet (existiendo 4096 identificadores distintos según el estándar 802.1Q, que no 4096 VLAN distintas).

VLAN Translation permite convertir un identificador de VLAN en otro, de forma que el identificador usado en una parte de la red sea distinto al usado en otra, evitando de esta manera posibles conflictos entre identificadores de distintos usuarios.

#### **41.8.2 MAN BASADA EN SDH**

Una Ethernet MAN basada en SDH (Synchronous Digital Hierarchy) es un paso intermedio entre redes tradicionales (basadas en división de tiempos) y redes más modernas (como Ethernet). En este modelo la infraestructura SDH existente es usada para transportar conexiones Ethernet de alta velocidad aportando una gran fiabilidad gracias a los mecanismos intrínsecos de las redes SDH (con un tiempo de recuperación inferior a 50 ms).

Este tipo de implantaciones se limitan a los casos donde ya existe una red SDH debido al alto coste de los equipos SDH y a las limitaciones de SDH para gestionar el tráfico (velocidad, ruta,...) originando muchas veces la instalación de switches Ethernet en la frontera SDH para aliviar parte de estas limitaciones

#### **41.8.3 MAN BASADA EN MPLS**

En este caso las tramas Ethernet enviadas por el usuario son empaquetadas en MPLS que transmite sus datos sobre (habitualmente)

Ethernet, creando una pila Ethernet sobre MPLS sobre Ethernet (aunque podría haber otro protocolo por debajo).

Este despliegue usa LDP (Label Distribution Protocol) punto a punto para la etiqueta interna (etiqueta del VC) y RSVP-TE (Resource reSerVation Protocol-Traffic Engineering) o LDP para la etiqueta externa usada en la red.

Uno de los mecanismos de restauración de MetroEthernet basadas en MPLS es Fast ReRoute (FRR) que permite un tiempo de restauración inferior a los 50ms. Esta es una de las cosas que más hay que tener en cuenta a la hora de decidirnos por MetroEthernet basadas en MPLS frente aquellas basadas en Ethernet, ya que si tenemos un tiempo de restauración equivalente usando una solución Ethernet pura no merece la pena introducir una basada en MPLS.

#### **41.9 BIBLIOGRAFÍA**

José Manuel Huidrobo. Todo sobre comunicaciones. PARANINFO, 1998

José Manuel Huidrobo. Manual de Telefonía. PARANINFO, 1996

Andrew S. Tanenbaum. Redes de computadoras. PRENTICE HALL, 1997

**Autor:** Matías Villanueva Sampayo

Director de Informática Asociación Provincial de Pensionistas y Jubilados de A Coruña

Colegiado del CPEIG

## **42. TECNOLOGÍAS DE TRANSPORTE: FRAME RELAY, ATM, DWDM, MPLS. REDES DE FIBRA ÓPTICA. REDES DE NUEVA GENERACIÓN (NGN).**

## **Tema 42. Tecnologías de transporte: Frame Relay, ATM, DWDM, MPLS. Redes de fibra óptica. Redes de nueva generación (NGN).**

### 42.1 Tecnologías de transporte

#### 42.1.1 Frame Relay

##### 42.1.1.1 Arquitectura de protocolos

###### 42.1.1.1.1 Protocolo LAPF

###### 42.1.1.1.2 Direccionamiento

###### 42.1.1.1.3 Control de la congestión

###### 42.1.1.1.4 Tipos de tráfico transportado

###### 42.1.1.1.5 Ventajas

#### 42.1.2 ATM

##### 42.1.2.1 Principios de operación

##### 42.1.2.2 Capas de ATM

###### 42.1.2.2.1 Capa física

###### 42.1.2.2.2 Capa ATM

###### 42.1.2.2.2.1 Parámetros del tráfico

###### 42.1.2.2.2.2 Clases de servicio

###### 42.1.2.2.2.3 Asignación de ancho de banda y control de la congestión

###### 42.1.2.2.3 Capa de adaptación (AAL)

###### 42.1.2.2.3.1 Estructura de la capa AAL

#### 42.1.3 DWDM

##### 42.1.3.1 Demultiplexadores

##### 42.1.3.2 Erbium Doped Fiber Amplifier

#### 42.1.4 MPLS

##### 42.1.4.1 Funciones de MPLS

##### 42.1.4.2 LSRS y LERS

##### 42.1.4.3 FEC

##### 42.1.4.4 Etiquetas

##### 42.1.4.5 Distribución de etiquetas

##### 42.1.4.6 LSP

#### 42.1.4.7 Pila de etiquetas

### 42.2 Redes de fibra óptica

#### 42.2.1 Jerarquía Digital Plesiócrons (PDH)

#### 42.2.2 Jerarquía Digital Síncrona (SDH) / SONET

##### 42.2.2.1 Frame SDH / SONET

##### 42.2.2.1.1 Framing

##### 42.2.2.1.2 Estructura del frame STM-1

### 42.3 Redes de nueva generación (NGN)

#### 42.3.1 Tecnologías de soporte

### 42.4 Bibliografía

## **42.1 TECNOLOGÍAS DE TRANSPORTE**

### **42.1.1 FRAME RELAY**

Frame relay es un protocolo de transmisión de paquetes de datos en ráfagas de alta velocidad a través de una red digital fragmentado en unidades de transmisión llamadas Frames. Requiere una conexión exclusiva durante el período de transmisión.

Frame relay es una tecnología de paquete-rápido ya que el chequeo de errores no ocurre en ningún nodo de la transmisión. Son los extremos los responsables de este chequeo de errores. (Sin embargo debido a que los errores en redes digitales son extremadamente menos frecuentes en comparación con las redes analógicas, esto no supone un verdadero inconveniente).

A diferencia de los paquetes que son de tamaño fijo, Frame Relay transmite Frames que son de tamaño variable (mil o mas bytes).

El estándar de Frame Relay (ITU-T I.122) es una extensión del estándar ISDN. Implementa varias interfaces físicas como V.35 para velocidades menores de 2 Mb y G.703 para 2 Mb.

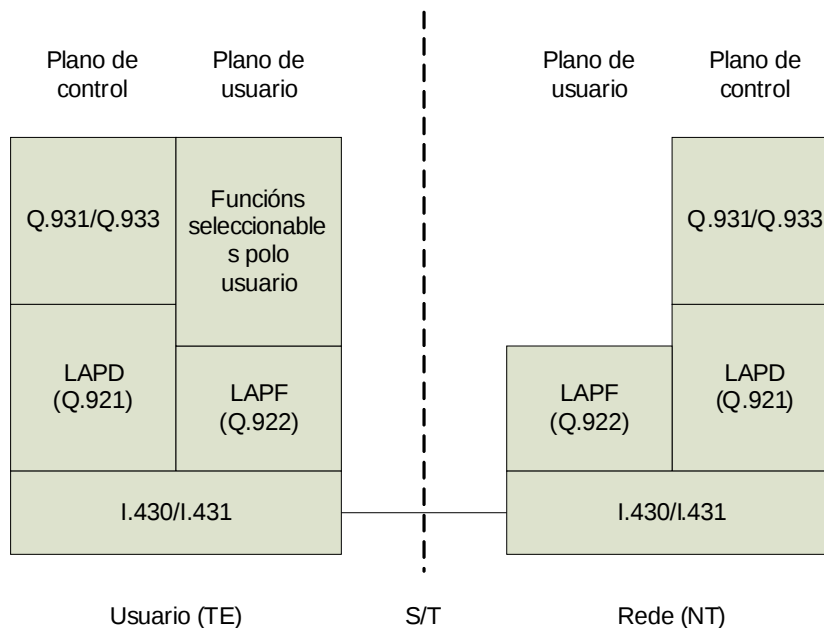
Una conexión Frame Relay es conocida como una conexión virtual. Una conexión virtual permanente es exclusiva al par origen-destino y puede transmitir por encima de 1,544 Mbps, dependiendo de las capacidades del

par origen-destino. Es posible también una conexión virtual conmutada usando la red pública y puede proporcionar elevados anchos de banda.

#### **42.1.1.1 ARQUITECTURA DE PROTOCOLOS**

En Frame Relay se consideran dos planos de operación:

- Plano de control (C): Involucrado en el establecimiento y liberación de conexiones lógicas. Los protocolos de este plano se implementan entre el usuario y la red. Es similar a la señalización en canales de servicios de conmutación de circuitos, ya que usa un canal lógico separado para la información de control. En la capa de enlace se usa el protocolo LAPD (Q.921) para proporcionar un servicio de control de datos fiable, mediante un control de flujo y de errores entre la red y el usuario. Este servicio de enlace de datos usa para el intercambio de mensajes de control el protocolo Q.931.
- Plano de usuario (U): Responsable de la transferencia de datos extremo a extremo mediante el protocolo LAPF (Procedimiento de Acceso al Enlace para Servicios en Modo Trama) definido en el estándar Q.922.



##### **42.1.1.1.1 PROTOCOLO LAPF**

Permite la retransmisión de tramas como un servicio orientado a conexión de la capa de enlace con las siguientes propiedades:



- Envío secuencial de las tramas a partir de la información de direccionamiento existente en la cabecera de control de cada trama.
- Existe una probabilidad pequeña de pérdida de tramas.
- Reduce al mínimo el trabajo de la red.

El formato de trama de LAPF de funcionamiento mínimo (conocido como protocolo central LAPF) es similar al LAPD y LAPB con una salvedad: no existe campo de control, lo que tiene las siguientes implicaciones:

- Existe un único tipo de trama, usada para transportar datos de usuario. No existen tramas de control.
- No es posible el uso de señalización en banda; una conexión lógica solo puede transmitir datos de usuario.
- No es posible llevar a cabo control de errores dado que no existen números de secuencia.
- Los campos indicadores y secuencia de verificación de trama (FCS), actúan como en LAPD y LAPB. El campo de información contiene datos de capas superiores. Si el usuario decide implementar funciones adicionales de control de enlace de datos extremo a extremo, debe incluirse en este campo una trama de enlace de datos.

#### **42.1.1.1.2 DIRECCIONAMIENTO**

El campo de dirección tiene implícitamente una longitud de 2 octetos, y puede ampliarse a 3 o 4 octetos (se indica mediante los bits de ampliación del campo de dirección (EA)). Este campo contiene un identificador de conexión de enlace de datos (DLCI) de 10, 17 o 24 bits. El DLCI proporciona la misma función que el número de circuito virtual en X.25: permite la multiplexación de varias conexiones lógicas de retransmisión de tramas a través de un único enlace físico. Como en X.25, el identificador de conexión solo tiene significado local.

La función realizada por cualquier red que soporte la técnica de retransmisión de tramas, consiste en el encaminamiento de las tramas de acuerdo con sus valores DLCI. Para esta función existe un gestor de tramas

encargado de tomar las decisiones de encaminamiento. Esta operación puede involucrar a múltiples gestores de tramas interconectados.

#### **42.1.1.1.3 CONTROL DE LA CONGESTIÓN**

Se utilizan dos tipos de señalización para el control de la congestión:

- Señalización implícita: se produce cuando la red descarta tramas, hecho que es detectado por los protocolos de nivel superior en la interfaz de usuario.
- Señalización explícita: es de carácter opcional y constituye una señalización de la red a la interfaz de usuario mediante dos bits. Ambos bits permiten que los dispositivos finales regulen la velocidad de transmisión de información a la red hasta que la congestión desaparezca. Los bits son el BECN (congestión en el sentido opuesto a la trama) y el FECN (congestión en el sentido de la trama).

Adicionalmente, existe otro mecanismo de control de la congestión (CLLM, “Consolidated Link Layer Management”), consistente en mensajes que la red envía a los dispositivos de acceso con códigos indicadores de las causas de la congestión, así como una lista de todos los DLCI’s que deben reducir su tráfico para disminuir el nivel de congestión.

En Frame Relay se definen dos clases de tráfico por Circuito Virtual Permanente:

- Clase de Caudal o CIR (Committed Information Rate): se define como el caudal de información que la Red se compromete a transmitir expresado en bit/sg. En un dimensionamiento correcto, debe corresponder a un tráfico “normal” o promedio. Se recomienda que no supere el 75 % de la velocidad de la línea, aun cuando hay poca simultaneidad entre los Circuitos Virtuales, se puede superar con la sobrecontratación que se recomienda no sobrepase del 200%.
- Exceso de Tráfico o EIR (Excess Information Rate): sin contratación, y que va dirigido a permitir la transmisión de ráfagas de gran intensidad de tráfico, sin coste adicional: se define como la cantidad de información en exceso del CIR contratado, que la Red es capaz de

gestionar durante un período de tiempo definido. Esta clase de tráfico, expresado en bit/sg, es señalada por la Red como de menor prioridad ( $DE=1$ ), y en condiciones normales será retransmitida y en condiciones de congestión puede ser descartada.

Para los Circuitos Virtuales Conmutados se define un CIR y un EIR Total Agregado que engloba a todos los CVC establecidos.

#### **42.1.1.1.4 TIPOS DE TRÁFICO TRANSPORTADO**

La necesidad de nuevas facilidades para las comunicaciones de área extendida, en particular para las comunicaciones de datos entre redes locales, impulsó enormemente el desarrollo y utilización de Frame Relay, debido a que satisface las dos características predominantes de este tipo de tráfico:

- Tráfico a ráfaga e impulsivo, que exigiría dimensionar en exceso el enlace para atender los picos de demanda. Frame Relay soluciona este problema, ya que requiere una pequeña cantidad de ancho de banda permanentemente reservada vía un circuito virtual permanente, mientras que dinámicamente, y siempre que exista ancho de banda disponible, es posible asignar mayor velocidad a la conexión para atender picos de demanda.
- La necesidad de interconexión remota de LANs incrementa la necesidad de mallado de las redes WAN resultantes. Frame Relay evita la necesidad de numerosos enlaces físicos, permitiendo la definición de múltiples circuitos virtuales permanentes a diferentes destinos, a través de un mismo puerto dedicado a un enlace físico.

#### **42.1.1.1.5 VENTAJAS**

Ahorro en los costes de telecomunicaciones: Con el servicio Frame Relay los usuarios podrán transportar simultáneamente, compartiendo los mismos recursos de red, el tráfico perteneciente a múltiples comunicaciones y aplicaciones, y hacia diferentes destinos.

Tecnología punta y altas prestaciones: Frame Relay proporciona alta capacidad de transmisión de datos por la utilización de nodos de red de

alta tecnología y bajos retardos, como consecuencia de la construcción de red (backbone) sobre enlaces a 34 Mbps. y de los criterios de encaminamiento de la Red de Datos, orientados a minimizar el número de nodos de tránsito.

**Flexibilidad del servicio:** Frame Relay es una solución adaptable a las necesidades cambiantes, ya que se basa en circuitos virtuales permanentes (CVP), que es el concepto de Red Pública de Datos, equivalente al circuito punto a punto en una red personal. Sobre una interfaz de acceso a la red se pueden establecer simultáneamente múltiples circuitos virtuales permanentes distintos, lo que permite una fácil incorporación de nuevas sedes a la Red de Cliente.

**Servicio normalizado:** Frame Relay es un servicio normalizado según los estándares y consejos de UIT -T, ANSI y Frame Relay Forum, con el que queda garantizada la interoperatibilidad con cualquier otro producto Frame Relay también normalizado.

#### **42.1.2 ATM**

Asynchronous Transfer Mode (ATM) es una técnica de conmutación que lleva a cabo la transmisión de datos por medio de paquetes (celdas), permite la multiplexación de varias conexiones lógicas sobre una única interfaz física y se trata de una técnica de conmutación de paquetes orientada a conexión.

El protocolo de ATM tiene una mínima capacidad de control de errores, de flujo y un tamaño de paquete fijo (celda) con lo que facilita el uso de nodos de conmutación a velocidades elevadas.

Sus principales características son:

- Capacidad de integración de diverso tipo de tráfico.
- Asignación dinámica y flexible del ancho de banda.
- Optimización del compromiso entre caudal y latencia.
- Ganancia estadística: capacidad de optimizar la relación entre la suma de las velocidades de pico de las fuentes y la velocidad del enlace.

#### **42.1.2.1 PRINCIPIOS DE OPERACIÓN**

Las conexiones lógicas en ATM están relacionadas con las conexiones de canales virtuales (VCC, “Virtual Channel Connection”). Una VCC es la unidad básica de conmutación en una red ATM. Una VCC se establece entre dos usuarios finales a través de la red, intercambiándose celdas de tamaño fijo a través de la conexión de un flujo full-duplex y de velocidad variable. Las VCC se utilizan también para señalización de control y gestión de red y encaminamiento.

Se introdujo una segunda capa de procesamiento en ATM para gestionar el concepto de camino virtual. Una conexión de camino virtual (VPC, “Virtual Path Connection”) es un conjunto de VCC con los mismos extremos, de manera que todas las celdas fluyendo a través de las VCC de una misma VPC se conmutan conjuntamente.

La técnica de camino virtual ayuda a contener el coste de control agrupando en una sola unidad conexiones que comparten caminos comunes a través de la red. Las acciones de la gestión de red pueden ser aplicadas a un pequeño número de grupos de conexiones, en lugar de a un gran número de conexiones individuales.

#### **42.1.2.2 CAPAS DE ATM**

Las normalizaciones ITU-U para ATM se basan en una arquitectura donde se contemplan los siguientes niveles:

- Capa física: Especifica el medio de transmisión y un esquema de codificación de la señal. Dividiéndose en dos capas:
  - Subcapa dependiente de un medio físico (PMD): que lleva a cabo funciones de transmisión y temporización de bits.
  - Subcapa de Convergencia de Transmisión (TC): responsable de las funciones relacionadas con la transmisión de células como control de HEC, delimitación de celdas, etc.
- Capa ATM: Define la transmisión de datos en celdas de tamaño fijo, al tiempo que establece el uso de conexiones lógicas. Realiza funciones de multiplexación de celdas y control de flujo.

- Capa de adaptación ATM (AAL): Capa de adaptación para admitir compatibilidad con protocolos de transferencia de información no basados en ATM. Se divide en dos:
  - Subcapa de Convergencia (CS).
  - Subcapa de Segmentación y reensamblado (SAR).

#### **42.1.2.2.1 CAPA FÍSICA**

La función básica del nivel físico es la codificación/decodificación de la información en formato eléctrico u óptico para la transmisión/recepción sobre el medio físico de comunicación utilizado. Otras funciones proporcionadas por este nivel son la desalineación de celdas y generación y proceso del checksum para el control de errores en la cabecera.

Los consejos ITU-T detallan la velocidad de transmisión y las técnicas de sincronización para la transmisión de celdas ATM. Las principales propuestas de capas físicas en redes ATM son las siguientes:

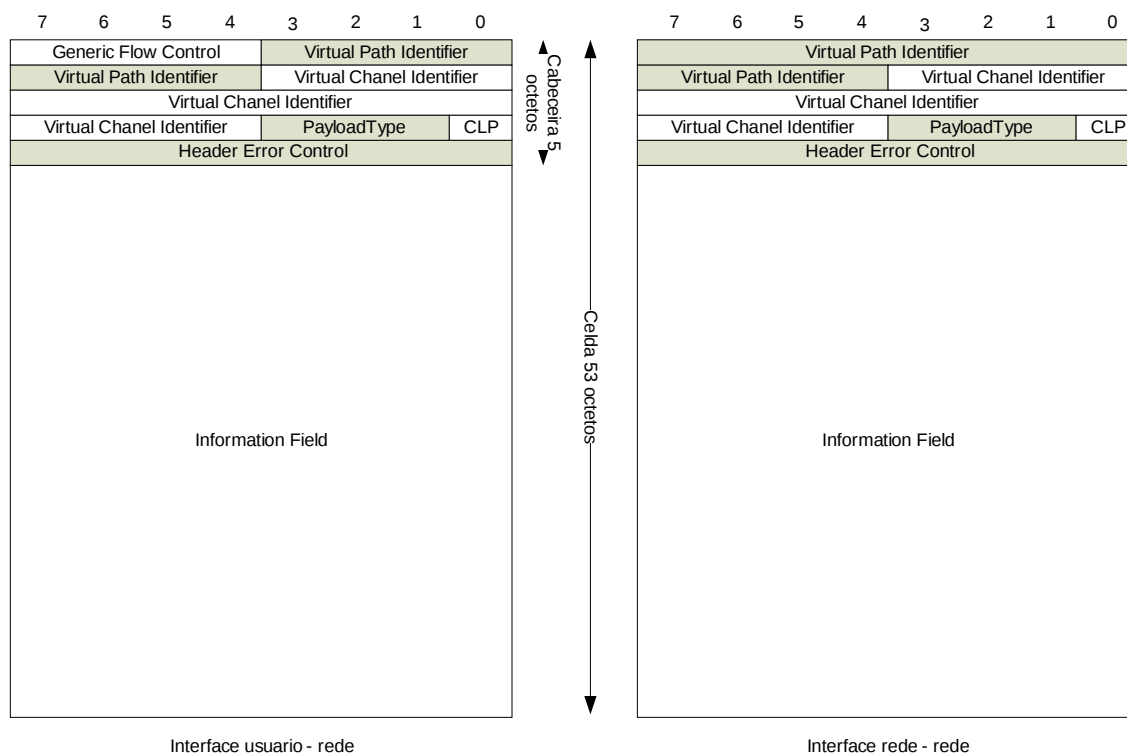
- ATM sobre SDH: STM-1 (155,52) y STM-4 (622,08)
- ATM sobre PDH: E1 (2,048), DS1 (1,548), DS2 (6,312), E3 (34,368), E4 (139,264) y DS3 (44,736).
- ATM a 100 Mbps sobre FDDI.
- ATM a 25,6 Mbps (propuesta por IBM).

#### **42.1.2.2.2 CAPA ATM**

El modo de transferencia asíncrono utiliza celdas de tamaño fijo, que constan de 5 octetos de cabecera y un campo de información de 48 octetos. La capa ATM es la encargada de incorporar la cabecera de 5 octetos al campo de información. La cabecera tiene el siguiente formato:

- Control de Flujo Genérico (GFC, “Generic Flow Control”). Se utiliza solo en la interfaz UNI (User-Network Interface).
- Identificador de camino Virtual (VPI, “Virtual Path Identifier”) e Identificador de Canal Virtual (VCI, “Virtual Channel Identifier”).
- Indicador de Tipo de Carga Útil (PTI, “Payload Type Indicator”). Indica si se trata de datos de usuario, información de gestión, información OAM, etc.

- Prioridad de Pérdida de Celda (CLP, “Cell Loss Priority”). Las celdas marcadas son las primeras en ser descartadas en caso de congestión.
- Control de Errores de Cabecera (“Checksum”).



#### 42.1.2.2.2.1 PARÁMETROS DE TRÁFICO

Cuando se establece una conexión ATM se constituye un “Contrato de tráfico” en el que se especifican los parámetros de tráfico y los de QoS, entre los más significativos están:

- PCR (“Peak Cell Rate”): Tasa máxima de celdas permitida sobre el circuito.
- MCR (“Minimun Cell Rate”): Mínima tasa de celdas sobre el circuito garantizada por el proveedor del servicio.
- CDVT (“Cell Delay Variation Tolerance”): Nivel de tolerancia para la diferencia entre la celda con el mínimo retardo y la celda con el máximo retardo.
- SCR (“Sustained Cell Rate”): Tasa media de transmisión de celdas que se mantendrá durante la duración de una transmisión.

- BT (“Burst Tolerance”): Límite que la transmisión puede alcanzar en su nivel más alto (PCR).

Se garantiza una calidad de servicio con respecto a un mínimo ancho de banda disponible, la cantidad de retardo que afectará a la transmisión y la máxima pérdida de celdas que se producirá en esta.

#### **42.1.2.2.2 CLASES DE SERVICIO**

A partir de estas especificaciones iniciales del ITU, se definieron cinco clases de servicio que la red ATM debería proporcionar (no es obligatorio):

- CBR (“Constant Bit Rate”): Es un servicio determinístico, diseñado para soportar emulación de circuitos, tráfico de voz y video (por ejemplo, MPEG/JPEG) en tasa constante de bits. Proporciona ancho de banda reservado, garantizado mínima pérdida de celdas y mínimas variaciones en retardos. El servicio CBR proporciona ancho de banda reservado hasta el PCR especificado para el circuito. Con CBR el usuario debe declarar el PCR y el CDTV en el momento de establecer la conexión.
- VBR (“Variable Bit Rate”): Se definieron dos tipos:
  - VBR-RT (“Real Time”), proporciona un estrecho control de los retardos para la transmisión de información como video y voz sin silencios.
  - VBR-NRT (“No Real Time”) tiene menos exigencias que el anterior respecto a las variaciones en retardos y se desarrolló para la transmisión de datos transaccionales.

El servicio VBR también requiere la especificación del PCR, si bien con un comportamiento diferente: el usuario puede utilizar el canal por encima del SCR (hasta el PCR) solo durante cortos períodos de tiempo determinados por el parámetro BT, pero debe mantener el SCR como una tasa media. Si el usuario sobrepasa el SCR durante un período de tiempo, a esto le seguirá un período similar por debajo del SCR. Si el usuario sobrepasa el PCR, esto se seguirá por un período



de inactividad antes de que el usuario pueda sobrepasar el SCR de nuevo. Con VBR el usuario debe declarar PCR, CDVT, SCR y BT.

- UBR (“Unspecified Bit Rate”): Se diseñó para permitir el uso de ancho de banda excedente no utilizado para los servicios CBR y VBR. No ofrece garantías en cuanto a la pérdida de celdas o variaciones en retardos; es decir, no incluye ningún mecanismo de control en el caso de congestión en la red. Con UBR no hay descriptores de tráfico, ni garantías de calidad de servicio ni mecanismos de realimentación en caso de congestión en la red. La estación puede enviar tráfico cuando lo necesite y la red lo aceptará; eso sí, en caso de congestión, la estación no es notificada de ningún modo y el conmutador eliminará celdas cuando sus buffers estén llenos. Esto significa que la tasa potencial de pérdida de celdas con UBR puede ser inaceptablemente alta.
- ABR (“Available Bit Rate”): Al igual que UBR se diseñó para aprovechar el ancho de banda excedente pero, al contrario que aquel, implementa mecanismos de control y control en caso de congestión en la red. ABR se concibió para transportar el tráfico a ráfagas sin las limitaciones de calidad de servicio de UBR, básicamente aplicaciones que no funcionen en tiempo real y por lo tanto poco sensibles a retardos. Utiliza básicamente los descriptores PCR y MCR; el usuario se compromete a no enviar información más rápido que el PCR y la red a proporcionar como mínimo el MCR requerido. El usuario no está obligado a especificar PCR y MCR; en ausencia de tal especificación, los valores por defecto serían el PCR a velocidad de acceso y el MCR a cero. Si se cumplen estos parámetros descriptores de tráfico, se garantiza la calidad de servicio en cuanto a mínimo nivel de pérdida de celdas y mínimo ancho de banda asegurado; el retardo de celdas será minimizado, pero no existe garantía absoluta respecto al retardo para el servicio ABR. Hay otro tipo de servicio en estudio por el ATM Forum (VBR+) que, como el

ABR, contempla un sistema de realimentación para control de la congestión en la red pero, adicionalmente, proporciona además garantías en cuanto a los retardos.

#### **42.1.2.2.3 ASIGNACIÓN DE ANCHO DE BANDA Y CONTROL DE CONGESTIÓN**

Una red ATM debe garantizar unos determinados parámetros de QoS, además de proporcionar ganancia estadística. Para conseguirlo utiliza métodos preventivos, denominados Control de Admisión de Conexión y de monitorización posterior, mediante una función de policía (UPC) que emplea diversos algoritmos para este fin.

También existen métodos reactivos como el CLP (Cell Loss Priority) o el GFC (Generic Flow Control) que controla el tráfico del usuario a la red.

Para el control de la congestión existen dos opciones:

- Control basado en créditos: el extremo receptor emite créditos, que indican el número de celdas que puede enviar el emisor, útil en entornos de área local.
- Control basado en la velocidad: basado en mensajes EFCI (Explicit Forward Congestion Indication), las estaciones y conmutadores ajustan la velocidad dinámicamente. Esta es la técnica más ampliamente utilizada.

#### **42.1.2.2.3 CAPA DE ADAPTACIÓN (AAL)**

La función básica del nivel de adaptación ATM (AAL) es proporcionar el enlace entre los servicios requeridos por los niveles superiores de red y el nivel ATM.

Hasta el momento, el ITU-T definió para la AAL cuatro clases de servicio, respondiendo esta clasificación a tres parámetros básicos: la relación de tiempo entre la fuente y el destino, tasa de bits constante o variable y modo de conexión. Las clases definidas son las siguientes:

<b>Clase</b>	<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>
<b>Relación origen / destino</b>	Sí	Sí	No	No

<b>Velocidad</b>	Constante	Variable	Variable	Variable
<b>Orientado a conexión</b>	Sí	Sí	Sí	No

#### **42.1.2.2.3.1 ESTRUCTURA DE LA CAPA AAL**

La capa AAL está organizada en dos subcapas:

- Subcapa de segmentación y reensamblado (SAR): Segmenta la información de las capas superiores para construir la carga útil de las celdas ATM y recíprocamente, reensambla los campos de información de las celdas en unidades de información para las capas superiores.
- Subcapa de convergencia (CS): tiene como misión realizar funciones específicas para cada servicio, como el tratamiento de la variación del retardo de celda, sincronización extremo a extremo, tratamiento de celdas mal insertadas o perdidas. Existen, por tanto, diferentes CS sobre la subcapa SAR. Debido a la gran cantidad de servicios propuestos sobre ATM, fue necesario distinguir entre una parte común CS (CPCS) y una parte específica de servicio (SSCS).

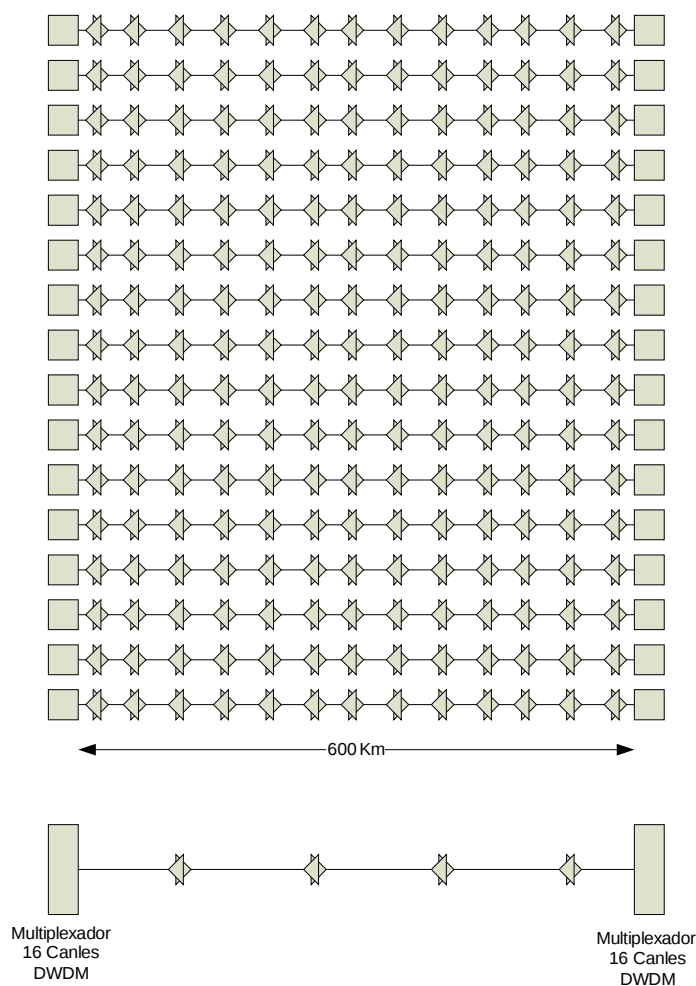
Inicialmente, el ITU-T recomendó cuatro tipos de protocolos AAL para soportar las cuatro clases de servicio definidas, los protocolos AAL de tipo 1, 2, 3, y 4. Así, el tráfico de clase 1 utilizará el protocolo AAL-1, el de clase 2 el AAL-2, y los de clase 3 y 4 el protocolo AAL-3/4. Siendo los protocolos de las clases 3 y 4 un protocolo único.

Debido a la complejidad del protocolo AAL-3/4 se propuso como alternativa el AAL-5, a veces denominado SEAL ("Simple and Efficient Adaptation Layer"). En consecuencia, las clases de tráfico 3 y 4 pueden utilizar el protocolo AAL-3/4 o el AAL-5.

#### **42.1.3 DWDM**

La tecnología DWDM usa una composición óptica de la señal de distintos flujos de datos, cada uno de los cuales en su propia longitud de onda óptica. A pesar de que la división y multiplexado usando el espectro óptico

es una tecnología que se conoce desde hace tiempo, sus primeras aplicaciones restringían su uso a proporcionar dos longitudes de onda muy gruesas y muy separadas o a la construcción de componentes capaces de incluso 4 canales. Solo recientemente la tecnología evolucionó hasta el punto de poder empaquetar e integrar en un sistema de transmisión una alta densidad de canales paralelos, simultáneos, a una frecuencia extremadamente alta (192 - 200 Terahertz). Conforme al plan de canales del ITU, este sistema asegura la interoperabilidad con otros equipos y permite a los operadores posicionarse bien para desplegar soluciones ópticas en su red. El sistema de 16 canales proporciona básicamente un cable virtual con 16 fibras.



Para transmitir 40Gb/s a 600 Km usando un sistema tradicional requerimos 16 pares de fibra óptica con regeneradores cada 35 km con un total de 272

regeneradores. Un sistema DWDM de 16 canales usa sólo un par de fibra óptica y 4 amplificadores posicionados cada 120Km.

La forma más común de DWDM usa un par de fibra (una para transmitir y otra para recibir). Aunque existen sistemas que solo usan una fibra para transmitir y recibir, estos sistemas deben sacrificar un poco de la capacidad de la fibra óptica para una banda de guardia y evitar así la mezcla de canales reduciendo también el rendimiento de los amplificadores.

Adicionalmente, existe un gran riesgo de que los reflejos producidos durante el mantenimiento o reparación puedan dañar los amplificadores.

La disponibilidad de tecnologías maduras de soporte como los multiplexadores precisos y los EDFA (Erbium Doped Fiber Amplifiers) permitió la disponibilidad comercial de sistemas DWDM con ocho, dieciséis, o incluso un mayor número de canales.

#### **42.1.3.1 DEMULTIPLEXADORES**

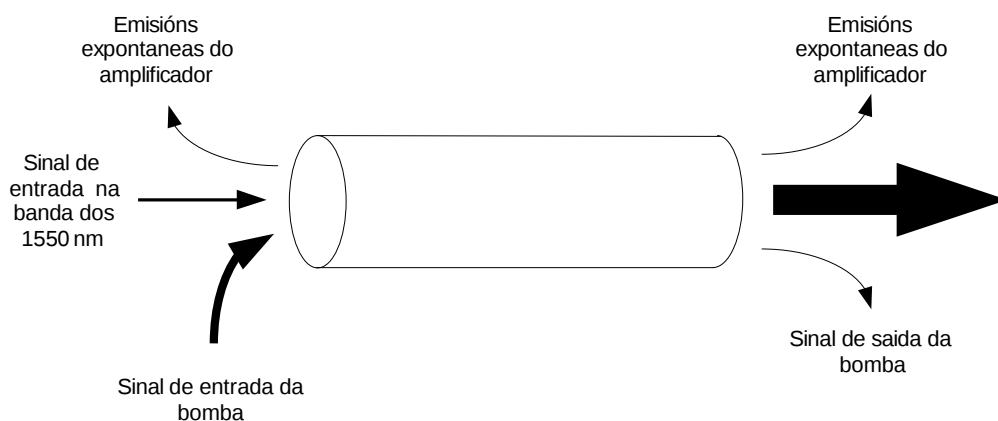
Con señales tan precisas y densas como las usadas en DWDM, tiene que existir una forma que aporte una precisa separación de las señales, o filtrado, en el receptor óptico. Esta solución también tiene que ser fácil de implementar y no requerir mantenimiento. Los sistemas primitivos de filtrado eran o demasiado imprecisos para DWDM o demasiado sensibles a variaciones de temperatura y polarización, demasiado vulnerables a cruces de comunicaciones en canales adyacentes o demasiado caros. Esto restringió la evolución de DWDM. Para conseguir satisfacer los requisitos de alto rendimiento se desarrolló una nueva tecnología de filtrado que hizo DWDM posible a un coste aceptable: la fibra con rejilla de Bragg.

El nuevo componente de filtro (fibra con rejilla) consiste en una determinada longitud de fibra óptica donde el índice de refracción del núcleo fue permanentemente modificado en puntos equidistantes, generalmente por exposición a un patrón de interferencia ultravioleta. El resultado es un componente que refleja la luz dependiendo de la longitud de onda y es útil para separar longitudes de onda. En otras palabras, la rejilla crea un filtro altamente selectivo para una longitud de onda muy estrecha que funciona de forma similar a un espejo y aporta mayor

selectividad de longitud de onda que cualquier otra tecnología. Como este es un dispositivo pasivo, fabricado en fibra de vidrio, es fuerte y durable.

#### **42.1.3.2 ERBIUM DOPED FIBER AMPLIFIER**

La llegada del EDFA permitió el desarrollo de sistemas DWDM comerciales proporcionando una forma de amplificar todas las longitudes de onda al mismo tiempo. Esta amplificación óptica se hace incorporando iones de Erblio en el núcleo de una fibra especial en un proceso conocido como dopado. Se usan bombas ópticas láser para transferir altos niveles de energía a la fibra especial, energizando los iones de Erblio que luego aumentan a las señales ópticas que pasan por la fibra. La estructura atómica del Erblio proporciona amplificación a los grandes rangos del espectro que se requieren para DWDM.



En vez de múltiples regeneradores electrónicos, que requieren que la señal óptica sea convertida a señal electrónica y viceversa, el EDFA amplifica directamente las señales ópticas. De esta forma la señal puede ser enviada hasta 600 Km sin regeneración y hasta 120Km entre amplificadores en un sistema DWDM disponible comercialmente.

#### **42.1.4 MPLS**

MPLS (MultiProtocol Label Switching) es una solución versátil para resolver los problemas que afrontan las redes actuales de velocidad, escalabilidad, gestión de la calidad de servicio (QoS) e ingeniería del tráfico. MPLS emergió como una solución elegante para aportar la gestión de ancho de banda y requerimientos de servicio para la nueva generación de redes troncales basadas en IP. MPLS resuelve problemas relacionados con la

escalabilidad y encaminamiento (basados en QoS y métricas de la calidad de servicio) y puede existir sobre redes ATM y Frame Relay ya existentes.

#### **42.1.4.1 FUNCIONES DE MPLS**

MPLS realiza las siguientes funciones:

- Especifica mecanismos para gestionar los flujos de tráfico de varias granularidades, como los flujos entre diferente hardware, máquinas o incluso entre diferentes aplicaciones.
- Se mantiene independiente de los protocolos de las capas 2 y 3.
- Aporta un método para mapear direcciones IP a simples etiquetas de tamaño fijo, usadas por diferentes tecnologías.
- Se integrara con protocolos existentes como RSVP (Resource Reservation Protocol) y OSPF (Open Shortest Path First).
- Da soporte a IP, ATM y Frame Relay.

En MPLS la transmisión de datos se produce en LSPs (Label Switching Paths ). Un LSP es una secuencia de etiquetas para cada uno de los nodos a lo largo del camino, desde el origen hasta el destino. Los LSPs se establecen antes de la transmisión de los datos (establecimiento por control) o ante la detección de un determinado flujo de datos (establecimiento por datos).

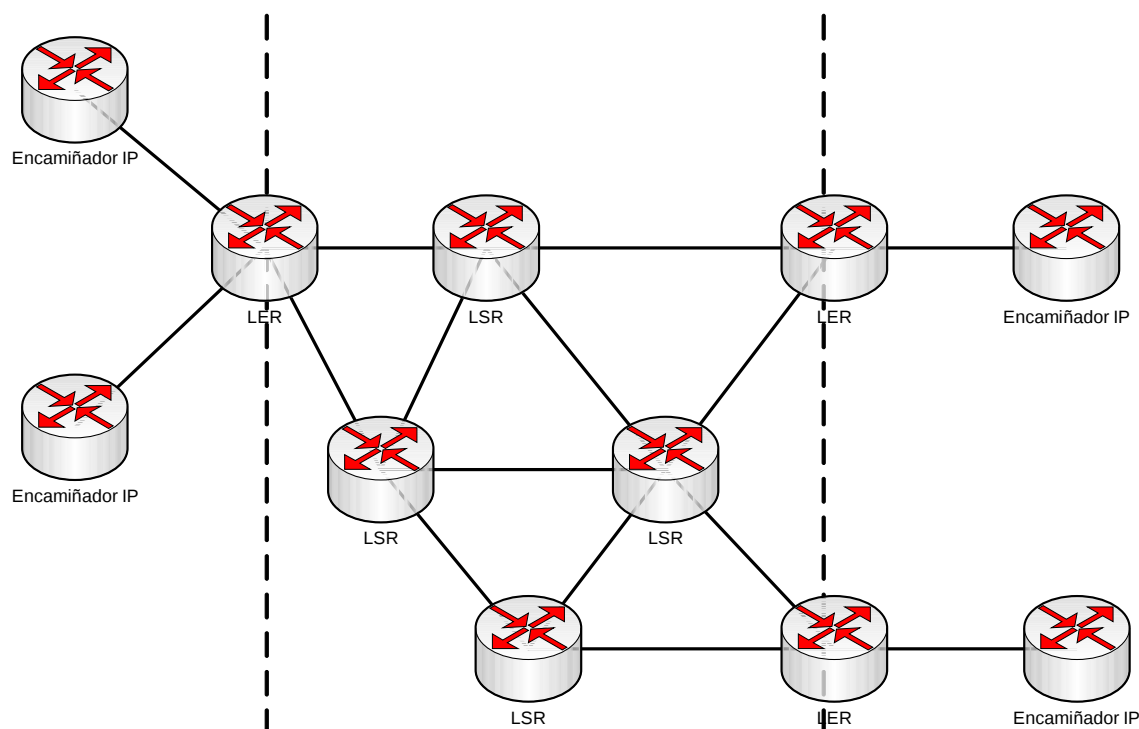
Las etiquetas (que son identificadores específicos de los protocolos subyacentes) se distribuyen usando LDP (Label Distribution Protocol) o RSVP o acompañando los datos de protocolos de encaminamiento como el BGP (Border Gateway Protocol) y OSPF. Cada paquete de datos encapsula y transporta las etiquetas durante su trayecto desde el origen hasta el destino. La alta velocidad de conmutación es posible por la longitud fija de las etiquetas y porque estas son insertadas al principio del paquete o celda y esto permite que hardware específico las use para conmutar paquetes a alta velocidad entre distintos enlaces.

#### **42.1.4.2 LSRS Y LERS**

Los dispositivos que participan en los mecanismos del protocolo MPLS pueden clasificarse en LERS (Label Edge Routers) y LSRS (Label Switching Routers).

Un LSR y un encaminador de alta velocidad en el núcleo de una red MPLS que participa en el establecimiento de los LSPs usando el protocolo de señales adecuado y proporciona conmutación de alta velocidad de los datos basándose en los caminos establecidos.

Un LER es un dispositivo que opera en el límite entre la red de acceso y de la red MPLS. LERs soportan múltiples puertos conectados a diferentes redes (Frame Relay, ATM, Ethernet,...) y envía el tráfico a la red MPLS después de establecer su LSP, también distribuye el tráfico a la red de acceso en el proceso inverso. El LER juega un papel muy importante en la asignación y renovación de etiquetas, cuando el tráfico entra o sale de una red MPLS.



#### 42.1.4.3 FEC

A FEC (Forward Equivalent Class) es la representación de un grupo de paquetes que comparten los mismos requisitos de transporte. Todos los paquetes de este grupo reciben el mismo tratamiento en su ruta hacia el destino. Al contrario que IP, en MPLS, la asignación de un determinado paquete a un FEC se realiza solo una vez, cuando el paquete entra en la red. Los FECs están basados en los requisitos del servicio para un conjunto de paquetes o simplemente en un determinado prefijo de red. Cada LSR



construye una tabla para saber cómo un paquete debe ser enviado. Esta tabla, llamada LIB (Label Information Base) se compone de relaciones FEC – etiqueta.

#### **42.1.4.4 ETIQUETAS**

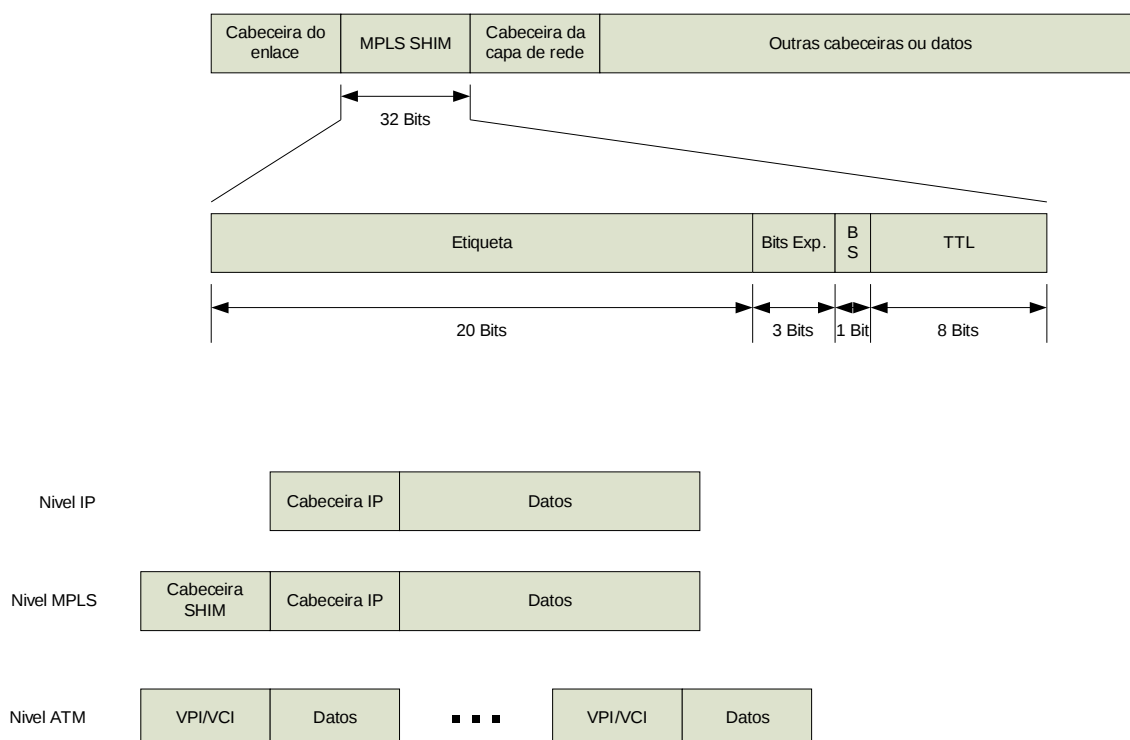
Una etiqueta en su forma más simple, identifica el camino que debe recorrer un paquete. Las etiquetas se encapsulan en una cabecera de nivel 2 junto con el paquete. El encaminador que lo recibe, examina el paquete para obtener su etiqueta para determinar el siguiente salto. Una vez que el paquete esta etiquetado, el resto del camino por la red troncal se basa en conmutación por etiquetas. Los valores de las etiquetas sólo tienen valor local, esto quiere decir que solo pertenecen a saltos entre LSRs concretos. Cuando un paquete se clasifica como un FEC nuevo o existente, le asigna una etiqueta. Los valores de la etiqueta se derivan de la capa de enlace inferior (DLCI para Frame Relay, VPI/ VCI para ATM,...).

Las etiquetas se enlazan con los FEC como resultado de un evento o política que indica la necesidad de ese enlace. Estos eventos pueden tener como origen las señales de control o las propias transmisiones de datos siendo estas últimas la mejor opción por sus propiedades de escalado.

La asignación de etiquetas se basa en criterios de encaminado como:

- Destino Unicast
- Ingeniería de tráfico
- Multicast
- VPN (Virtual Private Network)
- QoS

En la siguiente imagen podemos ver la estructura de una etiqueta MPLS y un ejemplo del proceso desde IP incluso la capa de enlace (en este caso sobre ATM).



#### 42.1.4.5 DISTRIBUCIÓN DE ETIQUETAS

MPLS no imponen un solo método de distribución de etiquetas. Existen protocolos de encaminado, como BGP y RSVP que fueron mejorados para poder incorporar (usando el método *poggyback*) la información de las etiquetas junto con la información del protocolo. El IETF (Internet Engineering Task Force) también definió un protocolo llamado LDP (Label Distribution Protocol) para gestionar las etiquetas. Extensiones de LDP permiten definir rutas explícitas basadas en requisitos de Qos. Estas extensiones están recogidas en la definición del protocolo CR-LSP (Constrint-based Routing-LDP).

Un resumen de los varios esquemas para el intercambio de etiquetas es el siguiente:

- LDP: mapea direcciones de unicast IP a etiquetas.
- RSVP, CR-LDP: se usan para ingeniería del tráfico y reserva de recursos.
- PIM (Protocol Independent Multicast): se usa para mapear multicasts a etiquetas.
- BGP: etiquetas externas (VPN)

#### 42.1.4.6 LSP

Una colección de dispositivos MPLS se define como un dominio MPLS.

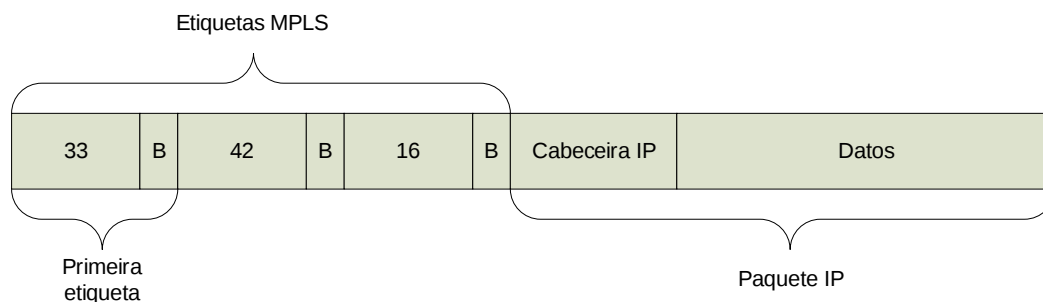
Dentro de un dominio MPLS se fija un camino para un paquete basándose en su FEC. El LSP se fija antes de la transmisión de los datos usando una de estas dos opciones:

- Encaminado salto a salto (hop-by-hop): cada LSR selecciona independientemente el siguiente salto para un FEC determinado. Esta es una metodología muy similar a la usada en redes IP. El LSR usa los protocolos de encaminamiento disponibles como OSPF, etc.
- Encaminado explícito (ER-LSP): El LSR de entrada (el LSR donde el flujo de datos comienza en la red) especifica una lista de nodos que el ER-LSP atraviesa. El camino podría no ser óptimo. Los recursos necesarios podrían ser reservados para garantizar la QoS. Esto facilita la ingeniería de tráfico en la red, y permite que se den servicios diferentes usando flujos basados en métodos de políticas o gestión de red.

Un LSP para un determinado FEC es unidireccional en naturaleza, el tráfico de retorno tendrá que usar otro LSP.

#### 42.1.4.7 PILA DE ETIQUETAS

El apilado de etiquetas (Label Stack) es un mecanismo que permite la operación jerárquica dentro de un dominio MPLS. Básicamente permite la MPLS ser usado simultáneamente para el encaminado a nivel fino (entre encaminadores individuales dentro de un ISP) y en un nivel grueso (dominio a dominio). Cada nivel, en una pila de etiquetas, pertenece a un nivel jerárquico. Esto facilita el uso de túneles en MPLS.



### 42.2 REDES DE FIBRA ÓPTICA

#### **42.2.1 JERARQUÍA DIGITAL PLESIÓCRONA (PDH)**

A PDH (Plesiochronous Digital Hierarchy) es una tecnología usada en redes de telecomunicaciones para transportar una gran cantidad de datos sobre redes de fibra óptica o radioenlaces. El término alude a que las diferentes partes de la red funcionan de forma casi sincronizada, pero no totalmente. La mayor parte de los operadores están actualizando PDH a SDH o SONET (capaces de transmitir a más velocidad) si no la reemplazan directamente por una tecnología completamente distinta.

PDH permite la transmisión de datos a una velocidad similar entre las transmisiones, pero permitiendo variaciones sobre la velocidad nominal, ya que no hay una sincronización exacta dentro de la red.

La velocidad de transferencia básica y de 2 Mb/s, para la transferencia de voz se divide en 30 canales de 64Kb/s y dos canales para señalización y sincronización; además todo el enlace (2Mb/s) se puede usar para transmitir datos.

La velocidad de transmisión está controlada por un reloj en el equipo que envía los datos, esta velocidad puede variar 50 ppm desde los 2 Mb/s esto quiere decir que las distintas transmisiones pueden estar sucediendo a diferentes velocidades (y probablemente lo están).

Para transmitir varios flujos de datos desde un origen, estos se multiplexan en grupos de 4 de forma que primero se transmite el bit 1 del flujo 1, luego el bit 1 del flujo 2, luego el bit 1 del flujo 3, luego el bit 1 del flujo 4 y así sucesivamente. El transmisor también añade información para poder reconstruir el flujo cuando se recibe. Como los flujos de datos transmitidos pueden no estar siendo recibidos a la misma velocidad en el multiplexador de origen, este supone que los está recibiendo a la máxima velocidad posible. Esta suposición provoca que a veces no se haya recibido el bit correspondiente de un flujo, esta situación debe ser notificada al multiplexador de destino para que este pueda reconstruir los flujos a la velocidad correcta.

La velocidad resultante de la multiplexación descrita es de 8.448 kbit/s, técnicas similares permiten combinar 4 streams de 8 Mb/s más bit de

relleno (proporcionando 34 Mb/s), 4 streams de 34Mb/s (proporcionando 140Mb/s) y 4 streams de 140 Mb/s (proporcionando 565Mb/s). 565 Mbit/s es la velocidad típica para transmitir datos sobre fibra óptica para largas distancias.

#### **42.2.2 JERARQUÍA DIGITAL SÍNCRONA (SDH) / SONET**

Synchronous optical networking (SONET) y Synchronous Digital Hierarchy (SDH) son protocolos de multiplexación que transmiten varios streams sobre una red de fibra óptica, aunque también se pueden usar sobre interfaces eléctricos (a menor velocidad). Esta técnica se desarrolló para reemplazar PDH en la tarea de transportar un ancho número de llamadas de teléfono y tráfico de datos sobre la misma fibra en problemas de señalización. La mayor diferencia con PDH es que SDH / SONET está sincronizada usando relojes atómicos reduciendo la necesidad de buffers en la red y aprovechándola mejor.

SONET y SDH, son básicamente idénticos, y, aunque SONET (ANSI TI.105) es anterior, dado que solo se usa en Canadá y EEUU mientras que SDH (ITU G.707, G.783, G.784 y G.803) se usa en el resto del mundo, SONET se considera una variación de SDH. Dada la gran similitud entre ellos es extremadamente fácil la interconexión entre los dos a cualquier velocidad.

##### **42.2.2.1 FRAME SDH / SONET**

La unidad básica para la transmisión en SDH es la Synchronous Transport Module, level 1 (STM-1), que es transmitida a 155,52Mb/s. SONET cambia el nombre de esta estructura a Synchronous Transport Signal 3 concatenated (STS-3c) u OC-3c dependiendo de si la señal se transporta eléctricamente (STS) u ópticamente (OC) pero su funcionalidad, velocidad y tamaño son iguales a STM-1.

SONET proporciona otra unidad STS-1 u OC-1 que opera a 51,84Mb/s (un tercio de STM-1) esto busca poder transmitir un canal DS-3 estándar (672 canales de 64Kb/s para voz).

##### **42.2.2.1.1 FRAMING**

En tecnologías como Ethernet la trama consiste en una cabecera y en un conjunto de datos, la cabecera es transmitida primero seguida de los datos

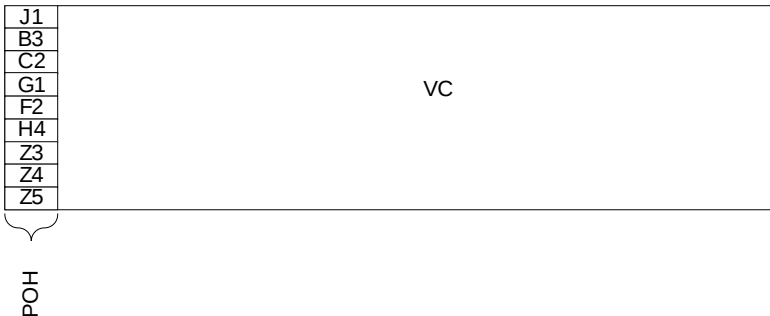
y posiblemente de una cola conteniendo el CRC o equivalente. En SDH esto se modifica ligeramente, la cabecera se denomina overhead y en vez de ser transmitida antes que el resto de los datos es entrelazada con ellos durante la transmisión: se transmite parte de la cabecera, luego parte de los datos,... hasta que se transmite todo el frame.

En el caso de STS-1, cada frame está compuesto de 810 octetos, mientras que en el caso de STM-1/STS-3 cada frame está compuesto de 2.430 octetos. STS-1 transmite 3 octetos de overhead seguidos de 87 de datos durante nueve veces hasta transmitir los 810 octetos en 125 microsegundos. En el caso de STM-1 (que opera a tres veces a velocidad de STS-1) se transmiten 9 octetos de overhead y 261 de datos, también 9 veces hasta que se transmite en los 2.430 octetos llevando también 125 microsegundos. Esto suele representarse gráficamente dibujando el frame como un bloque de 90 columnas y 9 filas para STS-1 y 270 columnas y 9 filas para STM-1 de esta forma a representación alinea toda la overhead y todos los datos de la payload.

La estructura interna de la overhead y de los datos transmitidos dentro del frame varían ligeramente para SONET y SDH, usando también diferentes nombres para describir estructuras.

#### **42.2.2.1.2 ESTRUCTURA DEL FRAME STM-1**

La Overhead Section y los Administrative Unit Pointers ocupan las 9 primeras columnas del frame. Estos punteros (los bytes H1, H2 y H3) identifican AUs (Administrative Units) dentro de la carga útil. Cada una de estas unidades administrativas puede albergar uno o varios Virtual Containers (VC) que su vez contienen una descripción del camino (Path OverHead, POH) y datos. La primera columna es para el POH y el resto son para datos que pueden ser a su vez otros VC. Las unidades administrativas pueden tener cualquier tipo de alineación y es esta alineación la que es indicada por el puntero de la fila 4.



Los componentes del frame son:

- Overhead que contiene:
  - o SOH (Section OverHead) contiene información para el sistema de transmisión (para control de calidad, detección de fallas,...) y está dividida en dos partes:
    - RSOH (Regenerator Section Overhead) también conocida como Section overhead: compuesta por 27 octetos que contiene información sobre la estructura del frame.
    - MSOH (Multiplex Section Overhead) también conocida por Line overhead : está compuesta 45 octetos con información sobre corrección de errores y mensajes Automatic Protection Switching (como podrían ser alarmas y mensajes de mantenimiento)

- o Puntero AU: Apunta a la localización del byte J1 en los datos (el primer byte del VC).
- Datos Path: los datos transmitidos de extremo a extremo se llaman datos path y están compuestos de 2 elementos:
  - o POH (Path OverHead): nueve octetos para señalización extremo a extremo.
  - o Datos de usuario: 2340 octetos de datos

### **42.3 REDES DE NUEVA GENERACIÓN (NGN)**

Según el ITU-T, una NGN es una red basada en paquetes que puede proporcionar servicios, incluyendo servicios de telecomunicaciones, y capaz de usar varias tecnologías de banda ancha que incorporen capacidades de calidad de servicio. En estas redes las funciones relacionadas con el servicio deben ser independientes de la tecnología de transporte usada. Ofrecen un acceso sin restricciones al usuario a diferentes proveedores de servicio. Soporta la movilidad lo que permitirá un acopio de servicio consistente y ubicuo a los usuarios.

Desde un punto de vista práctico, NGN implica tres cambios en la arquitectura:

- En la red troncal, NGN implica la consolidación de diversas redes de transporte, cada una de las cuales construida históricamente para un servicio diferente en una sola red troncal de transporte (normalmente basada en IP y Ethernet). Implica entre otras, la migración de la voz desde una arquitectura de conmutación de circuitos (RTC) a VoIP, y también la migración de servicios como X.25 y Frame Relay (ya sea una migración comercial a nivel de usuario con servicios como VPN sobre IP, o la migración técnica emulando estos servicios pero sobre la NGN).
- En la red de acceso por par de cobre, NGN implica la migración desde el sistema dual que la voz independiente del xDSL en la central a la que llega el bucle de abonado, a una configuración donde los DSLAMs (Digital Subscriber Line Access Multiplexer, el punto donde se



concentran todas a las conexiones DSL de la central) integran puertos de voz o VoIP, posibilitando la eliminación de la infraestructura de conmutación de voz.

- En la red de acceso de cable, la convergencia NGN implica la migración de la voz de servicios CBR a estándares de VoIP y SIP.

En una NGN existe una mayor distancia entre la parte de la red que proporciona el transporte (conectividad) y los servicios que se ejecutan sobre este. Esto tiene como consecuencia que cada vez que un operador quiere dar un nuevo servicio solo tiene que definir al nivel de la capa de servicios sin preocuparse por el transporte. Cada vez más los servicios (incluyendo los de voz) tienden a ser independientes de la red de acceso y residen más en los equipos de usuario final (PC, Set-Top Box,...).

#### **42.3.1 TECNOLOGÍAS DE SOPORTE**

Las redes de nueva generación están basadas en tecnologías como IP y MPLS. A nivel de aplicación SIP (Session Initiation Protocol) está reemplazando a H.323. Aunque originalmente H.323 era el protocolo de VoIP / videoconferencia /... sobre redes IP más popular, sus limitaciones para atravesar NAT (Network address translation) y firewalls reducen su implantación a nivel del bucle de abonado. Son estas algunas de las razones (junto con la complejidad de H.323) que están llevando a la implantación de SIP, sobre todo en el bucle de abonado. Sin embargo, en las redes de operador (donde todo está bajo su control) muchos de ellos usan H.323 para a sus redes troncales.

Con los nuevos cambios introducidos en H.323 es posible que dispositivos H.323 atraviesen NAT y firewalls fácilmente, esta circunstancia puede propiciar que H.323 pueda volver a ser usado en entornos donde esto sea necesario.

Como contrapartida muchos operadores están investigando y dando soporte a IMS (IP Multimedia Subsystem, una arquitectura estandarizada para servicios multimedia en Internet definida por el ETSI y la 3GPP) que daría a SIP la oportunidad de ser el protocolo más usado.

Para aplicaciones de voz el dispositivo más importante de la NGN y el Softswitch (este nombre y sus funciones aún son muy dependientes del fabricante), un dispositivo que controla las llamadas VoIP permitiendo la correcta integración de diferentes protocolos dentro de la NGN. Su función principal es crear la interfaz a las redes telefónicas existentes (RTC).

Uno de los términos más comúnmente usados es el de GateKeeper, que originalmente se refería a un dispositivo que transformaba voz y datos desde sus formatos analógicos a IP. Cuando este dispositivo comenzó a utilizar Media Gateway Control Protocol pasó a llamarse Media Gateway Controller (MGC).

Un Agente (Call Agent, SIP Agent,...) es un nombre general para dispositivos capaces de controlar llamadas.

#### **42.4 BIBLIOGRAFÍA**

- MultiProtocol Label Switching – The International Engineering Consortium
- Dense Wavelength Division Multiplexing - ATG's Communications & Networking Technology

**Autor:** Matías Villanueva Sampayo

Director de Informática Asociación Provincial de Pensionistas y Jubilados de A Coruña

Colegiado del CPEIG

# **INFRAESTRUCTURA DE SISTEMAS.**



# **44. CONCEPTO, EVOLUCIÓN Y TENDENCIAS DE LOS SISTEMAS OPERATIVOS. SISTEMA OPERATIVO UNIX- LINUX. SISTEMA OPERATIVO WINDOWS.**

## **Tema 44. Concepto, evolución y tendencias de los sistemas operativos. Sistema operativo UNIX-LINUX. Sistema operativo Windows**

---

### **ÍNDICE**

<b>44.1.- Sistema Operativo.....</b>	<b>1</b>
44.1.1 Concepto.....	1
44.1.2 Evolución de los Sistemas Operativos.....	2
44.1.3 Clasificaciones de los Sistemas Operativos.....	4
44.1.4 Estructura de los Sistemas Operativos.....	5
44.1.5 Funciones de un Sistema Operativo.....	5
44.1.6 Tendencias.....	6
<b>44.2.- Sistema Operativo UNIX-LINUX.....</b>	<b>7</b>
44.2.1 Características.....	7
44.2.2 Arquitectura de Unix-Linux.....	8
44.2.3 Gestión de procesos.....	8
44.2.4 Gestión de memoria.....	9
44.2.5 Gestión de E/S.....	9
44.2.6 Gestión de archivos.....	10
<b>44.3.- Sistema Operativo Windows.....</b>	<b>11</b>
44.3.1 Características.....	12
44.3.2 Arquitectura de Windows.....	12
44.3.3 Gestión de procesos.....	13
44.3.4 Gestión de memoria.....	14
44.3.5 Gestión de E/S.....	14
44.3.6 Gestión de archivos.....	15
44.3.7 Seguridad.....	16
<b>44.4.- Bibliografía.....</b>	<b>16</b>

## **44.1.- SISTEMA OPERATIVO.**

### **44.1.1 Concepto**

Se puede definir un sistema operativo (S.O.) como un conjunto de programas que controlan directamente los recursos hardware (HW) o físicos de un ordenador (CPU, memoria principal y periféricos) proporcionando una máquina virtual que oculta los detalles físicos de la máquina y ofrece al hombre un entorno más amigable. El sistema operativo es la capa de software más baja de un ordenador. Cada capa oculta a las capas superiores ciertos detalles de las capas inferiores. De esta forma se construye el software, basándonos en lo que ya existe

El sistema operativo tiene una serie de funciones, que se pueden agrupar en 3:

1. Inicializar la máquina. Prepara la máquina para el funcionamiento. Hay 2 formas de inicialización:

- a) TOTAL: Inicialización de todas las funciones y servicios que la máquina puede ofrecer. Por ejemplo, MS-DOS tiene inicialización total.
- b) PARCIALMENTE: Se va a ser selectivo con los tipos de servicios que se inician. Por ejemplo, Linux y Windows.

La principal ventaja/utilidad de la inicialización parcial es la recuperación de la máquina ante fallos: consiste en que si falla un servicio de la máquina no hace falta apagar toda la máquina, sólo hay que lanzar de nuevo el servicio.

2. Servir de máquina virtual. Se ocultan detalles de hardware proporcionando un entorno más amigable. Esto tiene 2 objetivos:

- a) La SEGURIDAD: En lugar de que el usuario acceda directamente a un recurso HW, lo hace el S.O. para que no se produzcan operaciones no

deseadas, también llamadas operaciones peligrosas: operaciones de entrada/salida (E/S), operaciones de acceso a memoria.

El HW tiene dos formas de actuar: modo supervisor y modo usuario.

Todos los programas actuarán en modo usuario hasta el momento en que haya que acceder al HW; será entonces cuando cambia a modo supervisor para evitar las operaciones que puedan causar problemas. Se generará una interrupción al realizarse una de estas operaciones susceptibles de fallo. Esa interrupción la coge el sistema operativo y actúa en consecuencia. El sistema operativo tomará el control del hardware y realizará la operación que se le indica.

Una interrupción es una señal física que generan los dispositivos del sistema y que la trata el sistema operativo.

Al conjunto de interrupciones se le llama **interface interna** del sistema operativo.

- b) **ABSTRACCIÓN**: Se abstraen las características físicas y reales de la máquina ofreciendo una serie de servicios incluso mayores de los que puede ofrecer la propia máquina. Por ejemplo para trabajar con ficheros se utilizan nombres, pero el ordenador no utiliza esos nombres para referirse a ellos; emplea una dirección.

La E/S: cuando tecleamos un dato lo vemos en pantalla tal como lo hemos imaginado, aunque realmente para la máquina son unos y ceros.

Esto constituye la **interface externa** del S.O., el lenguaje con el que nos vamos a comunicar con él. Se denomina SHELL.

Por tanto, tenemos 2 tipos de interface:

- La externa: forma de comunicación entre nosotros y el ordenador; a través de comandos/órdenes (abstracción).

- La interna: forma de comunicación del sistema operativo con el hardware (modo supervisor).

3. Administrar los recursos para su funcionamiento. Esta administración tiene que cumplir 3 características:

- tiene que ser CORRECTA: si hay 2 procesos que quieren acceder a un recurso, hay que darle acceso primero a uno y luego a otro, pero no mezclarlos.
- tiene que ser JUSTA: si tenemos 2 procesos hay que darle salida a los dos; un proceso no puede monopolizar.
- tiene que ser EFICIENTE: para mejorar el rendimiento del sistema.

Por último decir que un S.O. tiene que tener estas 2 características:

- DETERMINISMO: si se repite la misma operación con los mismos datos de entrada debemos obtener los mismos resultados.
- INDETERMINISMO: en el sentido de que tiene que responder oportunamente a las interrupciones, es decir, no sabe qué interrupción va a llegar primero, no conoce el orden pero debe saber tratarlas.

#### ***44.1.2 Evolución de los Sistemas Operativos***

Los sistemas operativos han evolucionado en paralelo al desarrollo del HW. Conforme el HW iba incorporando nuevas capacidades, los S.O. debían adaptarse para permitir gestionar eficientemente esas nuevas capacidades. La evolución de S.O. puede organizarse en generaciones con algunas características comunes:

##### **Primera Generación**

Abarca desde 1945 a 1955. Se caracteriza porque no existía un sistema operativo. Eran los propios diseñadores de las máquinas los que las programaban a través de cableado, y las manejaban.



## **Segunda Generación**

Desde 1955 a 1965. En el campo del hardware aparece el transistor. Se empiezan a utilizar las tarjetas perforadas. Se distinguen 2 tipos de sistema operativo:

- Monitor Residente: el S.O. se limitaba a cargar los programas a memoria, leyéndolos de tarjetas perforadas, y ejecutarlos. El problema era encontrar una forma de optimizar el tiempo entre la retirada de un trabajo y el montaje del siguiente.
- Trabajo por lotes: se utilizan las cintas magnéticas. Como solución para optimizar el tiempo de montaje surgió la idea de agrupar los trabajos en lotes, en una misma cinta o conjunto de tarjetas, de forma que se ejecutaran uno a continuación de otro sin perder apenas tiempo en la transición. En la memoria del ordenador existen dos ítems: a) el monitor de lotes: indicando qué trabajo se está ejecutando y b) el trabajo actual.

Estos dos tipos de sistema operativo se caracterizan por:

- No existe ningún planificador (el que decide qué trabajo se va a realizar): la razón de su inexistencia es que no es necesario, ya que sólo hay un trabajo en memoria principal
- Tampoco existe un reloj (mide el tiempo que un trabajo está ocupando la CPU): no es necesario porque sólo hay un proceso en memoria ejecutándose.

## **Tercera Generación**

Abarca desde 1965 a 1980. En el campo del hardware tenemos los circuitos integrados con tecnología LSI y VLSI. Antes de ver los distintos S.O., merece la pena dejar claros dos conceptos: Programa: código (algo estático) y Proceso: programa en ejecución (algo dinámico).



- S.O. Multiprogramación. En memoria principal va a haber más de un programa. La CPU ejecuta instrucciones de un programa, cuando el que se encuentra en ejecución realiza una operación de E/S, en lugar de esperar a que termine la operación de E/S, se pasa a ejecutar otro programa. De esta forma es posible, teniendo almacenado un conjunto adecuado de tareas en cada momento, utilizar de manera óptima los recursos disponibles.

Se va a definir el grado de multiprogramación como el número de programas que hay actualmente en memoria

OverHead es un parámetro que mide la diferencia de tiempo que hay entre que el sistema operativo está dedicado a hacer sus tareas y el tiempo dedicado al cambio de contexto entre procesos.

- S.O. Multiproceso. Tiene varios procesos en memoria principal. Hay que distinguir que:
  - o *Multiprogramación IMPLICA Multiproceso.* Multiprogramación: varios programas en memoria principal. Al estar en memoria principal se están ejecutando. Programas en ejecución = procesos.
  - o *Multiproceso NO IMPLICA Multiprogramación* Ahora bien, podemos tener únicamente un programa en memoria y este programa lanzar varios procesos. Un programa puede querer imprimir, leer algo por teclado, ... lanzar procesos. Varios procesos = multiproceso
- S.O. Multiprocesador. Se utilizan donde hay 2 CPU's o más.
- S.O. Interactivos. Sistemas que de alguna forma mantienen un diálogo con el usuario; mediante el SHELL (lenguaje de comandos).

Es muy importante el tiempo de respuesta: tiempo que transcurre entre que el usuario manda hacer algo al sistema hasta que obtiene la respuesta. Siempre se intenta minimizar el tiempo de respuesta.

- S.O. Multiusuario. Aquellos sistemas operativos en los cuales varios usuarios pueden acceder al mismo ordenador simultáneamente.

Por ejemplo: cualquier sistema UNIX o LINUX. Se puede tener una máquina y generar usuarios accediendo a esa máquina (a través de red local, Internet, ...).

- S.O. de Tiempo Compartido. Pretenden dotar a cada persona de una parte de la CPU. El usuario ve el ordenador como suyo propio, aunque no lo es.
- S.O. de Tiempo real. Estos sistemas se usan en entornos donde se deben aceptar y procesar en tiempos muy definidos un gran número de sucesos, en su mayoría externos al ordenador. Si el sistema no respeta las restricciones de tiempo en las que las operaciones deben entregar su resultado se dice que ha fallado.

### **Cuarta Generación**

Abarca desde 1980 hasta nuestros días, está marcada por los ordenadores personales.

Sistema Operativo de Red. El usuario es consciente de que existen otras máquinas. El usuario ya no quiere trabajar solo; quiere trabajar con otros usuarios. Tiene que acceder de forma explícita a esas máquinas.

Sistema Operativo Distribuido. El parámetro clave es la transparencia: el usuario no es consciente de que existen otras máquinas; no sabe en qué máquina está.

#### **44.1.3 Clasificaciones de los Sistemas Operativos**

- Según el número de usuarios:
  - o S.O. monousuarios: sólo aceptan un usuario en un momento determinado.
  - o S.O. multiusuarios: aceptan simultáneamente a más de un usuario.

- Según el hardware:
  - o Según el número de CPU'S:
    - ▢ S.O. monoprocesador: sólo controla una CPU.
    - ▢ S.O. multiprocesador: varios procesadores (más complicado de diseñar).
  - o Según la organización de la memoria principal:
    - ▢ S.O. centralizados: una memoria principal y los procesadores van a estar intentando acceder a esta memoria principal. Los procesos se comunican a través de la memoria.
    - ▢ S.O. distribuidos: cada procesador tiene su propia memoria principal. Los procesos tienen otros mecanismos de comunicación.
- Según el modo de trabajo con los sistemas operativos:
  - o Interactivo (on-line): el usuario dialoga con la máquina.
  - o Batch (off-line): no hay comunicación con la máquina cuando está realizando el trabajo.
- Según el objetivo para el que fueron diseñados:
  - o S.O. de propósito general: capaces de realizar cualquier tarea.
  - o S.O. de propósito específico: sólo pueden realizar una tarea específica, se instalan en microprocesadores que controlan el funcionamiento de electrodomésticos, vehículos, equipos de electrónica consumo, etc.
  - o S.O. de Tiempo Real: ofrecen una respuesta en un intervalo de tiempo bien definido.
  - o S.O. Virtuales: corren sobre HW de un ordenador ofreciendo a los niveles superiores copias exactas de la máquina real, de forma que en cada copia se puede ejecutar un sistema operativo distinto.

- o S.O. de dispositivos móviles: deben adaptarse a las limitaciones que estos dispositivos presentan: procesadores lentos, memoria limitada, pantallas pequeñas y consumo de energía limitado. Ejemplos típicos para estos dispositivos son iOS de Apple, Windows Mobile, Android, etc.

#### **44.1.4 Estructura de los Sistemas Operativos**

##### Sistemas operativos monolíticos

Estos S.O. no tienen una estructura definida. El S.O. se escribe como una colección de procedimientos entrelazados de tal forma que cada uno puede llamar a cualquier otro. Las características de este tipo de estructura son:

- Construcción del S.O. basado en procedimientos compilados separadamente que se unen en un solo fichero objeto a través del enlazador (linker).
- Buena definición de parámetros de enlace entre los distintos procedimientos existentes, lo que genera acoplamiento.
- Carece de protección al entrar a procedimientos que gestionan diferentes aspectos de los recursos del ordenador, como almacenamiento, E/S, etc.
- Son hechos a medida lo que tiene como ventaja que son eficientes y rápidos y como desventaja que carecen de flexibilidad para crecer.

##### Sistemas operativos con capas

El SO se organiza en una jerarquía de estratos, estando construido cada uno de ellos sobre el otro que tiene menor jerarquía que él. Ejemplos: THE, MULTICS.

##### Sistemas operativos Cliente-Servidor.

Minimizar el kernel (núcleo) del S.O., desplazando el código de todos sus servicios a estratos lo más superiores posibles. Para ello, la mayoría de sus funciones se implementan como procesos de usuario, denominados

procesos servidores, de forma que cuando un proceso de usuario llamado proceso cliente, necesita un servicio del S.O. lo que hace es enviar un mensaje al proceso servidor correspondiente, que realiza el trabajo y devuelve la respuesta.

#### ***44.1.5 Funciones de un Sistema Operativo***

Las principales funciones que tienen los S.O. son la gestión de procesos, la gestión de la memoria principal, la gestión del almacenamiento secundario y la gestión de los dispositivos de entrada/salida.

##### **Gestión de procesos**

La CPU es el recurso principal del ordenador de modo que es necesaria una gestión eficiente de la misma para garantizar su aprovechamiento.

El S.O. ha de cargar los distintos procesos, iniciarlos, supervisar su ejecución llevando a cabo los cambios de contexto necesarios y detectar su terminación normal o anormal. En los entornos multiusuario es fundamental la activación de mecanismos de protección que limiten las posibilidades de acceso de cada proceso a una serie de recursos para los que cuente con la debida autorización.

##### **Gestión de memoria principal**

En un sistema multiproceso los procesos tienen que compartir la CPU teniendo que encontrarse en memoria principal para poder pasar a ejecutarse inmediatamente; así, varios procesos tienen que compartir la memoria principal sin que unos puedan acceder a los recursos de otros.

Para ello hay que dividir la memoria en bloques y éstos se les asignaran a distintos procesos. Para hacer la división se utiliza la segmentación, la paginación o la segmentación paginada.

##### **Gestión de los sistemas de archivos**

En un sistema de archivos el S.O. tiene que hacerse cargo de: la gestión del espacio libre/ocupado, de los cachés de lectura y escritura, del vínculo entre nombres y archivos, de las asociaciones entre los bloques físicos de

los dispositivos y los bloques lógicos, de los permisos para el acceso y modificación a los distintos elementos.

### **Gestión de entrada/salida (E/S)**

La velocidad con que se comunican el procesador y la memoria principal contrasta con la velocidad cuando los programas deben interactuar con algún tipo de dispositivo de E/S; durante este proceso la ejecución del programa se ve interrumpida ya que la comunicación con dichos dispositivos es significativamente más lenta que con la memoria.

Conjuntamente con la multiprogramación surgen dos conceptos: el acceso directo a memoria (DMA) y las interrupciones. El procesador cede el control de la E/S a un módulo que se encarga de ejecutar este tipo de operaciones (el controlador de DMA) y esperar hasta que éstas se completen y cuando esto suceda avisar al procesador (que se encuentra mientras tanto ejecutando otras instrucciones -bien sea del mismo proceso o de algún otro-) que puede continuar con las operaciones subsiguientes que quedaron pendientes cuando se realizó la petición de E/S mediante una interrupción.

#### ***44.1.6 Tendencias***

Actualmente, la movilidad es lo primordial en nuestra sociedad, y asociado a esta movilidad está la seguridad. Se empieza a hablar de sistemas operativos en la nube (cloud computing) y se consolidan los sistemas operativos de los dispositivos móviles.

Otro aspecto a destacar derivado del momento económico es el ahorro de costes. Fruto de esto se puede ver una tendencia más que clara en la virtualización.

### **S.O. en dispositivos móviles.**

Si existe una carrera hoy en día en el desarrollo de S.O., ésta está en los S.O. para dispositivos móviles. Los nuevos sistemas operativos convierten al teléfono en un completo aparato multimedia. Hasta hace muy poco

tiempo la elección de un móvil venía determinada por sus características físicas: recepción de la señal, cámara... pero con la llegada de los smartphone, la elección del S.O. se ha convertido en algo muy importante.

A diferencia del mundo del ordenador personal, y debido quizás a su juventud en el mercado, no existe un dominador claro de S.O. móviles. Hay fabricantes hardware que son los fabricantes de sus propios sistemas operativos como por ejemplo Apple que sólo distribuye el S.O. iOS para iphones e iPad, lo mismo que RIM que distribuye su S.O. BlackBerry OS en dispositivos BlackBerry, etc. En el otro extremo están los fabricantes que utilizan S.O. de otras compañías como Android, Symbian, Windows Mobile...

Las compañías que fabrican y distribuyen su propio S.O. tienen a favor que las actualizaciones de los dispositivos son muy controladas.

Otra tendencia vendrá de la mano del despegue de los dispositivos tablets (tabletas), teniendo en cuenta los anuncios de lanzamientos de tabletas: Apple (iPad 2), RIM (PlayBook), Samsung (Galaxy Tab 2), etc. Esto dará aun más auge a los S.O. móviles.

### **S.O. en la nube o en red.**

Estos S.O. surgen del concepto de Computación en la Nube (Cloud Computing) que es un nuevo paradigma que básicamente permite tener servicios computacionales a través de Internet.

Una de las grandes ventajas que se pueden encontrar a este nuevo paradigma es la baja inversión a realizar en HW ya que toda la infraestructura de la computación en la nube se encuentra en los grandes proveedores de servicios de Internet. Bastaría con un hardware mínimo, un navegador y una buena conexión a internet.

Otra de las ventajas sería que las aplicaciones no se instalan en el pc, son aplicaciones Web lo que hace que sea compatible con la mayoría de los formatos conocidos.



Permite tener una única copia de un fichero disponible en cualquier lugar y momento.

Sus puntos débiles son la seguridad y la necesidad de una conexión a Internet.

Estos S.O. son una buena opción para los notebook que tienen poco hardware, e incluso podrían hacer que los fabricantes apostasen por modelos más baratos que permitiría difundir mucho más la informática.

Entre los S.O. más importantes destacan: eyeOS, ChromeOS de Google, oOS, iCloud, etc.

#### **44.2.- SISTEMA OPERATIVO UNIX-LINUX**

UNIX es un sistema operativo creado en 1969 por un grupo de investigadores de los laboratorios Bell de AT&T, entre ellos Ken Thompson, Dennis Ritchie y Douglas McIlroy, como una versión reducida del proyecto MULTICS; primero fue escrito en ensamblador, pero ello impedía la portabilidad a diferentes ordenadores. Después de que Dennis Ritchie en el 1973 crease el lenguaje C, se reescribe UNIX totalmente en este lenguaje de alto nivel, haciendo por tanto el código casi totalmente independiente del tipo de máquina, permitiendo la instalación de UNIX en diferentes plataformas.

Inicialmente los laboratorios AT&T Bell, consideran que UNIX era más bien un proyecto de investigación y lo llegó a distribuir de forma gratuita entre departamentos informáticos de las universidades, los cuales lo podían modificar y adaptar a sus necesidades. Pero, la gran demanda del sistema operativo hace que los laboratorios Bell iniciara su venta a través de distribuciones oficiales concediendo a los usuarios que lo requerían licencias de uso.

Debido a las múltiples versiones en el mercado de UNIX, el IEEE especifica una familia de estándares para definir una interfaz de programación de aplicaciones (API) para que todas las versiones fuesen 'compatibles'. A esta

familia se le conoce como POSIX (Portable Operating System Interface; la X viene de UNIX como seña de identidad de la API)

Linux, se creó en 1991 por Linus Torvalds basándose en otros dos sistemas operativos:

- El sistema abierto UNIX.
- El sistema educativo Minix creado en 1987 por Andrew S. Tanenbaum.

Torvalds crea sólo el Kernel, el núcleo del sistema sin la capa de servicios, gestores, aplicaciones graficas, etc. que serán creadas posteriormente por otros autores. El código del núcleo lo podemos encontrar en la dirección ([www.kernel.org](http://www.kernel.org)).

En la comunidad de programadores se crea el proyecto GNU (Gnu's Not Unix), proyecto para generar software libre, donde se generan editores, compiladores, etc. bajo la licencia pública general GPL (General Public License): usar, copiar, distribuir y modificar siempre que se conserve la firma del autor, pudiendo cobrar por ello.

Linux se crea con esta filosofía de libre distribución y el sistema operativo completo que se construye con este núcleo también. A todo el sistema se le da el nombre de GNU/Linux (distribución completa del sistema operativo con Linux), que contiene el núcleo más las otras capas del sistema operativo y utilidades. Si bien muchas veces se denomina a todo el sistema simplemente LINUX.

#### **44.2.1 Características**

Las características más relevantes del sistema UNIX son:

- UNIX ha sido diseñado como un sistema multiusuario en tiempo compartido; ofreciendo protección de los datos privados sobre ficheros y protección del entorno de ejecución.
- Portabilidad: UNIX fue escrito en el lenguaje C, un lenguaje de alto nivel, lo cual hace que sea relativamente fácil de leer, entender,

modificar y transportar a otras máquinas con una arquitectura física diferente.

- Código y funcionamiento escrito bajo la familia de estándares POSIX (Portable Operating System Interface).
- Interfaz de usuario simple e interactiva: el intérprete de órdenes (shell) es un programa independiente que el usuario puede sustituir. La sintaxis de utilización es idéntica para todas las órdenes.
- Modularidad: Proporciona primitivas que permiten construir grandes programas a partir de otros más sencillos, así como librerías para linkaje.
- Posee bibliotecas compartidas para facilitar el enlace dinámico.
- Protecciones de memoria.
- Soporta diferentes sistemas de archivos, incluidos los de Microsoft Windows
- Sistema de archivos con estructura de árbol invertido (de múltiples niveles que permite un fácil mantenimiento) y jerárquico (permite la unión de diversos sistemas de ficheros con el sistema principal, y una separación de directorios).
- Todos los archivos de usuario son simples secuencias de bytes (8 bits ), no tienen ningún formato predeterminado.
- Independencia de dispositivos: Los discos y los dispositivos de entrada y salida (E/S) se tratan todos de la misma manera: como meros archivos. Las peculiaridades de los dispositivos se mantienen en el núcleo (kernel).
- La arquitectura de la máquina es completamente transparente para el usuario, lo que permite que los programas sean fáciles de escribir y transportables a otras máquinas con hardware diferente

- UNIX no incorpora diseños sofisticados; de hecho, la mayoría de los algoritmos han sido seleccionados por su sencillez y no por su rapidez o complejidad.
- Incorpora todos los servicios de red, TCP/IP, DNS, sendmail, etc.
- Proporciona un completo entorno de programación: los filtros son utilidades simples que se concentran en realizar bien una sola función. Pueden combinarse de forma muy flexible utilizando las pipes (tuberías) y las redirecciones de E/S según las necesidades y preferencias de cada usuario.
- Mantenimiento fácil: consecuencia directa de la modularidad. El sistema sigue evolucionando y se perfecciona y enriquece con nuevas funcionalidades.
- Carácter abierto: permite ampliar fácilmente la funcionalidad con nuevos componentes sin tener que depender de un único fabricante.

#### **44.2.2 Arquitectura de Unix-Linux**

La arquitectura está basada en capas o niveles de forma que cada capa únicamente puede comunicarse con las capas que se hallan en los niveles inmediatamente inferior y superior.

En la capa inferior tenemos toda la parte del Hardware que el sistema operativo debe gestionar. Por encima de éste se sitúa el kernel de Unix que es el encargado de la administración de procesos, gestión del sistema de archivos, entradas / salidas, etc. A los procesos que trabajan a ese nivel se les llama procesos en modo kernel.

La biblioteca estándar se ubica por encima del kernel, se encarga, por ejemplo de las operaciones de apertura, cierre, etc. A este nivel se trabaja en modo usuario. La interface entre las dos capas, o el acceso de la capa

de biblioteca estándar a la del kernel se realiza a través de la interfaz de llamadas al sistema.

A un nivel superior tenemos los programas y utilidades como el shell, compiladores, etc., que sirven de ayuda a desarrolladores y usuarios que interactúan con el sistema operativo. La interface entre esta capa y la inmediatamente inferior es a través de la interfaz de biblioteca.

Por último, se ubicarían los usuarios que por medio de la interfaz de usuario se comunican con el shell, u otras utilidades del sistema Unix.

El núcleo de UNIX (kernel) es de tipo monolítico, diferenciándose dos partes principales: el núcleo dependiente de la máquina y el núcleo independiente. El núcleo dependiente se encarga de las interrupciones, los dispositivos de bajo nivel y parte de la administración de la memoria. El núcleo independiente es igual en todas las plataformas e incluye la gestión de llamadas del sistema, la planificación de procesos, la paginación e intercambio, la gestión de discos y del sistema de archivos

#### **44.2.3 Gestión de procesos.**

La gestión de procesos en UNIX es por prioridad y round robin. En algunas versiones se gestiona también un ajuste dinámico de la prioridad de acuerdo al tiempo que los procesos han esperado y al tiempo que ya han usado la CPU. El sistema provee facilidades para contabilizar el uso de CPU por proceso y una pila común para todos los procesos cuando necesitan estarse ejecutando en modo privilegiado (cuando hicieron una llamada al sistema).

Los procesos trabajan en modo usuario y en modo kernel. El paso de modo usuario a kernel o viceversa se realiza a través de traps que crean una interrupción para acceder a la interfase de llamadas al sistema y al resto de los componentes de nivel kernel. El paso del modo kernel a usuario es un retorno tras la realización de la petición que motivó el paso al modo kernel.

UNIX permite que un proceso haga una copia de sí mismo por medio de la llamada «fork», lo cual es muy útil cuando se realizan trabajos paralelos o concurrentes; también se proveen facilidades para el envío de mensajes entre procesos (pipes, signals).

Los procesos no interactivos se denominan daemons o procesos background. Cuando se inicia un proceso, se le asigna un identificador PID, se guarda el proceso que lo lanzó PPID, el propietario que lo lanzó UID y el grupo de pertenencia GID, lo que definirá el perfil de permisos de acceso a los que tendrá derecho. Existe la posibilidad de alterar el usuario o grupo efectivo de permisos, durante la ejecución del proceso, mediante la llamada a **setuid** o **setgid**, siempre que se disponga de los permisos apropiados. También existe una bandera de permisos **setuid** asociada al archivo del programa que permite ejecutar éste, con los permisos del propietario del archivo en lugar de los del usuario que lo ejecuta.

LINUX combina multiprogramación y tiempo compartido.

El gestor de procesos en el kernel del sistema UNIX, se encarga de la asignación de CPU, la programación de procesos, y las solicitudes de los procesos. Para realizar estas tareas, el kernel mantiene varias tablas importantes para coordinar la ejecución de estos procesos y la asignación de los dispositivos.

Utilizando una política predefinida, el programador de procesos selecciona un proceso de la cola de procesos listos y comienza su ejecución durante un pedazo de tiempo ya dado.

El algoritmo de programación de procesos selecciona el proceso con la mayor prioridad para ser ejecutado primero. Si varios procesos tienen la misma prioridad, se aplica el algoritmo round-robin.

#### **44.2.4 Gestión de memoria.**

Los sistemas UNIX utilizan el manejo de memoria virtual siendo el esquema más usado la paginación por demanda y combinación de segmentos paginados, en ambos casos con páginas de tamaño fijo.

En todos los sistemas UNIX se usa una partición de disco duro para el área de intercambio (swap). Esa área se reserva durante la instalación del sistema operativo.

Una regla muy difundida entre administradores de sistemas es asignar una partición de disco duro que sea al menos el doble de la cantidad de memoria real del ordenador. Con esta regla se permite que se puedan intercambiar flexiblemente todos los procesos que estén en memoria RAM en un momento dado por otros que estén en el disco.

Si no caben todos los programas en memoria principal se hace uso de la partición de intercambio (swapping)

- **Swap out** Cuando no caben en memoria procesos activos, se “expulsa” un proceso de memoria principal, copiando su imagen a swap, aunque no es necesario copiar todo el mapa. Existen diversos criterios de selección del proceso a intercambiar: dependiendo de la prioridad del proceso; preferencia a los procesos bloqueados; no intercambiar si está activo DMA sobre mapa del proceso.
- **Swap in.** Cuando haya espacio en memoria principal, se intercambia el proceso a memoria copiando la imagen desde swap.

Todos los procesos que forman parte del kernel no pueden ser intercambiados a disco.

Cada proceso dispone de su propio espacio de direcciones, organizado en segmentos según:

- **Text Segment:** que almacena el código

- **Data Segment:** que almacena los datos o variables que utilizan los procesos, éste segmento tiene dos partes: **Initialized Data** (datos inicializados) y **Uninitialized Data** (datos no inicializados).
- **Stack Segment:** que almacena la información referente a llamadas a otras funciones.

Es posible compartir código entre procesos mediante el empleo de **Shared Text Segments**. Dos procesos nunca comparten los segmentos de datos y de pila (salvo los **thread**), la forma de compartir información se lleva a cabo mediante el empleo de segmentos especiales de memoria compartida **Shared Segments**.

Linux comparte muchas de las características de los esquemas de gestión de memoria de otras implementaciones UNIX, pero tiene sus características propias.

En lo que respecta a memoria virtual, el direccionamiento de memoria virtual de Linux, hace uso de una estructura de tabla de páginas con tres niveles, formada por los siguientes tipos de tablas (cada tabla individual es del tamaño de una página): **Directorio de páginas** un proceso activo tiene un solo directorio de páginas que es del tamaño de una página. Cada entrada en el directorio de páginas apunta a una página del directorio intermedio de páginas. Para un proceso activo, el directorio de páginas tiene que estar en la memoria principal; **Directorio intermedio de páginas** este directorio puede ocupar varias páginas y cada entrada de este directorio apunta a una página de la tabla de páginas; **Tabla de páginas** esta tabla de páginas también puede ocupar varias páginas, y cada entrada de la tabla de página hace referencia a una tabla virtual del proceso.

Para utilizar esta estructura de la tabla de páginas a tres niveles, una dirección virtual en Linux se ve como un conjunto de cuatro campos. El campo más a la izquierda (más significativo) se utiliza como índice en el directorio de páginas. El siguiente campo sirve como índice en el directorio intermedio de páginas. El tercer campo sirve como índice en la tabla de



páginas. Y el cuarto y último campo, indica el desplazamiento dentro de la página seleccionada de la memoria

#### **44.2.5 Gestión de E/S**

Los dispositivos de entrada y salida son considerados ficheros especiales: Toda entrada/salida está basada en el principio de que todos los dispositivos se pueden tratar como ficheros simples que se acceden mediante descriptores de archivos cuyos nombres se encuentran generalmente en el directorio «/dev».

Cada proceso en UNIX mantiene una tabla de archivos abiertos (donde el archivo puede ser cualquier dispositivo de entrada/salida). Esa tabla tiene entradas que corresponden a los descriptores, los cuales son números enteros obtenidos por medio de la llamada del sistema.

Las llamadas al gestor de entrada/salida se hacen de dos formas: síncrona y asíncrona. El modo síncrono es el modo normal de trabajo y consiste en hacer peticiones de lectura o escritura y el proceso espera a que el sistema le responda.

El gestor de entrada/salida utiliza como elementos principales el buffer de cache; el código general de gestión de dispositivos, y drivers de dispositivos de hardware. Existen dos tipos de dispositivos:

Dispositivos de bloques:

- Usan secuencias de bytes (bloques)
- Utilizan buffer-cache
- Están estructurados en bloques de tamaño fijo (512 bytes)
- Permiten optimizar el rendimiento

Dispositivos de carácter:

- Son dispositivos sin estructura (terminales, impresoras, etc)
- No usan buffer

- La operaciones se realizan carácter a carácter

### **Interrupciones y excepciones**

UNIX permite interrumpir la CPU asíncronamente. Al recibir la interrupción, el kernel almacena el contexto actual, determina la causa y responde a la interrupción. Tras responder a ésta, devuelve el contexto interrumpido y sigue ejecutando. El HW asigna las prioridades a los dispositivos de acuerdo con el orden de actuación en las interrupciones.

Así como las interrupciones están causadas por factores externos a un proceso, las excepciones son sucesos inesperados producidos por procesos, tales como la ejecución de instrucciones reservadas, de forma que el sistema, al encontrarse con una, tiende a reiniciar la instrucción, en lugar de pasar a la siguiente.

No obstante, el kernel debe tener la posibilidad de impedir la aparición de interrupciones en momentos críticos, para evitar la degradación de los datos. El sistema que se utiliza es el de disponer de un conjunto de instrucciones restringidas que colocan el nivel de ejecución del procesador en el estado de palabra (status word). Al asignar un nivel de ejecución del procesado, todas las interrupciones de ese nivel e inferiores quedan suprimidas, permitiendo sólo las superiores.

#### **44.2.6 Gestión de archivos**

Un sistema de archivos permite realizar una abstracción de los dispositivos físicos de almacenamiento de la información para que sean tratados a nivel lógico, como una estructura de más alto nivel y más sencilla que la estructura de su arquitectura hardware particular.

El sistema de archivos UNIX se caracteriza porque posee una estructura jerárquica, realiza un tratamiento consistente de los datos de los archivos,

protege los datos de los archivos y trata a los dispositivos y periféricos (terminales, unidades de disco, cinta, etc.) como si fuesen archivos.

El sistema de archivos está organizado, a nivel lógico, en forma de árbol invertido, con un nodo principal conocido como nodo raíz ("/"). Cada nodo dentro del árbol es un directorio y puede contener a su vez otros nodos (subdirectorios), archivos normales o archivos de dispositivo.

Los nombres de los archivos (pathname) se especifican mediante la ruta (path), que describe cómo localizar un archivo dentro de la jerarquía del sistema. La ruta de un archivo puede ser absoluta (referida al nodo raíz) o relativa (referida al directorio de trabajo actual).

Todos los sistemas UNIX pueden manejar múltiples particiones de disco, cada una con un sistema de archivos distinto.

Una partición de disco clásica en UNIX contiene una estructura como la de la figura.

- Bloque de arranque: contiene el código para arrancar el ordenador con esa partición.
- Superbloque: Contiene información crucial acerca de la organización del sistema de archivos, incluido el número de nodos-i (i-nodes), el número de bloques de disco y el principio de la lista de bloques de disco libres. La destrucción del superbloque hace que el sistema de archivos ya no pueda leerse.
- Nodos-i: contiene la representación interna de un fichero que permite entre otras cosas localizar todos los bloques de disco que contienen los datos del archivo
- Bloques de Datos: almacenan la información. Lo normal es que un archivo ocupe más de un bloque de datos no siendo necesario que estén contiguos.

Dentro de la estructura de directorios de UNIX – Linux existen una serie de directorios comunes a todas las instalaciones que es preciso conocer:

- `/`: Directorio raíz, inicio del sistema de archivos.
- `/tmp`: Directorio de archivos temporales.
- `/dev` : Directorio de dispositivos. En él se encuentran todos los dispositivos de E/S, que se tratan como archivos especiales.
- `/etc`: Directorio para archivos de sistema diversos.
- `/bin`: Directorio para programas binarios (ejecutables).
- `/lib`: Directorio de bibliotecas del sistema.
- `/usr`: Directorio de usuarios.
- `/home`: Directorio base a partir del cual se ubican los directorios por defecto de las cuentas de usuario.

LINUX comenzó empleando el sistema de archivos de MINIX, pero estaba limitado a nombres de 14 caracteres y a archivos de 64 MB de tamaño. La primera mejora vino de la mano de un sistema de archivos denominado Ext que permitía nombres de archivo de 255 caracteres y 2 GB de tamaño por archivo, pero era muy lento. La evolución vino de la mano del sistema de archivos Ext2, con nombres de archivo largos, archivos grandes y mejor rendimiento. Ext2 evoluciono a Ext3 que trajo principalmente las transacciones (journaling) a Ext2. En ext3 se almacena la información necesaria para restablecer los datos afectados por la transacción en caso de que ésta falle. La evolución de ext3 es **ext4** que soporta un tamaño máximo de sistema de archivos de 1 ExaByte (1 ExaByte = 1024 PetaBytes = 1048576 TeraBytes) y un tamaño máximo por archivo de 16 TB para los archivos y por otro lado, modifica estructuras de datos importantes, como la destinada a almacenar los datos del archivo utilizando **“extent”** que es un conjunto de bloques físicos contiguos,

mejorando el rendimiento al trabajar con ficheros de gran tamaño y reduciendo la fragmentación.

### **44.3.- SISTEMA OPERATIVO WINDOWS**

#### **44.3.1 Características**

La familia de S.O. Windows es propiedad de Microsoft. Se podría decir que básicamente existen dos versiones distintas de S.O. Windows: aquellos enfocados al mundo empresarial y aquellos enfocados al consumidor final o usuario doméstico.

Dentro de esta simple clasificación podríamos hacer otras clasificaciones. Orientado al usuario domestico existen S.O. para equipos de sobremesa, portátiles, tablets, y dispositivos móviles. Y orientado al mundo empresarial podemos distinguir S.O. para servidores y S.O. para estaciones de trabajo. Aún podríamos dividir más los S.O. para servidores (Standard, Enterprise, Datacenter, Web, Storage, Small Business Server)...

Los últimos S.O. que han visto a luz son Windows 7 con las versiones Starter (sólo para 32 bits), Home Basic, Home Premium, Professional, Ultimate, Enterprise. Y a nivel de S.O. para servidores es Windows Server 2008 R2.

Es destacable el interés de mejora adquirido por Microsoft con los S.O. de servidor, comprometiéndose a un cambio cada 5 años (antes del 2008 estaba el 2003) y a una revisión cada 2 años aprox. (por eso existe Windows Server 2008 R2).

Windows Server 2008 es el último S.O. de servidor que incluye estas mejoras:

- Nuevo proceso de reparación de sistemas NTFS: proceso en segundo plano que repara los archivos dañados.

- Creación de sesiones de usuario en paralelo: reduce tiempos de espera en los Terminal Services y en la creación de sesiones de usuario a gran escala.
- Cierre limpio de Servicios.
- Sistema de archivos SMB2: de 30 a 40 veces más rápido el acceso a los servidores multimedia.
- Address Space Load Randomization (ASLR): protección contra malware en la carga de controladores en memoria.
- Windows Hardware Error Architecture (WHEA): protocolo mejorado y estandarizado de informe de errores.
- Virtualización de Windows Server: mejoras en el rendimiento de la virtualización.
- PowerShell: inclusión de una consola mejorada con soporte GUI para administración.
- Server Core: el núcleo del sistema se ha renovado con muchas y nuevas mejoras.

En los siguientes apartados se verán características de Windows, y si bien la mayoría son comunes para todas las versiones, las explicaciones estarán más centradas en Windows 2008 Server R2.

#### **44.3.2 Arquitectura de Windows**

La estructura modular de Windows 2008 aporta una gran flexibilidad. Su diseño le permite ejecutarse en una gran variedad de plataformas hardware. En esta estructura modular se distinguen dos capas principales:

- **Modo usuario**: Cuyos programas y subsistemas están limitados a los recursos del sistema a los que tienen acceso. Está formado por

subsistemas que pueden pasar peticiones de E/S a los controladores apropiados del modo núcleo a través del gestor de E/S.

- **Modo núcleo o kernel**: Tiene acceso total al hardware de la máquina, impidiendo a los servicios del modo usuario y a las aplicaciones acceder a al hardware que queda totalmente protegido por el sistema operativo. La arquitectura dentro del modo núcleo se compone de lo siguiente:
  - o El micronúcleo: situado entre la capa de abstracción de hardware y el Executive, proporciona la gestión *multiprocesador*: gestión de procesos, hilos y tratamiento de interrupciones, y de excepciones.
  - o Una capa de abstracción de hardware (en inglés Hardware Abstraction Layer o HAL), se encarga de ocultar las diferencias de hardware y por tanto proporciona una plataforma única donde pueda ejecutarse el S.O. independientemente del HW.
  - o Controladores o también llamados drivers: utilizados para interactuar con los dispositivos hardware.
  - o Executive: sobre el cual son implementados todos los servicios de alto nivel. Se relaciona con todos los subsistemas del modo usuario. Se ocupa de la entrada/salida, la gestión de memoria, Plug&Play, la seguridad y la gestión de procesos.

Dado que el enlace estático de los programas de usuario con las bibliotecas de la API Win32 conllevaría un tamaño enorme en los programas y un desperdicio de memoria, pues cada programa en ejecución tendría su copia de estas bibliotecas, todas las versiones de Windows manejan bibliotecas compartidas, llamadas bibliotecas de vínculos dinámicos (DLLs; Dinamic Link Libraries).

#### **44.3.3 Gestión de procesos.**

Los procesos se crean como objetos, y un proceso puede tener varios hilos. Dado que el proceso es un objeto, su composición será un conjunto de

datos sólo accesibles a través de un conjunto de funciones que los ocultan del resto de aplicaciones o funciones. Estas funciones o servicios se activan por medio de mensajes. El proceso tendrá al menos un hilo, que a su vez puede ejecutar otros hilos, pudiendo hacerlo en paralelo en un sistema multiprocesador.

Windows mantiene dos listas diferentes con la información de todos los procesos e hilos. Cada proceso tiene asociado un bloque o estructura de datos EPROCESS, que apunta al EPROCESS siguiente y al anterior (doble lista enlazada), y la otra estructura es ETHREAD para recoger la información de los hilos.

El algoritmo de planificación está basado en colas de retroalimentación de múltiples niveles con prioridades. Cada cola se gestiona con una política Round Robin.

La planificación se aplica sobre los hilos, no a los procesos (sin tener en cuenta a qué proceso pertenecen los distintos hilos que se ejecutan), y está basada en prioridades, es decir, siempre se ejecutará el hilo de mayor prioridad de la cola de hilos preparados.

Cuando se selecciona un hilo para su ejecución, se le concede un quantum, o intervalo de tiempo durante el cual se permite al hilo ejecutarse antes de que lo haga otro hilo del mismo nivel de prioridad. Los valores de quantum pueden variar.

Aunque se conceda un quantum a un hilo, éste podría no consumirlo completamente porque aparezca en el sistema un nuevo proceso de mayor prioridad, que obligaría al que se está ejecutando a abandonar el procesador.

El sistema trata igual a todos los hilos que tengan la misma prioridad, asignando a cada hilo de mayor prioridad un intervalo de tiempo de procesador con un Round Robin. Si ninguno de estos hilos estuviera preparado para ejecutarse, pasarían a ejecutarse, con la misma política, los de la prioridad inmediatamente inferior.



En cada uno de estos casos, Windows debe determinar qué hilo debe ejecutarse a continuación, y esta decisión es lo que se conoce como dispatcher.

Cada proceso recibe una prioridad base para todos sus hilos. El sistema se basa en 32 prioridades, del 0 (menor prioridad) al 31 (mayor prioridad):

- La prioridad 0 está reservada para el hilo de sistema responsable de poner a cero las páginas libres cuando no las necesiten ningún hilo.
- Las prioridades 1 a 15 son las reservadas para los procesos de usuario (prioridades variables).
- Las prioridades 16 a 31 están reservadas al sistema operativo (prioridades en tiempo real).

El planificador funciona accediendo a la tabla por el proceso de prioridad 31 y viendo si tiene hilos listos para ejecutar. Si los hay toma el primero de la lista y lo ejecuta durante un quantum. Mientras existan procesos preparados de una prioridad superior, el sistema les concederá todo el tiempo que precisen. Este comportamiento se repite para cada una de las entradas de la tabla de prioridades.

En ciertas condiciones un subproceso puede ver incrementada su prioridad base, pero nunca por encima de la prioridad 15 y nunca para subprocesos de prioridad mayor de 15. Si una operación de E/S libera un subproceso, éste ve incrementada su prioridad base de modo que pueda ejecutarse pronto.

También se produce aumento de prioridad si el subproceso estaba esperando por un semáforo, mutex u otro suceso. Estas elevaciones de prioridad van disminuyendo a medida que un subproceso beneficiado va consumiendo por completo su quantum, hasta volver a situarse en su prioridad base.

#### **44.3.4 Gestión de memoria.**

La gestión de memoria en Windows es de memoria virtual con paginación.

Las aplicaciones de 32 bits tienen un espacio de dirección del proceso de 4 GB de memoria. Los sistemas operativos de Microsoft Windows proporcionan a las aplicaciones acceso a 2 GB de espacio de dirección del proceso, específicamente conocido como espacio de dirección virtuales del modo de usuario. Todos los subprocesos pertenecientes a una aplicación comparten el mismo espacio de direcciones virtuales del modo de usuario. Los 2 GB restantes se reservan para el sistema operativo (también conocido como espacio de dirección del modo de kernel).

El espacio de direcciones virtual se pagina por demanda con tamaño fijo de páginas (mínimo 4KB para arquitecturas x86 y x64bits y 8KB para arquitecturas IA64 y máximo de 4MB para arquitectura x86, 2MB para x64 y 16MB para IA64).

Windows utiliza un algoritmo de **paginación por demanda anticipada**, es decir, cada vez que se produce un fallo de página, el sistema copiará en memoria la página correspondiente a la referencia a memoria que ha causado el fallo de página y además un conjunto de páginas próximas a ella, tanto anteriores como posteriores, al suponer que, debido a la localidad de las referencias, es casi seguro que en un futuro próximo también se hará referencia a estas páginas, que cuando se quieran utilizar ya estarán en memoria y, por lo tanto, no producirán fallos de páginas adicionales.

El mecanismo de paginación se apoya mucho en el concepto de **Conjunto de Trabajo (Working Set)** que asegura una cierta cantidad de memoria física para cada proceso.

Windows presta especial atención al momento de arranque de los procesos ya que, como no tienen ninguna página cargada en la memoria, hasta que carguen todas las páginas necesarias se producirán muchos fallos de

página. Para optimizar la carga de los procesos, Windows cuenta con lo que se conoce como “Prefetcher” cuya misión es acelerar el proceso de carga.

Si se produce un fallo de página y es necesario sustituir algún marco de página que está en memoria, Windows emplea el algoritmo LRU (aunque algunas versiones utilizan también FIFO).

Permite compartir páginas, al poder proteger contra lectura o escritura las mismas. Igualmente admite que se pueda bloquear una página en memoria que sea crítica, impidiendo que pueda sustituirse ante una falta de página, facilitando así la implementación de aplicaciones en tiempo real.

#### **44.3.5 Gestión de E/S**

El sistema de entrada/salida (E/S) de Windows es el que permite utilizar los dispositivos facilitando el acceso a los mismos e independizando a los programas de los dispositivos, ofreciendo además la seguridad en su uso y la escalabilidad del sistema.

Las entradas y salidas en Windows pueden ser síncronas (el proceso esperará hasta que se haya completado la operación en el dispositivo hardware) o asíncronas (el proceso lanza la operación y sigue con su ejecución y cuando la operación E/S finaliza el S.O. lo avisa).

En Windows se cargan y descargan los drivers en cualquier momento, evitando que consuman recursos si no se van a utilizar.

Esto se hace gracias al Plug and Play (PnP) que permite detectar cualquier dispositivo que se conecte al sistema y cargar el driver correspondiente.

El sistema de E/S se compone de los siguientes módulos:

- El gestor de E/S: define la infraestructura que soporta los drivers de dispositivos. Forma parte del sistema operativo.

- El driver de dispositivo: proporciona un interface de E/S para un determinado tipo de dispositivo. Los drivers reciben peticiones canalizadas a través del gestor de E/S y las dirigen al dispositivo concreto, e informan al gestor de que se ha completado la operación de E/S. Estos módulos los desarrollan los fabricantes.
- El gestor de PnP: detecta los dispositivos hardware al conectarse o desconectarse.
- El gestor de energía: facilita al sistema, así como a los drivers de dispositivo, los cambios de estado de consumo de energía eléctrica de acuerdo con la actividad del dispositivo.

#### **44.3.6 Gestión de archivos**

En Windows, la asignación del espacio la realiza el subsistema de ficheros en unidades “cluster”, cuyo tamaño depende de la capacidad del disco, normalmente oscila desde 512 bytes hasta 4 Kbytes. Utiliza 64 bits para direccionar los cluster y permite definir ficheros de 264TB (16.384 petabytes), aunque, lógicamente, el tamaño máximo de los ficheros está limitado por la capacidad de los discos.

Windows gestiona los discos y la información que contienen en base a particiones y volúmenes.

- **Particiones.** Cada disco se puede dividir en particiones primarias y extendidas. Las particiones primarias serán aquellas que puedan contener un S.O. y, por lo tanto, permitan el arranque del S.O. desde las mismas. De aquí podemos deducir, que el sistema requiere como mínimo una partición primaria en algún disco.

Una partición nunca podrá sobrepasar un disco. Sólo puede haber una partición extendida por disco, y como máximo sólo podrá contener 4 particiones en total.

Una vez creada la partición, es necesario darle formato para que pueda contener datos. Un volumen es sinónimo de partición formateada.

En Windows, las particiones las gestiona el gestor de particiones. Este gestor utiliza el gestor de E/S para identificar las particiones y crear los dispositivos que las representen, es decir, las unidades lógicas correspondientes.

Este gestor envía un comando al gestor de volúmenes (descrito más adelante) para saber si la partición tiene un volumen asociado, y si es así, a partir de ese momento cualquier acción sobre la partición se la notificará al gestor de volúmenes.

- **Volumen.** Un volumen desde el punto de vista del usuario es una partición formateada. Las particiones primarias sólo podrán contener un volumen, mientras que las extendidas podrán albergar varios, teniendo en cuenta que un sistema sólo podrá tener como máximo 24 volúmenes, ya que se identifican por medio de las letras del abecedario, y en inglés sólo tiene 24 letras.

En Windows Server podemos trabajar con dos tipos de discos:

- Discos básicos. Son los que se basan exclusivamente en tablas de particiones MBR (Master Boot Record) o tablas GPT (GUID Partition Table).
- Discos dinámicos. Se basan en *volúmenes dinámicos*, que permiten la creación de volúmenes de particiones múltiples tales como simples, distribuidos, espejos, stripes y RAID-5. Los discos dinámicos se particionan con el Administrador de discos lógicos (LDM – Logical Disk Manager).

Windows trabaja con los siguientes tipos de volúmenes dinámicos:

- Volúmenes distribuidos (spanned). Es un único volumen lógico compuesto por un máximo de 32 particiones libres en uno o más discos. Es una forma de juntar el espacio no asignado en un sistema con varios discos en una única unidad lógica.

- Volúmenes Espejo. En este tipo de volumen, el contenido de la partición se duplica en una partición idéntica en otro disco, aunque se ven como un único volumen y no como dos. Los volúmenes espejo se conocen como RAID de nivel 1 (RAID1) y son tolerantes a fallos.
- Volúmenes divididos (Striped). Similar al volumen distribuido, utiliza el espacio de varios discos y los convierte en una única unidad lógica. Utiliza un tipo especial de formato para escribir en el disco y tiene más rendimiento que el volumen distribuido. Los fallos de escritura suelen ser mayores que en el caso del volumen distribuido. Se conocen como RAID de nivel 0 (volúmenes RAID-0).
- Volúmenes RAID-5. Como los volúmenes striped, pero con tolerancia a fallos ya que distribuyen la información de paridad entre todos los discos miembros del volumen.

### **Sistemas de ficheros**

Windows soporta los siguientes formatos de sistemas de ficheros:

A) CDFS (sistema de ficheros de CD-ROM): sólo permite la lectura, y soporta los formatos de disco ISO-9660 y Joliet.

B) UDF: es un subconjunto del formato ISO-13346 con extensiones para formatos como CD-R y DVD-R/RW. UDF está incluido en la especificación DVD y es más flexible que el CDFS.

C) FAT12, FAT16, y FAT32: Windows es compatible con el sistema de ficheros FAT por compatibilidad con MS-DOS y otras versiones de Microsoft Windows. El formato FAT (File Allocation Table) incluye un mapa de bits que se utilizan para identificar clusters o bloques en el disco.

FAT32 tiene una capacidad teórica para direccionar volúmenes de 8 terabytes (TB), no obstante limita los volúmenes a un máximo de 32 GB.

D) NTFS (New Technology File System): es el formato nativo de Windows Server para los sistemas de ficheros. NTFS utiliza direcciones de disco de 64 bits con lo que podría gestionar volúmenes de hasta 16 exabytes, sin

embargo, Windows limita el tamaño de un volumen NTFS a lo que se pueda direccionar con 32 bits, que es un poco menos de 256 TB (con clústeres de 64 KB). NTFS admite ficheros de máximo 16 TB.

NTFS añade características de seguridad de ficheros y directorios, cuotas de disco, compresión de ficheros, enlaces simbólicos basados en directorios, y cifrado. Una de sus características más significativas es la recuperabilidad: registra los cambios que se realizan en los metadatos como si fueran transacciones con la finalidad de que se puedan recuperar en el caso de la pérdida de ficheros o de sus datos.

La estructura central de NTFS es la tabla maestra de archivos MFT (Master File Table), que es una sucesión lineal de registros de tamaño fijo (1 KB). Cada registro de MFT describe un archivo o un directorio, contiene los atributos del archivo, como su nombre y marcas de tiempo, y la lista de direcciones de disco donde están sus bloques. Si un archivo es demasiado grande puede ser necesario emplear más de un registro MFT para contener la lista de todos los bloques. En este caso al primer registro se le denomina registro base, y apunta a los demás registros MFT.

#### **44.3.7 Seguridad**

El administrador de seguridad, componente del Executive, hace que se respete el complejo mecanismo de seguridad de Windows 2008, que satisface los requisitos C-2 del Libro Naranja del Departamento de Defensa de Estados Unidos.

Cada usuario y grupo de Windows 2008 se identifica con un SID (Security ID) único a nivel mundial. Cada proceso lleva asociado una ficha de acceso que especifica su SID y otras propiedades.

Cada objeto tiene asociado un descriptor de seguridad que indica quién puede realizar qué operaciones con él. Un descriptor de seguridad está

formado por un encabezado, seguido de una DACL (discretionary Access Control List) con uno o más elementos de control de acceso (ACE). Los más importantes son Allow y Deny. Además del DACL, el descriptor tiene una SACL (System Access Control List) que no especifica quién puede usar el objeto sino qué operaciones con el objeto se asientan en el registro de sucesos de seguridad del sistema (función de auditoría).

En un sistema autónomo la validación corre por cuenta del proceso winlogon y la configuración de seguridad almacenada en la propia máquina en las claves del registro: SECURITY y SAM. Donde la primera establece las políticas globales de seguridad y la segunda la seguridad específica de cada usuario.

En un sistema en red, la autenticación de los usuarios está centralizada en ciertos servidores denominados controladores del dominio. Los equipos se organizan dentro de Dominios, pudiendo éstos estar gestionados mediante el empleo del Active Directory.

Windows 2008 dispone de administración centralizada de certificados. En las versiones anteriores se confiaba que cada aplicación mantenía su propia lista de claves o CA confiables.

El protocolo KERBEROS (RFC 1510), que es un estándar de Internet para autenticación, es el método nativo que emplean los sistemas Windows 2008. Cualquier servidor del Directorio Activo, automáticamente, tiene el servicio del Centro de distribución de claves de Kerberos (KDC- Kerberos Key Distribution Center).

#### **44.4.- BIBLIOGRAFÍA**

- Sistemas Operativos Modernos. Tanenbaum, Andrew. Prentice Hall, 2005
- Linux Bible 2008 Edition. Christopher, N. Wiley Publishing, Inc 2008.



- Windows internals 5<sup>th</sup> Edition. Mark E. Russinovich, David A. Solomon, Alex Ionescu. Microsoft Press, 2009

**Autor:** Francisco Javier Rodríguez Martínez

Subdirector de Sistemas Escola Superior Enxeñaría Informática Ourense

Colegiado del CPEIG



# **45. SERVIDORES DE MENSAJERÍA. SISTEMAS DE CORREO. SERVIDORES DE APLICACIONES Y SERVIDORES WEB.**

## **Tema 45. Servidores de Mensajería. Sistemas de Correo. Servidores de aplicaciones y servidores web.**

### **ÍNDICE**

#### **45.1 Servidores de Mensajería**

*45.1.1 Sistema de mensajería Centralizada*

*45.1.2 Sistema de mensajería Distribuido*

#### **45.2 Sistemas de Correo**

*45.2.1 Sistemas de correo electrónico: arquitectura*

*45.2.2 SMTP (Simple Mail Transfer Protocol)*

*45.2.3 POP (Post Office Protocol)*

*45.2.4 IMAP - Internet Message Access Protocol*

*45.2.5 Formato de mensajes en internet*

*45.2.6 Extensiones de SMTP*

*45.2.6.1 ESMTP*

*45.2.6.2 MIME*

#### **45.3 Servidores de aplicaciones y servidores web**

#### **45.4 Servidores de Aplicaciones**

*45.4.1 Servicios proporcionados por un servidor de aplicaciones*

*45.4.2 Estándar J2EE*

*45.4.3 Estructura de un servidor de Aplicaciones*

*45.4.4 BEA WebLogic*

#### **45.5 Servidores Web**

*45.5.1 Introducción Cliente/Servidor*

*45.5.2 Servidores Web*

*45.5.3 Características de los servidores web*

*45.5.4 Arquitectura*

*45.5.4.1 Funcionamiento del servidor web*

*45.5.5 Apache*

*45.5.5.1 httpd.conf*

45.5.5.2 Módulos de Apache

45.5.6 *Microsoft IIS*

45.5.6.1 Administración de IIS

45.5.7 *Lighttpd*

## **45.6 Bibliografía**

### **451 SERVIDORES DE MENSAJERÍA**

Un servidor de mensajería es una aplicación que posee la capacidad para manejar mensajes entre dos o más entidades, ya sean aplicaciones de usuario u otros sistemas de gestión de mensajes. Los mensajes de un servidor de mensajería son enviados a través de un middleware, el cual facilita la comunicación entre los distintos elementos del sistema, utilizando generalmente un conjunto de reglas y especificaciones que posibilitan la comunicación entre las distintas partes. Otra de las características de los servidores de mensajería es la capacidad de almacenaje de los mensajes, este almacenamiento se produce generalmente en una cola hasta que es posible el envío del mismo hacia su destinatario, que por lo general resulta ser otra aplicación.

Es muy habitual encontrarse en una empresa u organización un sistema de mensajería funcionando en un servidor y esperando el envío de mensajes a su cola de entrada. Desde allí, el middleware analiza mensaje a mensaje determinando el destino de cada uno. Una vez en el servidor, un mensaje sólo tiene dos posibilidades de entrega, o ser enviado de manera local, o que éste tenga que ser redirigido a otro servidor de mensajería para que sea él el que realice la entrega. Si el mensaje ha de ser entregado a un destino local, entonces es enviado inmediatamente al buzón local. Por el contrario, si el mensaje es determinado como remoto, el servidor de mensajería debe enviar el mensaje a otro servidor de mensajería dentro de

su entorno para que sea éste el que realice la entrega del mensaje.

Por lo general, si existen problemas de conexión entre los servidores o no es posible determinar la localización del servidor de mensajería remoto, el usuario el cual realizó el envío del mensaje es informado a través de un mensaje enviado por el servidor de mensajería, informando de la situación. Este tipo de mensaje suelen ser sólo de notificación de que se están teniendo problemas con el envío del mensaje, puesto que el servidor de mensajería continuará intentando enviar el mensaje hasta que se agoten el número máximo de intentos de envío o hasta que el mensaje caduque, es decir, exceda un límite de tiempo de estancia en el servidor.

Los modelos de los servidores de mensajería suelen adaptarse a una arquitectura centralizada o seguir una solución distribuida.

#### **451.1 SISTEMA DE MENSAJERÍA CENTRALIZADA**

Un sistema de mensajería centralizada se fundamenta en un núcleo de datos, el cual aloja todos los recursos y servicios de los servidores que conforman el sistema. Este núcleo de datos permite que cualquier usuario del sistema de mensajería se conecte a los servicios de mensajería, ya sea de forma local o remota.

Las características de un sistema de mensajería centralizado son:

- **Datos:** Todos los datos y la información se encuentra albergada y se gestiona desde el núcleo, incluso cuando los usuarios establecen una conexión remota para su utilización. Esta centralización facilita en gran medida la administración de los servicios, puesto que provoca que ésta sea más sencilla.
- **Actualizaciones:** Las actualizaciones se han de realizar únicamente en el núcleo central, en donde se encuentra todo el sistema.

- **Ubicación:** El centro de datos añade al sistema dispositivos de aislamiento de la alimentación o sistemas de alimentación ininterrumpida (SAI). Proporciona además la posibilidad de ofrecer servicios incluso cuando se produce alguna incidencia de carácter grave, ya que posibilita la réplica de todo el sistema de una manera eficaz.

#### **451.2 SISTEMA DE MENSAJERÍA DISTRIBUIDO**

Un sistema de mensajería distribuido está formado por una serie de sucursales repartidas en distintas ubicaciones conectadas entre sí. Cada sucursal posee un servidor o servidores de mensajería con todos sus servicios de manera independiente del resto de sucursales. Cada uno de los servidores de mensajería realiza el envío de sus mensajes locales y redirige a los otros servidores aquellos mensajes que no son de dominio local y que sí son capaces de resolver alguno de los otros servidores.

- **Datos:** La información se encuentra también distribuida entre cada una de las sucursales y cada una de ellas gestiona y administra esta información y sus servicios, lo que provoca un aumento en la complejidad de estas tareas.
- **Actualizaciones:** Cada vez que se lleva a cabo una tarea de actualización ésta ha de realizarse en cada una de las sucursales, para que tenga efecto en todo el sistema.
- **Ubicación:** Cada sucursal posee su propio centro de datos, los cuales pueden ofrecer los mismos servicios que en una arquitectura centralizada.

## **452 SISTEMAS DE CORREO**

Un servidor de correo es una aplicación que nos permite enviar mensajes (correos) de unos usuarios a otros, con independencia de la red que dichos usuarios estén utilizando.

Para lograrlo se definen una serie de protocolos, cada uno con una finalidad concreta:

- SMTP, Simple Mail Transfer Protocol: Es el protocolo que se utiliza para que dos servidores de correo intercambien mensajes.
- POP, Post Office Protocol: Se utiliza para obtener los mensajes guardados en el servidor y pasárselos al usuario.
- IMAP, Internet Message Access Protocol: Su finalidad es la misma que la de POP, pero el funcionamiento y las funcionalidades que ofrecen son diferentes.

Así pues, un servidor de correo consta en realidad de dos servidores: un servidor SMTP que será el encargado de enviar y recibir mensajes, y un servidor POP/IMAP que será el que permita a los usuarios obtener sus mensajes.

Para obtener los mensajes del servidor, los usuarios se sirven de clientes, es decir, programas que implementan un protocolo POP/IMAP. En algunas ocasiones el cliente se ejecuta en la máquina del usuario, sin embargo existe otra posibilidad: que el cliente de correo no se ejecute en la máquina del usuario; es el caso de los clientes vía web.

El correo electrónico es una de las aplicaciones TCP/IP más utilizadas en estos días. En su forma más sencilla, el correo electrónico, es una manera de enviar mensajes o cartas electrónicas de un computador a otro.

El correo electrónico de Internet se implementó originalmente como una función del protocolo FTP. En 1980 Suzanne Sluizer y Jon Postel realizaron trabajos con un protocolo que posteriormente se denominaría SMTP ("Simple Mail Transfer Protocol"). Hoy en día se sigue utilizando este protocolo, con los avances lógicos que requiere el tipo de transferencia actual.

El protocolo SMTP fue desarrollado pensando en que los sistemas que intercambiarían mensajes, eran grandes computadores, de tiempo compartido y multiusuario conectados permanentemente a la red Internet. Sin embargo, con la aparición de los computadores personales, que tienen una conectividad ocasional, se hizo necesaria una solución para que el correo llegase a estos equipos. Para solventar esta limitación, surgen los protocolos POP e IMAP.

Por lo tanto, podemos discriminar dos tipos de agentes que están involucrados en la transferencia de correo, MUA y MTA:

- Agente de usuario (MUA), interfaz para leer y escribir los mensajes, son los clientes finales.
- Agente de transporte (MTA o estafeta), encargado del transporte de los mensajes (SMTP). A la primera MTA a la que el cliente entrega su correo se la llama MSA (S de sending) y a la última que lo recibe y lo entrega al cliente destinatario se le llama MDA (D de delivering)
- Cuando un MTA no es el destinatario de un correo, se lo debe de entregar a otro y así hasta llegar a su destino. Este comportamiento es conocido como **Relay**.

Es muy importante configurar bien la función Relay de un MTA porque si se configura de forma abierta, se puede terminar siendo una fuente de SPAM.



## **452.1 SISTEMAS DE CORREO ELECTRÓNICO: ARQUITECTURA**

Los sistemas de correo electrónico se configuran para que trabajen de forma asíncrona en la comunicación, de forma que el cliente envía un mensaje y no tiene que esperar respuesta. De esta forma los buzones de correo y las funciones de transmisión y recepción de mensajes se ubican en un servidor de correo.

El servidor de correo permanece a la “escucha” de conexiones de otros servidores de correo para la recepción de mensajes con destino a sus usuarios, almacena los mensajes de correo recibidos en los buzones y realiza la transmisión de mensajes de los usuarios a otros servidores remotos, es decir, el servidor de correo actúa como un MTA.

Los usuarios de un servidor, interactuarán con él a través de un cliente de correo. Para la recuperación de mensajes se utiliza POP3 o IMAP4. El envío de mensajes no se realizará directamente a los servidores remotos, sino que en primer lugar se enviará el mensaje al servidor de correo que le da servicio, mediante SMTP y será este servidor el que realice la transmisión definitiva del mensaje al servidor de correo remoto que albergue el buzón destino.

Teniendo en cuenta esto, la arquitectura de un sistema de correo incluye:

- Servidores de correo para el envío, recepción y almacenamiento de la información de los usuarios. Deberán ser correctamente configurados para poder ser alcanzables en el DNS.
- Clientes de correo utilizados por los usuarios para, básicamente, componer y leer correos electrónicos.
- Soporte para plataformas de acceso: hoy en día se puede acceder

desde el puesto de trabajo en la intranet de la empresa, desde un equipo portátil en la extranet, hasta smartphones y PDA's.

- Sistema de almacenamiento de los mensajes en los respectivos buzones.
- Sistema de Directorio (Active Directory o LDAP) útiles para acceder a la información de los usuarios de la empresa (nombre, cargo, etc).

### **Clientes de correo**

Los clientes de correo permiten a los usuarios interaccionar con el sistema para enviar y recibir mensajes. Desde los inicios del correo electrónico, hasta hoy en día, podemos clasificar los clientes en:

- Clientes en modo texto: es el cliente accesible desde la interfaz de comandos Shell mediante una cuenta en la máquina que alberga el servidor de correo.
- Clientes pesados: software que se instala en el PC del usuario y gestionan todo el ciclo de vida de creación, envío y recepción de mensajes así como la posibilidad de almacenamiento local, por ejemplo Microsoft Outlook, Lotus Notes Thunderbird, etc.
- Clientes ligeros: no hace falta instalar ningún software en el PC, sino que el usuario accede al sistema de correo a través de una interfaz web con un navegador. La seguridad en la comunicación puede implementarse mediante SSL.
- Clientes en smartphones o PDAs: Permiten acceder a los mensajes en cualquier lugar y a cualquier hora. Suelen configurarse con IMAP4 como protocolo de recuperación para permitir que posteriormente, los mensajes procesados sean visibles también desde el PC.

## **Servidores de correo**

Hasta hace pocos años a un servidor de correo se le pedía que gestionase correctamente el servicio del correo. Actualmente, se tiende a proporcionar una solución unificada de mensajería para una organización que integre mensajería móvil, mensajería instantánea, groupware, etc. con lo que la frontera entre servidores puros de correo electrónico y servidores generales de mensajería o entorno colaborativo está poco clara.

Para escoger un servidor u otro deberían de tenerse en cuenta las siguientes consideraciones:

- El cliente que se asocia al servidor, ya que algunos servidores exigen un tipo determinado de cliente, como Lotus Notes.
- Estrechamente relacionado con el anterior punto está la integración de servidor y cliente y, generalmente, se obtendrá una mejor integración cuando el servidor y el cliente son del mismo fabricante, como sucede con Microsoft Outlook y Microsoft Exchange Server o Lotus Notes y Lotus Domino.
- El nivel de gestión requerido, la solución de almacenamiento y la disponibilidad. Para una organización, puede resultar insuficiente en términos de rendimiento, flexibilidad y escalabilidad utilizar los discos de una máquina para almacenar los buzones de correo con lo que es más habitual contar con dispositivos de almacenamiento dedicados como soluciones NAS o SAN con conexiones rápidas de fibra óptica y que además facilitan el uso de clústeres activo/activo al permitir que un servidor asuma los buzones gestionados por otro en caso de caída.
- Es importante utilizar técnicas de benchmarking para evaluar las siguientes características:

- o La capacidad de tratar correo concurrentemente.
  - o La velocidad de entrega.
  - o La extensibilidad y funciones implementadas.
  - o La estabilidad.
- Al tratarse de una aplicación crítica para una organización, el correo electrónico debe ser configurado como una solución de alta disponibilidad lo cual requiere el establecimiento de políticas de redundancia adecuadas para garantizar el servicio.

Actualmente los programas servidor de correo más extendidos son:

- Microsoft Exchange Server.
- Lotus Domino/Notes.
- Sendmail.

## **452.2 SMTP (SIMPLE MAIL TRANSFER PROTOCOL)**

El significado de las siglas de SMTP es Protocolo Simple de Transmisión de Correo. Este protocolo es el estándar de Internet para el intercambio de correo electrónico. SMTP necesita que el sistema de transmisión ponga a su disposición un canal de comunicación fiable y con entrega ordenada de paquetes, con lo cual, el uso del protocolo TCP (puerto 25) en la capa de transporte, es lo adecuado. Para que dos sistemas intercambien correo mediante el protocolo SMTP, no es necesario que exista una conexión interactiva, ya que este protocolo usa métodos de almacenamiento y reenvío de mensajes.

Realmente son tres los estándares que se aplican a un envío de correo de

esta clase. El termino SMTP es frecuente y erróneamente usado para referirse a la combinación del grupo de los tres estándares involucrados en el envío de correo electrónico. Ello se debe a que los tres están estrechamente relacionados, pero estrictamente hablando SMTP es uno de los tres estándares. Los tres estándares son:

- Un estándar para el intercambio de correo entre dos computadores, el cual especifica el protocolo usado para enviar correo entre "host" TCP/IP. Este estándar es SMTP y está definido originalmente en el RFC 821, siendo actualizado en los RFC 2821 y RFC 5321 (octubre 2008).
- Un estándar del formato del mensaje de correo, contenido en dos RFC:
  - o RFC 822 describe la sintaxis de las cabeceras del correo electrónico y describe la interpretación del grupo de campos de la cabecera. Este protocolo ha sido actualizado en los RFC 2821 y 5322.
  - o RFC 1049 describe como un conjunto de documentos de tipos diferentes del texto ASCII plano se pueden usar en el cuerpo del correo. Los estándares son PostScript, Scribe, SGML, TEX, TROFF y DVI. El nombre del protocolo oficial para este estándar es MAIL.
- Un estándar para el encaminamiento de correo usando el DNS (sistema de nombres de dominio), descrito en RFC 974. El nombre oficial del protocolo para este estándar es DNS-MX.

## **Funcionamiento**

El protocolo SMTP es un protocolo cliente/servidor, por lo que siempre es el usuario SMTP el que inicia la sesión y el servidor de correo el que responde.

El protocolo SMTP se basa en la entrega de mensajes extremo a extremo. Cuando un servidor de SMTP, requiere transmitir un mensaje a otro

servidor SMTP, el emisor (servidor que inicia la sesión SMTP) establece una conexión con el receptor (servidor que recibe petición de establecer sesión SMTP). Esta conexión es unidireccional, es decir, el emisor puede enviar correo al receptor, pero durante esa conexión, el receptor no puede enviar correo al emisor. Si el receptor tiene que enviar correo al emisor, tiene que esperar a que finalice la conexión establecida y establecer otra en sentido contrario, cambiando los papeles de emisor y receptor. Una vez establecida la conexión, el emisor envía comandos y mensajes.

El protocolo SMTP funciona con comandos y respuestas de texto escritas en ASCII-NVT (estándar USA - 7 bits).

Cada comando se envía al servidor SMTP al puerto 25 de manera predeterminada. A cada comando enviado por el cliente le sigue una respuesta del servidor SMTP compuesta por un código numérico de tres dígitos, seguido de un mensaje descriptivo. El número está pensado para un procesamiento automático de la respuesta, mientras que el texto permite que un humano intérprete la respuesta.

En el protocolo SMTP todas las órdenes, respuestas y datos son líneas de texto, delimitadas por el carácter CRLF. Todas las respuestas tienen un código numérico al comienzo de la línea.

## **Flujo**

Los pasos fundamentales para trabajar con el correo electrónico utilizando este protocolo son los siguientes:

- El cliente SMTP se conecta al servidor SMTP, realizando un telnet por el puerto 25 y espera respuesta.
- El servidor SMTP puede responder
  - o “220 Service Ready” junto con el nombre de dominio del

servidor, si el servicio de correo está disponible.

o “421 Service not available” si el destinatario es temporalmente incapaz de responder.

- Si el servicio está disponible el cliente se tiene que identificar. Para ello envía el comando HELO seguido por el nombre de dominio de su equipo. Desde abril de 2001, las especificaciones para el protocolo SMTP, definidas en RFC 2821, indican que el comando HELO sea remplazado por el comando EHLO.

o Un receptor SMTP que no soporte el RFC 2821 responderá con un mensaje "500 Syntax error, command unrecognized". El emisor SMTP debería intentarlo de nuevo con HELO o, si no puede retransmitir el mensaje, enviar un mensaje QUIT.

o Si un receptor SMTP soporta las extensiones de servicio, responde con un mensaje “250 OK” que incluye una lista de las extensiones de servicio que soporta.

- El emisor inicia ahora una transacción enviando el comando MAIL FROM: al servidor. Este comando contiene la ruta de vuelta al emisor que se puede emplear para informar de errores. Si se acepta el comando, el receptor responderá con un mensaje "250 OK". Cualquier otro código indica error.

- El segundo paso del intercambio de correo consiste en darle al servidor SMTP el destinatario del mensaje (puede haber más de un receptor). Esto se hace enviando uno o más comandos “RCPT TO: <destinatarios>” (si hay más de un destinatario éstos irán separados por comas. Cada uno de ellos recibirá una respuesta "250 Recipient OK" si el servidor conoce el destino, o un "550 No such user here" si no.

- El siguiente paso es informar al servidor que se va a empezar a introducir el cuerpo del mensaje, para ello se utiliza la orden DATA.
- El servidor contesta con "354 Start mail input, end with <CRLF>.<CRLF>". Donde se indica que el final del mensaje se debe finalizar con un punto en una única línea, seguido de un retorno de carro.
- El cliente envía los datos línea a línea, acabando con la línea <CRLF>. <CRLF> que el servidor reconoce con "250 OK" o el mensaje de error apropiado si cualquier cosa fue mal.
- Una vez que el servidor recibe el mensaje finalizado con un punto puede almacenarlo si es para un destinatario que pertenece a su dominio, o bien retransmitirlo a otro servidor para que finalmente llegue a un servidor del dominio del receptor.
- Ahora hay varias acciones posibles:
  - o El emisor no tiene más mensajes que enviar; cerrará la conexión con un comando QUIT, que será respondido con "221 Service closing transmission channel".
  - o El emisor no tiene más mensajes que enviar, pero está preparado para recibir mensajes (si los hay) del otro extremo. Mandará el comando TURN. Los dos SMTPs intercambian sus papeles y el emisor que era antes receptor puede enviar ahora mensajes.

Si se requiere autenticación TLS/SSL la conexión se realiza a los puertos 465 o 587, en vez del puerto 25.

### **452.3 POP (POST OFFICE PROTOCOL)**

El protocolo de oficina de correo, POP, es un protocolo cuya misión es la de



entrega final del correo al destinatario, no sirve para enviar correos ni para enviarlos. Su objetivo principal es poder gestionar los correos sin tener que estar conectado a Internet, es decir, permite a los usuarios con conexiones intermitentes ó muy lentas (p.ej. módem), descargar el correo electrónico mientras tienen conexión y revisarlo posteriormente incluso estando desconectados.

### **Modelo de comunicaciones POP**

La descripción del protocolo POP la podemos encontrar en el RFC 1939. La última versión del POP es la 3, por eso se suele referir a este protocolo como POP3.

El protocolo POP3 es un protocolo cliente/servidor, por lo que siempre es el usuario POP3 el que inicia la sesión y el servidor de correo el que responde.

El protocolo POP3 funciona con comandos y respuestas de texto escritas en ASCII.

El cliente POP se conecta con el servidor a través del puerto TCP, 110. Para conectarse al servidor, es necesario una cuenta de identificación en dicha máquina (lo que le permite tener un espacio reservado para sus correos). A continuación es necesario verificar que es dueño de la cuenta a través de una clave. Una vez conectado al sistema, el cliente POP puede dialogar con el servidor para saber, entre otros, si existen mensajes en la casilla, cuántos mensajes son o para solicitar la descarga de alguno de ellos.

Cuando la conexión TCP está establecida, POP3 continúa con tres fases:

- **Autorización:** Se envía el login y password para identificar al usuario que quiere leer el correo. Cuando se verifica que el nombre y la clave son correctos, el servidor pasa a un estado de transacción. Antes de pasar a este estado, el servidor POP bloquea el buzón para impedir que

los usuarios modifiquen o borren el correo antes de pasar al estado siguiente.

- **Transacción:** Se produce la manipulación del contenido del buzón del usuario.
- **Actualización:** Todas las modificaciones se realizan cuando el cliente finaliza el servicio (con el comando QUIT).

Por lo tanto, el protocolo POP3 administra la autenticación utilizando el nombre de usuario y la contraseña. Sin embargo, esto no es seguro, ya que las contraseñas, al igual que los correos electrónicos, circulan por la red como texto plano, sin cifrar. En realidad, según RFC 1939, es posible cifrar la contraseña utilizando un algoritmo MD5 y beneficiarse de una autenticación segura. Sin embargo, debido a que este comando es opcional, pocos servidores lo implementan. Además, el protocolo POP3 bloquea las bandejas de entrada durante el acceso, lo que significa que es imposible que dos usuarios accedan de manera simultánea a la misma bandeja de entrada.

## **Flujo**

Los pasos fundamentales para trabajar con el correo electrónico utilizando este protocolo son los siguientes:

- El cliente establece una conexión TCP en el puerto 110 del servidor POP.
- El servidor POP responderá con un indicador de estado y una palabra clave. Si el servicio está disponible responderá con el indicador de estado +OK, en caso contrario responderá con -ERR.
- Si el servicio está disponible se pasa a la fase de autorización y el

cliente se identificará con los comandos USER y PASS.

- Si la información es correcta, el servidor responderá con +OK y da acceso exclusivo al buzón.

- El cliente puede interactuar con el buzón, para ello utiliza los siguientes comandos

- o LIST muestra los correos que hay en el buzón y su tamaño.

- o STAT da el número de correos no borrados en el buzón y su longitud total.

- o TOP <nº\_mens> <num\_líneas> muestra n líneas del correo, cuyo número se da en el argumento. En el caso de una respuesta positiva del servidor, éste enviará de vuelta los encabezados del correo, después una línea en blanco y finalmente las primeras n líneas del correo.

- o RETR < nº\_mens > recoge un correo especificado por su número

- o DELE <nº\_mens> borra un correo especificado por su número.

- o RSET recupera los correos borrados (en la conexión actual).

- o UIDL obtiene el listado con todos los identificadores únicos de mensajes. El servidor asigna un identificador único a cada mensaje, de modo que no cambie su identificador entre sesiones. Este identificador es el UID.

- Para terminar la sesión POP se utiliza el comando QUIT. Se eliminan aquellos mensajes que han sido marcados con el comando DELE. Hasta que no se invoca a la orden QUIT los mensajes marcados no son

borrados del buzón.

Si se requiere autenticación TLS/SSL la conexión se realiza al puerto 995, no al puerto 110.

#### **452.4 IMAP - INTERNET MESSAGE ACCESS PROTOCOL**

El protocolo IMAP (Protocolo de acceso a mensajes de Internet) es un método utilizado por las aplicaciones cliente de correo electrónico para obtener acceso a los mensajes almacenados remotamente. En este caso, los mensajes no son recuperados por el gestor de correo, sino que se trabaja con ellos directamente sobre el servidor.

Es un protocolo más complejo que POP3. Algunas ventajas sobre el anterior son:

- Las transacciones IMAP pueden durar mucho más tiempo.
- El servidor guarda información del estado de los mails (si fueron leídos o no, si fueron guardados en una carpeta, etc).
- Se pueden definir distintas carpetas para acceder a distintos buzones.
- Se pueden devolver partes del mensaje al cliente, ahorrando ancho de banda.
- Se puede conectar más de un cliente al mismo buzón.
- Posee buscadores que se ejecutan en el servidor.
- A diferencia de POP (donde el Cliente debe estar conectado al Servidor para que se realicen los cambios), IMAP permite a los Clientes realizar cambios tanto estando estos conectados como desconectados.

- Es totalmente compatible con diferentes estándares de mensajes de Internet, como MIME.

Sin embargo posee ciertas desventajas:

- Es más complejo de implementar que POP3.
- El servidor debe ser más potente para atender a todos los usuarios. Consume más recursos de CPU, memoria, etc.

El protocolo IMAP es un protocolo cliente/servidor, por lo que siempre es el usuario IMAP el que inicia la sesión y el servidor de correo el que responde.

Los clientes IMAP pueden acceder siguiendo uno de estos tres modos de conexión:

- **Modo offline.** Periódicamente se conecta al servidor para descargar mensajes nuevos y sincronizar cualquier cambio que haya podido suceder en las diferentes carpetas. Existe la posibilidad de borrar los mensajes a medida que los descargamos, siguiendo un funcionamiento muy parecido a POP3.
- **Modo online.** Se accede directamente a la copia de los mensajes del servidor exactamente cuando hace falta, sincronizando los cambios prácticamente al vuelo.
- **Modo desconectado.** En este caso el cliente trabaja con una copia local mientras que no tiene acceso a internet, creando/borrando/leyendo sus emails. La próxima vez que se conecte a Internet estos cambios se sincronizarán con la copia maestra del servidor.

El protocolo IMAP funciona con comandos y respuestas de texto escritas en ASCII. Actualmente la versión operativa es la 4 por eso a este protocolo también se le conoce como IMAP4.

Dado que se parte de un modelo en el que los mensajes se guardan normalmente en el servidor después de ser leídos, IMAP define una manera sencilla de administrarlos: con buzones de correo, es decir, con carpetas. Éstas siguen una jerarquía de tipo árbol. Siguiendo el estándar siempre existirá un buzón de entrada que será el principal, pero podremos crear otras carpetas con diferentes atributos. Por ejemplo, existen atributos para especificar que una carpeta contiene sólo correos (*\NoInferiors*) o sólo carpetas (*\Noselect*), pero también pueden tener otros atributos que indiquen si existen o no mensajes nuevos desde la última vez que la abrimos (*\Marked* y *\Unmarked*).

Una clase parecida de etiquetas pueden tener los correos que se reciban y/o envíen. Uno de los más usados será el que indica si está leído o no (*\Seen*), pero también existen otros que indican que el mensaje ha sido contestado (*\Answered*), que el mensaje ha sido destacado (*\Flagged*), que es un borrador (*\Draft*), etc... Toda esta información se guarda directamente en el servidor y no en el cliente, lo que permite sincronizar perfectamente estos metadatos entre varios clientes.

En la RFC 2060 (actualmente la RFC 3501) se definen las instrucciones para poder interactuar con el servidor de correo y sus buzones.

Fases de una Sesión IMAP. Al igual que en el POP3, en una sesión IMAP existen las siguientes fases:

- Non-authenticated state: En este estado el Cliente aún no se ha autenticado con el Servidor.
- Authenticated state: El Cliente ha sido autenticado por el Servidor y debe seleccionar un buzón para interactuar.
- Selected state: El cliente ha seleccionado un buzón y se pueden realizar acciones sobre los correos contenidos en él.

- Logout state: La conexión ha sido finalizada.

## **Flujo**

Los pasos fundamentales para trabajar con el correo electrónico utilizando este protocolo son los siguientes:

- El cliente establece una comunicación TCP con el servidor IMAP por el puerto 143.
- El servidor responde con OK si el servicio está disponible; en caso contrario el servidor responderá con BAD.
- A continuación el cliente tiene que identificarse mediante el comando LOGIN <usuario> <password>, para poder acceder a los buzones. Esta es una forma no segura porque la password no va cifrada. Se puede utilizar el comando AUTHENTICATE para autenticar al usuario de forma segura.
- Si los datos son correctos el servidor responde OK. Si se produce un fallo de autenticación devolverá un NO. Si los argumentos no son válidos devolverá un BAD.
- Ahora el cliente puede interactuar con sus mensajes. Para ello usa los comandos:
  - o LIST para ver los buzones existentes.
  - o SELECT <nombre\_buzón> para ver el contenido de un buzón determinado.
  - o CREATE <nom\_buzón\_nuevo> para crear buzones.
  - o DELETE <nombre\_buzón> para borrar buzones.

- o `RENAME <nombre_buzón_old> <nombre_buzon_new>` para renombrar buzones.
- o `FETCH <num_mens> <parte_mens>` para ver las diferentes partes de los mensajes del buzón que se haya seleccionado.
- o `CLOSE` cierra el buzón y borra los mensajes marcados para borrar.
- o `EXPUNGE` borra todos los mensajes marcados para borrar.
- o `SEARCH` busca mensajes según algún criterio de búsqueda.
- o `COPY` copia los mensajes de una carpeta a otra.
- Con el comando `LOGOUT` se termina la sesión.

Si se requiere autenticación TLS/SSL la conexión se realiza al puerto 993, no al puerto 143.

## **452.5 FORMATO DE MENSAJES EN INTERNET**

La RFC 2822 define el estándar del formato de mensaje de Internet. El correo electrónico se divide en dos partes separadas por una línea en blanco.

- La cabecera del mensaje.
- El cuerpo del mensaje.

**Cuerpo del mensaje.** Contiene la información que se intercambian el emisor y el receptor. La forma en que está codificada viene determinada por el RFC 2231.

**Encabezados del mensaje.** Es la metainformación colocada antes del



cuerpo del mensaje. En general, el software de transporte de correo no revisa ni altera los encabezados del correo, a excepción de la cabecera Received. Están formadas por la tupla Palabra\_clave: valor. Las cabeceras más importantes son:

- From: Dirección del emisor del mensaje.
- Reply-to: Cuenta de correo a donde se dirigirán las respuestas al correo. En ausencia de este campo las respuestas se dirigirán a la/s dirección/es indicadas en el campo From.
- To: Este campo contiene la/s direcciones del/de los principal/es destinatario/s del mensaje.
- Cc: Copia a destinatarios. Campo que indica la/s dirección/es a las que se les hará llegar una copia del correo, aunque el contenido del mensaje puede que no vaya dirigido expresamente a ellos.
- Bcc: Copia oculta. Se manda una copia a los destinatarios aquí indicados sin que el resto de destinatarios tengan conocimiento de ello.
- Message-ID: Es un identificador único de cada mensaje. Este código es asignado por el servidor de donde sale el mensaje. Este identificador no se puede cambiar ni modificar.
- Reference: Contiene todos los Message-ID de los mensajes a los que éste hace referencia. Este campo es generado por la aplicación cliente.
- Keywords: Palabras clave que identifican el contenido del mensaje.
- Return-Path: Contiene la trayectoria de regreso al remitente.
- Received: Es la información que se utiliza para comprobar los problemas que hayan aparecido en el reparto de un mensaje. En ella se

muestran las direcciones de las máquinas por las que pasó el mensaje en dirección a su destino, junto con la fecha y hora en que lo hizo.

- Date: Fecha y hora en la que el mensaje es entregado a la cola del servidor SMTP para su envío. Este campo lo establece el servidor origen.
- Subject: Este campo contiene un pequeño texto con la descripción del asunto del mensaje.
- X- : Son campos definidos por el usuario. Siempre tienen que empezar por X-, seguidos del nombre que se le quiera asignar al campo. Por ejemplo X-mailer: "Mi gestor de correo". Se está utilizando una cabecera X-SPAM para marcar correos como presuntos correos basura.

Sólo las cabezas subrayadas son obligatorias según el estándar.

## **452.6 EXTENSIONES DE SMTP**

### **452.6.1 ESMTP**

El protocolo ESMTP es una extensión del protocolo SMTP, definido en la RFC 4954.

Se trata de un mecanismo para autenticar la identidad del cliente que se conecta al servidor, y además permite la negociación de una capa de seguridad para hacer más segura la comunicación. El protocolo SMTP permanece inalterado, lo que se hace es agregar los siguientes comandos:

- EHLO dominio: Hace que el servidor realice una consulta al DNS del reverso del dominio indicado para verificar que el mismo exista.
- ETRN dominio (Extended Turn). Este comando permite que el cliente le pida al servidor que le envíe todos los mensajes que posee destinados al cliente. Si hay mensajes para la máquina cliente, el servidor debe

iniciar una nueva sesión SMTP para enviarle los mensajes.

- AUTH: Comando que sirve para negociar un protocolo de seguridad para el intercambio de datos. Los posibles protocolos, para la capa de seguridad, que se pueden negociar los da como respuesta el servidor al comando EHLO.

#### **452.6.2 MIME**

El protocolo SMTP impone determinadas restricciones sobre el contenido de los mensajes:

- El contenido sólo debe estar compuesto de caracteres ASCII, no se puede enviar ficheros binarios como audio, video, documentos, etc.
- Las líneas no pueden exceder los 100 caracteres.
- El tamaño total del contenido no puede exceder una determinada dimensión.
- Además, también existen problemas a la hora de enviar mensajes en lenguajes distintos del inglés:
  - o Lenguajes sin alfabetos “occidentales” (chino, japonés)
  - o Lenguajes con alfabetos no latinos (ruso, árabe)
  - o Lenguajes con acentos (alemán, castellano)

Para solventar estas limitaciones se han definido las especificaciones MIME (Multipurpose Internet Mail Extensions) que son unas extensiones del correo electrónico utilizadas también en otros protocolos como el HTTP que permiten la transmisión de datos no ASCII, a través de e-mail, en el cuerpo del mensaje.

MIME no cambia a SMTP ni lo reemplaza, por lo que los mensajes a enviar con MIME también cumplirán este protocolo. Dado que SMTP utiliza para comandos y respuestas el ASCII de 7 bits, el camino seguido para transmitir cualquier fichero es transformar (codificar) el fichero no ASCII en ASCII de 7 bits (haciéndolo compatible con SMTP), transmitirlo en este formato y reconvertirlo en destino al formato original (decodificarlo).

MIME incorpora las siguientes características al servicio de correo electrónico:

- Capacidad de enviar múltiples adjuntos en un solo mensaje.
- Longitud ilimitada del mensaje.
- Uso de conjuntos de caracteres no pertenecientes al código ASCII.
- Uso de texto enriquecido (diseños, fuentes, colores, etc.).
- Adjuntos binarios (ejecutables, imágenes, archivos de audio o video, etc.), que se pueden dividir de ser necesario.

Las cabeceras descritas en la RFC 2822 son suficientes para enviar correo codificado en texto ASCII, pero no son adecuadas para mensajes multimedia. Para ello MIME añade unas cabeceras que describen el tipo de contenido del mensaje y el tipo de código. Estas son:

- **MIME-Version:** Contiene la versión de las extensiones MIME empleadas en el mensaje.
- **Content-Transfer-Encoding:** Señala como ha sido codificado el mensaje para su transmisión por e-mail, de forma que pueda viajar sin problemas de que sea corrompido desde el destinatario al receptor a través de los agentes de correo (MUAs). Para transferir datos binarios,

MIME ofrece cinco formatos de codificación:

- o 7bit: Significa que el fichero es SÓLO texto ASCII (caracteres no acentuados). Las líneas deben ser "cortas", de 100 caracteres o menos, terminando con CRLF.
- o Quoted-Printable: Utilizado por texto que es mayoritariamente US-ASCII (7 bit) pero con un pequeño porcentaje de caracteres "extraños" (8 bit). Este es el caso del castellano.

En esta codificación, cada carácter de 8-bits es codificado en tres caracteres de 7 bits, el primero el signo igual (=) y el valor hexadecimal del carácter. Por ejemplo, la "ñ", "F1" en hexadecimal, se codifica como "=F1".

- o base64: usado para codificar secuencias arbitrarias de octetos de forma que satisfaga las reglas de 7bit. Se utiliza para enviar binarios.
- o 8bit: formato de texto de 8 bits.
- o binary: envío de binarios.
- **Content-Type:** indica que tipos de datos contiene el mensaje. Un tipo de MIME está compuesto de la siguiente manera: tipo\_mime\_principal/subtipo\_mime Pueden encontrarse los siguientes tipos:
  - o text: texto con o sin formato.
  - o image: imágenes estáticas.
  - o video: imágenes dinámicas, puede incluir audio.
  - o audio: sonido.

- o **message:** significa que el contenido está configurado según el estándar RFC 822; esto puede ser usado para reexpedir mensajes.
- o **application:** se emplea para señalar que el contenido es para ser enviado a un programa externo, por ejemplo, texto para una impresora PostScript.
- o **multipart:** un mensaje también puede tener varias partes con varios contenidos separados, incluso de tipos diferentes (texto, audio e imágenes). Incluso cada parte puede tener subpartes (ser a su vez multiparte), puesto que el formato MIME puede ser recursivo.

Aparte del tipo, también se puede especificar un subtipo, ambos separados por una barra inclinada /. Por ejemplo image/gif es una imagen en formato GIF; el tipo es image y el subtipo gif; text/html, text/plain, etc.

Si el tipo es multipart, los subtipos admitidos son:

- o **mixed:** permite que en un solo mensaje contenga varios submensajes independientes, cada uno con su tipo y codificación. De esta forma se puede incluir en un mensaje imágenes, audio, video....
- o **parallel:** permite incluir en un mensaje subpartes que se pueden ver simultáneamente, por ejemplo reproducir audio y video.
- o **digest:** permite incluir en un mensaje varios mensajes.
- o **alternative:** permite que en un mismo mensaje se pueda incluir una única información pero en diversos formatos. Esto es útil cuando los destinatarios tienen distinto hardware y/o sistema operativo.

Finalmente, puede tener parámetros opcionales empezando por un punto y coma ;. Por ejemplo, el parámetro charset= en Content-type:

text/plain; charset=iso-8859-1, indica que el cuerpo del mensaje utiliza el juego de caracteres ISO- 8859-1.

### 453 SERVIDORES DE APLICACIONES Y SERVIDORES WEB

La diferencia básica entre el servidor web y servidor de aplicaciones es que el servidor web sirve para ver las páginas en un navegador web, mientras que un servidor de aplicaciones proporciona los métodos necesarios, que pueden ser llamados por las aplicaciones cliente. En otras palabras, las peticiones HTTP son manejados por los servidores web y la lógica de negocio se sirve a los programas de aplicación, a través de una serie de protocolos en el servidor de aplicaciones. En un servidor de aplicaciones, un cliente puede utilizar la GUI y los servidores web, mientras que en los servidores web el cliente puede usar HTML y HTTP.

	<b><i>Servidor Web</i></b>	<b><i>Servidor de Aplicaciones</i></b>
<i>¿Qué es?</i>	Un servidor que gestiona conexiones HTTP.	Un servidor que expone la lógica del negocio al cliente mediante una serie de protocolos, pero no exclusivamente HTTP
<i>¿Añade Funcionalidad?</i>	Un servidor Web no añade funcionalidad simplemente recibe una petición y envía	Añade funcionalidad, puesto que implementa una lógica de negocio intermedia.

	la respuesta al cliente.	
<i>¿Qué tipo de aplicaciones sirve?</i>	Sólo basadas en web	Aplicaciones basadas en web, pero también otras que no lo son, lo cual es posible ya que un servidor de aplicaciones incluye internamente un servidor web.
<i>¿Qué tipo de clientes permite?</i>	Navegadores.	Navegadores e interfaces pesadas.
<i>¿Cuáles son sus funciones?</i>	Almacenar ficheros escritos en HTML, PHP, etc., de tal forma que sean accesibles para los navegadores web cuando los sitios web necesiten acceder a ellos.	Ofrecer aplicaciones a otro sistema.



--	--	--

#### **454 SERVIDORES DE APLICACIONES**

Por servidor de aplicaciones entendemos a aquel que permite la ejecución de una serie de aplicaciones. Habitualmente se trata de un programa software que gestiona casi por completo las funciones de lógica de negocio y de acceso a los datos de la aplicación. Su propósito es gestionar centralizadamente la forma en que los clientes se conectan a la base de datos o a los servicios con los que éstos deben interactuar.

Los servidores de aplicaciones comienzan a surgir cuando se hace patente que las aplicaciones cliente/servidor iban a presentar problemas de escalabilidad cuando se tratase de servir a un gran número de usuarios. Además era necesario trasladar las reglas de negocio a un lugar intermedio entre los clientes y la base de datos.

El concepto de servidor de aplicaciones está muy ligado con el de sistema distribuido, los cuales permiten mejorar 3 aspectos fundamentales en una aplicación:

- *La alta disponibilidad:* se refiere a la necesidad de que un sistema funcione 24 horas al día, todos los días. Para poder cumplir con esta característica son necesarias técnicas de balanceo de carga y de recuperación ante fallos.
- *La escalabilidad:* consiste en la capacidad de hacer crecer un sistema cuando aumentan el número de peticiones. Cada sistema puede atender a un número limitado de peticiones, puesto que sus recursos son finitos, al añadir nuevos equipos la cantidad de recursos se multiplica y con ello el número de peticiones que pueden ser atendidas.

- *El mantenimiento:* tiene que ver con la facilidad para realizar actualizaciones, depurar fallos y mantener el sistema.

#### **454.1 SERVICIOS PROPORCIONADOS POR UN SERVIDOR DE APLICACIONES**

- *Gestión de la sesión:* el servidor debe conservar la información entre peticiones de un usuario mientras dure dicha sesión.
  - Esta es una característica fundamental para las aplicaciones de comercio electrónico, que requieren establecer al usuario a través de su navegación por el sitio web, sin embargo, el protocolo http es un protocolo sin sesión, por lo que no permite mantener una conexión abierta entre cliente y servidor más allá de lo que dura la transferencia de información. Por ello son los servidores de aplicaciones los que se encargan de todo lo relacionado con la gestión de la sesión.
- *Balanceo de carga:* un servidor de aplicaciones debe proporcionar técnicas para equilibrar su propia carga, es decir, debe ser capaz de repartir el procesamiento entre diversos servidores, lo cual es fundamental para su escalabilidad.
  - Las peticiones que realizan los clientes se transmiten a la máquina que esté menos ocupada en cada momento, lo cual mejorará el rendimiento global de la aplicación.
  - Con un buen balanceo de carga, además de conseguir un sistema más escalable, se consigue una mayor tolerancia a fallos.
- *Acceso a los datos:* un servidor de aplicaciones proporciona un acceso sencillo para realizar la administración de las conexiones a bases de datos relacionales.
  - Es habitual que también permitan realizar acceso a otros tipos de fuentes de datos como:
    - ERP
    - Repositorios XML
    - Sistemas heredados

- *Pooling de conexiones*: es habitual que los servidores de aplicaciones mantengan de forma permanente conexiones con las bases de datos. Estas conexiones se distribuyen entre los procesos de forma transparente, ya que sería muy costoso, además de influir negativamente en el rendimiento de la aplicación, el abrir una conexión por cada consulta que se quiera realizar.
- *Gestión de transacciones*: las transacciones son fundamentales en cualquier software y más aun en los de tipo comercial, puesto que evitan la aparición de información inconsistente.
  - Los servidores de aplicaciones suelen contar con esta característica, de forma que con indicar en qué momento se inicia una transacción y en cual se finaliza, el propio sistema se encargaría de deshacer los pasos intermedios en caso de que se produzca un error en la aplicación.

## **454.2 ESTÁNDAR J2EE**

Las plataformas más comunes en las que se asientan los servidores de aplicaciones son J2EE y .NET. J2EE está más extendida y hasta hace relativamente poco, era impensable implementar un servidor de aplicaciones que no siguiese este modelo.

El estándar J2EE permite desarrollar aplicaciones empresariales de forma eficiente y sencilla. El hecho de desarrollar una aplicación con tecnologías J2EE permite que esta sea desplegada en cualquier servidor de aplicaciones que cumpla con dicho estándar. Un servidor de aplicaciones es una implementación de la especificación J2EE que se compone de:

1. Cliente Web o contenedor de applets: es un navegador web que

interactúa con el contenedor web mediante HTTP.

- a. Puede ejecutar applets y código javascript.
  - b. Recibe páginas HTML o XML.
2. Aplicación Cliente: se trata de clientes que no se ejecutan dentro de un navegador.
- a. Pueden utilizar distintas tecnologías para comunicarse con el contenedor web.
  - b. Pueden comunicarse directamente con la base de datos.
3. Contenedor Web o servidor web: se corresponde con la parte visible de un servidor de aplicaciones.
- a. Emplea los protocolos HTTP y SSL
4. Servidor de aplicaciones: proporciona servicios que dan soporte a la ejecución y disponibilidad de las aplicaciones desplegadas.

Existen distintas implementaciones partiendo de este estándar, cada una con sus propias peculiaridades que las pueden hacer más adecuadas para un determinado sistema. Algunas de las más destacadas son:

- BEA WebLogic
- IBM WebSphere
- Sun-Netscape IPlanet
- Sun One
- Oracle IAS
- Borland AppServer
- HP Bluestone

### **454.3 ESTRUCTURA DE UN SERVIDOR DE APLICACIONES**

Un servidor de aplicaciones se asienta en una estructura en 3 capas que permite realizar una estructuración más eficiente del sistema.

- *Capa Cliente*: contiene los programas que ejecutan los usuarios,

como navegadores Web. Estos programas pueden estar implementados en cualquier lenguaje de programación.

- *Capa Media*: contiene el servidor de aplicaciones y otros que pueden ser direccionados por los clientes, como servidores proxy o servidores web existentes.
- *Capa Datos*: contiene los recursos, como sistemas de bases de datos, ERP, etc.

#### **454.4 BEA WEBLOGIC**

BEA WebLogic Server es un servidor de aplicaciones completo y basado en estándares, que proporciona el fundamento sobre el cual una empresa puede construir sus aplicaciones. Presenta un completo conjunto de características, como son, el cumplimiento de los estándares abiertos, la arquitectura de varios niveles, y el apoyo para desarrollo basado en componentes, etc.

BEA WebLogic Server proporciona todas las funciones básicas esenciales de un servidor de aplicaciones y servicios, tales como:

- Equilibrio de carga
- Tolerancia a fallos
- Servicios Web
- Transparencia en la red
- Integración de sistemas heredados
- Gestión de transacciones
- Seguridad
- Multi-threading
- Persistencia
- Conectividad con bases de datos
- Agrupación de recursos

Estas funcionalidades agilizan el desarrollo de aplicaciones y alivian los

esfuerzos de los desarrolladores.

Además de J2EE, WebLogic Server implementa todos los estándares importantes de programación, integración y trabajo en red que son la base para la construcción de una infraestructura de aplicaciones, incluyendo:

- XML: BEA WebLogic Server implementa la última versión del Api de Java API para el procesamiento de XML (JAXP), e incluye un analizador integrado Apache Xerces y un analizador XML de alto rendimiento diseñado específicamente para pequeñas y medianas empresas.
- SOAP: SOAP es el nuevo estándar para el intercambio de información en un entorno distribuido. Es el protocolo de comunicación para definir el formato de los datos para los servicios Web que se entregan a través de HTTP.
- WSDL: WSDL es un lenguaje basado en XML utilizado para describir un servicio Web publicado. BEA WebLogic Server tiene soporte incorporado para WSDL y genera un guión WSDL de forma automática cuando un servicio Web se implementa en el servidor WebLogic.
- UDDI: Un registro UDDI es un directorio de servicios web, distribuido y basada en Web, muy similar a una libreta de teléfonos.. BEA WebLogic Server incluye incrustado un registro UDDI y una API para la búsqueda y la actualización de este, o cualquier otro registro UDDI.
- JMX y SNMP: La infraestructura del servidor WebLogic se basa en el estándar abierto y extensible JMX. Además, el agente SNMP está disponible para la compatibilidad con sistemas que se basen en SNMP.
- Los administradores de sistemas pueden configurar sus políticas de seguridad de basadas en roles de acceso.

## **455 SERVIDORES WEB**

### **455.1 INTRODUCCIÓN CLIENTE/SERVIDOR**

La tecnología Cliente/Servidor consiste en el procesamiento cooperativo de la información mediante un conjunto de procesadores, en el cual múltiples clientes, geográficamente dispersos, pueden realizar peticiones a uno o más servidores centrales.

Desde una perspectiva funcional, podemos definir cliente-servidor como una arquitectura distribuida que permite a los usuarios obtener acceso de forma transparente a la información. Este tipo de arquitectura es la más extendida en los sistemas distribuidos.

Un sistema cliente servidor se basa en las siguientes características:

- *Servicio*: el servidor los proporciona y el cliente los utiliza.
- *Recursos compartidos*: son muchos los clientes que emplean los mismos servidores mediante los cuales comparten recursos no sólo lógicos sino también físicos.
- *Protocolos asimétricos*: los clientes son los encargados de iniciar la comunicación con el servidor, los cuales esperan el establecimiento de la conexión de forma pasiva.
- *Transparencia de localización*: los clientes no saben donde se localizan físicamente los recursos que desean utilizar.
- *Independencia de la plataforma*.
- *Sistemas débilmente acoplados*: Interacción basada en envío de mensajes.
- *Encapsulamiento de servicios*: la implementación que los servidores realizan de los servicios son transparentes para los clientes.
- *Escalabilidad horizontal*: incorporar nuevos clientes.

- *Escalabilidad vertical*: aumentar la potencia de los servidores.
- *Integridad*: tanto los datos como los programas están centralizados en servidores que facilitan su integridad y mantenimiento.

## **455.2 SERVIDORES WEB**

Un servidor http o servidor web es un programa que permite procesar las peticiones de los distintos navegadores, sirviendo los recursos que estos soliciten mediante los protocolos HTTP o HTTPS. De forma general, un servidor web funciona de manera muy simple, ejecutando constantemente las siguientes acciones:

1. Esperar peticiones en el puerto TCP indicado.
  - a. Por defecto se empleará el puerto 80.
2. Recibir una petición.
3. Procesar la solicitud.
4. Enviar al cliente la respuesta obtenida, empleando la misma conexión por la que se recibió la petición.
5. Volver a esperar nuevas peticiones.

Si un servidor web se ciñe al patrón anterior, cumplirá todos los requisitos básicos de los servidores HTTP, aunque se verá limitado a servir ficheros estáticos. Los servidores web existentes actualmente se han diseñado e implementado a partir del patrón anterior, donde la variación entre ellos sólo radica en el tipo de peticiones que van a atender, si son o no multiproceso, etc.

## **455.3 CARACTERÍSTICAS DE LOS SERVIDORES WEB**

- *Servir ficheros estáticos*: un servidor web debe ser capaz de servir ficheros estáticos que se localicen en algún lugar del disco (Requisito



imprescindible).

- o Debe ser posible especificar qué parte del disco se servirá.
- o Aunque un servidor pueda especificar un directorio por defecto, no debe obligar a emplear uno concreto.
- o En muchos servidores es posible especificar otros subdirectorios o directorios, indicando en qué punto del sistema de ficheros virtual del servidor irán localizados los recursos.
- *Seguridad*: un servidor web debe poder especificar directivas de seguridad, esto es, establecer quién puede acceder a qué recursos.
  - o Algunos servidores permiten especificar los ficheros que se considerarán como índice del directorio.
- *Contenido dinámico*: es una de las características fundamentales. Indica la capacidad del servidor para ofrecer contenido dinámico.
  - o La gran mayoría del contenido web que se sirve es dinámico.
  - o Punto fundamental a la hora de elegir un servidor.
- *Soporte para distintos lenguajes*: la mayor parte de los servidores ofrece soporte para algunos lenguajes de programación como:
  - o PHP
  - o JSP: para que un servidor atienda peticiones JSP requerirá algún tipo de software para funcionar, como un contenedor de Servlets.
  - o ASP
  - o CGI (sistema más antiguo y sencillo para generar contenido dinámico.)

Antes de seleccionar un lenguaje de programación de servidor, es necesario considerar, si se desea un lenguaje más estándar que pueda ser atendido por cualquier servidor genérico o bien, si se

prefiere una arquitectura concreta, etc.

#### **455.4 ARQUITECTURA**

La arquitectura de un servidor web se divide en:

1. Capa servidor
2. Capa soporte

**Capa servidor:** contiene 5 subsistemas, cuya función es la de implementar las funcionalidades del servidor.

- *Subsistema de recepción:* es el encargado de esperar las peticiones del cliente a través de la red.  
o Tiene la capacidad de manejar peticiones simultáneas, pudiendo analizarlas para determinar si son o no compatibles con el navegador.
- *Analizador de peticiones:* asocia al recurso de red un archivo local.
- *Control de acceso:* se encarga de validar y permitir el acceso.
- *Controlador de recursos:* fija el tipo de recurso que se ha solicitado, lo ejecuta y obtiene la respuesta.
- *Registro de transacción:* su función es registrar las peticiones junto con sus respuestas.

**Capa soporte:** conforma la interfaz entre el servidor web y el sistema operativo, y maneja los siguientes subsistemas.

- *Útil:* contiene las funciones que utilizan los otros subsistemas.
- *Capa abstracta del sistema operativo:* encapsula el funcionamiento del sistema operativo para facilitar la portabilidad del servidor entre las diferentes plataformas.

#### **455.4.1 Funcionamiento del servidor web**

Un servidor web se ejecuta esperando peticiones por parte de un navegador web (cliente) y atendiendo a dichas peticiones de forma adecuada, respondiendo mediante un mensaje de error o una página web con la respuesta a la petición formulada.

Por ejemplo, si tecleamos [www.xunta.es](http://www.xunta.es) en un navegador, el proceso que se desencadena es el siguiente:

- El navegador realiza una petición HTTP al servidor de la dirección solicitada.
- El servidor responde enviando el código HTML de la página solicitada.
- El cliente recibe el código de la página, lo interpreta y lo muestra en pantalla.

Es el cliente el encargado de interpretar el código HTML, mostrar los textos y objetos de la página, sus colores, fuentes, etc. El servidor por su parte se limita a transmitir el código de la página sin realizar ningún tipo de interpretación de dicha página.

Un servidor web, además de transmitir código HTML, también puede entregar aplicaciones web, que son segmentos de código que se ejecutan cuando se producen ciertas peticiones o respuestas HTTP.

Es necesario distinguir entre:

1. *Aplicaciones en el lado del cliente*: es el navegador web el encargado de ejecutar estas aplicaciones en el equipo del usuario (scripts). En esta categoría encontramos aplicaciones como Applets de Java o Javascript.
- El servidor proporciona el código de estas aplicaciones al cliente y es el navegador de éste, el que las ejecuta.

- Es necesario que el navegador del cliente disponga de la capacidad para ejecutar estas aplicaciones.
- Por defecto, la mayoría de los navegadores permite ejecutar scripts de java y javascript, aunque mediante plugins pueden añadirse más lenguajes.

2. *Aplicaciones en el lado del servidor*: es el servidor el encargado de ejecutar la aplicación, la cual, una vez ejecutada genera un código HTML que el servidor toma y lo envía al navegador del cliente mediante HTTP.

Generalmente la opción que se escoge es la de las aplicaciones en el lado del servidor, ya que al ejecutarse en el servidor y no en el equipo del cliente, éste no requiera de ningún tipo de software o funcionalidad añadida, mientras que en el caso de las aplicaciones en el lado del cliente, sí que es necesario.

## **455.5 APACHE**

El servidor Apache es un servidor web de código abierto, que se desarrolla dentro del proyecto HTTP Server de la Apache Software Foundation, y que puede ser instalado en plataformas Windows, Unix, Mac y otras. Este servidor viene instalado en la mayoría de distribuciones de Linux y en Mac OS X, Apache viene integrado como parte de su propio servidor web.

Se trata del servidor web más empleado, y si bien presenta algunas vulnerabilidades de seguridad, la gran mayoría de estas sólo podrían ser explotadas de forma local y no remotamente.

### **455.5.1 httpd.conf**

Apache puede ser configurado mediante el fichero *httpd.conf*. Cada vez

que se introduzca una modificación en este fichero, será necesario reiniciar el servicio apache. Se trata de un servidor altamente configurable, si bien su interfaz gráfica no es demasiado intuitiva.

El fichero de configuración `httpd.conf` puede dividirse en varias secciones:

- *Sección 1:* Entorno global. Sección del fichero donde se localizan las rutas a otros ficheros de configuración y se describe el funcionamiento general del servidor.
- *Sección 2:* Entorno servidor principal. Sección del fichero donde se describe la configuración que no atiende a las peticiones de los servidores virtuales. Se trata del comportamiento predeterminado del servidor.
- *Sección 3:* Entorno de servidores virtuales. Sección del fichero donde se pueden configurar servidores virtuales para trabajar con el mismo programa.

### ***Directivas de entorno global***

La configuración se realiza mediante directivas, variables almacenadas en un archivo de texto, que permiten alterar y controlar el funcionamiento de Apache en función de los valores que estas tomen.

- *ServerType:* permite indicar cómo será la respuesta del servidor.
  - o *Inetd:* se ejecuta cuando hay una petición.
  - o *Standalone:* siempre existe un proceso `httpd` en ejecución y este crea nuevos hijos para las conexiones con los diferentes clientes.
- *ServerRoot:* permite detallar el directorio que actuará como raíz del servidor.
- *Timeout:* permite especificar el número de segundos que se mantiene a la espera un servidor, desde que se recibe la petición hasta que se entiende la conexión como inactiva.
- *MaxClients:* limita el número máximo de clientes que se pueden

conectar de forma simultánea. Si este número se supera, los clientes son bloqueados.

- *Listen*: permite a Apache atender peticiones en otra dirección y/o puertos además de los establecidos por defecto.
- *BinAddress*: se emplea para especificar qué direcciones o IPs deben atenderse en el servidor. Permite dar soporte a servidores virtuales.
- *LoadModule*: permite cargar un nuevo módulo para aportar mayor funcionalidad al servidor.

### ***Directivas de configuración del servidor principal***

- *Port*: permite especificar el puerto en que escuchará el servidor. Sólo puede existir una directiva *Port*, mientras que se pueden especificar varias *Listen*.
- *User y Group*: permite indicar el usuario o grupo que puede iniciar la ejecución de httpd.
- *ServerAdmin*: establece la dirección de correo electrónico a la que enviar los problemas que puedan surgir. Esta dirección se mostrará en las páginas de error que genera el servidor.
- *ServerName*: permite asignar el nombre del servidor, que será mostrado a los clientes. No es aconsejable emplear el nombre real de la máquina.
- *ServerSignature on/off/email*: se emplea para que en caso de acceso a una página inexistente, el servidor devuelva una página de error indicando la versión de Apache y el nombre de la máquina.
- *DocumentRoot*: especifica el directorio en que se emplazan los documentos web que el servidor podrá a disposición de los clientes.

### **455.5.2 Módulos de Apache**

Apache es un servidor estructurado en módulos cuya configuración se realiza mediante la modificación de las directivas presentes en cada

módulo.

Los módulos de Apache pueden ser clasificados en los siguientes grupos:

- Módulos base: módulos que engloban las funciones básicas de Apache.
- Módulos multiproceso: módulos que se encargan de la interconexión con los puertos del ordenador, aceptando las peticiones y enviando a los distintos hilos a atender las peticiones. Módulos adicionales: cualquier módulo que incorpore una funcionalidad al servidor.

## **455.6 MICROSOFT IIS**

IIS (Internet Information Services) es un servidor web específico para el sistema operativo Microsoft Windows. IIS convierte un ordenador en un servidor Web, que permite publicar páginas web y hacerlas accesibles localmente, hacia una intranet o hacia Internet, además proporciona las funciones y herramientas necesarias para realizar de forma sencilla la administración de un servidor web seguro.

IIS se basa en diversos módulos que le proporcionan la capacidad de servir varios tipos de páginas, como ASP (Active Server Pages), ASP.NET, PHP o Perl.

### **455.6.1 Administración de IIS**

La última versión de IIS es la 7, aplicable a Windows 7, Windows Server 2008, Windows Server 2008 R2 y Windows Vista.

En IIS7 hay varias herramientas para realizar su administración y configuración. Entre las cuales se incluyen:

- Administrador de IIS.
- Herramienta de línea de comandos denominada Appcmd.exe.

- Almacén de configuración de IIS que consta de archivos ApplicationHost.config y Web.config.
- Espacio de nombres de Instrumental de administración de Windows (WMI).

## **455.7 LIGHTTPD**

Lighttpd es un servidor web libre, distribuido bajo la licencia BSD, diseñado para ser rápido, seguro, flexible, y respetuoso con los estándares. Está diseñado para entornos donde la velocidad es muy importante y se requieren respuestas rápidas y de alta escalabilidad. Consume menos memoria y procesador que otros servidores.

Algunas características de lighttpd son:

- Permite la comunicación con programas externos mediante SCGI o FastCGI.
- Tiene un módulo de reescritura y de redirección de URLs.
- Se han hecho mejoras específicas para su integración con PHP y Ruby on Rails.
- Permite módulos externos.
- Permite VirtualHosting.
- Puede servir tanto HTTP como HTTPS.
- Autenticación con LDAP, httpasswd o MySQL.
- Acepta Webdav.

Lighttpd puede usarse solo o combinado con otros, de hecho, es habitual emplearlo para liberar de carga a otros servidores más lentos, especialmente cuando hay que realizar el envío de ficheros grandes que suele ser mucho más rápido que en el resto de servidores. Es común encontrar Lighttpd en combinación con instalaciones de Apache, para



hacerlo más escalable y rápido en situaciones de carga.

## **456 BIBLIOGRAFÍA**

- Internet y Correo electrónico. Silva Salinas, Sonia; López Sanjurjo, Catherin, (aut.) Ideaspropias Editorial. 2007
- Correo electrónico. Romero Dueñas, Carlos y Gonzalez Hermonso, Alfredo. Edelsa, 2001
- Apache: The Definitive Guide, Third Edition. Ben Laurie, Peter Laurie
- Apache : soluciones y ejemplos para administradores de Apache. Ken Coar and Rich Bowen
- Web Server Technology. Nancy J. Yeager, Robert E. McGrath
- Linux Apache web server administration. Charles Aulds
- Managing Internet information services, Cricket Liu
- Client-server computing: architecture, applications and distributed systems management. Bruce R. Elbert and Bobby Martya

**Autor:** Francisco Javier Rodriguez Martínez

Subdirector de Sistemas Escola Superior Enxeñaría Informática Ourense

Colegiado del CPEIG

**46. ADMINISTRACIÓN Y GESTIÓN DE SISTEMAS Y ALMACENAMIENTO. VIRTUALIZACIÓN DE SERVIDORES. VIRTUALIZACIÓN DEL PUESTO CLIENTE. COMPUTACIÓN BASADA EN SERVIDOR (SBC). GRID COMPUTING. CLOUD COMPUTING. GREEN IT Y EFICIENCIA ENERGÉTICA. REDES SAN Y ELEMENTOS DE UM SAN. VIRTUALIZACIÓN DEL ALMACENAMIENTO. GESTIÓN DEL CICLO DE VIDA DE LA INFORMACIÓN (ILM). SISTEMAS DE BACKUP: HARDWARE Y SOFTWARE DE BACKUP. ESTRATEGIAS DE BACKUP A DISCO. REPLICACIÓN LOCAL Y REMOTA, ESTRATEGIAS DE RECUPERACIÓN.**

**Tema 46: Administración y gestión de sistemas y almacenamiento. Virtualización de servidores. Virtualización del puesto cliente. Computación basada en servidor (SBC). Grid Computing. Cloud computing. Green IT y eficiencia energética. Redes SAN y elementos de un SAN. Virtualización del almacenamiento. Gestión del ciclo de vida de la información (ILM). Sistemas de backup: hardware y software de backup. Estrategias de backup a disco. Replicación local y remota, estrategias de recuperación.**

---

#### **46.1 Administración y gestión de sistemas y almacenamiento**

##### *46.1.1 Administrador de red*

##### *46.1.2 Administración y gestión de redes*

#### **46.2 Virtualización de servidores**

##### *46.2.1 Funcionamiento*

#### **46.3 Virtualización del puesto cliente**

##### *46.3.1 Modos de Operación de VDI*

#### **46.4 Computación basada en servidor**

#### **46.5 Grid Computing**

##### *46.5.1 Características:*

##### *46.5.2 Funcionalidades:*

##### *46.5.3 Arquitectura Grid*

#### **46.6 Cloud computing**

##### *46.6.1 Arquitectura*

##### *46.6.2 Modelos de implementación*

#### **46.7 Green IT e eficiencia energética**

##### *46.7.1 Tecnologías verdes*

##### *46.7.2 Actividades relacionadas con Green IT*

## **46.8 Redes SAN y elementos de una SAN**

### *46.8.1 Estructura de las SAN*

## **46.9 Virtualización del almacenamiento**

### *46.9.1 Virtualización por bloques*

### *46.9.2 Virtualización a nivel de archivo*

### *46.9.3 Diferencias entre NAS y SAN*

## **46.10 Gestión del Ciclo de Vida de la Información (ILM)**

### *46.10.1 Gestión del Ciclo de Vida de los Datos*

### *46.10.2 Gestión del Ciclo de Vida de la Información*

### *46.10.3 Algunas soluciones para la gestión*

## **46.11 Sistemas de Backup: hardware y software de backup**

### *46.11.1 Hardware de Backup*

### *46.11.2 Software de Backup*

## **46.12 Estrategias de Backup a Disco**

## **46.13 Replicación local y remota, estrategias de recuperación**

### *46.13.1 Replicación Local*

### *46.13.2 La replicación remota*

## **46.14 Replicación local y remota, estrategias de recuperación**

### *46.14.1 Replicación Local*

## **46.15 Bibliografía**

# **461 ADMINISTRACIÓN Y GESTIÓN DE SISTEMAS Y ALMACENAMIENTO**

## **461.1 Administrador de red**

Un administrador de red como su nombre indica "administra" una red, es decir, se encarga de:

- Instalación y configuración de la red.
- Hardware de red, conexiones físicas, hubs, switches, routers, servidores y clientes.
- Software de red, como los sistemas operativos de red, servidores de correo electrónico, software para la realización de copias de seguridad, base de datos servidores y software de aplicación.

Lo más importante, el administrador tiene cuidado de los usuarios de la red, respondiendo a sus preguntas, escuchar sus problemas, y resolver sus problemas.

Cuando las tareas de administración se realizan en una red grande y compleja, este conjunto de tareas han de abarcarse de manera dedicada, es decir, poseer una o varias personas realizando únicamente las tareas de administración de la red. Esto debe ser así debido a que las redes tiende a ser volátiles en el sentido de:

- Los usuarios de la red cambian constantemente.
- Los equipos fallan.
- Se producen conflictos entre las distintas aplicaciones.
- En general, una red compleja sufre continuos estados de crisis.

Por el contrario, las redes de menor tamaño y por ello menos complejas son generalmente mucho más estables. Suele ser habitual que una vez puesta en funcionamiento una red sencilla no tenga que sufrir continuas y complejas tareas de administración ya sean de hardware o software.

En este tipo de redes pequeñas los problemas también aparecen, pero como intervienen un reducido número de equipos es normal que sean sencillos, pocos y distantes entre sí.

Independientemente del tamaño de una red, un administrador debe cubrir las siguientes tareas que son comunes a cualquier tipo de red:

- Involucrarse y formar parte en la toma de decisiones para la adquisición de nuevo equipamiento, servidores, equipos, impresoras, etc.
- Establecer las acciones necesarias para el correcto funcionamiento cada vez que se añada un nuevo equipo, es decir, un administrador de red cuando se integra un nuevo elemento a la red se encarga de introducir cambios en la configuración del cableado, de asignar un nombre de red al nuevo equipo, integrar a un nuevo usuario en el sistema de seguridad garantizando además sus privilegios.
- Estar al corriente de las actualizaciones de software que publiquen los proveedores y considerar si sus nuevas características son suficientes para justificar una posible actualización.

En la mayoría de los casos, la parte más difícil de un proceso de actualización de software es la determinación del camino a seguir, como llevar a cabo la actualización de toda la red afectando lo menos posible al funcionamiento de los usuarios. Esto suele ser aún más crucial si el software a actualizar es el sistema operativo de red, puesto que cualquier cambio en el puede afectar a toda la red.

Dentro de este procesos de actualización también intervienen afectando en menor medida a la estabilidad del sistema los parches y Service Packs que publican los proveedores para actualizar sus soluciones y que solventan problemas menores.

- Realizar tareas rutinarias como la realización de copias de seguridad de los servidores, la administración del historial de datos o la liberación de espacio en los discos duros. Gran parte de la tareas de administración de una red consisten en asegurarse de que todo funcione correctamente, buscando y corrigiendo los problemas que puedan tener los usuarios.
- Recopilar, organizar y controlar el inventariado de toda la red, para poder solventar en el menor tiempo posible cualquier imprevisto.

#### **461.2      *Administración y gestión de redes***

El concepto de administración tiene asociado muchos significados. Desde un punto de vista informal, la gestión de redes se refiere a las actividades relacionadas con el funcionamiento de una red, junto con la tecnología necesaria para apoyar estas actividades. Otro aspecto de importancia en la gestión de una red es la monitorización de la misma, es decir, entender en todo momento que es lo que está sucediendo en la red.

Desde un enfoque software, la gestión de redes hace referencia al conjunto de actividades, métodos, procedimientos y herramientas que intervienen en las operaciones de administración, mantenimiento y aprovisionamiento de los sistemas existentes dentro de la red.

Supone además garantizar toda la oferta operativa de servicios manteniendo la red en marcha y funcionando sin problemas. Para conseguir esto se hace imprescindible la utilización de herramientas para la monitorización de la red, que ofrezcan la detección de problemas tan pronto como sea posible, incluso antes de que algún usuario se vea afectado.

La administración abarca a su vez las tareas de seguimiento de los recursos en la red y de cómo estos se asignan, haciendo uso de todos los procesos o acciones de limpieza de la red que sean necesarias para

mantener todo bajo el control del administrador o administradores.

El proceso de *mantenimiento*, que se ocupa de realizar las operaciones de reparación y mejora, ha de llevar a cabo tareas como el reemplazo de una tarjeta de red, actualización del sistema operativo de un router, añadir un nuevo switch al entramado de red. El mantenimiento también implica medidas para la corrección y prevención, como por ejemplo, el ajuste de los parámetros necesarios de un dispositivo en función de las necesidades que se soliciten o intervenir cuando sea necesario para mejorar el rendimiento de la red en momentos puntuales.

Otro aspecto de la administración de una red es el *aprovisionamiento*, tarea que concierne a la configuración y adaptación de los recursos de red para dar soporte a los servicios ofertados. Un ejemplo de aprovisionamiento es el hecho de añadir las configuraciones necesarias en los sistemas para proporcionar el servicio de voz a un nuevo usuario.

#### 461.2.1 Tareas de la gestión de red

Las tareas de gestión de una red se pueden caracterizar de la siguiente manera:

- **QoS y Gestión del Rendimiento:** un administrador de red debe supervisar y analizar periódicamente los routers, hosts y el funcionamiento de los enlaces y luego en función de los resultados obtenidos realizar una redirección del flujo de datos para evitar la sobrecarga de ciertos puntos de la red. Para realizar esta tarea de seguimiento de la red, existen herramientas que detectan rápidamente los cambios que se producen en el tráfico de una red.
- **Gestión de fallos por la red:** cualquier fallo en la red, enlaces, nodos, routers, fallos de hardware o software, debe ser detectado, localizado y respondido por la propia red, es decir, la propia red debe



poseer mecanismos para intentar solventar por sí sola el mayor número de contingencias que se puedan producir.

- **Gestión de la configuración:** esta tarea implica el seguimiento de todos los dispositivos bajo gestión y la confirmación de que todos los dispositivos están conectados y funcionan correctamente. Si se produce un cambio inesperado en las tablas de enrutamiento, el administrador ha de descubrir el problema de configuración y solucionarlo lo antes posible para que ningún servicio ni usuario se vea afectado.
- **Gestión de la seguridad:** el administrador de red es el responsable de la seguridad de la red. Para poder manejar esta tarea se utilizan principalmente los firewalls, puesto que un firewall puede monitorizar y controlar los puntos de acceso a la red informando sobre cualquier intento de intrusión.
- **Gestión de facturación y contabilidad:** el administrador especifica a los usuarios de la red los accesos o restricciones sobre los recursos y se encarga de la facturación y de los cargos a los usuarios por el uso de los mismos.

#### **461.2.2**     *Elementos de la gestión de red*

La gestión de red está compuesta por tres componentes principales:

- ***Centro de gestión:*** compuesto por el administrador de red y sus oficinas o centros de trabajo. Normalmente el centro de gestión está compuesto por un grupo humano importante.
- ***Dispositivos a gestionar:*** conformado por el equipamiento de la red, incluido su software, que es controlado mediante el centro de gestión. Cualquier hub, bridge, router, servidor, impresora o módem es considerado un dispositivo que ha de ser gestionado.

- **Protocolo de gestión de la red:** es el conjunto de políticas que adopta el centro de gestión para controlar y manejar todos los dispositivos que conforman la red. El protocolo de gestión de red permite al centro de gestión conocer el estado de los dispositivos.

#### **461.2.2.1 Estructura de Gestión de la Información (SMI, Structure of Management Information):**

Define las reglas para nombrar los objetos y para codificarlos en un centro de gestión de una red, es decir, es un lenguaje mediante el cual se definen las instancias dentro de un centro de gestión de red.

El lenguaje SMI también ofrece construcciones del lenguaje de mayor nivel que, habitualmente, especifican los tipo de datos, el estado y la semántica de los objetos que contienen la información necesaria para realizar las tareas de gestión. Por ejemplo, la cláusula STATUS especifica si la definición del objeto es actual o está obsoleta.

Trabaja bajo el protocolo SNMP (Simple Network Management Protocol) definiendo los conjuntos de objetos dentro la gestión de información base (MIB).

#### **461.2.2.2 La Gestión de la Información Base (MIB, Management Information Base)**

Es un medio de almacenamiento de información que contiene los objetos que muestran el estado actual de una red. Debido a que los objetos tienen asociado información que se almacena en el MIB, este forma colecciones de objeto, en las que incluye las relaciones entre ellos, en el centro de gestión.

Los objetos se organizan de una forma jerárquica y se identifican por la notación abstracta ASN.1, lenguaje de definición de objetos. La jerarquía, conocida como ASN.1, es un árbol de identificadores de objeto en el cual cada rama tiene un nombre y un número, permitiendo así a la gestión de red identificar objetos por una secuencia de nombres o números desde la raíz al objeto.

#### **461.2.2.3 Protocolo SNMP (Simple Network Management Protocol)**

El Simple Network Management Protocol (SNMP) está diseñado para monitorear el rendimiento de los protocolos de red y de los dispositivos. Las unidades de datos del protocolo SNMP (PDUs) pueden ser transportadas en un datagrama UDP, por lo que su entrega en destino no está garantizada. Los dispositivos que se administran como los routers o hosts, son objetos y cada uno tiene una definición formal y MIB adapta una base de datos de información que describe sus características. Con este protocolo un gestor de red puede encontrar donde se localizan los problemas.

Se ejecuta sobre UDP y utiliza una configuración cliente-servidor. Sus comandos definen como realizar las consultas sobre la información de un servidor o como enviar esta hacia un cliente o hacia otro servidor.

La tarea principal del protocolo SNMP es la de transportar información entre los centro de gestión y los agentes que se ejecutan en representación de los centros de gestión. Para cada objeto MIB que se gestiona se utiliza una petición SNMP para obtener su valor o para modificarla. Si un agente recibe un mensaje no solicitado o si una interfaz o dispositivo deja de funcionar, entonces el protocolo puede informar al centro de gestión del fallo que se está produciendo.

La segunda versión de este protocolo, SNMPv2, corre por encima de varios protocolos y tiene más opciones de mensajería, lo que resulta en una gestión más eficaz de la red. Tiene siete unidades de PDU, o mensajes:

1.       **GetRequest.** Se utiliza para obtener un valor de objeto MIB.
2.       **GetNextRequest.** Se utiliza para obtener el siguiente valor de un objeto MIB.
3.       **GetBulkRequest.** Recibe múltiples valores, lo que equivale a GetRequests múltiples, pero sin necesidad de utilizar múltiples peticiones.
4.       **InformRequest.** Es un mensaje de director a director de comunicación que se envían entre sí dos centros de gestión a distancia el uno del otro.
5.       **SetRequest.** Es utilizado por un centro de gestión para inicializar el valor de un objeto MIB.
6.       **Response.** Es un mensaje de respuesta a una petición de tipo PDU.
7.       **Trap.** Notifica a un centro de gestión de un evento inesperado.

Hay dos tipos de representación de PDUs, Get o Set y Trap.

- El formato de PDU de Get o Set es el siguiente:
  - o       *PDU type*, indica uno de los siete tipos de PDU.
  - o       *Request ID*, es un ID que se utiliza para verificar la respuesta de una solicitud. Por lo tanto un centro de gestión puede detectar peticiones perdidas o duplicadas.
  - o       *Error status*, sólo es usado por PDUs *Response* para indicar tipos de errores reportados por un agente.

- o *Error index*, es un parámetro que indica a un administrador el nombre del objeto que ha causado el error.

Si las solicitudes o respuestas se pierden, el protocolo no realiza un reenvío. Los campos Error status and Error index son todo ceros excepto para las PDUs *GetBulkRequest*

- El formato de PDU de Trap es:
  - o *Enterprise*, para usar en múltiples redes.
  - o *Timestamp*, para realizar las mediciones de tiempo.
  - o *Agentadress*, para indicar que la dirección del agente gestor está incluida en la cabecera PDU.

## **462 VIRTUALIZACIÓN DE SERVIDORES**

Podemos definir virtualización como la técnica que consiste básicamente en agrupar diferentes aplicaciones y servicios de sistemas heterogéneos dentro de un mismo hardware, de forma que los usuarios y el propio sistema los vean como máquinas independientes dedicadas. Para ello, el sistema operativo virtualizado debe ver el hardware de la máquina real como un conjunto normalizado de recursos independientemente de los componentes reales que lo formen.

De esta forma, para virtualizar un sistema de servidores, los administradores deben, básicamente, optimizar los recursos disponibles, incluyendo el número y la identidad de los servidores físicos individuales, procesadores, y sistemas operativos, con el objetivo de producir una mejora tanto en la gestión como en el manejo de sistemas informáticos

complejos. El administrador del sistema virtual utilizará un software para la división del servidor físico en entornos virtuales aislados. Estos entornos son lo que se conoce técnicamente como servidores privados virtuales, pero también se pueden encontrar referencias como particiones, instancias, contenedores o emulaciones de sistemas.

En concreto, podemos decir que un servidor privado virtual es un término de marketing utilizado por los servicios de hosting para referirse a una máquina virtual para el uso exclusivo de un cliente individual del servicio. El término se utiliza para enfatizar que la máquina virtual, a pesar de ejecutarse en el mismo equipo físico que las máquinas virtuales de otros clientes, es funcionalmente equivalente a un equipo físico independiente, está dedicado a las necesidades individuales del cliente y puede ser configurado para ejecutarse como un servidor de internet (es decir, para ejecutar software de servidor). El término VDS o Virtual Dedicated Server (Servidor Virtual Dedicado) para el mismo concepto.

Cada servidor virtual puede ejecutar su propio sistema operativo y ser reiniciado de modo independiente.

#### **462.1      *Funcionamiento***

El servidor físico realiza una abstracción de los recursos que se denomina Hypervisor o VMM (Virtual Machine Monitor), elemento software que se instala en la máquina donde se va a llevar a cabo la virtualización y sobre la que se configuran las máquinas virtuales que es donde van a residir las aplicaciones. Es el encargado de gestionar los recursos de los sistemas operativos “alojados” (guest) o máquinas virtuales.

Desde un punto de vista lógico, el usuario percibe que son máquinas independientes y aisladas entre sí, pero desde una perspectiva física, todas las máquinas virtuales residen en un único servidor. A estas máquinas virtuales se les asigna un porcentaje de los recursos del servidor físico, que serán los únicos que el cliente conozca.

Se pueden encontrar tres modelos de virtualización: el modelo de máquina virtual o virtualización completa, el modelo paravirtual o virtualización parcial; y la virtualización a nivel de sistema operativo.

#### *462.1.1 Virtualización completa*

El modelo de máquina virtual está basado en la arquitectura cliente/servidor, donde cada cliente funciona como una imagen virtual de la capa hardware. Este modelo permite que el sistema operativo cliente funcione sin modificaciones. Además permite al administrador crear diferentes sistemas cliente con sistemas operativos independientes entre sí. La ventaja principal de este modelo radica en el desconocimiento por parte de los sistemas huésped del sistema hardware real sobre el que está instalado. Sin embargo, realmente todos los sistemas virtuales hacen uso de recursos hardware físicos. Estos recursos son administrados por un el hypervisor que coordina las instrucciones CPU, convirtiendo las peticiones del sistema invitado en las solicitudes de recursos apropiados en el host, lo que implica una sobrecarga considerable. Casi todos los sistemas pueden ser virtualizados utilizando este método, ya que no requiere ninguna modificación del sistema operativo. A pesar de esto, es necesaria una virtualización de la CPU como apoyo para la mayoría de los hypervisores que llevan a cabo la virtualización completa.

Ejemplos típicos de sistemas de servidores virtuales son VMware Workstation, VMware Server, VirtualBox, Parallels Desktop, Virtual Iron, Adeos, Mac-on-Linux, Win4BSD, Win4Lin Pro, y z/VM, openvz, Oracle VM, XenServer, Microsoft Virtual, PC 2007 y Hyper-V.

#### *462.1.2 Paravirtualización*

El modelo de máquina paravirtual (PVM) o virtualización parcial se basa, como el modelo anterior, en la arquitectura cliente/servidor, incluyendo también la necesidad de contar con un sistema monitor. Sin embargo, en este caso, el VMM accede y modifica el código del sistema operativo del

sistema huésped. Esta modificación se conoce como porting. El porting sirve de soporte al VMM para que pueda realizar llamadas al sistema directamente. Al igual que las máquinas virtuales, los sistemas paravirtuales son capaces de soportar diferentes sistemas operativos instalados en el hardware real. Esta técnica se utiliza con intención de reducir la porción de tiempo de ejecución empleada por el huésped empleado en realizar las operaciones que son mucho más difíciles de ejecutar en un entorno virtual en comparación con un entorno no virtualizado. Así se permite que el invitado(s) y el huésped soliciten y reconozcan estas tareas, que de otro modo serían ejecutados en el dominio virtual (donde el rendimiento de ejecución es peor). Una plataforma paravirtualizada exitosamente puede permitir que el VMM sea menos complejo (por la reubicación de la ejecución de las tareas críticas del dominio virtual en el dominio del servidor), y/o reducir la degradación del rendimiento global de la máquina virtual durante la ejecución de invitado.

UML, XEN, Xen, Virtuozzo , Vserver y OpenVZ (que es el código abierto y la versión de desarrollo de Parallels Virtuozzo Containers) son modelos de máquinas paravirtuales.

#### **462.1.3     Virtualización por S.O.**

La virtualización a nivel de sistema operativo se diferencia de las anteriores en que, en este caso, no existe un sistema cliente/servidor propiamente dicho. En este modelo el sistema principal exporta la funcionalidad del sistema operativo desde su propio núcleo. Por esta razón, los sistemas virtuales usan el mismo sistema operativo que el nativo (aunque en la mayoría de los casos pueden instalar distintas distribuciones). Esta arquitectura elimina las llamadas del sistema entre capas, lo que favorece una reducción importante en el uso de CPU. Además, al compartir los ficheros binarios y librerías comunes del sistema en la misma máquina, la posibilidad de escalado es mucho mayor, permitiendo que un mismo servidor virtual sea capaz de dar servicio a un gran número de clientes al



mismo tiempo.

La Virtualización de SO mejora el rendimiento, gestión y eficiencia. Podemos entenderlo como un sistema en capas. En la base reside un sistema operativo huésped estándar. A continuación encontramos la capa de virtualización, con un sistema de archivos propietario y una capa de abstracción de servicio de kernel que garantiza el aislamiento y seguridad de los recursos entre distintos contenedores. La capa de virtualización hace que cada uno de los contenedores aparezca como servidor autónomo. Finalmente, el contenedor aloja la aplicación o carga de trabajo.

Ejemplos de sistemas que usan virtualización a nivel de sistema operativo son Virtuozzo y Solaris.

### **463 VIRTUALIZACIÓN DEL PUESTO CLIENTE**

Esta técnica consiste en la separación del entorno de usuario de un ordenador personal de la máquina física con el modelo cliente-servidor. El modelo que sigue un servidor para implementar dicha característica se denomina VDI (Virtual Desktop Infrastructure, Infraestructura de Escritorio Virtual), también llamada Interfaz de Escritorio Virtual.

La mayoría de implementaciones comerciales de esta tecnología usan un servidor central remoto para llevar a cabo la “virtualización” del escritorio del cliente, en lugar de usar el almacenamiento local del cliente remoto. Esto implica que todas las aplicaciones, procesos, configuraciones y datos del cliente están almacenadas en el servidor y se ejecutan de forma centralizada.

El sistema cliente puede utilizar una arquitectura de hardware completamente diferente de la utilizada por el entorno de escritorio proyectado, y también puede estar basada en un sistema operativo

completamente diferente.

El modelo de virtualización del puesto cliente permite el uso de máquinas virtuales para que múltiples suscriptores de red puedan mantener escritorios individuales en un solo ordenador, el servidor central. Este servidor central puede operar en una residencia, negocio o centro de datos. Los usuarios pueden estar geográficamente dispersos, pero todos están conectados a la máquina central por una red de área local, una red de área amplia, o Internet.

#### **463.1      *Modos de Operación de VDI***

Básicamente existen cuatro modelos de operación VDI:

- Alojado (como servicio). Suelen contratarse a proveedores comerciales y normalmente proporciona una configuración del sistema operativo del puesto cliente administrado. Los principales suministradores son CITRIX, VMware y Microsoft.
- Centralizado. En este caso todas las instancias VDI están alojadas en uno o más servidores centralizados, los datos están en sistemas de almacenamiento conectados a estos. Este modelo a su vez puede distinguir dos tipos:
  - o VDI estático o persistente. Existe una única imagen de escritorio asignado por cliente y estos deben ser gestionados y mantenidos.
  - o VDI dinámico o no persistente. Existe una imagen maestra común para todos los clientes que se clona y personaliza en el momento de la petición con los datos y aplicaciones particulares de cada cliente.
- Remoto (o sin ataduras). Tiene como base el concepto de VDI centralizado pero permite trabajar sin la conexión a un servidor central o a Internet. Se copia una imagen al sistema local y se ejecuta sin

necesidad de más conexión. Las imágenes tienen un cierto periodo de vida y se actualizan periódicamente. Esta imagen se ejecuta en el sistema local que necesita un sistema operativo y un hipervisor (que ejecuta la instancia VDI). Esto implica que el dispositivo cliente tenga mayores necesidades de memoria, espacio en disco, CPU... La ventaja es la menor dependencia de conexión.

Los modelos alojado y centralizado necesitan de una red que conecte con el servidor donde se ejecuta la instancia VDI. El concepto base de este modelo es similar al de clientes ligeros debido a que el cliente sólo tiene que mostrar el escritorio virtual.

En el caso del modelo remoto, se permite a los usuarios copiar la instancia VDI en el sistema y luego se ejecutará el escritorio virtual sin necesidad de ningún tipo de conexión.

#### **464 COMPUTACIÓN BASADA EN SERVIDOR**

También conocida como SBC del inglés Server Based Computing, consiste en la separación del procesamiento de ciertas tareas como la gestión de datos que será realizado en un servidor central y otras tareas de procesamiento, como la presentación de aplicaciones de usuario e impresión de datos en el cliente. Lo único transmitido entre servidor y cliente son las pantallas de información. Esta arquitectura puede dar solución a los principales problemas que aparecen cuando se ejecutan aplicaciones en los clientes. Además simplifica procesos como pueden ser los entornos hardware, actualizaciones de software, despliegue de aplicaciones, soporte técnico, almacenamiento y respaldo de datos. Se centraliza la gestión de todos estos procesos en un único servidor.

Los clientes que actúan en esta arquitectura suelen llamarse thin clients, o clientes ligeros, este es un término general para dispositivos que se basan en un servidor para operar. El thin client proporciona pantalla, teclado, ratón y un procesador básico que interactúa con el servidor. Los thin client

no almacenan ningún dato localmente y requiere de pocos recursos de procesamiento. La característica más destacada de estos terminales es la reducción de costes asociados con el mantenimiento, administración, soporte, seguridad e instalación de aplicaciones comparándolo con un PC tradicional.

Esta tecnología está compuesta por tres componentes principales:

- Sistemas operativos multi-usuario que permiten el acceso y ejecución de modo concurrente, usando aplicaciones diferentes y con sesiones de usuario protegidas. Ejemplos de algunas terminales de servicio son: 2x Terminal Server para Linux, Microsoft Windows Terminal Server (Windows NT/2000), Microsoft Windows Terminal Services (Windows 2003), Citrix Presentation Server, Citrix XenApp Server, AppliDis Fusion, 2X Application Server, HOblink, Propalms TSE (antes Tarantella), Jethro cabina, GraphOn GO-Global, VMware View..
- El thin client se puede ejecutar con una cantidad mínima de software pero necesita al menos un programa de conexión a servicios de terminal. El thin client y el programa de servicios de terminal pueden ser ejecutados en sistemas operativos completamente diferentes.
- Un protocolo que permita al programa de servicios de terminal y al thin client comunicarse y enviar las pulsaciones de teclado, de ratón y las actualizaciones de pantalla a través de la red. Los protocolos más populares son RDP3 (Remote Desktop protocol), ICA y NX.

Entre las ventajas de la computación basada en servidor se puede nombrar:

- Reducción de los costes de administración. La gestión de clientes ligeros está casi en su totalidad centralizada en el servidor.
- Reducción de costes de hardware. El hardware en los clientes ligeros es generalmente más barato porque no es necesario tener memoria

para las aplicaciones o un procesador de gran alcance.

- Seguridad. Puede ser controlada centralmente.
- Menor consumo de energía. El hardware especializado en el cliente ligero tiene un consumo mucho menor de energía que los tradicionales.
- Reducción de la carga de red. El tráfico de red que generan los terminales ligeros sólo es el de los movimientos del ratón, teclado e información de pantalla desde / hacia el usuario. En el caso de que un cliente pesado abriese y guardase un documento ya implicaría el paso de este dos veces por la red. Usando protocolos eficientes de red tales como ICA y NX ya es posible usar esta tecnología en un ancho de banda de 28,8 Kbps.
- Actualización de hardware simple. Si el uso está por encima de un límite predefinido, es relativamente sencillo solucionar el problema, bastaría con un disco nuevo en un rack de servidores, aumentando así el número de recursos, exactamente la cantidad necesaria. Si ocurriese esto con clientes pesados habría que reemplazar un PC completo, lo que acarrearía tanto costes económicos como de recursos humanos.

A pesar de lo anterior, esta tecnología también presenta ciertos inconvenientes:

- Altos requerimientos de servidor. Al centrarse la carga de trabajo en el servidor, el sistema de clientes ligeros implica mayor consumo de recursos en los servidores, incluso es habitual que se use un gran número de servidores, lo que se denomina “granja de servidores”.
- Pobre rendimiento multimedia. El envío de datos de audio y video requieren mucho ancho de banda, por lo que estos sistemas son menos útiles para aplicaciones multimedia.
- Menos flexibilidad. No todos los productos software del mercado

pueden funcionar correctamente en un cliente ligero.

#### **465 GRID COMPUTING**

Arquitectura distribuida y paralela, de ámbito extenso geográficamente, en la que se premia la distribución, y a continuación la paralelización. Sus creadores fueron Ian Foster y Carl Kesselman. Su nombre proviene del paradigma de la red eléctrica (power grid).

Se basa en la compartición, selección y agregación de forma dinámica y en tiempo de ejecución de recursos autónomos, distribuidos geográficamente, dependiendo de criterios como la disponibilidad del hardware, la capacidad transaccional, el rendimiento que se pueda aportar a la solución final, el coste y los criterios de calidad del servicio que el demandante pueda proporcionar y exigir.

La red está formada por un conjunto de ordenadores independientes e interconectados que ponen a disposición del grid los excedentes de su procesamiento individual, es decir, los ciclos de reloj de sus CPUs no aprovechados por ellos mismos, sin poder superar un determinado porcentaje de dedicación configurado individualmente en cada nodo. A partir del porcentaje proporcionado por cada nodo, se virtualiza un recurso computacional único.

Los sistemas basados en grid computing están indicados para atender productividades sostenidas y sostenibles, sin poder nunca superar un determinado umbral. En estos sistemas se garantiza la escalabilidad como un criterio parametrizable. Es posible definir con qué criterio añadimos cada nuevo nodo a la solución final.

Actualmente, el único criterio que se tiene en cuenta es la capacidad de procesamiento (transaccionalidad), pero en el futuro, será posible tener en cuenta criterios más finos, referidos a la calidad del servicio.

Además, estos sistemas están dotados de un comportamiento dinámico,

según el cual, un determinado programa en ejecución en el sistema, puede modificar en tiempo real el dimensionamiento de la grid para adaptarlo a sus necesidades.

#### **465.1      *Características:***

- Podemos conseguir un máximo aprovechamiento de los nodos (100% de utilización de la CPU).
- Los nodos no tienen que estar dedicados. Además, al contrario que en el caso del cluster, nos aseguramos que la aportación al Grid no va a sobrepasar un determinado porcentaje de tiempo de procesamiento en cada nodo.
- Son sistemas heterogéneos, en los que podemos encontrar diversos HW y SW.
- La escalabilidad parametrizable es la característica más potente de esta arquitectura.

#### **465.2      *Funcionalidades:***

- Localización dinámica de recursos (máquinas con excedente).
- Optimización del acceso a datos, mapeando las estructuras de datos en cachés temporales locales (directorios).
- Autenticación del usuario (usr/pwd, certificados...).
- Monitorización de tareas y procesos desde cualquier nodo de la red, siempre que el usuario tenga permisos.
- Las máquinas se encuentran en situación paritaria.
- Si es posible, se paraleliza. Lo fundamental es la distribución de procesos débilmente acoplados.

### **465.3      *Arquitectura Grid***

Habitualmente se describe la arquitectura del Grid en términos de "capas", ejecutando cada una de ellas una determinada función. Como es habitual en este tipo de enfoque, las capas más altas están más cerca del usuario, en tanto que las capas inferiores lo están de las redes de comunicación.

Empezando por los cimientos, nos encontramos con la capa de red, responsable de asegurar la conexión entre los recursos que forman el Grid.

En la parte más alta está la capa de recursos, constituida por los dispositivos que forman parte del Grid: ordenadores, sistemas de almacenamiento, catálogos electrónicos de datos e incluso sensores que se conecten directamente a la red.

En la zona intermedia está la capa "middleware", encargada de proporcionar las herramientas que permiten que los distintos elementos (servidores, almacenes de datos, redes, etc.) participen de forma coordinada en un entorno Grid unificado. Esta capa es la encargada de las siguientes funciones:

Encontrar el lugar conveniente para ejecutar la tarea solicitada por el usuario.

- Optimiza el uso de recursos, que pueden estar muy dispersos.
- Organiza el acceso eficiente a los datos.
- Se encarga de la autenticación de los diferentes elementos.
- Se ocupa de las políticas de asignación de recursos.
- Ejecuta las tareas.
- Monitoriza el progreso de los trabajos en ejecución.
- Gestiona la recuperación frente a fallos.



- Avisa cuando se haya terminado la tarea y devuelve los resultados.

El ingrediente fundamental del middleware son los metadatos (datos sobre los datos), que contienen, entre otras cosas, toda la información sobre el formato de los datos y dónde se almacenan (a veces en varios sitios distintos).

El middleware está formado por muchos programas software. Algunos de esos programas actúan como agentes y otros como intermediarios, negociando entre sí, de forma automática, en representación de los usuarios del Grid y de los proveedores de recursos. Los agentes individuales presentan los metadatos referidos a los usuarios, datos y recursos. Los intermediarios se encargan de las negociaciones entre máquinas (M2M) para la autenticación y autorización de los usuarios y se encargan de definir los acuerdos de acceso a los datos y recursos y, en su caso, el pago por los mismos. Cuando queda establecido el acuerdo, un intermediario planifica las tareas de cómputo y supervisa las transferencias de datos necesarias para acometer cada trabajo concreto. Al mismo tiempo, una serie de agentes supervisores especiales optimizan las rutas a través de la red y monitorizan la calidad del servicio.

En la capa superior de este esquema está la capa de aplicación donde se incluyen todas las aplicaciones de los usuarios, portales y herramientas de desarrollo que soportan esas aplicaciones. Esta es la capa que ve el usuario.

Además, en las arquitecturas más comunes del Grid, la capa de aplicación proporciona el llamado "serviceware", que recoge las funciones generales de gestión tales como la contabilidad del uso del Grid que hace cada usuario.

Para poder hacer todo lo anterior, las aplicaciones que se desarrollen para ser ejecutadas en un PC concreto, tendrán que adaptarse para poder

invocar los servicios adecuados y utilizar los protocolos correctos. Igual que las aplicaciones que inicialmente se crearon para funcionar aisladamente se adaptan para poder ser ejecutadas en un navegador Web, el Grid requerirá que los usuarios dediquen cierto esfuerzo a "GRIDizar" sus aplicaciones.

Sin embargo, una vez adaptadas al Grid, miles de usuarios podrán usar las mismas aplicaciones, utilizando las capas de middleware para adaptarse a los posibles cambios en el tejido del Grid.

## **466 CLOUD COMPUTING**

Modelo que permite acceso a un conjunto compartido de recursos informáticos configurables a través de la red (por ejemplo, redes, servidores, almacenamiento, aplicaciones y servicios) que pueden ser desarrollados y desplegados rápidamente con mínimo esfuerzo de gestión o interacción con el proveedor de servicios.

Este término se refiere a la utilización y el acceso de múltiples recursos basados en servidores a través de una red. Los usuarios de la "nube" pueden acceder a los recursos del servidor utilizando un ordenador, netbook, pad computer, smart phone u otro dispositivo. En el cloud computing, el servidor presenta y gestiona las aplicaciones; los datos también se almacenan de forma remota en la configuración de la nube. Los usuarios no descargan ni instalan aplicaciones en su sistema, todo el procesamiento y almacenamiento se mantiene por el servidor. Los servicios on-line pueden ser ofrecidos a partir de un "proveedor de la nube" o por una organización privada.

### **466.1 Arquitectura**

Normalmente la arquitectura de los sistemas software implicados en el desarrollo de cloud computing incluyen múltiples componentes denominados "componentes cloud" que se comunican mediante

mecanismos de bajo acoplamiento, tales como las colas de mensajes.

Los dos componentes más significativos de la arquitectura cloud computing se conocen como el front-end y el back-end. El front-end es la parte vista por el cliente, es decir, el usuario del PC. Esto incluye la red del cliente y las aplicaciones utilizadas para acceder a la nube a través de una interfaz de usuario, como un navegador web. El back-end de la arquitectura es la propia nube, que comprende varios ordenadores, servidores y dispositivos de almacenamiento de datos.

Dentro de esta arquitectura se pueden distinguir las siguientes capas:

- Proveedor: Empresa responsable de proporcionar el servicio en la “nube”.
- Cliente: Serán el hardware y software diseñados para cloud computing, que permiten interactuar con los servicios remotos.
- Aplicación: Son los servicios en la “nube” o “Software as a Service” (SaaS), el software se proporciona a través de internet como si de un servicio se tratase. De este modo se evita la necesidad de instalar y ejecutar en el equipo del cliente la aplicación. Se reducen así el mantenimiento y el apoyo.
- Plataforma: Son los servicios de plataforma en la “nube”, también conocidos como “Platform as Service” (PaaS), proporcionan una plataforma de procesamiento y una pila de soluciones como un servicio, constituyen la base e infraestructura de las aplicaciones de la nube. Facilita el desarrollo de aplicaciones evitando el coste y la complejidad de comprar y mantener el hardware y las capas de software de base.
- Infraestructura. Servicios de infraestructura, también conocidos como “Infrastructure as a Service” (IaaS), proporciona la infraestructura como un servicio, suele ser una plataforma virtualizada. En lugar de comprar servidores, software, centro de datos especiales o equipos de red, los

clientes adquieren dichos recursos de servicios externos. La IaaS ha evolucionado a partir de las ofertas de servidores virtuales privados.

#### **466.2 Modelos de implementación**

- Nube pública o external cloud: Es el concepto tradicional donde los recursos se presentan a través de internet en función de la demanda, a través de aplicaciones o servicios web.
- Nube de la comunidad: Se da cuando varias organizaciones con las mismas necesidades comparten recursos. En este caso existen menos usuarios que en la nube pública y se ofrece mayor privacidad y seguridad. Un ejemplo puede ser el Google`s “Gov Cloud”.
- Nube híbrida. Es común que una empresa use tanto la nube pública como desarrollos privados para satisfacer sus necesidades con respecto a las TI. Existen varias empresas como HP, IBM, Oracle and VMware que ofrecen tecnologías para manejar la complejidad de mantenimiento, seguridad y privacidad consecuencia del uso del conjunto de estos servicios.
- Nube combinada. Se denomina al conjunto formado varios servicios cloud de distintos proveedores.
- Nube privada. Es trasladar el concepto de nube pública a una red de uso privado. Es decir, el uso de la nube única y exclusivamente dentro de la red de una empresa.

#### **467 GREEN IT E EFICIENCIA ENERGÉTICA**

El término Green Computing se acuñó posiblemente por primera vez tras el inicio del programa Energy Star en 1992, promocionado por el gobierno estadounidense.

Tenía por objetivo etiquetar monitores y equipamiento electrónico

caracterizados por su eficiencia energética. El término quedó registrado ya en 1992 en un grupo de noticias. Hoy en día el programa Energy Star es el motor de la eficiencia energética en los sistemas electrónicos (no sólo de procesamiento de la información, sino también del equipamiento electrónico doméstico).

La adopción de productos y aproximaciones más eficientes pueden permitir más equipamiento dentro del mismo gasto energético, lo que se denomina huella energética, o energy footprint. Las regulaciones se están multiplicando y podrían limitar seriamente a las empresas a la hora de construir centros de procesamiento de datos, ya que el efecto de las redes de suministro eléctrico, las emisiones de carbono por el incremento de uso y otros impactos medioambientales están siendo investigadas. Por tanto, las organizaciones deben considerar las regulaciones y tener planes alternativos para el crecimiento de sus centros de procesamiento de datos y de su capacidad.

Con el paso de los años, el número de servidores existentes en todo el mundo crece de forma casi exponencial. Consecuencia de esto es el creciente gasto energético para la refrigeración y gestión de los equipos. Hoy en día ya se están empezando a plantear soluciones que optimicen este gasto energético.

Este consumo energético no es el único problema ambiental relacionado con las TI. La etapa de fabricación de equipos presenta serios problemas relacionados con el medio ambiente: materiales de desecho tóxicos, producción de gases contaminantes, etc. La tendencia actual es la de minimizar el impacto contaminante (carbon footprint) presente en las tecnologías de fabricación de los sistemas electrónicos.

Finalmente, también tiene un impacto inmediato la eliminación de equipos para las TI, caracterizados por un tiempo de vida increíblemente breve de unos dos o tres años. Si no se reciclan de forma eficiente, terminan tirados en vertederos, y debido a la presencia de componentes tóxicos, son una

fuelle de contaminación terrestre y de las aguas. Todos estos aspectos deben ser considerados de manera global por los fabricantes y usuarios de equipos TI. La concienciación de la existencia de este problema ha llevado a la elaboración de numerosas y rígidas normativas a todos los niveles, lo que empieza a obtener algunos resultados.

GreenPeace Internacional realiza un ranking con los 18 principales fabricantes del sector electrónico (ordenadores personales, teléfonos móviles etc.) de acuerdo con sus políticas de reducción de emisiones tóxicas, reciclado o minimización de impacto en el cambio climático, y lo publica en su Guía para la Electrónica Verde (Guide to Greener Electronics), de publicación trimestral. Como se puede ver en los resultados de Diciembre de 2010, las empresas del sector obtienen unas calificaciones realmente bajas, siendo la mejor Nokia con un 7,5 sobre 10.

La mitad de estas 18 empresas suspenden un estudio que busca que las empresas analizadas:

- Limpíen sus productos al eliminar sustancias peligrosas. Los productos químicos peligrosos con riesgo impiden el posterior reciclado de los equipos.
- Reciclen de equipos/productos bajo su responsabilidad una vez quedan obsoletos.
- Reduzcan el impacto climático debido a sus operaciones y productos.

Por todo lo expuesto, la resolución efectiva del impacto ambiental de las tecnologías TI requiere un enfoque holístico del problema que englobe las cuatro vías:

- Utilización ecológica: principalmente a través de la reducción del consumo energético. La producción de energía eléctrica es la principal fuente de generación de gases de efecto invernadero.

- Diseño ecológico o eco-diseño: incluye diseño de equipos más eficientes energéticamente y respetuosos con el medio ambiente.
- Fabricación ecológica: eliminando completamente o minimizando el impacto del proceso de fabricación en el medio ambiente (emisiones, materiales de desecho, etc.).
- Eliminación ecológica: una vez finalizado el período de utilización de un equipo se deben poner en marcha las estrategias denominadas tres R: reutilización y renovación de equipos y, si no son aprovechables, reciclado.

La idea principal del enfoque holístico es que se cierre el ciclo de vida de los equipos TI de forma que no se perjudique el medio ambiente, lo que permitiría conseguir una mejora sustancial de cara al desarrollo sostenible.

#### **467.1      *Tecnologías verdes***

Hoy en día existen distintos enfoques tecnológicos que se acercan a un desarrollo sostenible de las TI.

- Monitores LCD. Con el paso de los años los monitores pasaron de ser CRT a LCD, este cambio no es sólo estético o de tamaño, sino que los niveles de consumo han disminuido notablemente. Un monitor CRT medio requiere 85W si está activo, frente a los 15W de uno LCD, 5W en modo bajo consumo para un CRT mientras que un LCD consumiría 1,5W. Apagados ambos consumirían 0,5W. En los últimos años se ha revolucionado el mercado de las pantallas de ordenador con la aparición de la tecnología OLED (Organic Light Emitting Diode), basadas en la utilización de diodos LED cuya capa electro-luminiscente se hace con un compuesto orgánico (un polímero que se ilumina al aplicarle un voltaje). La ventaja principal de este tipo de pantallas frente a las tradicionales de cristal líquido (LCD) es que los diodos OLED no necesitan retro-iluminación, por lo que el consumo de energía que requieren es muy

inferior.

- Discos duros. El consumo de los discos duros no es para nada despreciable, sobre todo en el arranque del sistema. Por ejemplo, el disco Seagate Barracuda 7200.8 requiere hasta 2,5 A de la línea de alimentación de 12 V. Si a esto le sumamos 3W que extrae desde la línea de +5 V se puede llegar a un consumo de pico en el arranque de 33 W. Si en lugar de sólo un disco duro hablamos de un equipo con dos o más empezamos a hablar de cifras muy comprometidas. Esto ha hecho que los fabricantes de discos duros comiencen a tener en cuenta el consumo en sus productos, creando casi todos una nueva gama denominada “verde” o “ecológica”. Por ejemplo Así, Western Digital con “Caviar Green”, Samsung con Eco Green, o Hitachi con eco-friendly Deskstar y Travelstar. Como alternativa a los discos tradicionales aparecen los discos en estado sólido (SSD), que presentan menores consumos de energía y es la tecnología a la que se espera evolucionen los sistemas de almacenamiento.
- CPDs. Aquí es donde se aloja toda la infraestructura de soporte a los diversos servicios computacionales, y una estructura adecuada permitirá buenos ahorros de energía, de espacio y de costos a mediano y/o largo plazo. Buscando la reducción de energía se puede empezar por la acción más simple que es apagar el equipo que no se esté utilizando, la reducción del hardware estudiando necesidades reales, o actuaciones específicas en función de la actividad de la empresa.
- Virtualización. La virtualización de servidores permite el funcionamiento de múltiples servidores en un único servidor físico. Esto ayuda a reducir la huella de carbono del centro de datos al disminuir el número de servidores físicos y consolidar múltiples aplicaciones en un único servidor con lo cual se consume menos energía y se requiere menos enfriamiento. Además se logra un mayor índice de utilización de recursos y ahorro de espacio.



- Cliente/Servidor. Estos sistemas mantienen el software, las aplicaciones y los datos en el servidor. Se puede tener acceso a la información desde cualquier ubicación y el cliente no requiere mucha memoria o almacenamiento. Este ambiente consume menos energía y enfriamiento.
- Cloud computing. Esto proporciona a sus usuarios la posibilidad de utilizar una amplia gama de recursos en red para completar su trabajo. Al utilizar computación en nube las empresas se vuelven más ecológicas porque disminuyen su consumo de energía al incrementar su capacidad sin necesidad de invertir en más infraestructura.
- Tele trabajo. Definido por Merrian-Webster como el trabajo en casa con el uso de un enlace electrónico con la oficina central. Al no desplazarse el empleado la contaminación es menor.

#### **467.2      *Actividades relacionadas con Green IT***

Existen varias actividades que promocionan e intentan solventar las cuestiones expuestas anteriormente. Estas actividades están patrocinadas bien desde administraciones públicas, bien desde empresas, que están entendiendo que Green IT, además de una necesidad, puede ser un negocio, desde el punto de vista de consultoría y servicios, o bien por consorcios de empresas.

The Green Grid (<http://www.thegreengrid.org>) es un consorcio global dedicado a avanzar en la eficiencia energética de los centros de procesamiento de datos y en ecosistemas de computación de negocio. En cumplimiento de su misión, The Green Grid se centra en:

- Definir métricas y modelos significativos y centrados en el usuario.
- Desarrollar estándares, métodos de medida, procesos y nuevas tecnologías para mejorar el rendimiento de los centros de procesamiento de datos frente a las métricas definidas.

- Promocionar la adopción de estándares, procesos, medidas y tecnologías energéticamente eficientes.

El comité de directores de The Green Grid está compuesto por las siguientes compañías miembros: AMD, APC, Dell, HP, IBM, Intel, Microsoft, Rackable Systems, Sun Microsystems y VMware.

Climate Savers. Iniciada por Google e Intel en 2007, Climate Savers Computing Initiative ([www.climatesaverscomputing.org](http://www.climatesaverscomputing.org)) es un grupo sin ánimo de lucro de consumidores y negocios con conciencia ecológica y organizaciones conservacionistas. La iniciativa se inició bajo el espíritu del programa Climate Savers de WWF (<http://www.worldwildlife.org/climate/projects/climateSavers.cfm>), que ha movilizado a una docena de compañías desde 1999 a recortar las emisiones de dióxido de carbono, demostrando que reducir las emisiones es bueno para el negocio. Su objetivo es promover el desarrollo, despliegue y adopción de tecnologías inteligentes que puedan mejorar la eficiencia de uso de la energía del computador y reducir su consumo cuando el computador se encuentra inactivo.

SNIA (Storage Networking Industry Association, <http://www.snia.org>) es una organización global sin ánimo de lucro compuesta por unas compañías de la industria del almacenamiento. SNIA Green Storage Initiative (<http://www.snia.org/green>) está llevando a cabo una iniciativa para avanzar en el desarrollo de soluciones energéticamente eficientes para el almacenamiento en red, incluyendo la promoción de métricas estándares, la formación y el desarrollo de buenas prácticas energéticas o el establecimiento de alianzas con organizaciones como The Green Grid.

Energy Star. En 1992 la Agencia de Protección Medioambiental de EEUU (U.S. Environmental Protection Agency) lanzó el programa Energy Star, que se planificó para promocionar y reconocer eficiencia energética en monitores, equipos de climatización y otras tecnologías. Aunque de carácter voluntario inicialmente, resultó pronto de amplia aceptación,

pasando a ser un hecho la presencia de un modo de descanso (sleep mode) en la electrónica de consumo.

Directiva Europea de Eco-Diseño. Siguiendo la misma línea que la iniciativa Energy Star de EEUU, la Unión Europea aprobó la directiva 2005/32/EC para el eco-diseño, nuevo concepto creado para reducir el consumo de energía de productos que la requieren, tales como los dispositivos eléctricos y electrónicos o electrodomésticos. La información relacionada con las prestaciones medioambientales de un producto debe ser visible de forma que el consumidor pueda comparar antes de comprar, lo cual está regulado por la Directiva de Etiquetado de la Energía (Energy Labelling Directive). Los productos a los que se conceda la Eco-etiqueta serán considerados como cumplidores con la implementación de las medidas, de forma muy similar a la etiqueta de Energy Star.

El Código de Conducta de la Unión Europea para Centros de Datos está siendo creado como respuesta al creciente consumo de energía en centros de datos y a la necesidad de reducir el impacto ambiental, económico y de seguridad de abastecimiento energético relacionado. El objetivo es informar y estimular a los operadores o propietarios de los centros de datos a que reduzcan el consumo de energía de una forma rentable sin dificultar su funcionamiento. Este código de conducta quiere conseguir esto mediante la mejora de la comprensión de la demanda de energía dentro del centro de datos, aumentando la concienciación, y mediante la recomendación de prácticas y objetivos energéticamente eficientes.

Grupo de trabajo sobre Green IT de la plataforma INES (Iniciativa Española de Software y Servicios, <http://www.ines.org.es>) es la Plataforma Tecnológica Española en el área de los Sistemas y Servicios Software y constituye una red de cooperación científico-tecnológica integrada por los agentes tecnológicos relevantes de este ámbito (empresas, universidades, centros tecnológicos, etc.).

Según la Agenda Estratégica de Investigación de INES, el plan de

dinamización para el Grupo de Trabajo de Green IT consiste en las siguientes acciones:

- Análisis de la influencia e importancia de las soluciones de Green IT.
- Difusión de las informaciones, noticias y existencia de este grupo de trabajo por Internet.
- Fomentar el interés y apoyar el desarrollo bajo Green IT.

Big Green Innovations (<http://www.ibm.com/technology/greeninnovations/>), programa de IBM. Dentro de este programa, y con fines educativos, IBM ha presentado un centro de datos virtual ecológico denominado Virtual Green Data Center.

La lista Green500 (<http://www.green500.org>) proporciona una clasificación de los supercomputadores más eficientes energéticamente del mundo, sirviendo como una visión complementaria a la lista Top500 (<http://www.top500.org>).

Otras empresas, como Google, Dell o Symantec, están desarrollando programas de eficiencia energética, tanto para sus propios procesos de TI como para los de sus clientes.

#### **468 REDES SAN Y ELEMENTOS DE UNA SAN**

Como resumen una SAN es una red donde se realiza el almacenamiento y se gestiona la seguridad de los datos. Las SAN (Storage Area Network, Redes de Almacenamiento) son redes en las que se conectan servidores de almacenamiento (especialmente arrays de discos). También hay que considerar como parte de las SAN las librerías necesarias para el uso de los arrays y los accesos a las redes. De forma contraria a las redes tradicionales, en las SAN se emplean protocolos orientados a la

recuperación de la información de los arrays de disco e inspirados en los propios estándares de comunicación con discos tradicionales (SCSI y SATA).

Normalmente los equipos diseñados para participar en estas redes suele ser especialmente caro aunque su precio depende, en una grande medida, de las tecnologías y protocolos empleados para la transmisión de los datos. Entre las tecnologías disponibles en la actualidad se encuentran: iSCSI (Internet Small Computer Storage Interconnect), Fibre Channel y AOE (ATA Over Ethernet, Advanced Technology Attachment Over Ethernet).

Entre las ventajas de la interconexión de redes de almacenamiento se resaltan las siguientes:

- Elimina los límites de distancia de discos introducidos por SCSI o ATA
- Consigue un mayor caudal de datos ya que los protocolos están específicamente diseñados para la transferencia de datos de dispositivos de almacenamiento.
- Permite un aprovechamiento mayor de los discos permitiendo que más de un servidor acceda al mismo disco.
- Capacidad para el uso de múltiples discos de forma transparente desde uno o varios servidores.
- Adquisición de discos diferida debido al mayor aprovechamiento
- Capacidades de recuperación ante desastres. Los arrays de discos empleados en las SAN suelen disponer de discos de reserva (para fallos de otros discos) y permitir distintos esquemas de RAID.
- Recuperación en caliente ante desastres
- Mejor capacidad de administración. La administración es más sencilla y está más centralizada.
- Reducción de los costos de administración y de almacenamiento de

datos

- Mejora de disponibilidad global ya que las SAN tienen menos fallos que los discos internos de los equipos.
- Reducción de servidores eliminando servidores de arquitecturas antiguas (NFS, SMB, etc).
- Reducción del caudal de las redes convencionales, pues las copias de seguridad se pueden hacer desde las SAN.
- Incremento de la rapidez de las operaciones de Entrada/Salida
- Reducción de los costes de administración de backups
- Protección de datos críticos
- Incremento de la capacidad de forma transparente
- Desarrollo y prueba de aplicaciones de forma más eficiente mediante el uso de copias de los datos de producción realizadas en la SAN.
- Facilita el empleo de clusters de servidores que tienen que disponer de un almacenamiento común.
- Permiten el almacenamiento bajo demanda de forma que cualquiera servidor puede solicitar espacio de almacenamiento segundo sus necesidades.

Dentro de una organización, se debería incluir en una SAN la siguiente información:

- La información almacenada por SGBDs (Sistemas Gestores de Bases de Datos). De hecho, algunos sistemas gestores como Oracle, Sybase, SQLServer, DB2, Informix o Adabase recomiendan esta alternativa
- La información almacenada por servidores de archivos. Los servidores de archivos funcionarán mejor y con menos recursos si los archivos están almacenados en una SAN.
- Servidores de backup. Si los servidores de backup están conectados a una SAN se conseguirá reducir los tiempos de copia de seguridad con

respecto a hacerlos en una LAN (Local Area Network, Red de Área Local) y reducir el tráfico de la LAN.

- Archivos de servidores de voz y video para streaming. Debido a que este tipo de servicios requiere grandes cantidades de disco, una SAN puede reducir los costes asociados al almacenamiento y desplazar el máximo posible el coste (incluir nuevos discos en los arrays cuando sean necesarios).
- Buzones de usuario (mailboxes) de servidores de correo permitiendo que los servidores de correo funcionen mas rápido y que se pueda realizar una restauración rápida en caso de que algún archivo se corrompa.
- Servidores de aplicaciones de alto rendimiento. Las SAN pueden mejorar el rendimiento de cualquier aplicación incluyendo gestores documentales, aplicaciones científicas, aplicaciones de datawarehouse y cuadros de mando integrales, aplicaciones para gestionar las relaciones con los clientes (CRM), etc.
- Soluciones de Virtualización.

Asi mismo, no es te conveniente usar una SAN para:

- Servidores web que no requieran grandes necesidades de almacenamiento (la mayoría)
- Servidores con servicios de red básicos como DNS, DHCP, WINS (Windows Internet Name Servers) y controladores de dominio de Windows (DC). Este tipo de servidores no requieren de las capacidades de almacenamiento permitidas por las SAN
- PCs de escritorio
- Servidores que necesitan menos de 10Gb de almacenamiento
- Servidores que no necesitan un acceso rápido a la información
- Servidores que no comparten archivos

### **468.1 Estructura de las SAN**

Habitualmente las SAN se conciben y se estructuran en tres capas:

1. La capa de hosts: Constituida en su mayoría por los servidores, los drivers y software necesarios para la conexión a la red y los HBAs (Host Bus Adapters) que son dispositivos (tarjetas) que se conectan a cada servidor para acceder al almacenamiento (en algunas soluciones concretas son adaptadores Ethernet simples y en el caso Fibre Channel llevan un conector GBIC-Gigabit Interface Connector).
2. La capa de estructura (fabric layer): Constituida por HUBs, Switches, Gateways y Routers se fuera necesario. Si se emplea la tecnología Fibre Channel, todos estos dispositivos emplean GBICs (Gigabit Interface Conectores) para la interconexión de los dispositivos de las capas superiores e inferiores.
3. La capa de almacenamiento (storage layer): Constituida por todo tipo de dispositivos de almacenamiento.

Un conjunto de discos situados en el mismo sitio y sin funcionalidades adicionales se conoce como JBOD (Just a Bunch Of Disks). Dentro de la capa de almacenamiento, los arrays no son simplemente JBODs, sino que incluyen ciertas funcionalidades interesantes implementadas en el firmware de la controladora como el RAID.

### **469 VIRTUALIZACIÓN DEL ALMACENAMIENTO**

Este tipo de virtualización permite una mayor funcionalidad y características avanzadas en el sistema de almacenamiento. Consiste en abstraer el almacenamiento lógico del almacenamiento físico y suele



usarse en SANs (Storage Area Network, Red de área de almacenamiento).

Este sistema de almacenamiento también se conoce como “storage pool”, matriz de almacenamiento, matriz de disco o servidor de archivos. Estos sistemas suelen usar hardware y software especializado, junto con unidades de disco con el fin de proporcionar un almacenamiento muy rápido y fiable para el acceso a datos. Son sistemas complejos, y pueden ser considerados como un ordenador de propósito especial diseñado para proporcionar capacidad de almacenamiento junto con funciones avanzadas de protección de datos. Las unidades de disco son sólo un elemento dentro del sistema de almacenamiento, junto con el hardware y el software de propósito especial incorporado en el sistema.

Los sistemas de almacenamiento pueden ser de acceso a nivel de bloque, o acceso a nivel de ficheros. El acceso por bloques suele llevarse a cabo por medio de Fibre Channel , iSCSI , SAS , FICON u otros protocolos. Para el acceso a nivel de archivo se usan los protocolos NFS o CIFS.

Dentro de este contexto nos podemos encontrar con dos tipos principales de virtualización: la virtualización por bloques y la virtualización por archivos.

#### **469.1      *Virtualización por bloques***

Este tipo de virtualización se basa en la abstracción (diferenciación) entre el almacenamiento lógico y el almacenamiento físico, consiguiendo que el acceso no tenga en cuenta el almacenamiento físico o estructura heterogénea.

Existen tres tipos de virtualización por bloques: basada en host, basada en dispositivos de almacenamiento, basada en red.

##### **469.1.1      Virtualización basada en host**

Esta virtualización requiere software adicional que se ejecuta en el host. En algunos casos la administración de volúmenes está integrada en el sistema

operativo, y en otros casos se ofrece como un producto separado. Los volúmenes (LUN) disponibles en el sistema son manejados por un controlador de dispositivos físicos tradicional. Por encima de este controlador se encuentra una capa software (el gestor de volúmenes) que intercepta las peticiones de E / S, y proporciona la búsqueda de meta-datos y mapeos de E / S.

Los sistemas operativos más modernos tienen algún tipo de gestor de volúmenes lógicos integrado (MVI en UNIX / Linux, o Administrador de discos lógicos o LDM en Windows), que realiza tareas de virtualización.

Existen varias tecnologías que implementan este tipo de virtualización, como pueden ser la gestión de volúmenes lógicos (Logical Volume Management, LVM), los sistemas de archivos (CIFS, NFS) o el montaje automático (autofs)

#### 469.1.2 Virtualización basada en dispositivos de almacenamiento

Se puede llevar a cabo la virtualización basada en medios de almacenamiento masivo utilizando un controlador de almacenamiento primario que proporcione los servicios de virtualización y permita conexión directa de los controladores de almacenamiento. En función de la implementación es posible usar modelos de distintos fabricantes.

El controlador primario proporcionará la puesta en común y los meta-datos de servicio de gestión. También puede ofrecer servicios de replicación y migración a través de los controladores que se virtualizan.

Una nueva generación de controladores de serie del disco permite la inserción posterior de los dispositivos de almacenamiento.

Los sistemas RAID pueden ser un ejemplo de esta técnica. Estos sistemas combinan varios discos en una sola matriz.

Las matrices avanzadas de disco, cuentan a menudo con clonación, instantáneas y replicación remota. En general, estos dispositivos no ofrecen

los beneficios de la migración de datos o de replicación a través de almacenamiento heterogéneo, ya que cada fabricante tiende a utilizar sus propios protocolos propietarios.

#### *469.1.3 Virtualización basada en red*

Esta es una virtualización de almacenamiento operando en un dispositivo basado en red (por lo general un servidor estándar o un smart switch) y el uso de redes iSCSI o FC de Fibre Channel para conectar como SAN (Storage Area Network). Este es el tipo de virtualización de almacenamiento más común.

El dispositivo de virtualización se encuentra en la SAN y proporciona la capa de abstracción entre los host, que permiten la entrada/salida, y los controladores de almacenamiento, que proporcionan capacidad de almacenamiento.

Hoy en día existen dos implementaciones distintas, la basada en el **dispositivo** y la basada en **conmutación**. Ambos modelos proporcionan los mismos servicios: gestión de discos, búsqueda de meta-datos, migración y replicación de datos. Igualmente, ambos modelos necesitan de un hardware específico que permita ofrecer dichos servicios.

La basada en dispositivos consiste en establecer el hardware especializado entre los hosts y la parte de almacenamiento. Las solicitudes de entrada/salida se redirigen al dispositivo, que realiza la asignación de meta-datos, mediante el envío de sus propias órdenes de E/S a la solicitud de almacenamiento subyacente. El hardware usado también puede proporcionar almacenamiento de datos en caché, y la mayoría de las implementaciones proporcionan algún tipo de agrupación de cada uno de los dispositivos para mantener un punto de vista atómico tanto de los meta-datos como de los datos de la caché.

Este tipo de almacenamiento también puede clasificarse en in-band (simétrica) o out-of-band (asimétrica).

#### **469.1.3.1 In-band (simétrica)**

En este caso los dispositivos de virtualización se asientan entre el host y el almacenamiento. Todas las peticiones de E/S y datos pasan a través del dispositivo. Los host nunca interactúan con el dispositivo de almacenamiento sino con el dispositivo de virtualización.

#### **469.1.3.2 Out-of-band (asimétrica)**

Los dispositivos usados en este tipo de virtualización también son llamados servidores de meta-datos. La única finalidad de estos dispositivos es proporcionar la asignación de meta-datos. Esto implica el uso de software adicional en el host, que es conocedor de la ubicación real de los datos. De este modo, se intercepta la petición antes de que salga del host, se solicita una búsqueda de meta-datos en el servidor (puede ser a través de una interfaz que no sea SAN) y se devuelve la ubicación real de los datos solicitados por el host. Finalmente se recupera la información a través de una solicitud de E/S común al dispositivo de almacenamiento. No se puede dar un almacenamiento en caché ya que los datos nunca pasan a través del dispositivo de virtualización.

### **469.2 Virtualización a nivel de archivo**

Con este tipo de virtualización se pretende eliminar las dependencias entre el acceso a datos a nivel de archivo y la ubicación física de los mismos. Esta técnica, conocida como NAS (Network-Attached Storage) o almacenamiento conectado a red, suele ser un equipo especializado pensado exclusivamente para almacenar y servir ficheros. Los equipos que funcionan como dispositivo NAS suelen incluir un sistema operativo específico para el propósito, como puede ser FreeNAS o FreeBSD.

Estos sistemas pueden contener uno o más discos duros, dispuestos a menudo en contenedores lógicos redundantes o arrays RAID.

NAS utiliza protocolos basados en archivos como NFS (sistemas UNIX), SMB / CIFS (Server Message Block/Common Internet File System) (sistemas MS Windows), o AFP (Apple Filing Protocol, sistemas Apple Macintosh). Las unidades NAS no suelen limitar a los clientes a un único protocolo. FTP, SFTP, HTTP, UPnP, rsync y AFS (Andrew File System) también lo soportan.

De este modo se consigue optimizar la utilización del almacenamiento y las migraciones de archivos sin interrupciones.

### **469.3 Diferencias entre NAS y SAN**

NAS proporciona almacenamiento y un sistema de archivos, lo que suele contrastar con SAN, que solamente proporciona almacenamiento basado en bloques y deja del lado del cliente la gestión del sistema de archivos.

NAS aparece en el sistema cliente como un servidor de archivos (se pueden asignar unidades de red a las acciones del servidor) mientras que un disco a través de una SAN se presenta al cliente como un disco más del sistema operativo, que podemos montar, desmontar, formatear...

	<b>NAS</b>	<b>SAN</b>
<b>Tipo de datos</b>	Archivos compartidos	Datos a nivel de bloque, por ejemplo, bases de datos.
<b>Cableado utilizado</b>	Ethernet LAN	Fibre Channel dedicado
<b>Clientes principales</b>	Usuarios finales	Servidores de aplicaciones
<b>Acceso a disco</b>	A través del dispositivo NAS (IP propia)	Acceso directo

## **4610 GESTIÓN DEL CICLO DE VIDA DE LA INFORMACIÓN (ILM)**

#### **4610.1      *Gestión del Ciclo de Vida de los Datos***

La gestión del ciclo de vida de los datos o DLM (Data Lifecycle Management) es un enfoque de la gestión de la información desde el punto de vista del manejo del flujo de los datos de un sistema de información durante todo su ciclo de vida, desde que se crean y se produce su primer almacenamiento, hasta que son declarados obsoletos y eliminados del sistema.

Los productos para la gestión del ciclo de vida de los datos tratan de automatizar los procesos que forman parte de este ciclo de vida. Organizan los datos en distintos niveles siguiendo unas políticas especificadas, y automatizan la migración o intercambio de los datos entre unos niveles y otros basándose para ello en los criterios especificados de cada uno.

Como norma general, los datos más recientes y aquellos a los que se accede con más frecuencia se tienden a almacenar en medios de almacenamiento más rápidos, pero también más caros, mientras que los datos de un nivel menos crítico se almacenan en los dispositivos más baratos y más lentos.

Las arquitecturas que gestionan el ciclo de vida de los datos suelen incluir un sistema de archivos que indexa toda aquella información crítica y aquella considerada no tan crítica pero que guarda relevancia o relación que esta. Con esta información crea copias de respaldo, los almacena en ubicaciones seguras para evitar manipulaciones pero que puedan ser accesibles de una manera segura y confiable.

Estas arquitecturas también se encargan de las posibles duplicaciones de datos y de la comprensión de los mismos para garantizar un correcto y eficiente uso del espacio de almacenamiento disponible.

Desafortunadamente, muchas implementaciones de DLM de negocios se han estancado, principalmente porque las empresas no han logrado definir ni las políticas de migración adecuadas ni el archivado de datos. Dado que esas políticas necesitan reflejar las prioridades de regulación y de negocio,

en sus definiciones es necesario una colaboración que involucre no solo a miembros del departamento de tecnologías de la información, sino también a miembros de diferentes departamentos del negocio.

Por otro lado, el criterio más sencillo para realizar una migración de la información a un sistema de almacenamiento más económico es el temporal, es decir, los datos más antiguos en los sistemas más lentos y baratos. Sin embargo, las empresas en industrias altamente reguladas a menudo quieren ir más lejos, estableciendo la clasificación de los datos en función de la rapidez con la que se precisen, o la frecuencia con la que se accede a ellos, o en base a quien los ha enviado o recibido, o en base a un conjunto de palabras clave o cadenas numéricas, etc. Entonces el reto está en conseguir definirlos de tal manera que sea viable realizarlo en el tiempo y mediante la menor intervención humana.

#### **4610.2      *Gestión del Ciclo de Vida de la Información***

La gestión del ciclo de vida de la información o ILM (Information Lifecycle Management) es un enfoque integral para el manejo del flujo de los datos de un sistema de información y los metadatos asociados desde su creación y almacenamiento inicial hasta el momento en que estos se vuelven obsoletos y son borrados.

A diferencia de anteriores enfoques para la gestión de almacenamiento de datos, ILM abarca todos los aspectos en los que se tratan los datos, partiendo de las prácticas de los usuarios, en lugar de la automatización de los procedimientos de almacenamiento y en contraste con los sistemas más antiguos, ILM permite criterios mucho más complejos para la realización de la gestión del almacenamiento que la antigüedad de los datos o la frecuencia de acceso a ellos.

Es importante destacar que ILM no es sólo una tecnología sino que integra los procesos de negocio y TI con el fin de determinar cómo fluyen los datos

a través de una organización, permitiendo a los usuarios y administradores gestionarlos datos desde el momento que se crean hasta el instante en el que ya no son necesarios.

Aunque los términos gestión del ciclo de vida de los datos (DLM) y gestión del ciclo de vida de la información (ILM) a veces se utilizan indistintamente, ILM a se considera un proceso más complejo.

La clasificación de los datos en función de valores del negocio es una parte integral y muy importante del proceso ILM. Esto quiere decir que ILM reconoce que la importancia de los datos no se basa únicamente en su antigüedad o en su frecuencia de acceso, sino que ILM espera que sean los usuario y los administradores los que especifiquen distintas directivas para que los datos vayan variando de una manera decreciente su relevancia o grado de importancia para la organización, o que puedan conservar su importancia durante todo su ciclo de vida, etc.

Para una exitosa y eficiente implementación de IML se necesita que la organización identifique requisitos de seguridad de los datos críticos e incluirlos en sus procesos de clasificación. Los usuarios de los datos, tanto los individuos como las aplicaciones, deben de ser identificados y categorizados en función de las necesidades asociadas con sus tareas.

Algunas de las mejores prácticas relacionadas con la implementación de IML comparten enfoques como:

- Se centran en la productividad del usuario con el fin de obtener una ventaja estratégica a través del acceso a los datos necesarios.
- Proteger los datos contra el robo, la mutilación, la divulgación involuntaria, o la eliminación.
- Crear múltiples capas de seguridad, sin crear una gestión excesivamente compleja.



- Asegurarse que los procesos de seguridad están incorporados en los procesos generales del negocio y en los procesos de TI.
- Utilizar estándares y modelos de referencias con el fin de satisfacer únicamente las necesidades de seguridad de la organización.

Por supuesto, cada organización deberá desarrollar e implementar su propia solución de seguridad de almacenamiento, que debe seguir evolucionando, adaptándose a las nuevas oportunidades, amenazas y capacidades.

### **4610.3    *Alguna soluciones para la gestión***

#### **4610.3.1    Microsoft**

Microsoft Identity Lifecycle Manager ofrece una solución integrada y completa para la gestión del ciclo de vida de las identidades de usuario y sus credenciales asociadas. Esta solución aporta la sincronización de identidades, los certificados y administración de contraseñas y suministro de usuarios. La solución funciona bajo plataformas Windows y otros sistemas organizacionales.

#### **4610.3.2    IBM**

Las soluciones de IBM para la gestión del ciclo de vida de la información se han agrupado en cinco categorías (IBM, 2008):

- *Archivo de correo electrónico* (IBM DB2 CommonStore, VERITAS Enterprise Vault, OpenText-IXOS Livelink)
- *Aplicación y base de datos de archivo* (Archivo Activo de Princeton Softech),

- *Gestión del ciclo de vida de los datos* (TotalStorage de IBM SAN File System)
- *Gestión de contenidos* (repositorio de administración de contenido, DB2 Content Manager)
- *Gestión de registros* (IBM DB2 Records Manager).

#### 4610.3.3 Oracle

Oracle ILM Assistant es una herramienta que se basa en una interfaz gráfica de usuario para la gestión de entorno de ILM. Ofrece la posibilidad de crear definiciones de ciclo de vida, que se asignan a las tablas en la base de datos. Posteriormente basándose en las políticas establecidas sobre el ciclo de vida, ILM Assistant informa cuando es el momento para mover, archivar o suprimir los datos. También muestra las necesidades de almacenamiento y el ahorro de costes asociados con el cambio de ubicación de los datos.

Otras capacidades de Oracle ILM Assistant incluyen la habilidad de mostrar cómo particionar una tabla basada en una definición del ciclo de vida, y poder simular los eventos para comparar el resultado en caso de que la tabla fuera particionada.

### **4611 SISTEMAS DE BACKUP: HARDWARE Y SOFTWARE DE BACKUP**

Un factor importante en todo sistema de backup es la elección de los sistemas hardware y software que lo componen.

#### **4611.1 Hardware de Backup**

En la categoría de elementos hardware de backup tenemos:

#### 4611.1.1 Cintas

Tradicionalmente, los cartuchos de cinta magnética son los medios de comunicación más habituales en los sistemas de backup. Como soporte de almacenamiento de los respaldos de datos, la cinta magnética tiene una larga historia de uso y es el medio de copia de seguridad con mayor nivel de madurez. La cinta magnética, o de una forma más abreviada, la cinta, es un componente basado en cartuchos que se hace típicamente de algún tipo de plástico rígido. Contiene uno o más bobinas de plástico flexible que se han impregnado con un material con comportamiento magnético.

Los cartuchos de cinta están fabricados en varios formatos. Cada formato tiene unas características diferentes que responden a las diferentes necesidades de almacenamiento físico y de tiempo de preservación de la copia de seguridad, tanto en términos de la cantidad de datos almacenados, como de vida útil de los medios de almacenamiento o su coste. Los formatos de cinta de uso común son los siguientes:

- DLT/ SDLT
- LTO
- AIT
- STK 9840/9940/T10000

Según el tipo de cada cartucho este posee distintas capacidades o características como la velocidad de funcionamiento. El mercado está renovando continuamente este tipo de dispositivos con el fin de mejorar ambos aspectos. Sin embargo, existen tres formatos que podemos considerar de los más comunes y tienen características particulares que se describen aquí como ejemplos de elementos arquitectónicos de diseño:

*DLT, LTO, T10000 y STK.*

El resto de formatos, aunque sean formatos comunes, se utilizan normalmente para entornos especializados, como el archivado y almacenamiento intermedio (nearline storage) empleado entre el almacenamiento online y el almacenamiento de backups.

#### **4611.1.1.1 Digital Linear Tape (DLT)**

Digital Linear Tape (DLT) es el formato de cinta más antiguo y por lo tanto uno de los productos más maduros del mercado. Originalmente fue diseñado e implementado por DEC en 1984, para posteriormente ser adquirida por Quantum y redistribuido en 1994.

DLT es el primer cartucho de cinta compacta para copias de seguridad de sistemas abiertos en la empresa. Mientras que otros tipos de medios se encontraban en uso (como la cinta media pulgada, 4mm/8mm, y otros), DLT proporciona el mejor compromiso entre todos los factores debido a su tamaño, la fiabilidad de su almacenamiento, la capacidad, y disponibilidad relativa.

La conectividad de DLT, se limita a los tradicionales de SCSI, y está limitado a 300 GB de capacidad nativa de almacenamiento y 160 MB /seg velocidad de transferencia (SDLT600). Existían otras variantes disponibles, pero nunca llegaron a popularizarse con carácter general. Hoy en día, DLT se encuentra normalmente como copia de seguridad de larga duración en entornos pequeños que no requieren mayor capacidad.

#### **4611.1.1.2 Linear Tape Open (LTO)**

Linear Tape Open (LTO) fue diseñado y concebido como una evolución y alternativa a los formatos DLT y otros ya existentes, y estaba destinado a proporcionar una plataforma común para los backups en cinta.

Seagate, HP e IBM fueron los iniciadores originales del consorcio LTO, encargado de realizar el desarrollo inicial y el cuál mantiene la licencia de la tecnología y la certificación del proceso. En teoría, se debería de haber producido un formato estándar de cinta, con el cual los fabricantes podrían seguir trabajando con el estándar en el mercado e incorporando sus propias características y funciones adicionales.

Sin embargo, entre el original LTO-1 y los formatos de LTO-2 hubo problemas de compatibilidad. Estos problemas abarcaban desde bloqueos en las cintas cuando se utilizan medios adquiridos a dos proveedores distintos hasta la incapacidad de una unidad LTO de un fabricante a leer los datos escritos en un cartucho de otra.

El LTO-1 inicial proporcionaba 100 GB de almacenamiento nativo y 15 MB /seg; con los actuales sistemas de LTO-4 se proporcionan 400 GB de almacenamiento nativo de 160 MB / seg. Por su parte, el LTO-5 proporciona 800 GB de capacidad de almacenamiento nativo a 160 MB / seg.

#### **4611.1.1.3 Sun StorageTek T10000 (T10k)**

El T10000 / StorageTek (T10k) de Sun representa uno de las tecnologías de almacenamiento en cinta que mejor se ha comportado en términos de capacidad. El T10k es un formato propietario producido únicamente por

StorageTek y se encuentra normalmente en entornos en los que se empleaban las tecnologías anteriores de Sun como el STK (9840/9940). También se han utilizado en sistemas abiertos de servidores o mainframe. El T10k está diseñado para 500 GB de almacenamiento nativo de 120 MB / seg.

#### **4611.1.1.4 Características de almacenamiento en cinta**

Aunque todos los datos anteriores indican un valor interesante en cuanto al rendimiento, todos los dispositivos de cinta con características similares de rendimiento deben tenerse en cuenta a la hora de diseñar entornos de backup.

La primera y más importante de ellas es el hecho de que todas las unidades de cinta son entornos serie. A diferencia de los dispositivos de disco, los dispositivos de cinta escriben los bloques de datos de forma lineal, uno tras otro. Las unidades de cinta sólo tienen una cabeza de escritura que escribe un bloque de datos de cada vez en la cinta, a medida que ésta se mueve por ella. Los dispositivos de disco tienen una serie de dispositivos de escritura, o cabezas, que se mueven a varios puntos del disco giratorio para situar los datos de una manera óptima. Esto permite que los dispositivos de disco puedan leer cualquier trozo de información solicitada. Dado que los discos tienen varias cabezas para obtener bloques de datos en paralelo, varios sistemas pueden acceder al disco al mismo tiempo

La lectura de los datos de una cinta, se realiza mediante el proceso inverso: La cinta debe rebobinarse hasta el principio, hacia adelante hasta bloque que se necesita, y leer así el bloque de datos. Al poder devolverse únicamente un segmento de datos con cada lectura, los dispositivos de

cinta no se pueden compartir de forma paralela entre sistemas sin un mecanismo para transferir el control entre los sistemas que usan dicho dispositivo.

El tipo de conectividad también tiene influencia sobre la utilización de dispositivos de cinta. Las unidades de cinta dependen de una conexión directa con el host para el transporte de los datos. Una vez más, esto se debe al hecho de que las unidades de cinta son dispositivos de serie que sólo aceptan una sola conexión a la vez.

#### *4611.1.2 Disco*

La cinta proporciona un método muy maduro, muy conocido, y de bajo coste para almacenar copias de seguridad. Sin embargo, las debilidades, tales como la naturaleza secuencial de la cinta, la complejidad mecánica, y la gran variabilidad del rendimiento de los dispositivos de cinta están rápidamente relegando a la cinta a medio de almacenamiento secundario o terciario en muchos entornos.

Con todos los problemas con la cinta, los administradores buscaban un medio que permitiera un rápido acceso a las copias de seguridad y que proporcionase una forma de tener un almacenamiento rápido y fiable: *el disco*.

Los backup a disco son simples sistemas de archivos que han sido situados aparte para que el software de backup los use. Aunque esto parece sencillo, la implementación y gestión de las soluciones basadas en disco pueden ser muy complejas.

El almacenamiento en disco supera algunas de las desventajas propias de las cintas. Por la capacidad de recibir datos de forma rápida, tiene múltiples flujos para almacenar copias de seguridad al mismo tiempo, y tiene la capacidad de presentar el almacenamiento de un número de maneras diferentes, dependiendo de la necesidad del sistema, por eso, el disco es muy empleado como un medio de almacenamiento de copia de seguridad

primario.

Pero el disco también tiene sus debilidades, el coste de los medios de comunicación, la falta de portabilidad, y la dificultad de asegurar la plena utilización de los medios de comunicación hacen que el disco no sea tan satisfactorio como parece a priori.

#### *4611.1.3 Medios Virtuales*

Los medios virtuales emulan el hardware físico de cinta con el objetivo de reducir o eliminar los problemas de gestión asociados a los medios físicos. Mediante la eliminación del hardware con una alta complejidad mecánica y de gestión y la eliminación de sus sistemas asociados y reemplazándolos por unidades de disco, los medios virtuales también tiene la ventaja de aumentar la fiabilidad general del entorno de backup. Los medios virtuales ofrecen estas ventajas sin cambiar los procedimientos operativos o exigir modificaciones del software de copia de seguridad. Además, en algunos casos, el rendimiento puede aumentarse a través de un mejor uso del ancho de banda en los medios de comunicación utilizados para conectar los medios virtualizados con los servidores de backup.

Los Medios virtuales de copia de seguridad se asocian tradicionalmente de forma exclusiva con bibliotecas de cintas virtuales (VTL) pero recientemente se han realizado nuevas implementaciones a través de protocolos que permiten la virtualización de otros tipos de sistemas de almacenamiento.

#### *4611.1.4 Medios Ópticos*

Los medios ópticos se sitúan entre las ventajas de las cintas y las del disco.



Sobresalen en las áreas de fiabilidad, flexibilidad, ciclo de trabajo e inamovilidad, mientras que sus retos los encontramos en las áreas de rendimiento, capacidad y coste.

#### **4611.1.4.1 CD**

CD, o compact disk, es un soporte digital óptico que se utiliza para el almacenamiento de prácticamente cualquier tipo de datos. En la actualidad el uso del CD está decayendo a favor del aumento del uso de un nuevo medio de similares características como el DVD.

El CD ha servido y sigue sirviendo como medio de almacenamiento de copias de seguridad gracias a su fiabilidad e inamovilidad. Proporciona en comparación con otros medios como la cinta magnética, mayor seguridad y protección de los datos, dado que el propio medio es mucho más robusto frente a interacciones físicas externas (por ejemplo los campos magnéticos).

Además de ser un medio habitual para el almacenamiento de pistas de audio, los CDs se utilizan habitualmente para la generación de copias de seguridad relacionadas con la recuperación de los sistemas.

Los sistemas de CD utilizan un dispositivo hardware específico para grabar información, conocido como grabadora/regrabadora de CD. Existen también dispositivos hardware similares que solamente permiten la lectura de este medio.

Las capacidades habituales de los CD estándar abarcan desde los 650MB hasta los 900MB.

#### **4611.1.4.2 DVD**

Los DVDs vienen a ser la evolución de la tecnología digital óptica de los CDs.

Al igual que los CDs, existen dos tipos de dispositivos para el uso de los DVDs que son las grabadoras y los lectores. Existen diferentes tipos de DVDs y diferentes categorizaciones, siendo la más importante la relativa al número de capas, factor que determina la capacidad final del dispositivo.

Las capacidades actuales abarcan desde los 4,3Gb hasta los 17Gb. Los DVD utilizan dos tipos de sistemas de ficheros que reemplazan el antiguo ISO 9660 de los CDs, y que son el UDF y el Joliet.

## **4611.2      *Software de Backup***

En la categoría de elementos software de backup tenemos herramientas de código abierto o software libre y software privativo o comercial. Las herramientas más comunes a nivel de software son:

### **4611.2.1      Herramientas de código abierto - AMANDA**

Amanda (Advanced Maryland Automated Network Disk Archiver), es el software de código abierto de copia de seguridad más conocido. Amanda se desarrolló inicialmente en la Universidad de Maryland en 1991 con el objetivo de proteger los archivos de un gran número de estaciones de trabajo cliente con un servidor de copia de seguridad único. James da Silva fue uno de sus desarrolladores originales.

El proyecto Amanda se registró en SourceForge.net en 1999. Jean-Louis Martineau, de la Universidad de Montreal ha sido el líder del desarrollo de Amanda en los últimos años. Durante años, más de 250 desarrolladores han contribuido al código fuente de Amanda, y miles de usuarios aportan pruebas y comentarios, lo que lo convierte en un paquete robusto y estable. Amanda se incluye con la mayor parte de las distribuciones Linux.

En un principio, Amanda fue utilizado mayoritariamente en las universidades, laboratorios técnicos, y departamentos de investigación. Hoy, con la amplia adopción de Linux en los departamentos de informática, Amanda se encuentra en muchos otros lugares, sobre todo cuando la atención se centra en aplicaciones LAMP (Linux+Apache+MySQL+PHP). Con los años, Amanda ha recibido múltiples premios de los usuarios.

Amanda permite configurar un único servidor backup maestro para realizar múltiples copias de seguridad de equipos Linux, Unix, Mac OS X, y Windows en una amplia variedad de dispositivos: cintas, discos, dispositivos ópticos, bibliotecas de cintas, sistemas RAID, dispositivos NAS, y muchos otros.

Las principales razones para la adopción generalizada de Amanda son:

- Se puede configurar un único servidor de copia de seguridad de varios clientes en red con cualquier dispositivo de almacenamiento: una cinta, disco o sistema de almacenamiento óptico.
- Está optimizado para el backup en disco y cinta, permitiendo escribir simultáneamente backup a cinta y disco.
- No utiliza drivers propietarios, cualquier dispositivo soportado por un sistema operativo también podrá funcionar en Amanda.
- Utiliza herramientas estándar, como dump y tar. Puesto que no son formatos propietarios, los datos se pueden recuperar con esas mismas herramientas.
- Se utiliza un planificador que optimiza niveles de seguridad para los diferentes clientes, de tal manera que el tiempo total del backup es aproximadamente el mismo para cada ejecución.
- Existe una amplia y activa comunidad de usuarios que crece día a día.

- El coste total de propiedad (TCO) de una solución de backup basada en Amanda es significativamente menor que el TCO de cualquier solución que utilice software privativo.

#### 4611.2.2 Herramientas de código abierto – BackupPC

BackupPC es un sistema de alto rendimiento que permite realizar copias de seguridad de sistemas Unix, Linux, Windows y MacOS en un disco. Es por tanto una herramienta basada totalmente en disco.

Ofrece una serie de ventajas como son:

- **Soporta cualquier sistema operativo cliente.** Esto se debe a que se utilizan herramientas estándar que o vienen con el SO o se pueden añadir al SO, sin necesidad de instalar cliente. Así resulta más fácil integrar un nuevo cliente.
- **Interfaz Web** con control de usuario para acceder a copias de seguridad. La mayoría de los SO trae un navegador web, así que usar una interfaz web es otra manera de acelerar el proceso de incorporación de nuevos clientes con diferentes sistemas operativos. La interfaz web está diseñada para dar el máximo control posible al cliente de forma segura. El usuario puede solicitar restauraciones, y navegar fácilmente y restaurar archivos individuales. Sin embargo, el usuario no podrá ver las máquinas de otro usuario.
- **Soporte de clientes DHCP.** Mediante el uso de servicios estándar, BackupPC soporta clientes DHCP, siempre y cuando el cliente esté registrado con un servicio de nombres como DNS, Active Directory o LDAP.

#### **Funcionamiento de BackupPC**

El modelo de BackupPC tiene un usuario por cliente. Esto es así porque

BackupPC fue específicamente diseñado para realizar copias de seguridad de PCs de varios usuarios (de ahí el nombre).

Normalmente, el usuario es el propietario de los datos de la máquina. Si se trabaja con un servidor de ficheros, el usuario deberá ser un administrador.

BackupPC envía mensajes de correo electrónico al propietario si no puede realizar la copia de seguridad después de un tiempo configurable; el propietario puede gestionar las restauraciones de las copias a través de una interfaz web.

En los siguientes puntos se describen algunas de las características proporcionadas por BackupPC:

- **Directo al disco.** BackupPC almacena todas sus copias de seguridad directamente en el disco. Los archivos idénticos en cualquier directorio o cliente se guardan sólo una vez, lo que reduce drásticamente los requisitos de almacenamiento del servidor. Estos archivos se almacenan en un conjunto de discos. Además del conjunto de discos, las copias de seguridad están en un árbol de directorios organizados por host.

BackupPC también tiene un proceso (que se lanza por las noches) que recupera espacio del conjunto de discos que no está referenciado por ningún backup, lo que evita un uso inadecuado del espacio en disco. Este es un proceso automático que el administrador no tiene que configurar.

- **Sistema operativo del servidor** La parte del servidor de BackupPC está diseñada para ejecutarse en un sistema tipo Unix con Perl y mod\_perl. Ofrece el mejor rendimiento con Apache, pero se puede ejecutar en cualquier servidor web que soporte Perl (se requiere mod\_perl o Perl setuid.) El servidor debe tener un disco con gran capacidad o RAID para almacenar los backups.
- **Sistema operativo del cliente.** Como se comentó anteriormente, soporta cualquier SO. Las versiones más modernas de las variantes

comerciales de Unix (Solaris, AIX, IRIX, HP-UX) traen en la propia distribución las herramientas tar, compress, gzip, rsync, y rsh y / o ssh. Otros sistemas operativos tipo Unix (Linux, FreeBSD, OpenBSD, NetBSD, Mac OS X) también cuentan con estas herramientas.

Los clientes de Windows pueden hacer copias de seguridad de diferentes formas dependiendo de si las políticas locales permiten o no la instalación de software. Si no se permite, BackupPC utilizar parte de la suite Samba (<http://www.samba.org>) para hacer backup de la información compartida mediante SMB o CFIS. Si se permite instalar software, se utiliza rsync junto con el conjunto de herramientas Cygwin (<http://www.cygwin.com>).

- **Soporte para herramientas nativas.** BackupPC utiliza las herramientas estándar de Unix para su funcionamiento interno. Esto incluye programas como Perl, tar, rsync, compress, gzip, bzip2, zip, apache y samba.

BackupPC no utiliza una base de datos o catálogo para almacenar la información de respaldo. En su lugar, utiliza el árbol de directorios para almacenar esta información. Esto simplifica las actualizaciones del sistema operativo del servidor de BackupPC o de la propia aplicación BackupPC.

- **Control de los backups y restauraciones a través de interfaz web.** La Web es la interfaz principal de BackupPC. Tras la configuración inicial, no es necesario acceder al servidor mediante línea de comandos para administrar BackupPC. La interfaz web está escrita en Perl y fue diseñada para funcionar tanto con mod\_perl como con CGIs o con Perl setuid.

La interfaz permite a los usuarios identificarse, acceder y controlar los respaldos y las restauraciones.

El usuario puede solicitar copias de seguridad de tipo one-time, de tipo

completa, o de tipo incremental.

Se pueden utilizar varias opciones para recuperar ficheros:

- o Los archivos individuales se recuperan mediante selección.
- o Los grupos de archivos o directorios se pueden restaurar a su ubicación original.
- o El usuario puede descargar los archivos como un archivo tar o zip.

El usuario tiene control absoluto sobre qué archivos o directorios se restauran y donde hay que restaurarlos. Un histórico muestra que archivos se han modificado durante cada copia de seguridad en cada directorio.

- **Soporte para clientes DHCP.** Los clientes BackupPC se referencian por nombre de host. Si la red de la copia de seguridad utiliza DHCP y se permite la resolución de nombres dinámica, no hay que hacer nada más para que el servidor BackupPC respalde a los clientes DHCP. Si este no es el caso, y los clientes son máquinas Windows, BackupPC se puede configurar para buscar un conjunto de direcciones de los clientes, localizándolos mediante SMB.

Si el cliente no está en línea durante el período de copia de seguridad normal, el servidor BackupPC no genera un error a menos que haya transcurrido un período de tiempo establecido desde la última copia de seguridad. En este punto, el servidor envía un email al propietario del cliente y le recuerda que se asegure que la máquina está en la red para hacer una copia de seguridad. (El servidor también puede enviar cualquier error al administrador.)

Los clientes que residen en otra LAN pueden ser gestionados a nivel local asumiendo que hay conectividad de entre las redes. Esto significa que se puede hacer backup de los clientes conectados a través de una red

privada virtual (VPN).

Si el usuario no desea realizar copias de seguridad en un momento dado, se conectaría a través de la interfaz web para cancelar la copia de seguridad.

- **Pool de Backups.** Cuando los clientes utilizan el mismo sistema operativo se duplican los archivos respaldados. Si se quiere mantener múltiples copias de seguridad completas aumenta el número de archivos duplicados, lo que aumenta los requisitos de capacidad de almacenamiento para el servidor. BackupPC almacena un árbol de directorios por cliente respaldado, pero comprueba si los archivos se han almacenado antes. Si es así, BackupPC utiliza un enlace que apunta al fichero existente en el conjunto de discos común, ahorrando una gran cantidad de espacio. Además, BackupPC puede comprimir opcionalmente para ahorrar más espacio.
- **Fácil configuración por cliente.** Una vez que el administrador haya definido cuáles deberían de ser las políticas de backup del sitio, es muy fácil anular cualquier opción de configuración en base a un cliente. Esto permite una gran flexibilidad sobre qué, cuándo, y cómo hacer copia de seguridad de un cliente. No hay clases de clientes por sí mismo.

#### *4611.2.3 Herramientas de código abierto – Bacula*

Bacula es un conjunto de programas Open Source, listos para ser utilizados en un entorno doméstico y profesional, que permiten administrar los backups, restauración y verificación de datos en una red heterogénea. Bacula es relativamente fácil de usar y eficiente, a la vez que ofrece muchas funcionalidades avanzadas para la administración de los datos almacenados, lo cual facilita hallar y recuperar archivos perdidos o



dañados. En términos técnicos, Bacula es un sistema de backups Open Source, orientado a la red y listo para la empresa.

Es capaz de realizar copias de seguridad en disco, cinta o medios ópticos. Bacula fue escrita originalmente por John Walker y Kern Sibbald en el año 2000. John dejó el proyecto no mucho tiempo después de su creación, y Kern, trabajó en él desde mediados del 2000 hasta el primer lanzamiento público de Bacula en abril de 2002. Desde entonces, otros desarrolladores han contribuido a su desarrollo.

Bacula está disponible bajo licencia AGPL versión 3. La página web del proyecto se encuentra en <http://www.bacula.org>, y los archivos descargables y un repositorio CVS se alojan en SourceForge.

### **Bacula Arquitectura**

Bacula es una solución distribuida de backups. Esto significa que Bacula está compuesto por varios elementos, que pueden o no residir en el mismo host. Por ejemplo, se puede tener un host con el catálogo y en otro el storage.

Se basa en una arquitectura Cliente-servidor que resulta eficaz y fácil de manejar, dada la amplia gama de funciones y características que brinda: copiar y restaurar ficheros dañados o perdidos. Además, debido a su desarrollo y estructura modular, Bacula se adapta tanto al uso personal como profesional.

Se puede utilizar TLS (Transport Layer Security) para proteger los datos durante la transmisión.

Los componentes principales de esta arquitectura son:

- **Director (DIR)** es el encargado de gestionar de forma centralizada la lógica de los procesos de backup y los demás servicios. Trabaja en base a una unidad básica denominada JOB (un cliente, un conjunto de archivos, ...)

de tal forma que el Director planifica, inicia y supervisa todos los jobs.

También es el encargado de mantener el catálogo, por lo que el servidor de la base de datos debe estar accesible desde la máquina que ejecuta el Director.

- **Storage** es el encargado de gestionar los dispositivos de almacenamiento; esto exige que esté instalado en la máquina que posea la conexión física a los dispositivos de almacenamiento, tales como: discos locales, grabadoras, unidades de cinta, volúmenes NAS o SAN, autocargadores o librerías de cinta.
- **File Daemon** es el agente que corre del lado del cliente, es decir, en la máquina cuyos datos se van a respaldar, y que tiene como objetivo empaquetar los datos y enviarlos al Storage, donde serán almacenados.
- **Consola** es la herramienta que permite al usuario o administrador controlar Bacula. Se comunica con el director vía red, iniciando los jobs, revisando la salida del job, haciendo consultas y modificaciones en el catálogo.

Existen consolas en modo texto, modo GUI para Windows y Linux/UNIX e interfaces web.

- **Catálogo** es una base de datos donde se guarda información sobre los jobs y sobre los datos respaldados. El catalogo permite dos cosas:
  - o Por un lado, como guarda información de los jobs, pools y volúmenes, Bacula lo usa para saber si hay un backup completo para un job, y si no lo hay, realizará para ese backup una copia completa.
  - o Por otro lado, el catálogo tiene todos los nombres de archivo (y sus atributos, como fecha de última modificación, etc.) que se respaldaron, y eso es lo que permite hacer una recuperación selectiva, es decir, seleccionar (marcar, en la jerga de Bacula) individualmente qué

archivos y/o directorios restaurar.

#### 4611.2.4 Software Propietario CommVault Simpana

Simpana comenzó como un proyecto dentro de AT & T Labs en 1987 y posteriormente fue adquirido por la empresa CommVault.

Simpania es un software de backup que realiza copias de seguridad de entornos Unix, Windows, Linux, servidores de correo Exchange, Lotus Notes, bases de datos Oracle, MySQL, SQLServer y máquinas virtuales VMware. Además permite funciones avanzadas como puede ser el archivado, la deduplicación y la replicación de ficheros.

El funcionamiento de la aplicación se basa en el uso de los bloques de disco, por lo que todos los módulos no utilizan la información del archivo, si no que trabaja a más bajo nivel. Con ello consigue mejores ratios de compresión y una importante reducción de la ventana de backup, al utilizar únicamente los bloques modificados y no el fichero entero para realizar estas operativas.

Otra característica que incide en el uso de almacenamiento de bajo coste es la capacidad de generar políticas como las de archivado, mediante las que automáticamente permite mover ficheros de un almacenamiento a otro con mayor capacidad a menor coste. De esta forma, por ejemplo se podrían pasar los datos de una cabina de fibra a otra con discos SATA, pudiendo llegar a un tercer nivel a cinta, en base a unos requisitos (fecha del archivo, último acceso al archivo, etc.). Todos los movimientos se realizan de forma transparente para el usuario, tanto en el archivado como en su recuperación (si fuese necesario).

A estas funcionalidades hay que sumar la capacidad de deduplicación, que realiza una compresión de los datos aprovechando las duplicidades de los datos a nivel de bloque, consiguiendo alcanzar ratios de hasta el 50% de ahorro en el uso de almacenamientos en datos de segundo nivel y hasta el 90% en los de tercer nivel.

Para terminar el repaso a las principales funcionalidades, la replicación, permite la utilización de snapshots a nivel de cabina permitiendo volver el almacenamiento replicado a un estado anterior o montar la imagen snapshot como un recurso compartido.

Todo se administra desde una única consola centralizada, que simplifica toda la administración de la plataforma. Adicionalmente el motor de búsqueda ofrece la opción de buscar rápidamente y recuperar datos sin necesidad de saber donde se ubican.

#### 4611.2.5 Software Propietario Symantec NetBackup

Symantec NetBackup es actualmente el titular de la mayor cuota de mercado del entorno de software de copia de seguridad.

Netbackup 7 es la nueva versión de la solución de copia de seguridad y recuperación de datos orientada a grandes corporaciones. Esta herramienta trata de simplificar la gestión de la información reduciendo el volumen de almacenamiento de datos con técnicas de deduplicación en los ordenadores cliente de la red además del propio servidor, ofreciendo protección para entornos virtualizados. Todo ello con el único propósito de agilizar los procesos de backup y recuperación de datos.

La nueva herramienta incluye eliminación de datos duplicados nativos dentro del cliente NetBackup y permite a los clientes multiplicar por diez la velocidad de las copias de seguridad en oficinas remotas, el propio centro de datos y los entornos virtuales. Esta eliminación de datos duplicados en el cliente y en el destino ofrece una mayor cobertura con menos herramientas.

El proceso de deduplicación se contempla para todos los sistemas físicos y virtuales, independientemente del método de copia de seguridad. De este modo se integra una mayor protección para los cada vez más extendidos entornos virtualizados bajo las plataformas Hyper-V y VMware. Es en el caso de esta última en la que se ha podido observar un incremento de

velocidad de hasta el 50% a la vez que disminuye el volumen de almacenamiento necesario en un 40%.

Otro de los aspectos notablemente mejorados en Netbackup 7 es la velocidad de recuperación de datos ante desastres. Permitiendo la restauración de grandes volúmenes de información en pocos segundos desde cualquier lugar y punto en el tiempo. Esta gestión se facilita al administrador de TI mediante un sistema centralizado de supervisión y alerta, que integra la administración de varios dominios de archivos con sus respectivas políticas de salvaguarda de datos.

La tecnología incluida en NetBackup acelerará la transición a un entorno virtual para las organizaciones empresariales que instalen un gran número de máquinas virtuales o que decidan crear una infraestructura de nube privada.

La solución NetBackup también ofrece una elaboración de informes simplificada y un mayor soporte a las aplicaciones de bases de datos de Oracle y MySQL.

Algunas de las prestaciones y beneficios incluidos en la última versión de la herramienta son:

- La tecnología Virtual Machine Intelligent Policy incorpora la automatización a la localización y la protección de máquinas virtuales y minimiza los esfuerzos de administración necesarios para realizar copias de seguridad de máquinas virtuales VMware de alto rendimiento.
- Un 50% más de rapidez en copias de seguridad de máquinas virtuales gracias a que la tecnología Granular Recovery Technology (GRT) se encuentra ahora disponible para sistemas Linux en entornos VMware. Esto permite a los clientes reducir los tiempos comparables de copias de seguridad de máquinas virtuales en un 50%, además de simplificar la administración y mejorar la velocidad de recuperación de

archivos individuales.

- Recuperación “a la carta” desde cualquier lugar con la nueva tecnología de replicación de imagen que permite a los clientes que replican datos entre múltiples sitios o dominios de NetBackup realizar backup de datos en un sitio alternativo.
- Recuperación acelerada: NetBackup RealTime ofrece soporte a entornos VMware para eliminar el espacio de tiempo entre copias de seguridad, además de reducir el impacto para grandes hosts de VMware y permitir la recuperación casi instantánea de sistemas completos.
- Satisfacer los requisitos normativos y de cumplimiento para seguimiento de auditorías.
- Incorpora informes mejorados de las políticas del ciclo de vida del almacenamiento, del seguimiento de las auditorías y del estado de las licencias.
- Deduplicación para Oracle mejorando el rendimiento de las copias de seguridad.
- Se añade un nuevo agente que presta soporte a MySQL para centralizar y automatizar las copias de seguridad y la recuperación de datos de las bases de datos de MySQL.
- Actualización simplificada de clientes con LiveUpdate que permite mejoras en equipos cliente para UNIX, Linux y Windows respecto a la versión NetBackup 6.5 y posterior desde una política única controlada por el administrador de NetBackup.

## **4612 ESTRATEGIAS DE BACKUP A DISCO**

Las estrategias de backup definen el plan que se ha de seguir para garantizar la integridad de la información. Los motivos por los que se debe establecer una correcta estrategia antes de comenzar a realizar las copias de seguridad pueden ser muy diversos, pero en esencia se trata de determinar la mejor manera para asegurar la información teniendo en cuenta las posibles dificultades de recuperación de parte de los datos, el coste de los medios que se emplearan y el tiempo que se necesitara.

Como no todos los sistemas son iguales, no todas las estrategias de backup son adecuadas para todos los sistemas. Partiendo de unas características comunes, algunas de las propiedades básicas de una estrategia backup son:

- **Tiempo de almacenamiento.** Define el tiempo máximo que una copia permanece almacenada en un dispositivo. Al finalizar este tiempo la copia puede cambiar de dispositivo o ser borrada para liberar espacio en el medio de almacenamiento y poder hacer uso del mismo.
- **Almacenamiento alternativo.** Posibilita realizar una o varias copias de seguridad en una ubicación externa al sistema y a la localización geográfica del mismo, manteniéndola durante un elevado período de tiempo, aumentando la seguridad ante cualquier catástrofe, ya sea a nivel de software o de hardware.
- **Protección ante fallo de los dispositivos.** Establece el número de medios que se emplean. Cuanto mayor es el número de medios utilizados, mayor es la seguridad contra posibles pérdidas de información producidas por un fallo en el dispositivo de almacenamiento.

- Tiempo de restauración. Esta característica especifica el tiempo de regeneración del sistema en caso de producirse algún fallo.
- El coste. Suele ser un factor determinante a la hora de seleccionar la estrategia a realizar.

Las estrategias para la realización de copias de seguridad pueden ser muy distintas, dependiendo del sistema en cuestión sobre el cual se realizan.

En algunos casos, solamente se efectúa un backup de todo el contenido. Esto se produce cuando por algún motivo especial y muy específico o por algún motivo técnico, cuestiones de tiempo o por que existe un elevado riesgo para los datos. Alguno de estos casos especiales pueden ser:

- No disponer del software original.
- Desconocimiento de la ubicación de los ficheros de configuración.
- Cambiar un disco de almacenamiento rígido.
- Realizar cambios en las particiones de uno o más discos de almacenamiento rígidos.

Es habitual que este tipo de situaciones concretas se produzcan a la hora de llevar a cabo tareas de reparación o actualización sobre sistemas no controlados.

Cuando se trata de cubrir alguno de estos casos la estrategia de backup a seguir es sencilla, realizar un resguardo o copia de seguridad de todo el contenido de las unidades involucradas para así garantizar que no se perderá ninguna información y que será posible realizar la restauración completa del sistema.

Por otro lado, cuando realmente se ha de diseñar un plan estratégico para la realización de las copias de seguridad de un sistema propio o de una organización externa, se deben tener en cuenta una serie de pautas que



ayudan a que el plan estratégico de backups sea el más conveniente y conseguir la mejor relación coste/beneficio posible.

Estas pautas aportan una reducción en el tiempo de respuesta a la hora de realizar una recuperación en caso de que se produzca cualquier tipo de contingencia.

Al intentar definir un plan de backups, surgen una serie de dudas:

***¿Qué datos se deberían resguardar en cada backup?*** Datos a resguardar.

Es un factor determinante para una estrategia de backup que se determine el grado o grados de importancia de la información, es decir, establecer que información resulta de mayor valor para la organización. No tienen la misma transcendencia un documento de trabajo que una copia de respaldo de la configuración de una aplicación.

***¿Cada cuánto se debería efectuar un backup de los datos?*** Frecuencia del backup.

Para determinar la periodicidad con la que se deben realizar las copias de seguridad no existe un criterio claramente definido. Sin embargo si se tienen en cuenta factores como:

- Tiempo empleado en la creación de la información.
- Coste invertido en la creación de la información.
- Posibles consecuencias derivadas de su pérdida.

***¿Cuánto tiempo deberían permanecer guardadas las copias de seguridad?*** Tiempo de Almacenamiento.

El período máximo de tiempo de estancia de una copia de seguridad en un dispositivo, es decir, el tiempo de retención, está directamente relacionado con los medios de almacenamiento disponibles, y por consiguiente por el

presupuesto de la estrategia de backup.

Otra de las decisiones importantes a tomar durante la elaboración de una estrategia para la realización de copias de seguridad es la de seleccionar y planificar los distintos tipos de copias de seguridad.

Los backups son copias exactas de la información. Se pueden definir como instantáneas de los datos en un momento determinado, almacenados en un formato estándar, se puede realizar un seguimiento a lo largo de su periodo de utilidad y con cada nueva copia se mantiene la independencia con copia inicial. Se pueden crear múltiples niveles de backups, siendo los principales:

- **Copias de seguridad completas (Full backups):** representan una copia exacta en un momento dado, de los datos que se pretende proteger. Proporcionan la base para todos los demás niveles de backup.
- Por otro lado, están dos niveles de backup que capturan únicamente los cambios realizados sobre una copia de seguridad completa.

o **Copia de seguridad diferencial**, también conocida como la *copia de seguridad incremental acumulativa*, captura copias de seguridad que se han producido desde el último backup completo y suele utilizarse en entornos en los que no se produce un elevado número de cambios. La copia de seguridad diferencial se debe utilizar con cuidado debido a que puede crecer con rapidez e igualar e incluso superar el tamaño de la copia de seguridad completa.

La ventaja de utilizar las copias de seguridad diferenciales viene dada en el momento de la restauración puesto que en el momento de restaurar una copia de seguridad diferencial sólo se necesita el backup completo y la última copia diferencial realizada. Debido a que únicamente se precisan dos imágenes para la restauración,

la probabilidade de que ambas imaxes sufran algún percance, perda, corrupción, etc., se reduce significativamente.

o ***Copia de seguridad incremental***, es capaz de capturar los cambios que se han producido desde la última copia de seguridad realizada, independientemente del tipo que sea. Es la forma más utilizada para la realización de copias de seguridad, evidentemente combinada con una copia de seguridad completa.

Este tipo de copia de seguridad contiene la menor cantidad de datos necesarios durante cada ciclo de backup, reduciendo la cantidad de datos que se transfieren y el tiempo que se necesita para la creación de una copia de seguridad.

Sin embargo las copias de seguridad incrementales tienen aspectos negativos. Si se está recuperando un grupo de archivos de un conjunto de copias de seguridad completas e incrementales, es probable que se requieran más de dos imágenes de copias de seguridad diferentes para completar la restauración, lo que aumenta la probabilidad de que alguna de estas partes sufra algún tipo de problema y no se pueda completar la restauración.

#### **4613 REPLICACIÓN LOCAL Y REMOTA, ESTRATEGIAS DE RECUPERACIÓN**

La replicación es el proceso de creación de una copia exacta de los datos. La creación de una o varias réplicas de los datos de producción es una de las maneras de proporcionar continuidad al del negocio (BC).

Estos modelos pueden ser utilizados para operaciones de recuperación y reinicia de los sistemas en caso de que se produzca una pérdida de datos.

Una réplica ha de proporcionar:

- **La capacidad de recuperación:** permite la restauración de los datos de los volúmenes de producción en caso de que se produzca una pérdida de los datos. Se ha de proporcionar un mínimo de y RTO y un RPO concreto que nos garanticen la reanudación de las operaciones comerciales en los volúmenes de producción.
- **La capacidad de reinicio:** garantiza la coherencia de los datos de la réplica, posibilitando la reanudación de las operaciones de negocio utilizando para ello la información contenida en las réplicas.

La replicación se pueden clasificar en dos grandes categorías: **locales y remotos**

### **4613.1      *Replicación Local***

La replicación local hace referencia al proceso creación de réplicas dentro del mismo array de discos o el mismo centro de datos.

#### **4613.1.1      Tecnologías de replicación local**

Las replications Host-based (basadas en replicación en host local) y Storage-based (basadas en almacenamiento) son las dos principales tecnologías adoptadas para la replicación local. La replicación de archivos del sistema y la replicación basada en LVM son ejemplos de la tecnología Host-based de replicación local. La replicación de almacenamiento basada en matrices de disco puede llevarse a cabo con soluciones distintas, la duplicación de todo el volumen, la replicación pointer-based de todo el volumen, y la replicación basadas en punteros y virtual.

##### **4613.1.1.1 Basada en replicación en host local**

En este tipo de replicación, los administradores del sistema llevan a cabo el proceso de copia y restauración en la propia máquina, pudiendo basarse la recuperación en una replicación integral del volumen mediante LVM (Logical Volume Manager), o bien mediante instantáneas del sistema de ficheros.

- Replicación del volumen mediante LVM: El LVM se encarga de crear y controlar el volumen de host a nivel lógico y está formado por tres componentes: los discos físicos, los volúmenes lógicos y los grupos de volúmenes. En la replicación de volúmenes basado en LVM, cada partición lógica en un volumen se asigna a dos particiones físicas en dos discos diferentes. De esa forma se consigue un espejo que permite redundancia y recuperación directa en caso de necesitar replicar.
- Instantánea de archivos del sistema: Consiste en crear una réplica a base de instantáneas del sistema de ficheros mediante la utilización de metadatos almacenados en un mapa de bits. Estos metadatos van

reflejando el cambio que se va produciendo en el sistema de ficheros y van almacenando un registro de las direcciones accedidas mediante operaciones de lectura/escritura. Este sistema requiere de una fracción del espacio utilizado por el sistema de ficheros original.

#### **4613.1.1.2 Basada en arrays de discos**

En este tipo de replicación se hace uso de matrices de discos que pueden estar distribuidas dentro del CPD. El entorno operativo es el que lleva a cabo el proceso de replicación de un determinado sistema de ficheros, sin necesidad de que los recursos de acogida (CPU y memoria) del anfitrión intervengan en el proceso de replicación.

### ***4613.2 La replicación remota***

La replicación remota consiste en el proceso de creación de réplicas del conjunto de datos en lugares con otra ubicación física. Las réplicas remotas ayudan a las organizaciones a mitigar los riesgos asociados a las interrupciones regionales del servicio, que pueden estar provocadas por diferentes causas, por ejemplo, desastres naturales. La infraestructura en la que los datos se almacenan inicialmente se llama fuente. La réplica, o infraestructura remota en la que se almacena la copia se le llama blanco.

#### ***4613.2.1 Tecnologías de replicación remota***

La más habitual es la tecnología de replicación basada en host remoto, que utiliza uno o más componentes de la máquina para realizar y gestionar la operación de replicación. Existen dos enfoques fundamentales para la replicación basada en host remoto: Replicación remota basada en LVM y replicación de bases de datos a través de trasvase de registros.

#### **4613.2.1.1 Replicación remota basada en LVM**

En este modelo, la replicación se efectúa y gestiona a nivel de grupo de volúmenes. El LVM de la máquina origen es el encargado de gestionar y transmitir la información del volumen al LVM de la máquina remota. El LVM de la máquina remota se encarga de recibir los datos y realiza la operación de réplica del volumen.

Antes del inicio de la replicación, se deben configurar los sistemas fuente y remoto para que los sistemas de archivos, los volúmenes y la agrupación de volúmenes sea idéntica en ambos. El punto de partida, o sincronización inicial, se puede realizar de diferentes formas, siendo la más frecuente la restauración en el punto remoto de una copia de seguridad de los datos de origen.

En la replicación remota basada en LVM se soportan dos modos de transferencia de datos, que son el síncronico y el asíncronico. En el modo asíncrono, las operaciones de escritura se van almacenando en una cola de registros gestionada por el LVM y se van enviando al host remoto en el orden en el que son recibidas. En caso de fallo de la red, las operaciones siguen acumulándose en la cola de registros.

En la replicación síncrona, las operaciones de escritura deben estar comprometidas tanto en origen como en destino. Las operaciones de escritura consecutivas no pueden ocurrir en fuente ni destino hasta que las operaciones previas hayan finalizado. Esto garantiza que los datos de la fuente y destino son exactamente los mismos en todo momento. Esto hace posible que el RPO en caso de fallo sea cero o cercano a cero. Sin embargo, como contraprestación al nivel de seguridad, el tiempo de respuesta es mucho mayor. El grado de impacto en el tiempo de respuesta depende de la distancia entre ambos sitios (fuente y destino), del ancho de banda disponible y de la infraestructura de conectividad de red.

#### **4613.2.1.2 Basada en trasvase de registros**

La replicación de bases de datos a través de trasvase de registros consiste en la captura de las transacciones realizadas en la base de datos fuente, que son almacenadas en registros que se transmiten periódicamente de un host fuente a un host destino. El host destino recibe el conjunto de registros y realiza las operaciones oportunas en la base de datos replicada. El proceso inicial de producción y reproducción requiere que todos los componentes importantes de la base de datos se repliquen en el sitio remoto.

Los sistemas gestores de bases de datos permiten definir un intervalo de tiempo para el envío de los ficheros de registro, o bien configurar un tamaño predeterminado de los mismos. Cuando un registro supera el intervalo de tiempo establecido o alcanza su tamaño máximo, se cierra, y se abre un nuevo fichero para registrar las transacciones. Los registros cerrados van siendo enviados desde la fuente al destino garantizando que la base de datos replicada en destino sea consecuente con la fuente hasta el último registro cerrado. El RPO en el sitio remoto dependerá del tamaño del fichero de registro y de la frecuencia de cambio de registro en la fuente.

#### ***4614 REPLICACIÓN LOCAL Y REMOTA, ESTRATEGIAS DE RECUPERACIÓN***

La replicación es el proceso de creación de una copia exacta de los datos. La creación de una o varias réplicas de los datos de producción es una de las maneras de proporcionar continuidad al del negocio (BC).

Estos modelos pueden ser utilizados para operaciones de recuperación y reinicia de los sistemas en caso de que se produzca una pérdida de datos.



Una réplica ha de proporcionar:

- **La capacidad de recuperación:** permite la restauración de los datos de los volúmenes de producción en caso de que se produzca una pérdida de los datos. Se ha de proporcionar un mínimo de  $RTO$  y un  $RPO$  concreto que nos garanticen la reanudación de las operaciones comerciales en los volúmenes de producción.
- **La capacidad de reinicio:** garantiza la coherencia de los datos de la réplica, posibilitando la reanudación de las operaciones de negocio utilizando para ello la información contenida en las réplicas.

La replicación se pueden clasificar en dos grandes categorías: ***locales y remotos***

### **4614.1      *Replicación Local***

La replicación local hace referencia al proceso creación de réplicas dentro del mismo array de discos o el mismo centro de datos.

#### **4614.1.1    Tecnologías de replicación local**

Las replications Host-based (basadas en replicación en host local) y Storage-based (basadas en almacenamiento) son las dos principales tecnologías adoptadas para la replicación local. La replicación de archivos del sistema y la replicación basada en LVM son ejemplos de la tecnología Host-based de replicación local. La replicación de almacenamiento basada en matrices de disco puede llevarse a cabo con soluciones distintas, la duplicación de todo el volumen, la replicación pointer-based de todo el volumen, y la replicación basadas en punteros y virtual.

##### **4614.1.1.1 Basada en replicación en host local**

En este tipo de replicación, los administradores del sistema llevan a cabo el proceso de copia y restauración en la propia máquina, pudiendo basarse la recuperación en una replicación integral del volumen mediante LVM (Logical Volume Manager), o bien mediante instantáneas del sistema de ficheros.

- Replicación del volumen mediante LVM: El LVM se encarga de crear y controlar el volumen de host a nivel lógico y está formado por tres componentes: los discos físicos, los volúmenes lógicos y los grupos de volúmenes. En la replicación de volúmenes basado en LVM, cada partición lógica en un volumen se asigna a dos particiones físicas en dos discos diferentes. De esa forma se consigue un espejo que permite redundancia y recuperación directa en caso de necesitar replicar.
- Instantánea de archivos del sistema: Consiste en crear una réplica a base de instantáneas del sistema de ficheros mediante la utilización de metadatos almacenados en un mapa de bits. Estos metadatos van

reflejando el cambio que se va produciendo en el sistema de ficheros y van almacenando un registro de las direcciones accedidas mediante operaciones de lectura/escritura. Este sistema requiere de una fracción del espacio utilizado por el sistema de ficheros original.

#### **4614.1.1.2 Basada en arrays de discos**

En este tipo de replicación se hace uso de matrices de discos que pueden estar distribuidas dentro del CPD. El entorno operativo es el que lleva a cabo el proceso de replicación de un determinado sistema de ficheros, sin necesidad de que los recursos de acogida (CPU y memoria) del anfitrión intervengan en el proceso de replicación.

#### **4614.1.2 La replicación remota**

La replicación remota consiste en el proceso de creación de réplicas del conjunto de datos en lugares con otra ubicación física. Las réplicas remotas ayudan a las organizaciones a mitigar los riesgos asociados a las interrupciones regionales del servicio, que pueden estar provocadas por diferentes causas, por ejemplo, desastres naturales. La infraestructura en la que los datos se almacenan inicialmente se llama fuente. La réplica, o infraestructura remota en la que se almacena la copia se le llama blanco.

#### **4614.1.3 Tecnologías de replicación remota**

La más habitual es la tecnología de replicación basada en host remoto, que utiliza uno o más componentes de la máquina para realizar y gestionar la operación de replicación. Existen dos enfoques fundamentales para la replicación basada en host remoto: Replicación remota basada en LVM y replicación de bases de datos a través de trasvase de registros.

##### **4614.1.3.1 Replicación remota basada en LVM**

En este modelo, la replicación se efectúa y gestiona a nivel de grupo de

volúmenes. El LVM de la máquina origen es el encargado de gestionar y transmitir la información del volumen al LVM de la máquina remota. El LVM de la máquina remota se encarga de recibir los datos y realiza la operación de réplica del volumen.

Antes del inicio de la replicación, se deben configurar los sistemas fuente y remoto para que los sistemas de archivos, los volúmenes y la agrupación de volúmenes sea idéntica en ambos. El punto de partida, o sincronización inicial, se puede realizar de diferentes formas, siendo la más frecuente la restauración en el punto remoto de una copia de seguridad de los datos de origen.

En la replicación remota basada en LVM se soportan dos modos de transferencia de datos, que son el síncrono y el asíncrono. En el modo asíncrono, las operaciones de escritura se van almacenando en una cola de registros gestionada por el LVM y se van enviando al host remoto en el orden en el que son recibidas. En caso de fallo de la red, las operaciones siguen acumulándose en la cola de registros.

En la replicación síncrona, las operaciones de escritura deben estar comprometidas tanto en origen como en destino. Las operaciones de escritura consecutivas no pueden ocurrir en fuente ni destino hasta que las operaciones previas hayan finalizado. Esto garantiza que los datos de la fuente y destino son exactamente los mismos en todo momento. Esto hace posible que el RPO en caso de fallo sea cero o cercano a cero. Sin embargo, como contraprestación al nivel de seguridad, el tiempo de respuesta es mucho mayor. El grado de impacto en el tiempo de respuesta depende de la distancia entre ambos sitios (fuente y destino), del ancho de banda disponible y de la infraestructura de conectividad de red.

#### **4614.1.3.2 Basada en trasvase de registros**

La replicación de bases de datos a través de trasvase de registros consiste

en la captura de las transacciones realizadas en la base de datos fuente, que son almacenadas en registros que se transmiten periódicamente de un host fuente a un host destino. El host destino recibe el conjunto de registros y realiza las operaciones oportunas en la base de datos replicada. El proceso inicial de producción y reproducción requiere que todos los componentes importantes de la base de datos se repliquen en el sitio remoto.

Los sistemas gestores de bases de datos permiten definir un intervalo de tiempo para el envío de los ficheros de registro, o bien configurar un tamaño predeterminado de los mismos. Cuando un registro supera el intervalo de tiempo establecido o alcanza su tamaño máximo, se cierra, y se abre un nuevo fichero para registrar las transacciones. Los registros cerrados van siendo enviados desde la fuente al destino garantizando que la base de datos replicada en destino sea consecuente con la fuente hasta el último registro cerrado. El RPO en el sitio remoto dependerá del tamaño del fichero de registro y de la frecuencia de cambio de registro en la fuente.

#### **4615 BIBLIOGRAFÍA**

- Windows Server 2008 Hyper-V : kit de recursos. Larson, Robert. Anaya, D.L. 2009
- Grid computing : experiment management, tool integration, and scientific workflows. Prodan, Radu Berlin: Springer, cop. 2007
- Virtualización na Wikipedia: <http://es.wikipedia.org/wiki/Virtualizaci%C3%B3n>
- Green IT: Tecnologías para la Eficiencia Energética en Sistemas TI.

Marisa López-Vallejo, Eduardo Huedo Cuesta y Juan Garbajosa Sopeña.

- Dot-cloud : the 21st century business platform built on cloud computing. Fingar, Peter Tampa (FL) : Meghan-Kiffer Press, cop. 2009
- System & Disaster Recovery Planning. Richard Dolewski
- Information Storage and Management: Storing, Managing, and Protecting Digital Information. G. Somasundaram y Alok Shrivastava.
- Backup & Recovery. W. Curtis Preston y O'Reilly Media.
- Redes de área de Almacenamiento na Wikipedia.  
[http://es.wikipedia.org/wiki/Red\\_de\\_área\\_de\\_almacenamiento](http://es.wikipedia.org/wiki/Red_de_área_de_almacenamiento)
- Cristopher Poelker y Alex Nikitin. Storage Area Networks for Dummies.
- G. Somasundaram, Alok Shirvastava "Information, Storage and Management: Storing, Managin and Protecting Digital Information". John Wiley & Sons. April 06, 2009.
- · Jason Buffington "Data protection for Virtual Data Centers". Sybex. August 02, 2010.
- · Doug Lowe "Networking for Dummies". John Willey & sons. May 29, 2007.

**Autor:** Francisco Javier Rodriguez Martínez

Subdirector de Sistemas Escola Superior Enxeñaría Informática Ourense

Colegiado del CPEIG

