

## **NOTA INFORMATIVA EN RELACIÓN COA PUBLICACIÓN DOS TEMARIOS DOS PROCESOS SELECTIVOS CONVOCADOS POLA XUNTA DE GALICIA NO DIARIO OFICIAL DE GALICIA Nº 142 DE 26 /07/11**

O Diario Oficial de Galicia nº 142, de 26 de xullo de 2011, publica distintas ordes da Consellería de Facenda polas que se convocan diferentes procesos selectivos para o ingreso na Administración autonómica de Galicia.

Cumprindo co compromiso adquirido, a EGAP, continúa coa publicación dos temarios correspondentes aos distintos procesos selectivos convocados formalmente.

Tendo en conta, por unha banda, o volume e complexidade na elaboración dun material didáctico que sirva de referencia básica e, por outra, o interese da EGAP para que os posibles usuarios dispoñan á maior brevidade posible de dito material, a publicación do mesmo na páxina web da Escola (<http://egap.xunta.es>), irase producindo da seguinte maneira:

- 1) **Lexislación**, actualizada e consolidada á data de publicación no DOG do nomeamento do tribunal do proceso (Base II.1 da convocatoria<sup>1</sup>), correspondente aos procesos selectivos para o ingreso no corpo superior da Administración da Xunta de Galicia, subgrupo A1; corpo de xestión da Administración da Xunta de Galicia, subgrupo A2; corpo superior da Administración da Xunta de Galicia, subgrupo A1, escala de sistemas e tecnoloxía da información; corpo de xestión da Administración da Xunta de Galicia, subgrupo A2, escala de xestión e sistemas de información e corpo auxiliar da Xunta de Galicia, subgrupo C2.

A data prevista para dita publicación é de 5 de agosto de 2011.

- 2) **Temarios específicos**, iranse publicando na páxina web da Escola, a medida que os procesos de elaboración e revisión vaian concluíndo.

Sen prexuízo da súa publicación nas dúas linguas oficiais e a fin de facilitar á maior brevidade posible este material aos usuarios, os temas iranse publicando na web no idioma orixinalmente empregado por cada un dos autores.

Para maior información pódense poñer en contacto co servizo de Estudos e Publicacións a través do correo electrónico [temarios.egap@xunta.es](mailto:temarios.egap@xunta.es), e teléfono 881 997 251.

A Escola reitera que os temarios por ela facilitados non teñen carácter oficial, polo que en ningún caso vincularán aos opositores ou aos tribunais; senón que se trata de instrumentos complementarios que servirán de apoio e axuda como textos de referencia pero nunca de forma exclusiva e excluínte.

Santiago de Compostela, 4 de agosto de 2011

---

<sup>1</sup> II. Proceso selectivo.  
II.1. Procedemento de oposición.  
(...)

Teranse en conta as normas de dereito positivo relacionadas co contido do programa que no momento de publicación no DOG do nomeamento do tribunal do proceso contén con publicación oficial no boletín ou diario correspondente.



**10. DIRECCIÓN E XESTIÓN DE  
PROXECTOS. XESTIÓN DA  
INTEGRACIÓN. O PLAN XERAL  
DO PROXECTO. XESTIÓN DO  
ALCANCE. XESTIÓN DO CUSTO.  
ORZAMENTOS. XESTIÓN DO  
TEMPO. TÉCNICAS DE  
PLANIFICACIÓN. XESTIÓN DA  
CALIDADE. PLAN DE  
CALIDADE. CAPACIDADES DO  
XEFE DE PROXECTO. XESTIÓN  
DAS COMUNICACIÓNS. XESTIÓN  
DO RISCO. CONTINXENCIAS.  
XESTIÓN DA  
SUBCONTRATACIÓN E  
ADQUISICIÓNS.**



**Tema 10. Dirección e xestión de proxectos. Xestión da integración. O plan xeral do proxecto. Xestión do alcance. Xestión do custo. Orzamentos. Xestión do tempo. Técnicas de planificación. Xestión da calidade. Plan de calidade. Capacidades do xefe de proxecto. Xestión das comunicacións. Xestión do risco. Continxencias. Xestión da subcontratación e adquisicións.**

## **ÍNDICE**

- 10.1 Dirección e xestión de proxectos
  - 10.1.1 Introducción
  - 10.1.2 Project Management Institute (PMI®)
  - 10.1.3 O Project Management Body of Knowledge (PMBOK®)
    - 10.1.3.1 Introducción
    - 10.1.3.2 Relación do PMBOK® coa dirección das TIC
    - 10.1.3.3 Estrutura do PMBOK®
- 10.2 Xestión da integración do proxecto
  - 10.2.1 O plan xeral do proxecto
- 10.3 Xestión do alcance do proxecto
- 10.4 Xestión do tempo do proxecto
  - 10.4.1 Técnicas de planificación
- 10.5 Xestión dos custos do proxecto
  - 10.5.1 Orzamentos
- 10.6 Xestión da calidade do proxecto
  - 10.6.1 Plan de calidade
- 10.7 Capacidades do xefe de proxecto
  - 10.7.1 Xestión dos Recursos Humanos do proxecto
  - 10.7.2 Capacidades do xefe de proxecto
- 10.8 Xestión das comunicacións do proxecto
- 10.9 Xestión dos riscos do proxecto
  - 10.9.1 Continxencias
- 10.10 Xestión das adquisicións do proxecto



## **10.1 DIRECCIÓN E XESTIÓN DE PROXECTOS**

### **10.1.1 INTRODUCCIÓN**

Un proxecto é unha actividade temporal deseñada para producir un único produto, servizo ou resultado. Un proxecto ten un comezo e un final, así como un alcance e recursos definidos.

Un proxecto é único no sentido de que non é unha operación repetitiva, senón un conxunto específico de operacións deseñadas para lograr un obxectivo. O equipo de proxecto inclúe xente que normalmente non traballa xunta, e en moitos casos pertencen a distintas organizacións e países.

O desenvolvemento de software, a construción dunha ponte ou edificio, o esforzo de reconstrución logo dunha catástrofe natural, todo son proxectos, e deben ser xestionados de forma experta para ser completados a tempo, segundo o presupostado, e coa calidade esixida polas organizacións.

A xestión de proxectos é a aplicación de coñecementos, habilidades e técnicas para executar os proxectos de forma eficaz e eficiente. É algo estratéxico para as organizacións, capacitándoas para aliñar os resultados do proxecto cos obxectivos do negocio —e por iso sendo máis competitivas nos seus mercados.

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

PMI é unha das asociacións profesionais máis grandes do mundo, con medio millón de membros en máis de 185 países. É unha organización sen ánimo de lucro que intenta mellorar a profesión da xestión de proxectos por medio de estándares e certificacións recoñecidas globalmente,



comunidades colaborativas, un extenso programa de investigación, e oportunidades de desenvolvemento profesional.

### Antecedentes

PMI Internacional fundouse en 1969 con socios voluntarios. Durante os anos setenta, PMI desenvolveuse principalmente no campo da enxeñaría, mentres o mundo dos negocios desenvolvía os seus proxectos a través de especialistas da mesma empresa e formaban grupos de traballo chamados “Task Force”. Nos anos oitenta, o mundo dos negocios comezou gradualmente a dirixir os seus esforzos por proxectos.

Durante este tempo, o PMI, a través do comité de estándares e colaboradores (entre eles empresas, universidades, asociacións de profesionais, especialistas e consultores en proxectos) realizou o estudo, avaliación e revisión dos estándares xeralmente aceptados a nivel internacional, dando como resultado os estándares que representan o corpo de coñecementos da Dirección de Proxectos, cuxo título orixinal é “Project Management Body of Knowledge” (PMBOK). En 1987 publicouse a súa primeira edición.

### Estándares

Desde 1987, o PMI encargouse de investigar, recompilar e publicar as boas prácticas xeralmente aceptadas para a maioría dos proxectos. Desde entón, publicou 14 libros de estándares. Un deles, o PMBOK, ten en circulación máis de 2.000.000 de exemplares. Tense acceso a estes estándares como socio do PMI®.

### Misión

A misión do PMI é servir a súa comunidade de asociados e profesionais interesados, desenvolvendo a arte de dirixir e levar á práctica a Dirección de Proxectos como disciplina profesional.

### Obxectivos

Os principais obxectivos do PMI son:



- Promover a dirección de proxectos.
- Compartir a experiencia internacional a través do desenvolvemento de profesionais.
- Desenvolver calidade nos recursos humanos para a dirección de proxectos.
- Compartir os coñecementos xeralmente aceptados que lle dan recoñecemento á profesión.
- Consolidar estándares internacionais.
- Certificación de profesionais en proxectos recoñecidos a nivel mundial.

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

#### **10.1.3.1 INTRODUCCIÓN**

O Project Management Body of Knowledge (PMBOK) é un termo que describe a suma dos coñecementos involucrados na profesión da administración de proxectos.

O coñecemento e as prácticas descritas no PMBOK son aplicables á maioría dos proxectos. Con todo, o equipo administrador do proxecto é sempre o responsable de determinar o que é apropiado para cada proxecto.

PMBOK prové a terminoloxía común da administración de proxectos.

PMBOK describe os métodos e prácticas que se deben ter en consideración desde que se inicia un proxecto ata a súa finalización. A aplicación destas prácticas permitirá levar unha boa xestión do proxecto e manter un maior control, permitíndolle ao Project Manager e ao seu equipo



realizaren proxectos de xeito eficaz e eficiente (en alcance, tempo, custo), así como aseguraren a calidade e transparencia ao longo de toda a vida do proxecto.

A guía do PMBOK é integrada mediante un proceso de desenvolvemento de normas por consenso voluntario. Este proceso reúne a voluntarios e/ou trata de obter as opinións de persoas que teñen interese no tema cuberto por esta publicación. Aínda que PMI administra o proceso e establece regras para promover a equidade no desenvolvemento do consenso, non redacta o documento e non proba, nin avalía, nin verifica de xeito independente a exactitude ou integridade de ningunha información nin a solidez de ningún xuízo contido nela.

### Finalidade do PMBOK

A finalidade principal do PMBOK é identificar, concentrar e publicar as mellores prácticas xeralmente aceptadas na Dirección de Proxectos.

'Xeralmente aceptadas' refírese a que os coñecementos e as prácticas descritos son aplicables á maioría dos proxectos, a maior parte do tempo, e que existe un amplo consenso sobre o seu valor e utilidade.

'Mellores prácticas' refírese a que existe un acordo xeral en que a correcta aplicación destas habilidades, ferramentas e técnicas pode aumentar as posibilidades de éxito dunha ampla variedade de proxectos diferentes.

### Como se organiza o PMBOK?

A base do PMBOK son os procesos, os cales se clasifican por grupos e áreas de coñecemento:



- Grupos de Procesos: modo lóxico de agrupar os procesos de dirección de proxectos, necesarios para calquera proxecto, con dependencias entre eles, e que se levan a cabo na mesma secuencia sempre. Son os seguintes: Iniciación, Planificación, Execución, Seguimento e Control e Peche.
- Áreas de Coñecemento: categoría que agrupa elementos en común. Son 9 en total: Integración, Alcance, Tempo, Custo, Calidade, Comunicacions, Recursos Humanos, Risco e Adquisicións.

### A quen está dirixido o PMBOK?

Os coñecementos contidos no PMBOK proporcionan unha referencia internacional para calquera que estea interesado na profesión da Dirección de Proxectos. Entre eles pódense mencionar: altos executivos, xerentes de programa e xerentes de directores de proxectos, directores do proxecto e outros membros do equipo do proxecto, membros dunha oficina de xestión de proxectos, clientes e outros interesados, consultores formadores, empresas, etc.

### Fortalezas de PMBOK

- ✓ A guía de PMBOK é un marco e un estándar.
- ✓ Está orientada a procesos.
- ✓ Indica o coñecemento necesario para manexar o ciclo vital de calquera proxecto, programa e portafolios a través dos seus procesos.
- ✓ Define para cada proceso as súas entradas, ferramentas, técnicas e informes necesarios.
- ✓ Define un corpo de coñecemento no que calquera industria poida construír as mellores prácticas específicas para a súa área de aplicación.



### Limitacións de PMBOK

- ✓ Complexo para os pequenos proxectos.
- ✓ Ten que ser adaptado á industria da área de aplicación, tamaño e alcance do proxecto, o tempo e o orzamento e as esixencias de calidade.

### Que necesidades satisfai PMBOK?

Polo que respecta ás TI, PMBOK proporciona principalmente elementos para satisfacer as necesidades de administración de proxectos de TI.

Os proxectos son unha forma de organizar as actividades que non poden ser tratadas dentro dos límites operativos da organización. Así mesmo, os proxectos utilízanse para lograr os obxectivos definidos no plan estratéxico da empresa, con independencia de se o proxecto é administrado pola mesma organización, ou se corresponde a un provedor de servizos externo.

Polo anterior, as mellores prácticas integradas en PMBOK axudan ao departamento de TI a lograr os resultados esperados dos proxectos informáticos de forma máis efectiva, garantindo que estes serven como indutores para a consecución dos obxectivos definidos no plan estratéxico.

### **10.1.3.2 RELACIÓN DO PMBOK COA DIRECCIÓN DAS TIC**

Nos seguintes apartados analízase a relación do PMBOK coas diversas normas, modelos, marcos, prácticas, etc. que se poden aplicar ao ámbito da dirección das TIC.



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

Contempla un sistema de administración do desempeño que lles permite ás empresas conduciren a súa estratexia de acordo co planeado mediante a monitorización continua, complementando os indicadores financeiros tradicionais con criterios de medición de desempeño orientados a: “Clientes”, “Procesos Internos” e “Aprendizaxe e Crecemento”. As mellores prácticas definidas en PMBOK serven como referencia para a administración dos proxectos que sustentan a consecución da estratexia da empresa, a cal pode ser definida e levada á práctica mediante Balanced Scorecard.

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

É un compendio de obxectivos de control para a Tecnoloxía de Información que inclúe ferramentas de soporte que lle permiten á administración cubrir a brecha entre os requirimentos de control, os aspectos tecnolóxicos e os riscos de negocio.

As mellores prácticas definidas en PMBOK están relacionadas cos obxectivos de control “Administrar os Investimentos de TI”, “Administrar a Calidade”, “Avaliar e Administrar os Riscos de TI”, “Administrar os Proxectos de TI” e “Aprovisionamento dos Recursos de TI” definidos en COBIT.

### **IT Governance**

É un conxunto de mecanismos utilizados pola administración dunha organización para dirixir e controlar o seu desenvolvemento tecnolóxico, asegurando que as metas do negocio sexan alcanzadas de forma efectiva mediante a detección e control dos riscos asociados. IT Governance pode tomar como referencia as áreas de coñecemento definidas en PMBOK para determinar as habilidades e coñecementos necesarios para a administración efectiva dos proxectos dentro dun ambiente de TI.



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

É un modelo de mellora de procesos de construción de software que pode tomar como referencia PMBOK para administrar os proxectos orientados a mellorar a capacidade e madurez dos procesos involucrados na construción de software.

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

É o marco de referencia para a Xestión de Servizos de TI máis aceptado e utilizado no mundo. Proporciona un conxunto de mellores prácticas en materia de administración de TI extraídas de organismos do sector público e privado que están na vangarda tecnolóxica a nivel internacional.

## **ISO 9000**

É o estándar para establecer sistemas de xestión de calidade máis recoñecido e adoptado no mundo, debido aos beneficios que brinda o uso das súas normas definidas para establecer, documentar, controlar, medir e mellorar os procesos e produtos dentro da organización.

### **10.1.3.3 ESTRUCTURA DO PMBOK**

A Guía do PMBOK está dividida en tres seccións:

**Sección I. Marco Conceptual da Dirección de Proxectos:** proporciona unha estrutura básica para entender a Dirección de Proxectos.

Defínense os termos clave e proporciona unha descrición xeral do resto da Guía do PMBOK. Tamén se describe o Ciclo de Vida do Proxecto e a Organización.

**Sección II. Norma para a Dirección de Proxectos aplicable a un proxecto:** especifica todos os procesos de Dirección de Proxectos que usa o equipo do proxecto para xestionalo.



Descríbense os cinco Grupos de Procesos de Dirección de Proxectos aplicables a calquera proxecto e os procesos de Dirección de Proxectos que compoñen tales grupos.

Un grupo de procesos é un modo lóxico de agrupar os procesos de dirección de proxectos, necesarios para calquera proxecto, con dependencias entre eles, e que se levan a cabo na mesma secuencia sempre. Son os seguintes:

Procesos de Inicio: procesos mediante os cales se leva a cabo a autorización formal para comezar un proxecto.

Procesos de Planificación: procesos que deberán refinar os obxectivos propostos durante o grupo de procesos de Inicio e planificar o curso de acción requirido para lograr os obxectivos e o alcance pretendido do proxecto.

Procesos de Execución: procesos que se despregan para completar o traballo definido no grupo de procesos de Planificación con obxecto de cumprir os requisitos do proxecto.

Procesos de Control: procesos realizados para medir e supervisar regularmente o avance do proxecto, de maneira que se poidan identificar as variacións respecto da planificación e adoptar, cando sexa necesario, as accións correctivas, preventivas e de control de cambios para cumprir cos obxectivos do proxecto.

Procesos de Peche: procesos requiridos para pechar formalmente un proxecto e que aseguran que as experiencias adquiridas durante o proxecto queden rexistradas e a disposición de futuros usos.

**Sección III: Áreas de Coñecemento da Dirección de Proxectos:** organiza os 42 procesos de Dirección de Proxectos en nove Áreas de Coñecemento. A introdución da Sección III describe a lenda dos diagramas



de fluxo de procesos que se usan en cada capítulo de Área de Coñecemento e na introdución de todas as Áreas de coñecemento.

Un equipo de proxectos funciona en 9 áreas de coñecemento cun número de procesos básicos segundo o resumo que presentamos a seguir:

- **Integración.** Desenvolvemento da carta do proxecto, a declaración do alcance e o plan. Dirección, supervisión e control do proxecto.
- **Alcance.** Planificación, definición, creación, verificación e control da estrutura de desagregación de traballo (EDT).
- **Tempo.** Definición, secuenciación, estimación de recursos necesarios e da duración, desenvolvemento e control do cronograma.
- **Custo.** Planificación de recursos, custos estimados, orzamento e control.
- **Calidade.** Planificación da calidade, aseguranza de calidade e control de calidade.
- **Recursos Humanos.** Planificación, contratación, desenvolvemento e administración dos Recursos Humanos.
- **Comunicacións.** Planificación de comunicacións, distribución da información, difusión do desempeño, xestión de *stakeholders* (interesados).
- **Riscos.** Planificación e identificación de riscos, análise de riscos (cualitativa e cuantitativa), planificación da resposta ante riscos (acción), e supervisión e control do risco.
- **Adquisicións.** Plan de contratacións e adquisicións, selección e incentivos dos vendedores, administración e peche de contratos.

Nos seguintes apartados coméntase cada unha das devanditas áreas de coñecemento, describindo cada un dos procesos que as compoñen.

### **3.1 XESTIÓN DA INTEGRACIÓN DO PROXECTO**



Describe os procesos e actividades que forman parte dos diversos elementos da Dirección de Proxectos, que se identifican, definen, combinan, unen e coordinan dentro dos Grupos de Procesos de Dirección de Proxectos. Componse dos procesos:

- Desenvolver a **Acta de Constitución do Proxecto**: autorízase formalmente o proxecto ou unha fase deste, documentándose os requisitos iniciais que satisfagan as necesidades e expectativas dos interesados.
- Desenvolver o **Plan para a Dirección do Proxecto**: documéntanse as accións necesarias para definir, preparar, integrar e coordinar todos os plans subsidiarios.
- **Dirixir e Xestionar** a Execución do Proxecto: lévase a cabo o traballo definido no plan para a dirección do proxecto para así cumprir cos seus obxectivos.
- **Monitorar e Controlar** o Traballo do Proxecto: monitórase, revísase e regúlase o avance para ver as posibles desviacións respecto das liñas base definidas no plan para a dirección do proxecto.
- Realizar o **Control Integrado de Cambios**: revísanse todas as solicitudes de cambio, ademais de aprobar e xestionar os cambios.
- **Pechar** o Proxecto ou unha Fase: finalízanse todas as actividades, dando por completado formalmente o proxecto ou unha fase deste.

#### **10.2.1 O PLAN XERAL DO PROXECTO**

Desenvolver o Plan para a Dirección do Proxecto é o proceso que consiste en documentar todas as accións necesarias para definir, preparar, integrar e coordinar todos os plans subsidiarios. O plan para a dirección do proxecto define o modo en que o proxecto se executa, se monitora, se controla e se pecha. En función da área de aplicación e da complexidade do proxecto, o contido do plan para a dirección do proxecto poderá variar. O



plan para a dirección do proxecto desenvólvese a través dunha serie de procesos integrados ata chegar ao peche do proxecto.

Este proceso dá lugar a un plan para a dirección do proxecto que se elabora de forma gradual por medio de actualizacións, e se controla e aproba a través do proceso Realizar o Control Integrado de Cambios.

### **10.3 XESTIÓN DO ALCANCE DO PROXECTO**

Describe os procesos necesarios para asegurarse de que o proxecto inclúa todo o traballo requirido para completarse satisfactoriamente. Componse dos procesos:

- **Recompilar Requisitos:** documéntanse as necesidades dos interesados para convertelas en requisitos do proxecto.
- **Definir o Alcance:** desenvólvese o enunciado do alcance detallado, o "qué".
- Crear a **Estrutura de Desagregación do traballo** ou EDT: descompoñer o proxecto en partes máis pequenas.
- **Verificar o Alcance:** conseguir a aceptación formal do alcance por parte do cliente ou patrocinador.
- **Controlar o Alcance:** xestionar os cambios no alcance.

No contexto do proxecto, o termo alcance pódese referir a:

- **Alcance do produto:** as características e funcións que definen un produto, servizo ou resultado.
- **Alcance do proxecto:** o traballo que se debe realizar para entregar un produto, servizo ou resultado coas características e funcións especificadas.

### **10.4 XESTIÓN DO TEMPO DO PROXECTO**



Describe os procesos relativos á puntualidade na conclusión do proxecto. Componse dos procesos:

- **Definir** as Actividades: identifícase cada unha das actividades que se deben realizar para lograr un proxecto exitoso.
- **Secuenciar** as Actividades: analízase que tipo de dependencia existe entre as distintas actividades.
- Estimar os **Recursos** das Actividades: determínase cales son os recursos necesarios e dispoñibles para levar a cabo cada actividade.
- Estimar a **Duración** das Actividades: estímase o tempo necesario para completar as actividades.
- Desenvolver o **Cronograma**: analízase a integración existente entre a secuencia, os recursos necesarios, as restricións e a duración de cada actividade.
- **Controlar o Cronograma**: adminístranse os cambios no cronograma.

#### **10.4.1 TÉCNICAS DE PLANIFICACIÓN**

Método da Ruta Crítica: o método da ruta crítica calcula as datas teóricas de inicio e finalización temperás e tardías para cada unha das actividades, sen ter en conta as limitacións de recursos, realizando unha análise que percorre cara a adiante e cara atrás a rede do cronograma.

Método da Cadea Crítica: consiste en modificar o cronograma do proxecto tendo en conta a restrición de recursos.

Nivelación de recursos: modifícase a programación do proxecto para mellorar a eficiencia en canto a asignación de recursos.



Análise “que pasa se”: realízanse simulacións de como cambiaría o cronograma do proxecto se cambiásemos algunha das variables que o afectan.

Aplicación de Adiantos e Atrasos: os adiantos e atrasos son unha técnica de refinamento que se aplica durante a análise da rede para desenvolver un cronograma viable.

Compresión do cronograma: consiste en acurtar o cronograma do proxecto sen modificar o alcance. Dúas das técnicas máis empregadas son:

- Compresión (Intensificación ou *Crashing*): agréganselle máis recursos ao proxecto para acurtar a duración. Polo xeral, esta técnica implicará maiores custos.
- Execución rápida (*fast-tracking*): realízanse actividades en paralelo para acelerar o proxecto, agregándolle riscos a este.

## **10.5 XESTIÓN DOS CUSTOS DO PROXECTO**

Describe os procesos involucrados na planificación, estimación, orzamento e control de custos de xeito que o proxecto se complete dentro do orzamento aprobado. Componse dos procesos:

- **Estimar** os Custos: calcúlanse os custos de cada recurso para completar as actividades do proxecto.
- Determinar o **Orzamento**: súmanse os custos de todas as actividades do proxecto a través do tempo.
- **Controlar** os Custos: inflúese sobre as variacións de custos e adminístranse os cambios do orzamento.

Existen diversos tipos de custos; a seguir mencionaremos os principais.



- **Custos variables:** dependen do volume de produción.
- **Custos fixos:** non cambian co volume de produción.
- **Custos directos:** pódense atribuír directamente ao proxecto.
- **Custos indirectos:** benefician a varios proxectos e normalmente non se pode identificar con exactitude a proporción que lle corresponde a cada un.
- **Custo de oportunidade:** o custo de oportunidade dun recurso é a súa mellor alternativa deixada de lado.
- **Custos afundidos ou enterrados:** custos que xa foron devindicados e non cambiarán coa decisión de facer ou non facer o proxecto.

### **10.5.1 ORZAMENTOS**

Para realizar os orzamentos cóntase coas seguintes ferramentas e técnicas:

#### **Suma de Custos**

As estimacións de custos súmanse por paquetes de traballo, de acordo coa EDT; a continuación vanse sumando os niveis superiores de compoñentes da EDT, tales como as contas de control, e finalmente o proxecto completo.

#### **Análise de Reserva**

A análise de reserva do orzamento pode establecer tanto as reservas para continxencias como as de xestión do proxecto. As reservas para continxencias son asignacións para aqueles cambios que non foron planificados previamente, pero que son potencialmente necesarios. As reservas de xestión son orzamentos reservados para cambios non planificados ao alcance e ao custo do proxecto. As reservas non forman parte da liña base de custo, pero pódense incluír no orzamento total do proxecto.



### **Xuízo de Expertos**

É aquel que se obtén en función da experiencia nunha área de aplicación, unha área de coñecemento, unha disciplina, unha industria, etc., segundo resulte apropiado para a actividade que se está desenvolvendo, e que se debe utilizar para determinar o orzamento. O xuízo de expertos pode proceder de diversas fontes, entre outras:

- outras unidades dentro da organización executante,
- consultores,
- interesados, incluíndo clientes,
- asociacións profesionais e técnicas,
- grupos industriais.

### **Relacións Históricas**

Calquera relación histórica que dea como resultado estimacións paramétricas ou análogas implica o uso de características (parámetros) do proxecto para desenvolver modelos matemáticos que permitan predicir os custos totais do proxecto. Estes modelos poden ser simples (p. ex., construír unha vivenda custará unha certa cantidade por metro cadrado de espazo útil) ou complexas (p. ex., un modelo de custo de desenvolvemento de software utiliza varios factores de axuste separados, con múltiples criterios por cada un deses factores).

## **10.6 XESTIÓN DA CALIDADE DO PROXECTO**

Describe os procesos necesarios para asegurarse de que o proxecto cumpra cos obxectivos para os cales foi emprendido. Componse dos procesos:

- **Planificar a Calidade:** establécese qué normas son relevantes e como se van satisfacer.
- **Asegurar a Calidade:** utilízanse os procesos necesarios para cumprir cos requisitos do proxecto. Dito doutro xeito, asegura que se estean utilizando os plans para a xestión de calidade.



- **Controlar a Calidade:** supervísase que o proxecto estea dentro dos límites pre-establecidos.

En todo proxecto é sumamente importante dedicarlle tempo á xestión de calidade para:

- Previr erros e defectos.
- Evitar realizar de novo o traballo, o que implica aforrar tempo e diñeiro.
- Ter un cliente satisfeito.

A xestión da calidade implica que o proxecto satisfaga as necesidades polas cales se emprendeu. Para iso será necesario o seguinte:

- Converter as necesidades e expectativas dos interesados en requisitos do proxecto.
- Lograr a satisfacción do cliente cando o proxecto produza o planificado e o produto cubra as necesidades reais.
- Realizar accións de prevención sobre a inspección.
- Buscar de forma permanente a perfección: mellora continua.

A xestión moderna da calidade complementa a dirección de proxectos. Ambas as disciplinas recoñecen a importancia de:

- **A satisfacción do cliente.** Entender, avaliar, definir e xestionar as expectativas, de xeito que se cumpran os requisitos do cliente.
- **A prevención antes que a inspección.** Un dos preceptos fundamentais da xestión moderna da calidade establece que a calidade se planifica, se diseña e se integra (e non se inspecciona).



- **A mellora continua.** O ciclo planificar-facer-revisar-actuar é a base para a mellora da calidade, segundo a definición de Shewhart, modificada por Deming.
- **A responsabilidade da dirección.** O éxito require a participación de todos os membros do equipo do proxecto, pero proporcionar os recursos necesarios para lograr este éxito segue sendo responsabilidade da dirección.

### **10.6.1 PLAN DE CALIDADE**

O plan de xestión de calidade describe cómo o equipo de dirección do proxecto levará á práctica a política de calidade da organización executante. É un compoñente ou un plan subsidiario do plan para a dirección do proxecto. O plan de xestión de calidade proporciónalle entradas ao plan xeral para a dirección do proxecto e comprende: o control de calidade, a aseguranza da calidade e métodos de mellora continua dos procesos do proxecto.

O plan de xestión de calidade pode ser redactado de modo formal ou informal, moi detallado ou formulado de maneira xeral. O formato e o grao de detalle determínanse segundo os requisitos do proxecto. O plan de xestión de calidade débese revisar nas etapas iniciais do proxecto, para asegurarse de que as decisións estean baseadas en informacións precisas. Os beneficios desta revisión poden incluír a redución do custo e sobrecustos no cronograma ocasionados polo reproceso.

## **10.7 CAPACIDADES DO XEFE DE PROXECTO**

### **10.7.1 XESTIÓN DOS RECURSOS HUMANOS DO PROXECTO**

A Xestión dos Recursos Humanos do Proxecto inclúe os procesos que organizan, xestionan e conducen o equipo do proxecto. O equipo do proxecto está conformado por aquelas persoas ás que se lles asignaron



roles e responsabilidades para completar o proxecto. Componse dos procesos:

- Desenvolver o **Plan de Recursos Humanos**: defínense os roles, responsabilidades e habilidades dos membros do equipo, así como as relacións de comunicación.
- **Adquirir** o equipo: obtéñense os recursos humanos necesarios para levar a cabo as actividades do proxecto.
- **Desenvolver** o equipo: mellóranse as competencias e as habilidades de interacción entre os membros do equipo.
- **Xestionar** o equipo: monitórase o desempeño individual e de grupo de cada persoa e resólvense os conflitos que adoitan ocorrer entre os membros do equipo.

### **10.7.2 CAPACIDADES DO XEFE DE PROXECTO**

Os directores de proxecto usan unha combinación de habilidades técnicas, humanas e conceptuais para analizaren as situacións e interactuar de xeito apropiado cos membros do equipo. Mediante estas habilidades, os directores de proxecto poderán sacar proveito dos puntos fortes dos membros do equipo.

Algunhas das habilidades interpersoais usadas con maior frecuencia polos directores do proxecto descríbense brevemente a seguir.

**Liderado.** Os proxectos exitosos requiren fortes habilidades de liderado. O liderado é importante en todas as fases do ciclo de vida do proxecto. É fundamental comunicar a visión e inspirar o equipo do proxecto co fin de lograr un alto desempeño.

Existen distintos estilos de **liderado**, como por exemplo:

- Directivo: dicir qué hai que facer.
- Consultivo (*Coaching*): dar instrucións.



- Participativo (*Supporting*): ofrecer asistencia.
- Delegativo (*Empowerment*): o empregado decide por si mesmo.
- Facilitador: coordinar os demais.
- Autocrático: tomar decisións sen consultar.
- Consenso: resolución de problemas grupais.

**Influencia.** Dado que a miúdo a autoridade directa dos directores do proxecto sobre os membros do seu equipo é escasa ou nula nunha organización de tipo matricial, a súa capacidade de influír nos interesados resulta vital para o éxito do proxecto. Entre as habilidades clave de influencia atópanse:

- Ter a habilidade para persuadir e expresar con claridade os puntos de vista e as posicións asumidas.
- Contar con gran habilidade para escoitar de xeito activo e eficaz.
- Ter en conta as diversas perspectivas en calquera situación.
- Recompilar información relevante e crítica co fin de abordar os asuntos importantes e lograr acordos.

**Toma de decisións eficaz.** Isto implica ter a habilidade de negociar e influír sobre a organización e o equipo de dirección do proxecto. Algunhas pautas en materia de toma de decisións inclúen:

- centrarse nos obxectivos perseguidos,
- seguir un proceso de toma de decisións,
- desenvolver as calidades persoais dos membros do equipo,
- fomentar a creatividade do equipo,
- xestionar as oportunidades e os riscos.

## **10.8 XESTIÓN DAS COMUNICACIÓNS DO PROXECTO**



Describe os procesos relacionados coa xeración, distribución, almacenamento e destino final da información do proxecto en tempo e forma. Componse dos procesos:

- **Identificar** os Interesados: identifícanse todas as persoas ou organizacións que dalgún xeito se verán afectadas polo proxecto.
- **Planificar** as Comunicacións: determínanse cales serán as necesidades de información do proxecto.
- **Distribuír** a Información: colócase a información a disposición dos interesados.
- **Xestionar as Expectativas** dos Interesados: satisfáanse os requisitos dos interesados e resólvense os conflitos entre os recursos humanos.
- **Informar** o Desempeño: comunícase o estado de avance do proxecto.

No momento de distribuír a información, hai que ter en conta as distintas dimensións da comunicación:

- Interna: entre as persoas que forman parte do proxecto.
- Externa: cara aos interesados externos do proxecto.
- Vertical: entre xefe-empregado e viceversa.
- Horizontal: entre colegas do proxecto.
- Escrita formal: plans, solicitude, etc.
- Escrita informal: memos, e-mails, notas.
- Oral formal: presentacións.
- Oral informal: reunións, conversas.

A maioría das habilidades de comunicación son comúns á dirección en xeral e á dirección de proxectos. Entre estas habilidades, inclúese:

- escoitar de xeito activo e eficaz,
- formular preguntas, sondar ideas e situacións para garantir unha mellor comprensión,



- educar para aumentar o coñecemento do equipo co fin de que sexa máis eficaz,
- investigar para identificar ou confirmar a información,
- identificar e xestionar expectativas,
- persuadir a unha persoa ou organización para levar a cabo unha acción,
- negociar co fin de lograr acordos entre partes que resulten mutuamente aceptables,
- resolver conflitos para previr impactos negativos.

## **10.9 XESTIÓN DOS RISCOS DO PROXECTO**

Non se debería comezar coa execución do proxecto sen unha análise de risco. A planificación dos riscos é unha área integradora do resto das áreas do coñecemento. Por exemplo, non podemos afirmar que temos un cronograma e orzamento realista se aínda non rematamos a análise de risco. Coa análise de risco determinaranse as reservas para continxencia de prazos e custos que se deben incluír no plan para a dirección do proxecto. Componse dos procesos:

- **Planificar** a Xestión de Riscos: defínense as actividades de xestión dos riscos para un proxecto.
- **Identificar** os Riscos: unha vez realizado o plan de xestión de riscos, é necesario comezar coa identificación dos eventos con risco que, se ocorresen, afectarían ao resultado do proxecto, ben sexa para ben ou para mal.
- Realizar a **Análise Cualitativa de Riscos**: priorizar os riscos para realizar outras análises ou accións posteriores, avaliando e combinando a probabilidade de ocorrencia e o impacto dos devanditos riscos. Para iso pódense usar as seguintes ferramentas e técnicas:



- **Avaliación de probabilidade e impacto:** estímase cal é a probabilidade de ocorrencia e o impacto de cada risco identificado.
- **Matriz de probabilidade e impacto:** táboa de dobre entrada onde se combina a probabilidade e o impacto para poder facer unha priorización dos riscos.
- **Categorización dos riscos:** agrúpanse os riscos por causas comúns.
- Realizar a **Análise Cuantitativa de Riscos:** analízase numericamente o efecto dos riscos identificados sobre os obxectivos xerais do proxecto. Para iso pódense usar as seguintes ferramentas e técnicas:
  - **Distribucións de probabilidade:** uniforme, triangular, beta, normal, log normal, poisson, F, Chi-cadrada, etc.
  - **Valor monetario esperado:** obtense de multiplicar a probabilidade de ocorrencia polo impacto en valor monetario.
  - **Árbore de decisión:** diagrama que describe as implicacións de elixir unha ou outra alternativa entre todas as dispoñibles.
- **Planificar a Resposta** aos Riscos: desenvólvense as accións para mellorar as oportunidades e reducir as ameazas aos obxectivos do proxecto. Para os **riscos negativos** adóitanse utilizar as seguintes estratexias ou ferramentas: evitar, transferir, mitigar ou aceptar. Pola súa banda, para os **riscos positivos** adóitanse utilizar as seguintes estratexias ou ferramentas: explotar, compartir, mellorar, aceptar.
- **Monitorar e Controlar** os Riscos: execútanse plans de resposta aos riscos, rastréxanse os riscos identificados, monitóranse os riscos residuais, identifícanse novos riscos e avalíase a efectividade do proceso contra riscos a través do proxecto.

### 10.9.1 CONTINXENCIAS



Algunhas estratexias están deseñadas para seren usadas unicamente se se presentan determinados eventos. Para algúns riscos, resulta apropiado para o equipo do proxecto elaborar un plan de resposta que só se executará baixo determinadas condicións predefinidas, se se cre que haberá suficientes sinais de advertencia para levar a cabo o plan. Os eventos que disparan a resposta para continxencias, tales como non cumprir con puntos clave intermedios ou obter unha prioridade máis alta cun proveedor, débense definir e rastrexar.

### **Reservas para continxencias**

Para aqueles riscos coñecidos, identificados e cuantificados, pódese estimar unha reserva monetaria para continxencias, que non forma parte da liña base de custo do proxecto. Pola súa banda, os riscos descoñecidos non se poden xestionar de xeito proactivo e poderíanse considerar asignándolle unha reserva de xestión xeral ao proxecto, que non forma parte da liña base de custo, pero si se inclúe no orzamento total do proxecto.

## **3.10 XESTIÓN DAS ADQUISICIÓNS DO PROXECTO**

Describe os procesos para comprar ou adquirir produtos, servizos ou resultados, así como para contratar procesos de dirección. Componse dos procesos:

- **Planificar** as Adquisicións: documéntanse as decisións de compra para o proxecto, especificando cómo facelo e identificando os posibles vendedores.
- **Efectuar** as Adquisicións: obtéñense as respostas dos vendedores, seleccionando os máis vantaxosos para o proxecto e adxudicando os contratos.



- **Administrar** as Adquisicións: xestiónanse as relacións de adquisicións, monitórase a execución dos contratos, e efectúanse cambios e correccións segundo sexa necesario.
- **Pegar** as Adquisicións: complétase cada adquisición para o proxecto.

Os procesos de Xestión das Adquisicións do Proxecto implican a realización de contratos, que son documentos legais que se establecen entre un comprador e un vendedor. Este documento representa un acordo vinculante para as partes en virtude do cal o vendedor está obrigado a prover os produtos, servizos ou resultados especificados, e o comprador debe proporcionar diñeiro ou calquera outra contraprestación válida. Un contrato de adquisición inclúe termos e condicións, e pode incorporar outros aspectos especificados polo comprador para establecer o que o vendedor debe realizar ou proporcionar. É responsabilidade do equipo de dirección do proxecto asegurar que todas as adquisicións cumpren as necesidades do proxecto, ao mesmo tempo que se respectan as políticas da organización en canto a adquisicións se refire.



Bibliografía:

*Guía de los fundamentos de la dirección de proyectos* (Guía do PMBOK), 4ª Edición.

Sitios web:

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

Autor:

Ramón Seoane Freijido

Director de Sistemas de Información Molduras del Noroeste

Colexiado do CPEIG



**11. SISTEMAS DE XESTIÓN DE  
CALIDADE. NORMALIZACIÓN E  
CERTIFICACIÓN. EFQM. SERIE  
ISO  
9000. CMMI.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 11. Sistemas de Xestión de Calidade. Normalización e certificación. EFQM. Serie ISO 9000. CMMI.**

### **ÍNDICE**

#### **11.1 Normalización e certificación**

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

##### **11.1.2 Certificación. Pasos**

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

##### **11.2.1 Introducción ao modelo**

##### **11.2.2 Que é a EFQM?**

##### **11.2.3 Historia da EFQM**

##### **11.2.4 Que é o modelo EFQM?**

##### **11.2.5 Para que serve o modelo EFQM?**

##### **11.2.6 Composición do modelo EFQM**

##### **11.2.7 Vantaxes de adoptar o modelo EFQM**

##### **11.2.8 Cambios que orixina a excelencia en xestión**

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

##### **11.3.1 Introducción**

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

##### **11.3.3 Principios de xestión**

##### **11.3.4 Versións específicas da norma ISO 9000**

##### **11.3.5 Custos e beneficios de establecer un sistema de xestión da calidade**

##### **11.3.6 Posta en práctica dun sistema de xestión da calidade**

#### **11.4 CMMI**

### **11.1 NORMALIZACIÓN E CERTIFICACIÓN.**

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

Que se entende por normalización?



A normalización é unha actividade colectiva encamiñada a establecer solucións a situacións repetitivas.

En particular, esta actividade consiste na elaboración, difusión e aplicación de normas.

A normalización ofrece importantes beneficios, como consecuencia de adaptar os produtos, procesos e servizos aos fins aos que se destinan, protexer a saúde e o ambiente, previr os obstáculos ao comercio e facilitar a cooperación tecnolóxica.

### Que é unha norma?

As normas son documentos técnicos coas seguintes características:

- Conteñen especificacións técnicas de aplicación voluntaria.
- Son elaborados por consenso das partes interesadas:
- Están baseados nos resultados da experiencia e o desenvolvemento tecnolóxico.
- Son aprobados por un Organismo Nacional/Rexional/Internacional de Normalización recoñecido.
- Están dispoñibles ao público.

As normas ofrecen unha linguaxe común de comunicación entre as empresas, a Administración e os usuarios e consumidores, establecen un equilibrio socioeconómico entre os distintos axentes que participan nas transaccións comerciais, base de calquera economía de mercado, e son un patrón necesario de confianza entre cliente e provedor.

### Que vantaxes ofrece a normalización?

a) Para os consumidores:

- Establece niveis de calidade e seguridade dos produtos e servizos.
- Informa das características do produto.



- Facilita a comparación entre diferentes ofertas.
- b) Para os fabricantes:
- Racionalizar variedades e tipos de produtos.
  - Diminúe o volume de existencias en almacén e os custos de produción.
  - Mellora a xestión e o deseño.
  - Axiliza o tratamento dos pedidos.
  - Facilita a comercialización dos produtos e a súa exportación.
  - Simplifica a xestión de compras.
- c) Para a Administración:
- Simplifica a elaboración de textos legais.
  - Establece políticas de calidade, ambientais e de seguridade.
  - Axuda ao desenvolvemento económico.
  - Axiliza o comercio.

### Que se pode normalizar?

O campo de actividade das normas é tan amplo como a propia diversidade de produtos ou servizos, incluídos os seus procesos de elaboración.

Así, normalízanse os Materiais (plásticos, aceiro, papel, etc.), os Elementos e Produtos (parafusos, televisores, ferramentas, tubaxes, etc.), as Máquinas e Conxuntos (motores, ascensores, electrodomésticos, etc.), Métodos de Ensaio, Temas Xerais (ambiente, calidade da auga, regras de seguridade, estatística, unidades de medida, etc.), Xestión e Aseguranza da Calidade, Xestión Ambiental (xestión, auditoría, análise do ciclo de vida, etc.), Xestión de prevención de riscos no traballo (xestión e auditoría), etc.

### Que clases de normas existen?

Os documentos normativos poden ser de diferentes tipos dependendo do organismo que os elaborou.



Na clasificación tradicional de normas distínguese entre:

- Normas nacionais. Son elaboradas, sometidas a un período de información pública e sancionadas por un organismo recoñecido legalmente para desenvolver actividades de normalización nun ámbito nacional. En España estas normas son as normas UNE, aprobadas por AENOR, que é o organismo recoñecido pola Administración pública española para desenvolver as actividades de normalización no noso país (Real decreto 2000/1995).
- Normas rexionais. Son elaboradas no marco dun organismo de normalización rexional, normalmente de ámbito continental, que agrupa un determinado número de Organismos Nacionais de Normalización. As máis coñecidas, inda que non as únicas, son as normas europeas elaboradas polos Organismos Europeos de Normalización (CEN, CENELEC, ETSI), e preparadas coa participación de representantes acreditados de todos os países membros. AENOR é o organismo nacional de normalización español membro de CEN e CENELEC e, polo tanto, a organización a través da cal canalizar os intereses e a participación dos axentes socioeconómicos do noso país na normalización europea.
- Normas internacionais. Teñen características similares ás normas rexionais en canto á súa elaboración, pero distínguense delas en que o seu ámbito é mundial. As máis representativas polo seu campo de actividade son as normas CEI/IEC (Comité Electrotécnico Internacional) para a área eléctrica, as UIT/ITU (Unión Internacional de Telecomunicacións) para o sector das telecomunicacións e as normas ISO (Organización Internacional de Normalización) para o resto. AENOR é o organismo nacional de normalización español membro de ISO e CEI e, polo tanto, a organización a través da cal canalizar os intereses e a participación dos axentes socioeconómicos do noso país na normalización internacional.



### Que é unha norma UNE?

Unha norma UNE é unha especificación técnica de aplicación repetitiva ou continuada cuxa observancia non é obrigatoria, establecida con participación de todas as partes interesadas, que aproba AENOR, organismo recoñecido a nivel nacional e internacional pola súa actividade normativa (Lei 21/1992, do 16 de xullo, de industria).

### Como se elabora unha norma UNE?

A elaboración dunha norma UNE, incluída a adopción de normas europeas, lévase a cabo no seo dos Comités Técnicos de Normalización (CTN) a través das seguintes fases:

- Traballos preliminares (recompilación de documentación, discusión sobre o contido...) previos á toma en consideración dunha nova iniciativa.
- Elaboración do proxecto de norma; inclúe todas aquelas actividades que desenvolve o Comité ata a aprobación dun documento como proxecto de norma, buscando sempre o consenso de todas as partes.
- Información pública no BOE; anuncio da existencia do proxecto de norma, tanto nacional como europea, para que calquera persoa, física ou xurídica, poida remitir as observacións a este que estime oportunas.
- Elaboración da proposta de norma; unha vez superada a fase anterior, e recibidas en AENOR as posibles observacións ao proxecto, o CTN procede ao seu estudo e á aprobación da proposta de norma final, para a súa consideración e adopción por AENOR.



- Rexistro, edición e difusión da norma UNE; publicación da norma UNE por AENOR, notificación ao BOE, promoción e comercialización, a través dos servizos comerciais de AENOR.

### **11.1.2 CERTIFICACIÓN. PASOS.**

O proceso de certificarse con base en ISO 9001, e de manter este status unha vez conseguido, preséntase nos pasos seguintes:

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

As organizacións que desexen obter un certificado, deben presentar unha solicitude ante o organismo de certificación da súa elección. Os aspectos que cómpre considerar ao seleccionar o organismo de certificación inclúen:

- Se a natureza da acreditación do organismo de certificación é aceptable no mercado ao cal a organización desexa exportar.
- A imaxe do organismo de certificación no mercado.
- Cotizacións das tarifas de certificación e auditorías, etc.

#### 2. Preparación para a avaliación

De acordo coa ISO 9001, o primeiro requisito é definir os procesos da organización que afectan á calidade, de maneira que o primeiro paso é que o auditor do organismo de certificación se reúna coa alta dirección da organización, co fin de que aquel obteña unha comprensión clara acerca dos procesos da organización.

#### 3. Auditoría

Os auditores recollen evidencia de conformidade ou non conformidade mediante a observación de actividades, o exame de procedementos/rexistros, observacións das condicións de manexo da empresa, a través de entrevistas cos directores e persoal involucrado da organización, etc. A información recompilada mediante as entrevistas é



verificada ou ensaiada polos auditores mediante a recolección da mesma información doutras fontes, tales como observacións físicas ou medicións realizadas no produto e os seus rexistros relacionados. Os auditores visitan e verifican a conformidade co SGC en todos os departamentos e funcións dentro do alcance do SGC.

#### 4. Non conformidades

A evidencia recollida polos auditores é comparada cos criterios da auditoría (políticas e obxectivos da compañía, manuais, procedementos, instrucións, contratos, regulamentacións, etc.) e os resultados das auditorías, incluídas as non conformidades, se as hai, son aclaradas e presentadas ante a alta dirección ao final da auditoría no sitio, nunha reunión formal coa alta dirección, chamada “Reunión de Peche”. As non conformidades (NC) son clasificadas polos auditores como “maiores” ou “menores”. As “observacións” tamén se rexistran.

#### 5. Outorgamento do certificado ISO 9000

En función das recomendacións do auditor e logo da revisión independente destas recomendacións polo organismo certificador, este expídelle un certificado á organización. O certificado expídese para o alcance específico do negocio e para os produtos ou servizos para os cales a organización puxo en práctica un SGC.

#### 6. Auditorías de seguimento

O certificado outórgase inicialmente por un período de tres anos. Durante este tempo, o organismo de certificación realiza auditorías de seguimento periódicas (unha ou dúas veces ao ano), en datas acordadas mutuamente. O organismo de certificación informa previamente un plan de auditoría de tres anos, en que se indique o alcance de cada auditoría de seguimento. Estas auditorías planifícanse de maneira que todos os aspectos do SGC se auditen nun período de tres anos. Despois dos tres



anos, lévase a cabo unha auditoría de re-certificación usando os pasos 2 e 5 anteriores.

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

### **11.2.1 INTRODUCCIÓN AO MODELO**

O Modelo EFQM de Excelencia é un marco de traballo non-prescritivo que ten nove criterios. Os criterios que fan referencia a un Axente Facilitador tratan sobre o que a organización fai. Os criterios que fan referencia aos Resultados tratan sobre o que a organización logra. Os Resultados son consecuencia dos Axentes Facilitadores.

O Modelo, que recoñece que a Excelencia en todo o referente a resultados e rendemento dunha organización se pode lograr de maneira sostida mediante distintos enfoques, fundaméntase en que:

"Os resultados excelentes con respecto ao Rendemento da Organización, aos Clientes, ás Persoas e á Sociedade lógranse mediante un Liderado que dirixa e impulse a Política e Estratexia, as Persoas da organización, as Alianzas e Recursos, e os Procesos."

### **11.2.2 QUE É A EFQM?**

EFQM (European Foundation for Quality Management ou Fundación Europea para a Xestión da Calidade).

MISIÓN: Ser a forza que impulsa a Excelencia nas organizacións europeas de forma sostida.

VISIÓN: Un mundo en que as organizacións europeas sobresaian pola súa Excelencia.

EFQM é unha organización sen ánimo de lucro cuxo ámbito é Europa e a súa sede está en Bruxelas.

EFQM é o creador e o xestor do premio á Excelencia EEA (EFQM Excellence Award) que recoñece a Excelencia en Xestión nas organizacións.



EFQM é a propietaria do Modelo de Excelencia EFQM e é a encargada de actualizala coas boas prácticas que se están levando a cabo nas organizacións punteiras no tema da Excelencia en Xestión.

### **11.2.3 HISTORIA DA EFQM**

1988 Foi creada a Fundación Europea para a Xestión da Calidade (EFQM), sendo unha organización sen ánimo de lucro formada por 14 organizacións europeas (Bosch, BT, Bull, Ciba-Geigy, Dassault, Electrolux, Fiat, KLM, Nestlé, Olivetti, Philips, Renault, Sulzer e Volkswagen).

1989 Foron establecidos a misión, visión e obxectivos do EFQM e comézanse os traballos de desenvolvemento do Modelo Europeo de Calidade. Ademais, engadíronse outras 53 empresas.

1991 Nace o Modelo de Excelencia EFQM e lánzase o primeiro Premio Europeo de Calidade para empresas.

1992 Preséntase o Premio Europeo de Calidade.

1995 Adáptase o Modelo e lánzase o Premio Europeo para o sector público.

1996 Simplifícase o Modelo e lánzase o Premio Europeo para pemes e unidades operativas.

2003 Actualízase o Modelo de Excelencia.

2005 Lánzase o sistema 2005+ para a presentación de memorias e avaliación para o Premio EFQM á Excelencia (EEA).

### **11.2.4 QUE É O MODELO EFQM?**

O Modelo EFQM de Excelencia é un instrumento práctico que axuda as organizacións a estableceren un sistema de xestión apropiado, medindo en que punto se atopan dentro do camiño cara á excelencia, identificando posibles carencias da organización e definindo accións de mellora.

### **11.2.5 PARA QUE SERVE O MODELO EFQM?**



É un marco que as organizacións poden utilizar para se axudaren a desenvolver a súa visión e as metas para o futuro dun xeito tanxible.

É un instrumento que as organizacións poden utilizar para identificar e entender a natureza do seu negocio, é dicir, as relacións entre os distintos axentes presentes na actividade, e as relacións causa-efecto.

É unha ferramenta que permite establecer unha mesma linguaxe e modo de pensar en toda a organización.

É unha ferramenta de diagnóstico para determinar a saúde actual da organización, detectando puntos de mellora e implantando accións que a axuden a mellorar.

É a base para a concesión do Premio EFQM á Excelencia, isto é, un proceso de avaliación que lle permite a Europa recoñecer as súas organizacións mellor xestionadas e promovelas como modelos de excelencia, dos que as demais organizacións poidan aprender.

### **11.2.6 COMPOSICIÓN DO MODELO EFQM**

O Modelo de Excelencia EFQM é un marco non preceptivo baseado en nove criterios.

Cinco destes son “Axentes Facilitadores” (o que a organización fai; inclúe 24 subcriterios) e catro son “Resultados” (o que a organización logra; inclúe 8 subcriterios). Total: 9 CRITERIOS, 32 subcriterios e 298 áreas que cómpre contemplar.

#### **CRITERIO 1: LIDERADO**

Cómo os líderes desenvolven e facilitan a consecución da misión e a visión, desenvolven os valores necesarios para alcanzar o éxito a longo prazo e implantan todo iso na organización mediante as accións e os comportamentos adecuados, estando implicados persoalmente en asegurar que se desenvolve e implanta o sistema de xestión da organización.

#### **Subcriterios**



- 1a. Desenvolvemento da misión, visión e valores por parte dos líderes, que actúan como modelo de referencia dentro dunha cultura de Excelencia.
- 1b. Implicación persoal dos líderes para garantir o desenvolvemento, implantación e mellora continua do sistema de xestión da organización.
- 1c. Implicación dos líderes con clientes, socios e representantes da sociedade.
- 1d. Reforzo por parte dos líderes dunha cultura de Excelencia entre as persoas da organización.
- 1e. Os cambios na organización son definidos e impulsados polos líderes.

## CRITERIO 2: POLÍTICA E ESTRATEXIA

Cómo implanta a organización a súa misión e visión mediante unha estratexia claramente centrada en todos os grupos de interese e apoiada por políticas, plans, obxectivos, metas e procesos relevantes.

### Subcriterios

- 2a. As necesidades e expectativas actuais e futuras dos grupos de interese son o fundamento da política e estratexia.
- 2b. A información procedente das actividades relacionadas coa medición do rendemento, investigación, aprendizaxe e creatividade son o fundamento da política e estratexia.
- 2c. Desenvolvemento, revisión e actualización da política e estratexia.
- 2d. Comunicación e posta en práctica da política e estratexia a través dun esquema de procesos clave.

## CRITERIO 3: PERSOAS

Cómo xestiona, desenvolve e aproveita a organización o coñecemento e todo o potencial das persoas que a compoñen, tanto a nivel individual, como de equipos ou da organización no seu conxunto; e cómo planifica estas actividades en apoio da súa política e estratexia e do eficaz funcionamento dos seus procesos.



#### Subcriterios

- 3a. Planificación, xestión e mellora dos recursos humanos.
- 3b. Identificación, desenvolvemento e mantemento do coñecemento e a capacidade das persoas da organización.
- 3c. Implicación e asunción de responsabilidades por parte das persoas da organización.
- 3d. Existencia dun diálogo entre as persoas da organización.
- 3e. Recompensa, recoñecemento e atención ás persoas da organización.

#### CRITERIO 4: ALIANZAS E RECURSOS

Cómo planifica e xestiona a organización as súas alianzas externas e os seus recursos internos en apoio da súa política e estratexia e do eficaz funcionamento dos seus procesos.

#### Subcriterios

- 4a. Xestión das alianzas externas.
- 4b. Xestión dos recursos económicos e financeiros.
- 4c. Xestión dos edificios, equipos e materiais.
- 4d. Xestión da tecnoloxía.
- 4e. Xestión da información e do coñecemento.

#### CRITERIO 5: PROCESOS

Cómo diseña, xestiona e mellora a organización os seus procesos para apoiar a súa política e estratexia e para satisfacer plenamente, xerando cada vez maior valor, os seus clientes e outros grupos de interese.

#### Subcriterios

- 5a. Deseño e xestión sistemática dos procesos.
- 5b. Introducción das melloras necesarias nos procesos mediante a innovación, co fin de satisfacer plenamente a clientes e outros grupos de interese, xerando cada vez maior valor.



5c. Deseño e desenvolvemento dos produtos e servizos baseándose nas necesidades e expectativas dos clientes.

5d. Produción, distribución e servizo de atención, dos produtos e servizos.

5e. Xestión e mellora das relacións cos clientes.

#### CRITERIO 6: RESULTADOS NOS CLIENTES

Qué logros está alcanzando a organización en relación cos seus clientes externos.

##### Subcriterios

###### 6a. Medidas de percepción

Refírense á percepción que teñen os clientes da organización, e obtéñense, por exemplo, das enquisas a clientes, grupos focais, clasificacións de provedores existentes no mercado, felicitacións e reclamacións.

###### 6b. Indicadores de rendemento

Son medidas internas que utiliza a organización para supervisar, entender, predicir e mellorar o rendemento, así como para anticipar a percepción dos seus clientes externos.

#### CRITERIO 7: RESULTADOS NAS PERSOAS

Qué logros está alcanzando a organización en relación coas persoas que a integran.

##### Subcriterios

###### 7a. Medidas de percepción

Refírense á percepción da organización por parte das persoas que a integran, e obtéñense, por exemplo, de enquisas, grupos focais, entrevistas e avaliacións de rendemento estruturadas.

###### 7b. Indicadores de rendemento

Son medidas internas que utiliza a organización para supervisar, entender, predicir e mellorar o rendemento das persoas que a integran, así como para anticipar as súas percepcións.



## CRITERIO 8: RESULTADOS NA SOCIEDADE

Qué logros está alcanzando a organización na sociedade.

### Subcriterios

#### 8a. Medidas de percepción

Refírense á percepción da organización por parte da sociedade, e obtéñense, por exemplo, de enquisas, informes, reunións públicas, representantes sociais e autoridades gobernamentais.

#### 8b. Indicadores de rendemento

Son medidas internas que utiliza a organización para supervisar, entender, predicir e mellorar o seu rendemento, así como para anticipar as percepcións da sociedade.

## CRITERIO 9: RESULTADOS CLAVE

Qué logros está alcanzando a organización con relación ao rendemento planificado.

### Subcriterios

#### 9a. Resultados Clave do Rendemento da Organización

Estas medidas son os resultados clave planificados pola organización e, dependendo do obxecto e dos obxectivos dela, poden facer referencia a:

- Resultados económicos e financeiros.
- Resultados non económicos.

#### 9b. Indicadores Clave do Rendemento da Organización

Son as medidas operativas que emprega a organización para supervisar, entender, predicir e mellorar os probables resultados clave do rendemento desta.

### **11.2.7 VANTAXES DE ADOPTAR O MODELO EFQM**

Aumentar a competitividade da organización:



- Sendo máis rendibles.
- Logrando un bo clima de traballo.
- Ofrecendo unha excelente calidade de servizo, tendo en conta tanto os requisitos legais como as necesidades e expectativas dos clientes.

### **11.2.8 CAMBIOS QUE ORIXINA A EXCELENCIA EN XESTIÓN**

#### Concepto tradicional

- Descoñecemento do cliente.
- Os empregados buscan satisfacer os xefes.
- A calidade refírese á produción e ás materias primas.
- O departamento de calidade é o que asegura a calidade.
- Existe unha reticencia cara ao cambio.
- A organización está dividida en departamentos.
- Non hai implicación entre departamentos.
- A participación e a implicación non é prioritaria e mesmo é sancionada.
- Os xefes son os que deciden.
- Xestión cualitativa.

#### Concepto Excelente

- O cliente é o que manda.
- Toda a organización busca satisfacer os clientes.
- A calidade concírnelles a todas as persoas da organización.
- Cada empregado garante a calidade.
- A contorna é cambiante, polo tanto o cambio é natural nas empresas.
- A organización está integrada e cohesionada.
- Estimúlase e prémiase a participación e a implicación.
- Os líderes delegan.
- Xestión con datos, os indicadores sinalan oportunidades de mellora.



## **11.3 SERIE ISO 9000**

### **11.3.1 INTRODUCCIÓN**

A Norma Internacional UNE EN ISO 9001 é un método de traballo considerado como o mellor para a mellora da calidade e da satisfacción do cliente. Na súa última revisión, ISO 9001:2008, clarifícanse algúns aspectos da súa anterior revisión (ISO 9001:2000), mantendo a súa esencia, sen ampliar a súa especificación.

O Estándar ISO 9000 está baseado nun modelo de xestión por procesos que desenvolve os oitos principios da Xestión da Calidade.

A nova versión da norma ISO 9001:2008 foi publicada en 2008, froito do traballo realizado polo Comité ISO TC/176/SC2.

A norma ISO 9001:2008 mantén de forma xeral a filosofía do enfoque a procesos e os oito principios de xestión da calidade, á tempo que seguirá sendo xenérica e aplicable a calquera organización independentemente da súa actividade, tamaño ou do seu carácter público ou privado.

Malia que os cambios abarcan practicamente a totalidade dos apartados da norma, estes non supoñen un impacto para os sistemas de xestión da calidade das organizacións baseados na ISO 9001:2000, xa que fundamentalmente están enfocados a mellorar ou resaltar aspectos como:

- Importancia relevante do cumprimento legal e regulamentario.
- Aliñamento cos elementos comúns dos sistemas ISO 14001.
- Maior coherencia con outras normas da familia ISO 9000.
- Mellora do control dos procesos subcontratados.
- Aumento de comprensión na interpretación e entendemento dos elementos da norma para facilitar o seu uso.



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

*ISO 9000, Quality management systems – Fundamentals and vocabulary (Sistemas de xestión da calidade – Fundamentos e vocabulario)*

Esta norma describe os conceptos dun Sistema de Xestión da Calidade (SGC) e define os termos fundamentais empregados na familia ISO 9000. A norma tamén inclúe os oito principios de xestión da calidade que se usaron para desenvolver a ISO 9001 e a ISO 9004.

*ISO 9001, Quality management systems - Requirements (Sistemas de xestión da calidade – Requisitos)*

Esta norma especifica os requisitos dun SGC, co cal unha organización busca avaliar e demostrar a súa capacidade para fornecer produtos que cumpran os requisitos dos clientes e os regulamentarios aplicables, e con iso aumentar a satisfacción dos seus clientes.

*ISO 9004, Quality management systems – Guidelines for performance improvements (Sistemas de xestión da calidade – Directrices para a mellora do desempeño)*

Esta norma proporciona orientación para a mellora continua e pódese usar para mellorar o desempeño dunha organización. Mentres que a ISO 9001 busca brindar aseguranza da calidade aos procesos de fabricación de produtos e aumentar a satisfacción dos clientes, a ISO 9004 asume unha perspectiva máis ampla de xestión da calidade e brinda orientación para melloras futuras. As directrices para autoavaliación incluíronse no Anexo A da ISO 9004. Este anexo brinda un enfoque sinxelo e de fácil uso para determinar o grao relativo de madurez do SGC dunha organización e identificar as principais áreas de mellora.

A ISO 9000 é un punto de partida para entender as normas, xa que define os termos fundamentais empregados na “familia” ISO 9000, ou no grupo de



normas relativas a xestión da calidade. A ISO 9001 especifica os requisitos para un sistema de xestión da calidade co cal se poida demostrar a capacidade de fornecer produtos que cumpran os requisitos dos clientes, do mesmo xeito que os requisitos aplicables; tamén busca incrementar a satisfacción dos clientes. A ISO 9004 brinda orientación sobre a mellora continua do seu sistema de xestión da calidade, de maneira que se cumpran as necesidades e expectativas de todas as partes interesadas. Dentro das partes interesadas inclúense os clientes e os usuarios finais; os directores e persoal da organización; os propietarios e investidores; os provedores e socios, e a sociedade en xeral.

A ISO 9001 e a ISO 9004 son un “par coherente” de normas que relacionan a xestión da calidade moderna cos procesos e actividades dunha organización, e destacan a promoción da mellora continua e o logro da satisfacción do cliente. A ISO 9001, que se centra na eficacia do sistema de xestión da calidade para cumprir os requisitos dos clientes, úsase para certificación ou para acordos contractuais entre provedores e compradores. Por outra banda, a ISO 9004 non se pode usar para certificación, xa que non establece requisitos senón que proporciona orientación sobre a mellora continua do desempeño dunha organización. A ISO 9001 céntrase na “eficacia”, é dicir, en facer o correcto, mentres que a ISO 9004 fai énfase tanto na “eficacia” como na “eficiencia”, é dicir, en facer o correcto na forma correcta.

### **11.3.3 PRINCIPIOS DE XESTIÓN**

A ISO 9000 baséase nos 8 principios de xestión:

- Enfoque ao cliente, que dá como resultado o cumprimento dos requisitos dos clientes e o esforzarse por excedelos.
- Liderado, que apunta a crear un ambiente interno onde as persoas estean totalmente implicadas.
- Participación do persoal, que é a esencia dunha organización.



- Enfoque baseado en procesos, que dá como resultado a mellora da eficiencia para obter os resultados desexados.
- Enfoque de sistema para a xestión, que conduce á mellora da eficiencia e a eficacia por medio da identificación, comprensión e xestión de procesos interrelacionados.
- Mellora continua, que se converte nun obxectivo permanente da organización.
- Enfoque baseado en feitos para a toma de decisións, baseado na análise de datos e información.
- Relacións mutuamente beneficiosas co proveedor, baseadas na comprensión da súa interdependencia.

Para o manexo dunha organización, a ISO 9000 estimula a adopción do enfoque baseado en procesos. Para o modelo de procesos revisado na ISO 9000 considéranse cinco áreas principais:

- Sistema de xestión da calidade.
- Responsabilidade da alta dirección.
- Xestión de recursos.
- Realización do produto.
- Medición, análise e mellora.

O modelo de proceso usado nas normas é completamente compatible co ben coñecido ciclo de PLANEAR, FACER, VERIFICAR, ACTUAR.

A xestión de calidade debe incluír os procesos requiridos para lograr calidade, e resaltar a interacción entre eles. A alta xerencia debe asumir a responsabilidade polo liderado, compromiso e participación activa para desenvolver e manter o sistema de calidade. A alta dirección debería fornecer os recursos adecuados, de maneira que os clientes obteñan o que se acordou mutuamente. É necesario contar con procesos ben definidos, tanto operacionais como de soporte, para poder realizar o produto. A



satisfacción dos clientes débese medir e analizar de maneira que a organización poida mellorar continuamente.

#### **11.3.4 VERSIÓNS ESPECÍFICAS DA NORMA ISO 9000**

As normas para “sectores específicos” son normas de xestión da calidade destinadas a unha industria específica, un produto ou grupo de produtos. Por exemplo, existen normas de xestión de calidade específicas para a industria automotriz, a industria de alimentos e bebidas, a industria das telecomunicacións, etc.

A familia de normas ISO 9000, xenérica por natureza, é aplicable a calquera tipo de produto ou servizo e pode ser implantada por calquera industria. Xa que logo, a ISO (Organización Internacional de Normalización), busca limitar a proliferación de normas no campo da xestión da calidade. O comité técnico ISO 176 (ISO/TC 176), responsable do desenvolvemento da familia de normas ISO 9000, apoia o desenvolvemento de normas para sectores específicos, unha vez se estableceu que hai necesidade delas.

#### **11.3.5 CUSTOS E BENEFICIOS DE ESTABLECER UN SISTEMA DE XESTIÓN DA CALIDADE**

##### 1. Custos...

Os custos en que incorren as compañías pódese pormenorizar en custos directos e indirectos.

Os custos directos inclúen, entre outros, os seguintes:

- Contratación de formadores ou consultores externos, se se requiren.
- Envío de persoal para recibir formación externa.
- Adquisición das normas nacionais e internacionais pertinentes da familia ISO 9000, e os libros e publicacións relacionadas.
- Adquisición de equipos adicionais, instrumentos e outros recursos que identifique a compañía.



Os custos indirectos inclúen, entre outros, os seguintes:

- Tempo empregado pola dirección e demais persoal para o desenvolvemento do sistema.
- Reorganización dos procesos, incluídas as melloras no manexo da empresa, se se requiren.
- Custos de calibración externa dos equipos, co fin de asegurar a trazabilidade das medicións comparados con patróns de medición trazables a patróns de medición nacionais ou internacionais.
- Organización da formación interna.
- Tempo gastado polos auditores internos para as auditorías internas periódicas.
- Accións correctivas, incluída a actualización de manuais e procedementos, se se require.
- Gastos en dixitalización de documentos, papelería e outros artigos de consumo requiridos para a preparación de manuais e documentación de procesos, etc.

Algúns factores que poden axudar a reducir os custos anteriores inclúen:

- Facer que o persoal da compañía se familiarice cos requisitos do SGC.
- Contar con actividades documentadas relacionadas co sistema, por exemplo instrucións de traballo, plans de calidade, procedementos, etc., xa implantadas.
- A contratación de consultores só para actividades específicas tales como "análise de brechas", formación de auditores, auditorías de preavaliación, etc., e contar con persoal interno para supervisar as actividades restantes.

Doutra banda, hai factores que poden significar custos de implantación maiores para a compañía. Por exemplo, se a súa compañía realiza actividades en diferentes lugares, ou está involucrada no deseño e desenvolvemento de produtos, isto pode aumentar os custos.



## 2. ... e beneficios de obter unha certificación baseándose en ISO 9000

A implantación dun sistema de xestión de calidade xéralle beneficios internos á maioría de organizacións, do mesmo xeito que oportunidades con relación ao mundo exterior.

Os beneficios internos para a compañía inclúen:

- Enfoque mellorado cara ao cliente e orientación aos procesos dentro da compañía.
- Maior compromiso da dirección e mellor toma de decisións.
- Condicións de traballo melloradas para os empregados.
- Aumento de motivación por parte dos empregados.
- Custo reducido de fallas internas (menores tarifas de reprocesos, rexeitamento, etc.) e fallas externas (menos devolucións dos clientes, substitucións, etc.), e o último, inda que non o menos importante:
- A mellora continua do sistema de xestión da calidade.

Xéranse os seguintes beneficios externos:

- Os clientes teñen máis confianza en que recibirán produtos conformes aos seus requisitos, o que, pola súa vez, redunda en maior satisfacción do cliente.
- Unha mellor imaxe da compañía.
- Publicidade máis agresiva, xa que os clientes poden estar informados dos beneficios de realizar negocios cunha compañía que manexa a calidade dos seus produtos.
- Máis confianza en que os produtos da compañía cumpren os requisitos regulamentarios pertinentes.
- Mellor evidencia obxectiva para defenderse contra demandas por obriga civil, se os clientes chegasen a presentar algunha.

### **11.3.6 IMPLANTACIÓN DUN SISTEMA DE XESTIÓN DA CALIDADE**



Un sistema de xestión de calidade baseado en ISO 9000 pódese implantar nos seguintes pasos:

### 1. Avaliar a necesidade e metas da organización con relación á implantación dun SGC

A necesidade pode xurdir a raíz de queixas repetidas dos clientes, devolucións frecuentes por garantía, entregas atrasadas, altos inventarios, atrasos frecuentes na produción, un alto nivel de reprocesos, ou rexeitamento de produtos ou servizos. Nesta etapa, identifique as metas que quere alcanzar a través dun SGC, tales como a satisfacción dos seus clientes, unha maior participación no mercado, mellores comunicacións e moral da organización, unha maior eficiencia e rendibilidade, etc.

Outro obxectivo de implantar un SGC pode ser a demostración de conformidade por medio dunha certificación por terceira parte, que pode solicitar un cliente importante, ou que se esixe para se poder rexistrar como provedor de grandes compañías, por exemplo, os fabricantes de equipos orixinais (OEM).

### 2. Obter información acerca da familia ISO 9000

As persoas identificadas para iniciar o desenvolvemento dun SGC baseado en ISO 9000 necesitan entender os requisitos da ISO 9001, xunto coa ISO 9000 e a ISO 9004.

A información de soporte, por exemplo os principios de xestión de calidade, preguntas frecuentes (FAQ), orientación sobre o numeral 1.2 (aplicación) da ISO 9001, orientación sobre os requisitos de documentación da ISO 9001 e outros folletos, están dispoñibles na páxina web de ISO: <http://www.iso.org>.

### 3. Nomear un consultor, se é necesario

Se dentro da organización non se conta coa competencia adecuada para desenvolver un SGC, pódese contratar un consultor. Antes de facelo, é conveniente verificar os seus coñecementos e experiencia; o coñecemento



deste acerca dos procesos de realización do produto da súa organización, e a súa experiencia en axudar a outras organizacións a alcanzar as súas metas establecidas, incluída a certificación.

#### 4. Toma de conciencia e formación

Hai que espertar a conciencia acerca dos requisitos do SGC entre todo o persoal que realiza actividades que afectan a calidade. Tamén planificar e brindar formación específica sobre como desenvolver Manuais de Calidade, como planear un SGC, como identificar e implantar procesos de mellora, e sobre como auditar a conformidade co SGC.

#### 5. Realizar a análise de brechas (*Gap analysis*)

Débense avaliar as brechas que hai entre o sistema de xestión da calidade existente e os requisitos da ISO 9001 para o SGC, e preparar o xeito de pechar estas brechas, incluída a planificación dos recursos adicionais requiridos. A análise destas brechas pódese levar a cabo mediante unha autoavaliación ou un consultor externo.

#### 6. Procesos de realización do produto

Examinar o numeral 7 da ISO 9001 relativo á realización "do produto", para determinar cómo os requisitos se aplican ou non ao SGC da compañía. Os procesos comprendidos por este numeral inclúen:

- Procesos relacionados co cliente.
- Deseño e desenvolvemento.
- Compras.
- Produción e abastecemento do servizo.
- Control de dispositivos de medición e seguimento.

#### 7. Fornecer o persoal

Decidir sobre as responsabilidades das persoas que estarán involucradas no desenvolvemento e documentación do seu SGC, incluído o nomeamento dun representante da dirección, quen supervisará a



implantación do SGC. A creación dun Comité Director do proxecto tamén pode ser útil para supervisar o progreso e fornecer os recursos cando se requiran.

### 8. Elaborar o cronograma

Preparar un plan completo para pechar as brechas identificadas no paso 5 para desenvolver os procesos do SGC. Neste plan incluír as actividades que quedan por realizar, os recursos requiridos, as responsabilidades e un tempo de finalización estimado para cada actividade. Os numerais 4.1 e 7.1 da ISO 9001 brindan información que se debería usar ao desenvolver o plan. O tempo total requirido para cada fase (planificación, documentación, implantación e avaliación) depende da extensión das brechas no seu SGC existente.

### 9. Redactar o Manual de Calidade

No Manual de Calidade:

- Incluír cómo se aplica o SGC aos produtos, procesos, instalacións e departamentos da organización.
- Excluír calquera requisito que se decidiu no paso 6, coa súa respectiva xustificación.
- Facer referencia ou incluír procedementos documentados para o seu SGC.
- Describir a interacción entre os procesos do SGC, por exemplo, a interacción entre os procesos de realización do produto e outros procesos de xestión, medición e mellora.
- Redactar a política de calidade e os obxectivos de calidade da organización.

O persoal implicado na organización debería revisar o Manual de Calidade e os procedementos documentados, de maneira que os seus comentarios e suxestións poidan ser tidos en conta antes de que o Manual de Calidade e os procedementos sexan aprobados para a súa publicación e



uso. Tamén se debería chegar a unha decisión acerca da data de implantación.

#### 10. Realización de auditorías internas

Durante a fase de implantación, de tres a seis meses aproximadamente despois de que se escribe a documentación, os auditores adestrados deberían levar a cabo unha ou dúas auditorías internas que cubran todas as actividades do SGC, e a dirección involucrada debería emprender sen demora as accións correctivas sobre os resultados da auditoría. Cando se requira, actualizar os manuais, os procedementos e os obxectivos. Logo de cada auditoría interna, a alta dirección debería revisar a eficacia do sistema e fornecer os recursos necesarios para as accións correctivas e melloras.

#### 11. Solicitude da certificación

Unha vez finalizado satisfactoriamente o paso 10, e se a compañía decide obter unha certificación por terceira parte, pódese solicitar unha certificación a un organismo de certificación acreditado.

#### 12. Realización de avaliacións periódicas

Logo da certificación, a organización debería realizar periodicamente auditorías internas para revisar a eficacia do SGC e ver como se pode “mellorar continuamente”. A organización debería avaliar periodicamente se o propósito e metas (ver o paso 1) para os cales se desenvolveu o SGC se están a lograr, incluída a súa mellora continua.

### **11.4 CMMI**

O Modelo de Madurez da Capacidade do Software **SW-CMM** (*Software Capability Maturity Model*) foi definido por Paulk, Curtis, Chrisis e Weber en 1993 como un modelo que establece os niveis polos cales as organizacións de software fan evolucionar as súas definicións, implantacións, medicións,



controis e melloras dos seus procesos de software. O modelo CMM permite definir o grao de madurez das prácticas de xestión e enxeñaría de software destas organizacións e determinar cales son as accións de mellora prioritarias para os seus procesos de software.

O modelo CMM componse de **cinco niveis de madurez** de acordo coa capacidade do proceso de software, definidos polos obxectivos dos procesos que, cando son satisfeitos, permiten avanzar ao próximo nivel, xa que un ou máis compoñentes importantes do proceso de software foron estabilizados.

En cada nivel defínese un conxunto de **áreas clave do proceso** que describen as funcións de enxeñaría do software que se deben levar a cabo para o desenvolvemento dunha boa práctica. Mediante un amplo conxunto de métricas determínase a calidade de cada unha das áreas clave, obténdose unha visión precisa do rigor, a eficacia e a eficiencia da metodoloxía de desenvolvemento dunha organización produtora de software.

Cada unha das áreas está organizada en cinco seccións, denominadas **características comúns**. Estas son as seguintes:

- **Compromiso.** É o conxunto de accións que a organización debe realizar para poder asegurar que o proceso é repetible e duradeiro. Normalmente está relacionado coas políticas da organización e o liderado da dirección.
- **Capacidade.** Describe as precondicións que se deben dar nun proxecto ou na organización para implantar de forma efectiva os procesos de software. Habitualmente afecta aos recursos, á estrutura e á formación.
- **Actividades.** Describen os roles e os procedementos necesarios para implantar unha área clave de proceso. Habitualmente inclúen procedementos relacionados coa planificación e o seguimento do traballo, así como as accións correctivas necesarias.



- **Medidas e análises.** Describen a necesidade de realizar medicións dos procesos e analizan estas medidas. Adoitan incluír exemplos das medidas tomadas para determinar o estado e a eficacia das actividades realizadas.
- **Verificación.** Describe os pasos que se deben levar a cabo para asegurar que as actividades se realizan segundo os procesos establecidos. Habitualmente inclúe revisións e auditorías por parte da dirección e do grupo de aseguranza da calidade.

Pola súa vez, en cada característica común especifícanse unhas **prácticas clave**, que son normas, procedementos e actividades cuxa realización leva á consecución dos obxectivos da área. Nalgúns casos detállanse subprácticas máis específicas, guías e interpretacións da práctica e, cando procede, exemplos e referencias cruzadas a outras prácticas. Así mesmo, o modelo define **indicadores clave**, que son aquelas prácticas clave ou compoñentes de prácticas clave que ofrecen unha visión maior da consecución dos obxectivos dunha área clave de proceso.

Os cinco niveis de madurez do modelo son:

- 1- **Inicial:** o proceso de software é un proceso improvisado e caótico. Non se definiron procesos metodolóxicos, ou definíronse mais non se seguen. O éxito que se poida obter depende das habilidades, coñecementos e motivacións do persoal. Non existen calendarios nin estimacións de custos e as funcionalidades e calidade do produto son impredecibles, nin un ambiente estable para o desenvolvemento e mantemento do software. O proceso do software é impredecible polo continuo cambio ou modificación a medida que avanza o traballo.
- 2- **Repetible:** establécense políticas e procedementos de administración e implantación do proceso básico para determinar custos, calendarios e funcionalidades. A madurez metodolóxica da organización permite estimar de xeito fiable o tamaño funcional ou



físico do sistema, así como os recursos, o esforzo, os custos e p calendario. Sentáronse as bases para repetir éxitos anteriores en proxectos con aplicacións similares. A organización mostra problemas de calidade e carece dunha estrutura adecuada para melloralala. Neste nivel, as áreas clave, cuxo estado se pode coñecer mediante diversas métricas, son as seguintes:

- Xestión de requisitos.
- Planificación do proxecto software.
- Seguimento e control do proxecto.
- Xestión da subcontratación do software.
- Aseguranza da calidade do software.
- Xestión da configuración do software.

3- **Definido:** o proceso de software para as actividades administrativas e técnicas está documentado, homoxeneizado e integrado nun proceso de software estándar dentro da organización, que axudará a obter un desempeño máis efectivo. O grupo que traballa no proceso enfoca e guía os seus esforzos á mellora do seu desenvolvemento, facilita a introdución de técnicas e métodos e informa á administración do estado do proceso. A capacidade do proceso está baseada nunha ampla comprensión común dentro da organización das actividades, roles e responsabilidades definidas no desenvolvemento de software. As áreas clave definidas para este nivel son as seguintes:

- Desenvolvemento e mellora dos procesos da organización.
- Definición dos procesos da organización.
- Programa de formación.
- Xestión integrada do software.
- Enxeñaría de produto software.
- Coordinación intergrupos.



- Revisión conxunta.
- 4- **Xestionado:** recompílanse medidas detalladas do proceso de software e da calidade do produto. Os dous son cuantitativamente entendidos e controlados. Este nivel de capacidade permítelle á organización predicir as tendencias na calidade do produto dentro dos límites establecidos e tomar as accións necesarias en caso de que sexan excedidos. Pódese predicir que os produtos desta categoría son de alta calidade. As áreas clave definidas para este nivel son as seguintes:
- Xestión cuantitativa do proxecto.
  - Xestión de calidade do software.
- 5- **Optimizado:** a mellora continua do proceso é garantida pola retroalimentación cuantitativa e desde as probas de técnicas e ferramentas innovadoras. A organización ten os medios para identificar os puntos débiles e coñecer como fortalecelos. A súa actividade clave é a análise das causas de defectos e a súa prevención. As áreas clave definidas para este nivel son:
- Prevención de defectos.
  - Xestión de cambios tecnolóxicos.
  - Xestión de cambios nos procesos.

A estrutura do CMM ofrece un camiño progresivo recomendado para as organizacións dedicadas ao desenvolvemento de software que queren mellorar a capacidade do seu proceso de software. De forma xeral identifícanse os seguintes **usos fundamentais**:

- Equipos de asesores, que usan o modelo para identificaren os puntos fortes e débiles na organización.
- Equipos de avaliación, que utilizan CMM para identificar o risco de seleccionar entre diferentes contratos de negocio e para monitoralos.



- Persoal técnico e de dirección, que usa CMM para entender as actividades necesarias que axuden a planear e implantar o programa de mellora do proceso de software da organización.
- Grupo de mellora do proceso, que empregan como guía para se axudaren a definir e mellorar o proceso de software na organización.

Ademais do Modelo de Madurez da Capacidade do Software, existen o Modelo de Madurez da Capacidade na Adquisición de Software (SA-CMM), o Modelo de Madurez da Capacidade das Persoas (P-CMM), etc.

Desde o ano 1991, o modelo CMM foise adaptando a múltiples disciplinas: enxeñaría de sistemas, enxeñaría do software, compras, desenvolvemento de procesos e produtos integrados, etc., derivando modelos diferentes. As organizacións que desexaban mellorar os seus procesos en todas estas disciplinas atopábanse con que o modelo presentaba grandes diferenzas de arquitectura, enfoque, contido e aplicación. Este feito provocaba un grande incremento do custo da implantación de CMM en termos de formación, avaliacións e actividades de mellora, xa que non existía unha integración de todos estes modelos. O **proxecto de integración de CMM ou CMMI** (Capability Maturity Mode of Software Integration) xurdiu como solución aos problemas de falta de integración, e foi posto en marcha para desenvolver un marco de traballo simple de mellora de procesos, para organizacións que perseguen a mellora en todos os ámbitos e niveis da empresa.

**CMMI** contén un conxunto de produtos que, ademais de numerosos modelos adaptables ás diferentes áreas de coñecemento, contén métodos de avaliación segundo cada modelo así como material de formación. O obxectivo inicial de CMM, *“obter produtos de calidade dentro das marxes temporais previstas co mínimo custo”*, non cambiou en CMMI. Pola contra, CMMI basea a aplicación de todos os principios de CMM ao longo de todo o ciclo de vida de enxeñaría, non só no ciclo de vida do desenvolvemento do



produto software. Ademais, o conxunto de produtos CMMI foi deseñado para manter compatibilidade e correspondencia co modelo **SPICE**.

En resumo, CMMI pode ser considerado como unha extensión do CMM-SW.

**As diferenzas principais son:**

- Engadíronse novas áreas de proceso.
- Engadíronse mellores e máis modernas prácticas clave.
- Engadiuse un obxectivo xenérico para cada área de proceso.

Se se analizan estas diferenzas en función do nivel de madurez en que se encontran, pódense atopar as seguintes áreas de proceso no modelo CMMI que non se atopan no modelo CMM:

- **Nivel 2.** Medición e análise. Foron illadas de CMM todas as prácticas relacionadas con este obxectivo e foron agrupadas dentro desta nova área de proceso.
- **Nivel 3.** A área de enxeñaría do produto software de CMM foi substituída en CMMI por múltiples e máis detalladas áreas de proceso:
  - o Desenvolvemento de requisitos.
  - o Solucións técnicas.
  - o Integración do produto.
  - o Verificación e Validación.

Na área de xestión integrada do proxecto de CMM contemplábase a xestión de riscos, pero agora foi considerada como unha área de proceso independente. Finalmente, a este nivel engadíuselle unha nova área denominada análise de decisións e resolución, que non se atopaba en CMM.

- **Nivel 4.** Este nivel sufriu unha reestruturación e as áreas de xestión cuantitativa de procesos e xestión da calidade de software foron



convertidas a xestión cuantitativa do proxecto e rendemento ou realización do proceso organizacional, respectivamente.

- **Nivel 5.** Tampouco houbo grandes cambios neste nivel, simplemente unha fusión das áreas xestión dos cambios tecnolóxicos e xestión do cambio nos procesos nunha única área de proceso: innovación organizacional e desenvolvemento. A área de prevención de defectos foi reestruturada e denominada análise causal e resolución.

CMMI v1.1 ten catro disciplinas dispoñibles:

- Enxeñaría de software: CMMI-SW (Software Engineering).
- Enxeñaría de sistemas: CMMI-SE (Systems Engineering).
- Desenvolvemento integrado do produto e do proceso: CMMI-IPPD (Integrated Product and Process Development).
- Provedores externos: CMMI-SS (Supplier Sourcing).

CMMI presenta dúas representacións do modelo:

- **Continua:** capacidade de cada área de proceso. Esta representación céntrase na capacidade de cada área de proceso para establecer unha liña a partir da cal se pode medir a mellora individual en cada área. Do mesmo xeito que o modelo por etapas, o modelo continuo ten áreas de proceso que conteñen prácticas, pero estas organízanse de maneira que soportan o crecemento e a mellora dunha área de proceso individual. Hai seis niveis de capacidade, do 0 ao 5. A representación continua céntrase en seleccionar unha certa área de procesos que mellorar e en fixar o nivel de capacidade desexado para esa área.
- **Por etapas:** madurez organizacional. Nesta representación dáse un mapa predefinido dividido en etapas (os niveis de madurez), para a mellora organizacional e que se basea en procesos probados, agrupados e ordenados e as súas relacións asociadas. Cada nivel de



madurez ten un conxunto de áreas de proceso que indican onde unha organización debería centrar a mellora do seu proceso. Cada área de proceso descríbese en termos de prácticas que contribúen a satisfacer os seus obxectivos. As prácticas describen as actividades que máis contribúen á implantación eficiente dunha área de proceso; auméntase o “nivel de madurez” cando se satisfán os obxectivos de todas as áreas de proceso dun determinado nivel de madurez.

Ambas as representacións inclúen Metas (Xenéricas e Específicas, definicións de resultados que cómpre obter pola implantación efectiva dos grupos de prácticas) e Prácticas (Xenéricas e Específicas, accións que cómpre realizar para cumprir obxectivos de área de proceso).



### Bibliografía:

- *Ingeniería del Software. Un enfoque práctico*. ROGER S. PRESSMAN. Ed. McGraw Hill.
- <http://modelosdegestiondelacalidad.blogspot.com/>
- Apuntamentos e papeis de traballo de Enxeñaría de Sistemas de Información. RAMÓN ORTIGOSA.
- Temario das probas selectivas para o ingreso no Corpo Superior de Sistemas e Tecnoloxías da Información da Administración do Estado. ASTIC.
- *Enginyeria del Software III*. Antònia Mas Pichaco. Universitat de les Illes Balears.

### Sitios web:

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

**Autor: Ramón Seoane Freijido**

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

**Colexiado do CPEIG**

**Autor: Hernán Vila Pérez**

**Xefe do Servizo de Informática. Instituto Galego de Vivenda e Solo**

**Vicepresidente do CPEIG**



**12. A BIBLIOTECA DE  
INFRAESTRUTURA TI (ITIL).  
SOPORTE AO SERVIZO.  
ENTREGA DE SERVIZOS. ISO  
20000-1:2005, ISO 20000-2:2005.  
OBXECTIVOS DA NORMA.  
MAPA E DESCRICIÓN DOS  
PROCESOS.**



**Tema 12. A biblioteca de infraestrutura TI (ITIL). Soporte ao servizo. Entrega de servizos. ISO 20.000-1:2005, ISO 20.000-2:2005. Obxectivos da norma. Mapa e descrición dos procesos.**

## **ÍNDICE**

### **12.1 A biblioteca de infraestrutura TI (ITIL)**

#### **12.1.1 Introducción**

#### **12.1.2 Antecedentes**

#### **12.1.3 Que é ITIL?**

##### **12.1.3.1 Obxectivos**

##### **12.1.3.2 Beneficios**

##### **12.1.3.3 Estrutura**

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

#### **12.1.5 Conclusións**

### **12.2 Soporte ao servizo. Entrega de servizos.**

#### **12.2.1 Soporte ao servizo**

##### **12.2.1.1 Procesos**

#### **12.2.2 Entrega de servizos**

##### **12.2.2.1 Procesos**

### **12.3 ISO 20.000. Obxectivos da norma**

#### **12.3.1 Que é ISO 20.000?**

#### **12.3.2 Utilidades do certificado ISO 20.000**

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

#### **12.3.4 Que representa exactamente ser conforme a ISO 20.000?**

### **12.4 Deseño do mapa de procesos ITIL**

#### **12.4.1 Utilidade do modelo de referencia de ITIL**

## **12.1 A BIBLIOTECA DE INFRAESTRUTURA TI (ITIL)**

### **12.1.1 INTRODUCCIÓN**



Na actualidade a dependencia polas TI é un factor crítico no desenvolvemento das organizacións. Dependendo das TI só poderá ser visto como positivo sempre e cando exista un xeito de operar que permita aproveitar as TI e as converta nunha vantaxe que proporcione funcionalidade e flexibilidade institucional. Para lograr o anteriormente dito, é indispensable que existan estándares internacionais que orienten ás organizacións respecto a como é posible organizar e estruturar do mellor xeito e aos mellores custos, todos os servizos de TI que xiran en torno á organización, ademais de lograr que se comuniquen os principais actores que interveñen no desenvolvemento da estratexia do negocio.

É por iso que actualmente existen unha serie de estándares ou trazos definidos por diversas institucións de recoñecido prestixio que lles pretenden ofrecer ás organizacións un marco de traballo que lles permita adoptar novas políticas interinstitucionais para a administración organizada e estruturada dos servizos de TI. Entre os estándares comunmente recoñecidos en todo o mundo encóntrase ISO 9000, COBIT, BS-15000, ITIL, ISO 20000, etc.

ITIL (Information Technology Infrastructure Library), considerado como un modelo que permite adoptar "mellores prácticas" nas organizacións, é o resultado da unión de varias librarías estandarizadas dedicadas a unha práctica en particular dentro da xestión de servizos de TI; ofrece ferramentas que facilitan a administración e optimización das TI nas organizacións. É unha estratexia que pode ser aplicable en todo tipo de organizacións pois o seu propósito final é implantar boas prácticas na prestación de servizos de TI.

### **12.1.2 ANTECEDENTES**

O incremento na dependencia das tecnoloxías da información (TI), así como a adopción de estándares propios para xestionar a información, incorrendo algunhas



veces na duplicidade de esforzos nos proxectos ou en maiores custos e a calidade dos servizos TI que ofrecía o goberno británico, levaron consigo que no ano 1980 a Central Computer and Telecommunications Agency (CCTA) desenvolvese unhas primeiras recomendacións que facilitasen a administración e optimización das TI, recomendacións que actualmente se coñecen como ITIL (Information Technology Infrastructure Library). Esta iniciativa converteuse nun marco de traballo conformado por numerosos volumes que probou a súa utilidade non só en organizacións gobernamentais, senón en todos os sectores, converténdose na base para todo tipo de empresas, grandes ou pequenas, que tivesen a disposición de implantar mellores prácticas.

Inicialmente, CCTA enfocouse a recompilar información tendente a verificar como as organizacións dirixían a administración dos seus servizos, logrando analizar e filtrar os diferentes problemas que se xeraban; despois, comprobaron a utilidade e os beneficios das súas recomendacións. Na década dos 90, moitas empresas do goberno europeo adoptaron este marco de traballo, converténdose nunha boa práctica para a administración dos servizos de TI.

No ano 2001, a CCTA e todas as actividades que estaban baixo o seu control pasaron a formar parte da OGC (Office of Government Commerce), oficina do Ministerio de Facenda británico, converténdose, deste xeito, na nova entidade propietaria de ITIL, que tiña por finalidade axudar a modernizar a provisión de TI do goberno británico a través do uso de boas prácticas, logrando deste xeito o mellor valor monetario nas súas relacións comerciais. Posteriormente, a OGC publicaría novas versións de librarías de boas prácticas, escritas por expertos internacionais de diversas organizacións do sector público e privado.

En 1991, créase no Reino Unido a rede mundial de grupos de usuarios das TI que ofrecen mellores prácticas e guías baseadas en estándares para a xestión de servizos de TI; esta rede denomínase itSMF (Information Technology Service



Management Forum); é o único grupo dedicado exclusivamente a este tipo de xestión polo que está recoñecido internacionalmente.

Está presente en varios países de Europa e nalgúns de América Latina, traballando en asociación coa OGC, British Standard Institute (BSI), a Information Systems Examination Board (ISEB) e Examination Institute of the Netherlands (EXIN) e contribuíndo deste xeito á industria das mellores prácticas. Os capítulos desenvolvidos por itSMF fomentan o intercambio de información e experiencias vividas, orientando as organizacións de TI na posta en marcha e melloras dos servizos que ofrecen.

### **12.1.3 QUE É ITIL?**

ITIL (Information Technology Infrastructure Library) considérase unha colección de guías especializadas en temas de organizacións enfocadas ao planeamento, a subministración e soporte de servizos de TI. Recoñécese como un estándar global que resume as mellores prácticas para o ámbito da xerencia de servizos de TI, enfocadas especificamente a describir que funcións ou procesos son os que se recomenda desenvolver, mais non no como desenvolvelos. Para isto último, é responsabilidade da organización definir as estratexias e métodos necesarios para implantala, sempre e cando se adaptan ao tamaño, á cultura e ás necesidades internas da organización.

ITIL ofrece un marco de traballo para as actividades do ámbito das TI, proporcionando unha descrición dos roles, tarefas, procedementos e responsabilidades que poden ser adaptados en organizacións de TI coa finalidade de mellorar a xestión dos seus servizos. Grazas á cantidade de temas que abrangue, considérase un elemento de referencia útil para as organizacións, xa que permite fixar novos obxectivos de mellora para a organización de TI, baseándose na calidade do servizo e no desenvolvemento dos procesos dun xeito eficaz e eficiente.



En varias ocasións, as mellores prácticas consideráronse como procesos que abranguen as actividades máis importantes que se deben ter en conta dentro das organizacións de servizos de TI, polo que se pode afirmar que ITIL é unha colección coherente das mellores prácticas desenvolvidas na industria e non só se pode adaptar ao sector público senón tamén ao privado.

As publicacións de ITIL describen como poden ser optimizados e coordinados dun mellor xeito todos aqueles procesos que foron previamente identificados e que interveñen na administración e operación da infraestrutura de TI, tales como o desenvolvemento do servizo, a xestión da infraestrutura e a provisión e soporte dos servizos; de igual xeito, revelan como estes poden ser formalizados dentro dunha organización, brindando un marco de traballo que facilita unificar a terminoloxía relevante dentro da organización e que lles permite ás persoas falar unha linguaxe común, co que así poden definir obxectivos claros e identificar os recursos e o esforzo necesarios para o seu cumprimento.

#### **12.1.3.1 OBXECTIVOS**

Especificamente ITIL concéntrase en ofrecer servizos de alta calidade, dando especial importancia ás relacións establecidas cos clientes, para o cal o departamento de TI debe prover e cumprir con todos os acordos de servizos previamente definidos con eles, e para logralo é necesario que exista unha forte relación entre estes dous; é por isto que algúns dos obxectivos de ITIL están relacionados con:

- Promover a visión de IT como provedor de servizos.
- Xerar melloras na calidade da subministración dos servizos de TI.
- Fomentar a redución de custos dos servizos de TI.
- Aliñar a prestación dos servizos de TI coas necesidades actuais e futuras do negocio das organizacións, ademais de mellorar a relación cos clientes.
- Estandarizar os procesos que forman parte da xestión de servizos de TI.



- Promover o uso dunha linguaxe común por parte das persoas para mellorar a comunicación dentro das organizacións.
- Servir de base para a certificación das persoas e das organizacións que desexan adoptar este estándar.
- Mellorar a eficiencia, aumentando a efectividade.
- Reducir os posibles riscos que se poden presentar.

### **12.1.3.2 BENEFICIOS**

ITIL centra os seus esforzos na satisfacción dos requirimentos da organización coa mellor relación custo/beneficio a través da descrición dun enfoque sistémico e profesional da xerencia de servizos de TI. Algúns dos beneficios que se conseguen coa adopción das mellores prácticas manexadas en ITIL están relacionados directamente co cliente e coa organización e principalmente teñen que ver con:

- A subministración dos servizos de TI oríéntase especialmente ao cliente e os acordos sobre a calidade do servizo melloran a relación entre o departamento de TI e o cliente.
- A mellora nos niveis de satisfacción dos clientes por medio de medidas obxectivas e eficacia na dispoñibilidade e desempeño da calidade dos servizos de TI.
- Implantación de estándares que permitan realizar o control, administración e operación dos recursos da organización.
- Os servizos ofrecidos son descritos nunha linguaxe máis cómoda e con maiores detalles para os clientes.
- Xestiónanse dun mellor xeito a calidade, dispoñibilidade, fiabilidade e custo dos servizos ofrecidos na organización.
- Melloras na comunicación co departamento de TI no momento de acordar os puntos de contacto.
- O departamento de TI xera unha estrutura clara e centrada nos obxectivos corporativos dun xeito eficaz.



- Soporte aos procesos de negocio e as actividades dos decisores de TI.
- O departamento de TI conta cun maior control sobre a infraestrutura e os servizos que ten ao cargo, obténdose unha visión clara da capacidade do departamento; ademais, permite administrar os cambios dun xeito sinxelo e doado de manexar.
- A definición de funcións, roles e responsabilidades no ámbito dos servizos.
- É posible identificar, a través dunha estrutura procedemental, a externalización dalgúns dos elementos dos servizos de TI.
- Subministración de servizos de TI que satisfagan as necesidades do negocio da organización.
- Incremento e melloras na produtividade e eficiencia da organización a través das experiencias vividas e os coñecementos adquiridos.
- Xera un cambio cultural cara á provisión de servizos e sustenta a introdución dun sistema de xestión de calidade.
- Establece un marco de traballo coherente para as comunicacións tanto internas como externas, permitindo contar coa identificación e estandarización dos procedementos que cómpre seguir.
- Melloras na satisfacción do persoal da organización reducindo notablemente a súa rotación.
- Melloras na comunicación entre o persoal de TI e os seus clientes.
- Xera o intercambio de experiencias obtidas coa súa adopción.

### **12.1.3.3 ESTRUCTURA**

O marco de traballo de **ITIL** está conformado por cinco (5) elementos principais que teñen directa relación entre si, xa que o éxito de cada un deles depende da súa interacción e coordinación cos demais.

Estes elementos son:

- The Business Perspective (A perspectiva do negocio)
- Managing Applications (administración de aplicacións)
- Deliver IT Services (Entrega de servizos de TI)



- Support IT Services (Soporte aos servizos de TI)
- Manage the Infraestructure (Xestión da infraestrutura)

Cada unha das publicacións de ITIL céntrase en documentar un a un os elementos do marco de traballo, realízase unha descrición xeral do que se require para estruturar a xestión de servizos de TI e defínense os obxectivos, as actividades, os roles, os fluxos de comunicación necesarios e as entradas e saídas de cada un dos procesos que son indispensables nunha organización de TI.

Realizáronse tres publicacións das mellores prácticas de ITIL, a primeira versión (V1) foi inicialmente desenvolvida na década de 1980, estaba conformada por dez (10) libros básicos que se centraban na xestión do servizo, especificamente nas súas dúas áreas principais: (i) a entrega do servizo de TI e (ii) o soporte a eses servizos, sendo soportados posteriormente por trinta (30) libros complementarios, que abranguían diversos temas, dende o cableado, ata a xestión de continuidade do negocio. Debido á cantidade de información existente, xorde a segunda versión (V2), que se empezou a reestruturar entre 1999 e 2001, cando ITIL se converte na pedra angular para a xestión do servizo, reorganizándose dun xeito máis sinxelo, onde a maioría da información relacionada coa entrega do servizo e o soporte dos servizos se converte na base do marco de traballo e se agrupa en dous grandes volumes, eliminando desta forma a duplicidade na información existente na primeira versión, de aí que esta versión quedase reorganizada aproximadamente en dez (10) libros. Na terceira versión (V3) liberada en maio de 2007, reducíronse as publicacións a cinco (5) volumes articulados, que principalmente se centran no concepto e desenvolvemento do ciclo de vida do servizo de TI. Ese ciclo iníciase cunha definición da estratexia do servizo, despois concéntrase en realizar o deseño do servizo, posteriormente inicia un período de transición onde se busca realizar o desenvolvemento e a implantación do servizo para axiña realizar a operación do servizo e, finalmente, concéntrase en prover unha mellora continua do servizo, a cal está relacionada permanentemente coas demais etapas do ciclo de vida.



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

##### Service Strategy

Encárgase de asegurar que a estratexia do servizo sexa definida, se manteña e se execute; introdúcense novos conceptos, tales como a consecución do valor, a definición do mercado e o espazo de solución; céntrase no desenvolvemento de prácticas que permitan tomar decisións, baseado na comprensión do servizo activo, as estruturas e os servizos da economía co obxectivo final de incrementar a vida económica dos servizos; busca obter o aliñamento entre as TI e o negocio, non como anteriormente se viña traballando, onde só as TI eran as que se debían adaptar ao negocio.

Algúns dos conceptos xerais que se abordan neste libro teñen que ver coa definición do servizo, a estratexia do Service Management e a planificación do valor, a identificación da dirección e do goberno dos servizos das TI, a correspondencia existente entre os plans de negocio e as estratexias dos servizos de TI, algúns arquetipos de servizos e tipos de provedores de servizos e, o máis importante, que debe formularse, implantarse e revisarse como estratexia do negocio.

##### Service Design

Este libro concéntrase en definir como se deseñará o servizo identificado previamente na estratexia, a través do desenvolvemento de plans que a convertan en realidade; para o deseño de servizos de TI axeitados e innovadores é necesario establecer e implantar políticas de TI, arquitecturas e algunha documentación pertinente. Dentro dos aspectos abordados no novo proceso de xestión de provedores que forma parte do deseño do servizo, encóntranse o aproveitamento da dispoñibilidade, a capacidade, a continuidade e administración do SLA, así como os conceptos de garantía do servizo e utilidade, que son considerados como aspecto fundamental polos Clientes.



Outros conceptos que son traballados neste volume están asociados co ciclo de vida do servizo, obxectivos e elementos no deseño dos servizos, selección dun modelo de servizos apropiado, identificación de servizos, persoas, procesos, ferramentas, etc., modelo de custos, unha análise de riscos e beneficios, e a implantación do deseño do servizo.

### Service Transition

Ten como obxectivo minimizar dun xeito eficaz a fenda existente entre os proxectos e as operacións; concéntrase nas accións que interveñen cando o servizo deseñado se debe poñer en produción, centrándose especialmente no papel que desempeña o Change Management e explicando as prácticas existentes para unha correcta Release Management dun xeito amplo e con visión a longo prazo, permitindo que se consideren todos os factores que participan, tales como mecanismos de entrega, riscos, beneficios e facilidade na posterior operación continua do servizo deseñado.

A transición do servizo ten que ver coa calidade e o control da entrega das operacións e proporciona modelos para apoiar a transición orientando a forma para reducir variacións na entrega.

### Service Operation

O ciclo de vida de calquera servizo culmina coa súa operación, a cal debe ser tan robusta e efectiva que permita obter unha estabilidade na xestión do servizo en todo momento e de extremo a extremo. Neste volume explícanse as actividades necesarias para garantir operatividade no día a día, abrangue moitas das disciplinas e conceptos definidos na V2 de ITIL, especificamente concéntrase nos libros de Service Support e Service Delivery. A través dos coñecementos que se adquiren coa prestación real do servizo pode chegar a influír na estratexia do servizo, o deseño, a transición e a mellora continua do servizo.



### Continual Service Improvement

Este libro abrangue a calidade do servizo no contexto de mellora continua, ademais céntrase tamén en ofrecer mellora continua do servizo aínda cando este se encontre próximo a ser retirado. Un dos maiores beneficios deste libro é que indica explicitamente as accións que se deben realizar para a revisión e mellora dos procesos, información, que na V2 non era tan clara.

Algúns dos conceptos que contempla este libro están relacionados cos principios da mellora continua do servizo, a implantación da mellora de servizos, algúns elementos do negocio e da tecnoloxía que poden levar consigo á mellora continua do servizo e os beneficios xerados que favorecen o negocio, a organización e o aspecto financeiro.

#### **12.1.5 CONCLUSIÓNS**

As mellores prácticas de ITIL ofrecen un marco de traballo que lles permite ás organizacións mellorar o nivel de calidade nos servizos de TI ofrecidos; é por isto que a súa adopción é un paso fundamental e transcendental que se debe tomar, xa que os beneficios que se van obter, non só no medio senón no longo prazo, van permitir que se perciba unha mellora continua a nivel institucional.

A adopción dun estándar como o de ITIL implica o desenvolvemento disciplinado dun proceso enfocado no ciclo de vida do servizo de TI, no cal interveñen varios actores da organización, tanto internos coma externos e onde a achega e contribución de cada un deles co cumprimento das políticas e actividades definidas para todos os procesos favorecen de xeito significativo o éxito da súa implantación. A estrutura das publicacións que foron liberadas desta metodoloxía no seu V3 permítelles ás organizacións adaptalo dun mellor xeito, xa que ofrece as ferramentas necesarias que se deben ter en conta durante a definición e implantación dos procesos de TI. Como o obxectivo central é o ciclo de vida dos servizos de TI,



presenta dun xeito organizado e centralizado toda a actividade que se debe considerar durante a súa implantación, dende a definición da estratexia do servizo, o deseño do servizo, o período de transición polo que debe pasar, a operación do servizo ata chegar a obter unha mellora continua do servizo.

Cada un dos procesos que se identificaron para ITIL teñen estreita relación entre eles mesmos e da súa interacción e comunicación depende en gran parte o éxito da implantación das mellores prácticas; o simple feito de que algúns dos elementos necesarios nos procesos non cumpra cos trazos establecidos no estándar, incrementa as posibilidades de que a adopción se converta nun fracaso.

Cada vez son máis as organizacións que entran a formar parte da familia de ITIL, o que ocasionou que agora os xerentes empecen a preocuparse pola identificación da maneira de como deben planear, implantar e administrar con éxito estas mellores prácticas, converténdose nun dos seus retos laborais e persoais. Tamén son varias as organizacións que xa teñen implantadas con éxito as mellores prácticas de ITIL e grazas ás experiencias que cada un deles viviu coa súa adopción púidose enriquecer este modelo, o que ofrece valiosos consellos ás novas organizacións que a están a implantar.

## **12.2 SOPORTE AO SERVIZO. ENTREGA DE SERVIZOS.**

A xestión dos servizos de TI está conformada por dúas grandes áreas: entrega do servizo (Service Delivery) e soporte ao servizo (Service Support). En libros independentes trabállase todo o relacionado coa perspectiva do negocio, a xestión da infraestrutura, o planeamento que se require para executar a xestión do servizo, a xestión da seguridade e a xestión das aplicacións.

### **12.2.1 SOPORTE AO SERVIZO**

É considerado como un dos eixes principais da xestión do servizo de TI. O contido do libro céntrase en describir os procesos necesarios para manter as



operacións funcionando no día a día, explica como o Service Desk é o responsable e soporta a xestión de incidentes, proporcionando unha base para o soporte ás solicitudes e problemas que se lle poden presentar aos usuarios nunha organización. Así mesmo, encárgase de explicar como a xestión de problemas necesita ser, en certo modo, proactiva e reactiva, ademais de expoñer os beneficios que se poden obter cando se realiza unha análise efectiva das causas fundamentais que ocasionan os problemas, ofrecendo unha ampla visión na redución do impacto que se xera cando existe unha suspensión no servizo aos usuarios.

En resumo, encárgase de describir o xeito en que os clientes e usuarios poden acceder aos servizos que lles permitan apoiar o desenvolvemento das súas actividades diarias e as do negocio, así como a forma en que eses servizos deben ser soportados; ademais, céntrase en todos os aspectos que interveñen para garantir que o servizo ofrecido aos usuarios sexa un servizo continuo, que estea dispoñible e que sexa de calidade.

#### **12.2.1.1 PROCESOS**

##### Service Desk

É o único punto de contacto entre o cliente e os usuarios cos provedores de servizos de TI para todo o relacionado coa subministración de servizos de TI; tamén é o punto de partida encargado de informar sobre os incidentes e a toma de solicitudes de servizo realizadas polos usuarios. A súa obriga é manter informados os usuarios do servizo sobre os eventos, accións e oportunidades que poden chegar a afectar dalgún xeito á dispoñibilidade do servizo e, por conseguinte, á continuidade na subministración do servizo no día a día. Todo o anterior é posible a través do rexistro, resolución e monitorización de problemas.

Dentro do marco de traballo das mellores prácticas de ITIL, o Service Desk non foi concibido como un proceso senón como unha función por desenvolver dentro da



organización do servizo. Algunhas das tarefas ao seu cargo inclúen: ser o punto primario de contacto (SPOC) cos clientes e usuarios; recibir e atender todas as solicitudes, consultas e inquietudes dos clientes e usuarios relacionadas coa subministración dos servizos de TI; documentar, priorizar e realizar un seguimento axeitado das solicitudes de modificacións ou cambios realizadas polos usuarios; atender todos os procesos da xerencia de servizos definidos por ITIL; manter informados sobre o estado e progreso das solicitudes os usuarios que as realizaron; clasificar as solicitudes recibidas e iniciar o seu proceso segundo os acordos do nivel de servizo (SLA) e procedementos establecidos; cando se requira dun soporte de segundo nivel, deberá encargarse de realizar a coordinación do soporte e a subministración do servizo, ao igual que a dos provedores ou participación de terceiros; xestionar a restauración dos servizos co mínimo impacto no negocio, segundo os SLA e prioridades do negocio establecidas; pechar as solicitudes de servizo realizadas polos usuarios e aplicando a avaliación de satisfacción do cliente; realizar seguimento aos SLA definidos tomando as accións necesarias en caso de presentarse incumprimentos; subministrar a información solicitada pola xerencia de TI para manter e mellorar a calidade nos servizos ofrecidos.

De acordo coas mellores prácticas de ITIL, o Service Desk clasifícase en tres tipos: Call Center (centro de chamadas), Help Desk (mesa de axuda) e Service Desk (centro de soporte). Pola súa banda, o Help Desk esta categorizado en: Service Desk Local, a través do cal se busca canalizar localmente todas as necesidades do negocio, moi práctico en varios sitios que requiren servizos de soporte, pero co que se pode chegar a incorrer en grandes custos, xa que o servizo é ofrecido en diferentes lugares, o que implica a definición dun estándar operacional; o Service Desk Central, busca que todos os requirimentos do servizo sexan rexistrados nunha localización central, o que minimiza os custos operacionais xa que existe soamente unha única mesa de axuda en toda a organización que atende todos os requirimentos; e o Service Desk Virtual que ten por finalidade ofrecer o servizo en calquera parte do mundo a través da rede, non importa a súa localización física xa que o servizo se encontra dispoñible



en todo momento, e o seu éxito dáse sempre e cando todos os usuarios da organización contan con infraestrutura tecnolóxica para poder acceder a ela. En conclusión, a implantación con éxito e a execución do proceso de Service Desk xerará maiores beneficios na organización, representado na satisfacción dos clientes, minimización de custos, compromiso persoal e profesionalidade.

### Configuration Management

Coñecida tamén como xestión da configuración, é parte integral de todos os demais procesos da xestión do servizo e ten por obxectivo controlar os activos e elementos de configuración que forman parte da infraestrutura de TI; xa que logo, encárgase de todos os procesos, ferramentas e técnicas necesarias para logralo; tamén é a súa responsabilidade proporcionar información fiable e actualizada non só dos elementos específicos da infraestrutura (elementos de configuración ou CI) necesarios para executar os procesos do negocio, senón tamén sobre as relacións entre eles mesmos, asegurando a integración coas demais disciplinas da xestión do servizo; permite o desenvolvemento dos servizos informáticos de mellor calidade dun xeito viable economicamente e subministra información importante para o cálculo dos custos e a facturación dos servizos executados. As solicitudes de cambio sobre os CI rexístranse nunha base de datos creada para a xestión da configuración denominada Configuration Management Database (CMDB); nesta base de datos encóntranse rexistrados todos os datos dos CI requiridos para a prestación do servizo, dende a súa descrición e interconexión, ata un nivel de detalle que inclúe a categoría, as relacións, os atributos e os posibles estados nos cales pode estar en determinado momento; é necesario actualizar a CMDB cada vez que se realiza un cambio na infraestrutura e ese cambio estea relacionado coa xestión da configuración.

### Incident Management

O obxectivo deste proceso é resolver calquera incidente que xere unha interrupción na prestación do servizo, restaurándoo de novo do xeito máis rápido e efectivo posible; este proceso non se para en encontrar, avaliar e analizar cales foron



as causas que desencadearon a ocorrencia do incidente que xerou unha interrupción no servizo, senón que simplemente se limita a solucionalo temporalmente e a restaurar o servizo de calquera xeito, o que probablemente pode chegar a xerar novas interrupcións no servizo polo mesmo incidente; os incidentes repórtanse sobre os CI.

Unha das maiores contribucións que se lle atribúen ás mellores prácticas de ITIL foi a de establecer a diferenza que existe entre os incidentes e os problemas, onde se distingue entre a restauración rápida do servizo (Incident Management) e a identificación e corrección total da causa que ocasionou o incidente (Problem Management), pero destácase a articulación que debe existir entre estes dous procesos, ao igual que con Change Management (xestión de cambios) e o Service Desk; recoméndase que todos as modificacións realizadas nos incidentes sexan relacionados na mesma CMDB o mesmo que os rexistros de problemas, erros coñecidos e cambios, pois deste xeito será máis doado identificar se a ocorrencia do incidente se pode converter posteriormente nun problema, o que permite analizar e buscar a súa solución definitiva ou corrección total, ademais de evitar a ocorrencia de novos incidentes como consecuencia dos cambios realizados. Algunhas das tarefas a cargo deste proceso teñen que ver con: identificación e documentación de todas as noticias que se realizan dos incidentes acontecidos, incluíndo informes e investigacións; priorizar e categorizar as noticias dos incidentes que se xeraron; proporcionar unha análise inicial do incidente ofrecendo un soporte de primeiro nivel; escalar, cando sexa necesario, a execución de soporte de segundo e terceiro nivel; cando estea en risco o cumprimento nos SLA, é necesario incrementar a asignación de recursos que traballan na solución do incidente; resolver a situación no menor tempo posible, restaurando o servizo; pechar e documentar os incidentes acontecidos; realizar seguimento exhaustivo a cada un dos incidentes que se presentan (monitorización, revisión e comunicación do progreso); realizar unha avaliación dos incidentes reportados, analizalos e xerar informes sobre posibles melloras ao servizo.

### Problem Management



Este proceso dedícase a identificar as causas (orixe) que ocasionan os problemas que se presentan na infraestrutura de TI e a súa solución definitiva para evitar novas ocorrencias. Cando existe un incidente que se repite máis dunha vez, é posible que posteriormente se poida converter nun problema, aínda que a idea é evitar que isto suceda sendo proactivos e previndo novas ocorrencias cando sexa posible; de aí que se fale da xestión de problemas proactiva, onde os incidentes son detectados con demasiada anterioridade de tal maneira que permita tomar as accións preventivas necesarias garantindo que o servizo permaneza dispoñible e non se vexa afectado en ningún momento.

Como resultado da identificación temperá dos incidentes, as accións preventivas que se recomendan realizar tradúcense na execución de cambios nos CI interactuando desta forma coa xestión de cambios; este proceso relaciónase tamén coa xestión de incidentes, xa que require dun rexistro preciso e completo de todos os incidentes co fin de identificar eficiente e eficazmente a súa causa e as tendencias na súa ocorrencia; ademais, despois de atopar a solución a un problema, os incidentes que previamente foran reportados e que tiñan directa relación coas causas do problema poderán pasar a un estado pechado ou dados de baixa.

### Change Management

Este proceso ten unha estreita relación co proceso de Configuration Management, xa que da exactitude dos datos dos elementos da infraestrutura (CMDB) é posible garantir que a análise do impacto é realizada e coñecida, logrando tramitar deste xeito os cambios necesarios a través de procesos e procedementos estandarizados e consistentes. Encárgase de dirixir a aprobación para realizar calquera cambio, así como de controlar a implantación dos cambios da infraestrutura de TI.

Dentro dos seus obxectivos encóntranse: realizar unha valoración dos cambios e garantir que se poden executar ocasionando o mínimo impacto na prestación dos servizos de TI e na infraestrutura actual ou nova, e asegurar de xeito simultáneo a



trazabilidade dos cambios; implantar os cambios autorizados e requiridos para o cumprimento dos SLA de xeito eficiente, efectivo, económico e oportuno; minimizar os cambios, evitando que se implanten cambios non autorizados.

### Release Management

Coa implantación dos cambios pódese xerar como resultado a instalación de novo hardware, a instalación de novas versións de software ou simplemente a actualización ou xeración de nova documentación; é por iso que todas estas accións deben ser controladas e distribuídas dun xeito organizado, como parte dun novo paquete ou versión.

Este proceso está asociado coa correcta implantación de todas as versións dos CI requiridas para a prestación dun SLA, proporcionando un marco de traballo para a coordinación, o control e a introdución física dun cambio. Encárgase de levar o control de todos os cambios e novas versións que se xeraron como resultado da implantación dun cambio ou dunha nova adquisición (p. ex. novo software instalado nunha máquina). É importante ter claridade nas relacións que existen entre os CI para que, cando se realice un cambio de versión, se coñeza con certeza que accións ou consideracións cómpre ter en conta e a que outros CI se está a afectar, ademais de manter os rexistros actualizados na CMDB.

### **12.2.2 ENTREGA DE SERVIZOS.**

Considerado outro dos eixes fundamentais da xestión do servizo de TI, o Service Delivery concéntrase en describir todos os aspectos que se deben ter en conta para realizar unha planificación e mellora continua do servizo de TI a longo prazo e en todos os procesos que interveñen para que a prestación do servizo se manteña e se subministre de tal maneira que satisfaga as necesidades actuais e futuras do negocio. Algúns dos aspectos que describe están relacionados coa xestión dos niveis do



servizo, os niveis de seguridade requiridos, a viabilidade financeira dos servizos, a súa capacidade, continuidade e dispoñibilidade, entre outros.

### **12.2.2.1 PROCESOS**

#### Availability Management

Este proceso encárgase de garantir que os servizos de TI poidan ser accedidos dun xeito fiable e estean dispoñibles e funcionando correctamente cada vez que os clientes ou usuarios así o requiran, enmarcados nos SLA que se definiran para a prestación do servizo.

Dentro dos obxectivos definidos para este proceso encóntrase realizar o deseño dos servizos de TI co nivel de dispoñibilidade requirido polo negocio, garantir que exista unha dispoñibilidade non só nos servizos de TI senón tamén na súa infraestrutura, de tal forma que cumpra cos SLA establecidos, xerar noticias de dispoñibilidade que demostren a fiabilidade e mantibilidade do sistema e minimizar a frecuencia e duración que tarda a solución dun incidente.

#### Capacity Management

O obxectivo deste proceso é asegurar a existencia de certa capacidade a nivel de infraestrutura de TI, que se debe encontrar dispoñible constantemente para satisfacer os requirimentos do negocio a nivel do volume de transaccións, o tempo de proceso, o tempo de resposta e ante todo contemplando a súa viabilidade cuantitativa e económica para non incorrer en custos desproporcionados.

Como o seu nome indica, a xestión da capacidade concéntrase en verificar e garantir que todos os servizos de TI estean soportados coa suficiente capacidade de proceso e almacenamento e que ademais estea dimensionada de tal maneira que non implique custos innecesarios para a organización, pero que tampouco xere insatisfacción nos clientes ou usuarios debido á escasa calidade na prestación do servizo.



### Financial Management for IT Services

Céntrase en realizar un axeitado manexo do recurso financeiro (ingresos e gastos) asociado ao que implica a prestación dos servizos de TI, sempre enfocándose no cumprimento dos SLA definidos; determina cal é o manexo financeiro asociado para cada un dos recursos que participan na subministración dun servizo, buscando manter un balance permanente. Mantén unha estreita relación coa xestión da capacidade, a xestión da configuración e a xestión de niveis de servizo, e a través da información que cada un deles prové, é posible determinar exactamente cal é o custo real dun servizo.

### IT Service Continuity Management

A función principal deste proceso é evitar que unha grave e imprevista interrupción no servizo atente contra a continuidade do negocio, polo que se centra na preparación e planificación das medidas que se deben tomar para recuperar o servizo en caso de que aconteza algún desastre.

Busca asegurar a dispoñibilidade do servizo a través da toma de medidas preventivas que se oriente a reducir a probabilidade de fallas e no caso de que aconteza algún fenómeno considerado como catastrófico ou desastre, o servizo poida ser restablecido no menor tempo posible e coas menores perdas de información para a organización.

### Service Level Management (SLM)

Encárgase de definir os servizos de TI ofrecidos, formalizándoos a través de acordos de niveis de servizo (SLA) e acordos de nivel operativo (SLO); realiza unha avaliación do impacto que ocasionan os cambios sobre a calidade do servizo e os SLA despois de que estes cambios sexan propostos e implantados, asegurando deste xeito que calquera impacto negativo sobre a calidade nos servizos de TI sexa relativamente



baixo; tamén se encarga da creación de plans e emisión de informes respecto á calidade do servizo que se está a ofrecer.

Explica a importancia de establecer unha boa relación cos clientes e deste xeito de asegurar que as necesidades das empresas sexan entendidas; é por isto que o Service Level Management se enfoca tamén en coñecer as necesidades dos clientes, en definir correctamente os servizos que serán ofrecidos e monitorar a calidade dos servizos ofrecidos por medio dos SLA definidos.

Algúns dos aspectos máis importantes que se deben considerar na definición dos SLA están relacionados coa descrición do servizo e as súas características de funcionamento, coa dispoñibilidade do servizo, é dicir, cal é o tempo que a organización se compromete a manter o servizo dispoñible aos seus clientes, para o cal tamén é indispensable que sexan acordados tempos de reacción (mínimos e máximos) na resolución de incidentes, de aí que os SLA dependan da solución dos incidentes nos tempos acordados. Outro aspecto que se debe ter en conta ten que ver cos obxectivos de dispoñibilidade, seguridade e continuidade do servizo, as obrigas que recaen tanto nos clientes como nos provedores e as horas críticas do negocio, entre outros.

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

### **12.3.1 QUE É ISO 20.000?**

ITIL mostra todo o que se debe facer para que os usuarios ofrezan servizos de TI axeitados cumprindo os procesos da súa empresa. Para persoas individuais é posible obter certificacións de ITIL pero ata o momento non foi posible para unha organización de TI presentar probas de que traballa segundo as recomendacións de ITIL.



As normas ISO foron concibidas para encher este baleiro. Sobre a base de ITIL as organizacións itSMF e BSI (British Standard Institute) elaboraron unha normativa que define os requisitos da xestión de servizos a organizacións de TI.

Na actualidade, a normativa do BSI coñécese internacionalmente como normativa ISO 20000 e une os enfoques de ITIL e COBIT. ISO 20000 ábrelles as portas ás organizacións de TI para que poidan obter por primeira vez unha certificación.

Aquelas organizacións que aspiren a lograr unha certificación segundo ISO 20000 deben cumprir os requisitos formulados na normativa —UNE-ISO/IEC 20000, Parte 1: Especificacións, nos que se fixan os requisitos obrigatorios que debe cumprir toda organización que desexe unha certificación segundo esta normativa.

Os requisitos centrais da normativa ISO 20000 para unha organización de TI son:

- O aliñamento dos procesos de TI segundo as normas de ISO 20000, que corresponden esencialmente ás recomendacións da xestión de servizo de ITIL (en especial despois da introdución de ITIL V3).
- O uso dun método de xestión na organización de TI segundo as normas ISO 9001, baseado nos principios da xestión de procesos e dirixido a unha mellora continua da calidade.

A norma ISO/IEC 20000 está estruturada en dous documentos:

- ISO/IEC 20000-1. Este documento da norma inclúe o conxunto dos "requisitos obrigatorios" que debe cumprir o provedor de servizos TI, para realizar unha xestión eficaz dos servizos que responda ás necesidades das empresas e os seus clientes.



- ISO/IEC 20000-2. Esta parte contén un código de prácticas para a xestión de servizos ("Code of Practice for Service Management") que trata cada un dos elementos contemplados na parte 1 analizando e aclarando o seu contido. En síntese este documento pretende axudar ás organizacións a establecer os procesos de forma que cumpran cos obxectivos da parte 1.

### **12.3.2 UTILIDADES DO CERTIFICADO ISO 20.000**

Un certificado ISO 20000 demostra que unha organización de TI:

- está orientada ás necesidades dos clientes,
- está en condicións de prestar servizos que cumpren cos obxectivos de calidade fixados,
- utiliza os seus recursos de forma económica.

Este certificado supón sempre unha vantaxe sobre a competencia. Cada vez hai máis clientes que esperan unha certificación ISO 20000 do seu provedor de TI, co cal o certificado se converte nunha condición imprescindible para gañar cota de mercado.

Pero tamén para a empresa mesma, traballar segundo os principios diso 20000 (e ITIL) leva consigo unha serie de beneficios. A normativa ten como obxectivo prover os negocios cos servizos de TI que realmente necesite e ocuparse sempre de que isto suceda de forma eficiente.

Empezar unha iniciativa ISO 20000 é unha boa forma de impulsar a introdución de mellores prácticas na organización de TI e manter co tempo a motivación para a súa implantación.

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



ISO 20000 fixa requisitos aos procesos sen ocuparse de como deben ser conformados tales procesos de forma concreta.

Aí é onde aparece ITIL en escena: ITIL (e máis especialmente a nova versión 3) oríéntase á normativa ISO 20000 e presenta un grande abano de recomendacións de mellores prácticas, o que supón unha base de partida ben fundamentada para deseñar procesos conforme a ISO 20000.

A introdución de ITIL é polo tanto a mellor forma de prepararse para unha certificación ISO 20000.

#### **12.3.4 QUE REPRESENTA EXACTAMENTE SER CONFORME A ISO 20000?**

Obter unha certificación ISO 20000 é un proxecto laborioso. A condición máis importante ao iniciar un proxecto así é determinar que obxectivo se persegue. Concretamente debe responderse á pregunta: Como debe ser a organización de TI ao final do proxecto?

A normativa deixa aberta non obstante esta cuestión, cinguíndose só a nomear os requisitos sen especificar como se deben cumprir. Por iso non existe unha resposta válida á pregunta sobre o que representa "ser conforme a ISO 20000"

Así as cousas, non é de estrañar que ao empezar unha iniciativa ISO 20000 se faga evidente un gran problema: Non queda claro como debe estruturarse de forma concreta o labor dunha organización de TI para cumprir os requirimentos da normativa ISO e polo tanto non é doado determinar os cambios necesarios para iso.

Aquí é onde ITIL pode prestar unha axuda decisiva xa que ISO 20000 está orientada a ITIL.



Os coñecementos sobre ITIL adquirense normalmente mediante libros ou, de forma alternativa, co mapa de procesos ITIL® V3 en combinación co ITIL - ISO 20000 Bridge

## **12.4 DESEÑO DO MAPA DE PROCESOS ITIL**

O mapa de procesos ITIL® V3 é un modelo de referencia íntegro de ITIL. Contén a descrición completa de forma gráfica de todos os procesos estándar na xestión de servizos de TI segundo ITIL V3. O modelo de procesos mostra como funciona ITIL na práctica: afórralle traballo á hora de converter en procesos implantables as múltiples normas recollidas na bibliografía sobre ITIL.

O mapa de procesos ITIL® V3 foi creado para organizacións de TI e provedores de servizos de TI que

- teñan previsto introducir por primeira vez, parcial ou totalmente, a xestión de servizos de TI segundo ITIL V3,
- desexen volver valorar os procesos xa introducidos de ITIL sobre a base de ITIL V3
- queiran orientarse segundo ISO 20000 e/ou
- estean a se preparar para unha certificación segundo ISO 20000.

### **12.4.1 UTILIDADE DO MODELO DE REFERENCIA DE ITIL**

O mapa de procesos ITIL® V3 está organizado para prestar o apoio óptimo en todos os pasos de calquera proxecto ITIL ou ISO 20000, dende a primeira planificación ata unha organización de TI que funcione segundo os principios das mellores prácticas. O seu uso ofrece vantaxes decisivas:

- O tratamento gráfico e navegable dos contidos de ITIL facilita a comprensión dos procesos ITIL e das súas interrelacións. Co modelo de procesos de ITIL pode aclarar



ITIL a todos os colaboradores da súa organización de TI dunha forma moi efectiva e económica.

- Os modelos de procesos, claramente estruturados, e as guías complementarias serven de fío condutor á hora de incorporar o seu proxecto á implantación de ITIL e levalo a cabo. Na definición e documentación de procesos reducirase o seu traballo, xa que adaptará os procesos de referencia existentes ás necesidades da súa organización sen ter que empezar cunha folla en branco.
- O mapa de procesos é unha documentación de procesos profesional coa que a xestión de TI estará na vantaxosa situación de poder demostrarlles aos clientes que a organización de TI realiza o seu labor de forma planificada, orientada ao cliente e de calidade.



Bibliografía:

Sitios web:

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

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

<http://iso20000enespanol.com/>

Autor:

Ramón Seoane Freijido

Director de Sistemas de Información Molduras del Noroeste

Colexiado do CPEIG





**XUNTA  
DE GALICIA**



*ESCOLA GALEGA  
DE ADMINISTRACIÓN  
PÚBLICA*

# **ASPECTOS LEGAIS**



**13. LEI DE ACCESO  
ELECTRÓNICO DOS CIDADÁNS  
AOS SERVIZOS PÚBLICOS.  
DECRETO 198/2010 POLO QUE SE  
REGULA O DESENVOLVEMENTO  
DA ADMINISTRACIÓN  
ELECTRÓNICA NA XUNTA DE  
GALICIA E NAS ENTIDADES  
DEPENDENTES. REAL DECRETO  
3/2010 POLO QUE SE REGULA O  
ESQUEMA NACIONAL DE  
SEGURIDADE NO ÁMBITO DA  
ADMINISTRACIÓN ELECTRÓNICA.  
REAL DECRETO 4/2010 POLO QUE  
SE REGULA O ESQUEMA  
NACIONAL DE  
INTEROPERABILIDADE NO  
ÁMBITO DA ADMINISTRACIÓN  
ELECTRÓNICA.**



**Tema 13.- Lei de acceso electrónico dos cidadáns aos servizos públicos. Decreto 198/2010 polo que se regula o desenvolvemento da Administración electrónica na Xunta de Galicia e as súas entidades dependentes. Real decreto 3/2010 polo que se regula o esquema nacional de seguridade no ámbito da Administración electrónica. Real decreto 4/2010 polo que se regula o esquema nacional de interoperabilidade no ámbito da Administración electrónica.**

## **ÍNDICE**

### 13.1 Lei de acceso electrónico dos cidadáns aos servizos públicos

#### 13.1.1 Introducción

#### 13.1.2 Obxecto da lei

#### 13.1.3 Ámbito de aplicación

#### 13.1.4 Finalidades

#### 13.1.5 Principios

### 13.2 Decreto 198/2010 polo que se regula o desenvolvemento da Administración electrónica na Xunta de Galicia e as súas entidades dependentes.

#### 13.2.1 Introducción

#### 13.2.2 Obxecto

#### 13.2.3 Estrutura

### 13.3 Real decreto 3/2010 polo que se regula o esquema nacional de seguridade no ámbito da Administración electrónica.

#### 13.3.1 Introducción

#### 13.3.2 Principios

#### 13.3.3 Obxectivos

#### 13.3.4 Ámbito de aplicación

### 13.4 Real decreto 4/2010 polo que se regula o esquema nacional de interoperabilidade no ámbito da Administración electrónica.

#### 13.4.1 Introducción

#### 13.4.2 Obxectivos



### 13.4.3 Ámbito de aplicación e análise

## **1.- A LEI 11/2007 DE ACCESO ELECTRÓNICO DOS CIDADÁNS AOS SERVIZOS PÚBLICOS. A CALIDADE DOS SERVIZOS PÚBLICOS E DE ATENCIÓN AO CIDADÁN.**

### 1.1 Introducción

Durante os últimos anos producíronse numerosos cambios moi importantes nas relacións entre Administración e Goberno e cidadáns; o progreso social, económico e tecnolóxico fomentaron o desexo de cambio e apuraron a Administración para se adaptar aos novos problemas, ás novas competencias e ás necesidades cidadás.

A Administración pública, no cumprimento do seu deber de servir con obxectividade os intereses xerais e actuar de acordo cos principios de eficacia, xerarquía, descentralización, desconcentración e coordinación con sometemento pleno á lei e ao Dereito (art. 103.1 CE), debe ser aquí unha peza fundamental para a implantación das políticas de modernización das administracións públicas. Ten que ser capaz de se adaptar ás novas realidades para formar parte do proceso de desenvolvemento económico e social das sociedades occidentais.

A mellora e, consecuentemente, a modernización das administracións públicas debe ser un proceso continuo, dinámico e constante, no que participen todos os que forman parte do sector público.

En España, o punto de partida supúxoo o Acordo do Consello de Ministros do 15 de novembro de 1991, xa que en 1992 se aprobaría o Plan de Modernización da Administración do Estado, composto por



unha serie de medidas que tiñan como obxectivo mellorar e modernizar a Administración pública para responder ás necesidades cambiantes dos cidadáns.

Na actualidade, as políticas de modernización están estreitamente ligadas ao desenvolvemento da Administración electrónica.

Seguindo a definición dada pola Comisión Europea: a Administración electrónica non é senón “o uso das tecnoloxías nas administracións públicas, combinado con cambios organizativos e novas aptitudes, co fin de mellorar os servizos públicos e os procesos democráticos e reforzar o apoio ás políticas públicas”.

Actualmente un elemento artellador do desenvolvemento das tecnoloxías na Administración constitúeo o denominado PLAN AVANZA II 2011-2015 (aprobado polo Consello de Ministros do 16 de xullo do 2010).

Tomando como punto de partida o Plan Avanza aprobado no ano 2005, así como o marco europeo en que se encadran este tipo de iniciativas, identificáronse 34 retos concretos que debe abordar España no ámbito das TIC. Neste contexto, a Estratexia 2011-2015 do Plan Avanza 2 vai centrar os seus esforzos na consecución dos seguintes 10 obxectivos que facilitarán a superación dos retos definidos:

1. Promover procesos innovadores TIC (tecnoloxías da información e da comunicación) nas administracións públicas.
2. Estender as TIC na sanidade e o benestar social.
3. Potenciar a aplicación das TIC no sistema educativo e formativo.



4. Mellorar a capacidade e a extensión das redes de telecomunicacións.

5. Estender a cultura da seguridade entre a cidadanía e as empresas.

6. Incrementar o uso avanzado de servizos dixitais pola cidadanía.

7. Estender o uso de solucións TIC de negocio na empresa.

8. Desenvolver as capacidades tecnolóxicas do sector TIC.

9. Fortalecer o sector de contidos dixitais garantindo a mellor protección da propiedade intelectual no actual contexto tecnolóxico e dentro do marco xurídico español e europeo.

10. Desenvolver as TIC verdes.

Dentro dese campo de actuación xoga un papel decisivo no seu desenvolvemento a Lei 11/2007, do 22 de xuño, de acceso electrónico dos cidadáns aos servizos públicos.

A Lei 30/1992, do 26 de novembro, de réxime xurídico das administracións públicas e do procedemento administrativo común (LRXAP-PC) na súa primeira versión recolleu, no seu artigo 45, o impulso ao emprego e aplicación das técnicas e medios electrónicos, informáticos e telemáticos, por parte da Administración para o desenvolvemento da súa actividade e o exercicio das súas competencias, e que lles permitiu aos cidadáns se relacionaren coas administracións cando fose compatible cos “medios técnicos de que dispoñan”. Ademais, o seu artigo 38, e posteriormente a Lei 24/2001, pasaron da informatización dos rexistros e arquivos aos rexistros telemáticos como forma de relacionarse coa Administración, sempre que o interesado sinalase este medio como preferente.

Pásase dunha declaración de impulso da Administración electrónica á obriga de empregar medios telemáticos, xa que a Lei



11/2007 reconece o dereito dos cidadáns a se relacionaren coa Administración a través destes medios.

O servizo ao cidadán esixe consagrar o seu dereito a se comunicaren coas Administracións por medios electrónicos, xa que estas están obrigadas a facelo mediante o recoñecemento da Lei do dereito dos cidadáns a establecer relacións electrónicas. Por iso, cada Administración (estatal, autonómica e local) débelle facilitar ao cidadán, entre outros:

- O acceso á información e servizos da súa competencia.
- Presentar solicitudes e recursos.
- Os medios para se dirixiren ás demais administracións, o que implica a colaboración entre administracións.
- Efectuar pagamentos.
- Acceder ás notificacións e comunicacións que lles remita a Administración.
- Atopar información nun punto de acceso único sobre os servizos multicanle ou aqueles que se ofrezan por máis dun medio ou plataforma.

Os puntos máis destacables da Lei son:

- Os cidadáns verán recoñecidos novos dereitos nas súas relacións coas administracións públicas.
- A creación da figura do Defensor do Usuario.
- As administracións terán a obriga de facer estes dereitos efectivos a partir do 2009.
- Os trámites e xestións poderanse facer desde calquera lugar, en calquera momento.
- A Administración será máis fácil, máis áxil e máis eficaz.
- Os cidadáns pasan a tomar o mando nas súas relacións coa Administración.



- É unha lei de consenso. Na súa elaboración participaron todas as administracións, de cidadáns, de partidos, de empresas e asociacións.

## 1.2 Obxecto da Lei

- Recoñecer o dereito dos cidadáns a se relacionaren coas Administracións públicas por medios electrónicos.

- Regular os aspectos básicos da utilización das tecnoloxías da información na actividade administrativa, nas relacións entre as administracións públicas, así como nas relacións dos cidadáns con estas, coa finalidade de garantir os seus dereitos, un tratamento común ante elas e a validez e eficacia da actividade administrativa en condicións de seguridade xurídica.

- Utilizar, por parte das AA.PP., as tecnoloxías da información de acordo co disposto na Lei, asegurando a dispoñibilidade, o acceso, a integridade, a autenticidade, a confidencialidade e a conservación dos datos, informacións e servizos que xestionen no exercicio das súas competencias.

## 1.3 Ámbito de aplicación (disposición derradeira primeira)

- As administracións públicas, entendendo por tales a Administración xeral do Estado, as administracións das comunidades autónomas e as entidades que integran a Administración local, así como as entidades de dereito público vinculadas ou dependentes delas.

- Os cidadáns nas súas relacións coas administracións públicas.
- As relacións entre as distintas administracións públicas.

A Lei non será de aplicación ás administracións públicas nas actividades que desenvolvan en réxime de dereito privado.



#### 1.4 Finalidades da Lei

- Facilitar o exercicio de dereitos e o cumprimento de deberes por medios electrónicos.
- Facilitar o acceso por medios electrónicos dos cidadáns á información e ao procedemento administrativo, con especial atención á eliminación das barreiras que limiten este acceso.
- Crear as condicións de confianza no uso dos medios electrónicos, establecendo as medidas necesarias para a preservación da integridade dos dereitos fundamentais, e en especial os relacionados coa intimidade e a protección de datos de carácter persoal, por medio da garantía da seguridade dos sistemas, os datos, as comunicacións, e os servizos electrónicos.
- Promover a proximidade co cidadán e a transparencia administrativa, así como a mellora continua na consecución do interese xeral.
- Contribuír á mellora do funcionamento interno das administracións públicas, incrementando a súa eficacia e a eficiencia mediante o uso das tecnoloxías da información, coas debidas garantías legais na realización das súas funcións.
- Simplificar os procedementos administrativos e proporcionar oportunidades de participación e maior transparencia, coas debidas garantías legais.
- Contribuír ao desenvolvemento da sociedade da información no ámbito das administracións públicas e na sociedade en xeral.

#### 1.5 Principios xerais

- Limitacións da utilización das tecnoloxías da información:
- As establecidas pola Constitución.
- O resto do ordenamento xurídico.
- Principios:



1. O respecto ao dereito á protección de datos de carácter persoal nos termos establecidos pola Lei orgánica 15/1999, de protección dos datos de carácter persoal, nas demais leis específicas que regulan o tratamento da información e nas súas normas de desenvolvemento, así como aos dereitos ao honor e á intimidade persoal e familiar.

2. Principio de igualdade con obxecto de que en ningún caso o uso de medios electrónicos poida implicar a existencia de restricións ou discriminacións para os cidadáns que se relacionen coas administracións públicas por medios non electrónicos, tanto respecto do acceso á prestación de servizos públicos como respecto de calquera actuación ou procedemento administrativo sen prexuízo das medidas dirixidas a incentivar a utilización dos medios electrónicos.

3. Principio de accesibilidade á información e aos servizos por medios electrónicos nos termos establecidos pola normativa vixente nesta materia, a través de sistemas que permitan obtelos de xeito seguro e comprensible, garantindo especialmente a accesibilidade universal e o deseño para todos dos soportes, canles e contornos con obxecto de que todas as persoas poidan exercer os seus dereitos en igualdade de condicións, incorporando as características necesarias para garantir a accesibilidade daqueles colectivos que o requiran.

4. Principio de legalidade en canto ao mantemento da integridade das garantías xurídicas dos cidadáns ante as administracións públicas establecidas na Lei 30/1992, de réxime xurídico das administracións públicas e do procedemento administrativo común.

5. Principio de cooperación na utilización de medios electrónicos polas administracións públicas co obxecto de garantir tanto a interoperabilidade dos sistemas e solucións adoptados por cada unha delas como, se é o caso, a prestación conxunta de servizos aos cidadáns. En particular, garantirase o recoñecemento mutuo dos



documentos electrónicos e dos medios de identificación e autenticación que se axusten ao disposto na presente Lei.

6. Principio de seguridade na implantación e utilización dos medios electrónicos polas administracións públicas, en virtude do cal se esixirá polo menos o mesmo nivel de garantías e seguridade que se require para a utilización de medios non electrónicos na actividade administrativa.

7. Principio de proporcionalidade, en virtude do cal só se esixirán as garantías e medidas de seguridade adecuadas á natureza e circunstancias dos distintos trámites e actuacións. Así mesmo, só se lles requirirán aos cidadáns aqueles datos que sexan estritamente necesarios en atención á finalidade para a que se soliciten.

## **2. DECRETO 198/2010 POLO QUE SE REGULA O DESENVOLVEMENTO DA ADMINISTRACIÓN ELECTRÓNICA NA XUNTA DE GALICIA E NAS SÚAS ENTIDADES DEPENDENTES:**

### **2.1 Introducción**

A administración da Comunidade Autónoma de Galicia non pode permanecer allea aos incesantes e cada vez máis frecuentes cambios no seo das relacións coa Administración desde o punto de vista tecnolóxico; é neste ámbito onde, tal e como se recolle na exposición de motivos do Decreto, se pretende conseguir unha Administración diferente, que terá á electrónica como elemento central na súa modernización, onde os seus efectos reais sobre a poboación irán encamiñados á utilización de medios e formas que reduzan a brecha tecnolóxica, creando as condicións de confianza precisas para o uso das tecnoloxías da información e da comunicación.

### **2.1 Obxecto**

Ten por obxecto regular o dereito dos cidadáns a se relacionaren coas administracións públicas por medios electrónicos, a tramitación



dos procedementos administrativos incorporados á tramitación telemática, a creación e regulación da sede electrónica, a creación da edición electrónica do Diario Oficial de Galicia e do Rexistro Electrónico, o impulso e desenvolvemento dos servizos electrónicos e o establecemento de infraestruturas e servizos de interoperabilidade.

### 2.3 Estrutura

O Decreto consta de 40 artigos, agrupados en nove capítulos, con tres disposicións adicionais, tres transitorias, unha derogatoria e catro derradeiras.

No capítulo I da norma recolle o seu obxecto, o de regular o dereito dos cidadáns a se relacionaren coas administracións públicas por medios electrónicos, a tramitación dos procedementos administrativos incorporados á tramitación telemática, a creación e regulación da sede electrónica, a creación da edición electrónica do Diario Oficial de Galicia e do Rexistro Electrónico, o impulso e desenvolvemento dos servizos electrónicos e o establecemento de infraestruturas e servizos de interoperabilidade.

Establece como medidas de carácter xeral

- Ordenar e impulsar a Administración electrónica, co fin de mellorar a eficiencia interna, as relacións intra- e interadministrativas e as relacións cos cidadáns.
- Garantir o dereito dos cidadáns a se relacionaren por medios electrónicos coa Administración pública autonómica.
- Contribuír ao desenvolvemento da sociedade da información no ámbito das administracións públicas de Galicia.
- Preservar a integridade dos dereitos fundamentais relacionados coa intimidade das persoas, para a garantía da



seguridade dos datos e das comunicacións e para a protección dos servizos prestados en soporte electrónico.

- Facilitar o acceso dos cidadáns aos servizos da Administración electrónica nas oficinas telemáticas integradas de atención aos cidadáns, baseadas na cooperación interadministrativa, ofrecéndolles servizos aos cidadáns en oficinas públicas, con independencia de cal sexa a Administración competente para coñecer o asunto.
- Posibilitar a intermediación entre administracións públicas para a resolución de trámites administrativos solicitados aos cidadáns cando sexan de competencia da Xunta de Galicia.

O capítulo II establece que a sede electrónica é o enderezo electrónico a través do cal os cidadáns acceden á información, servizos e trámites electrónicos, que representa unha fonte de información auténtica en que o organismo titular identificado coa sede garante responsablemente a integridade, veracidade e actualización da información e os servizos aos que se poida acceder a través desta.

O enderezo electrónico de referencia da sede electrónica da Xunta de Galicia será <https://sede.xunta.es>, que será accesible directamente, así como a través do portal [www.xunta.es](http://www.xunta.es), configurándose como un conxunto de páxinas web que asegurará:

- A calidade da información e a coherencia na navegación.
- A identificación e comunicación segura, mediante os correspondentes certificados electrónicos admitidos pola Xunta de Galicia.



- O acceso ao Rexistro Electrónico, ás comunicacións e notificacións e aos formularios para iniciar os procedementos administrativos ou solicitar a prestación de servizos.
- Os principios de accesibilidade de acordo coas normas establecidas, estándares abertos e, no seu caso, aqueloutros que sexan de uso xeral polos cidadáns.

O capítulo III regula a creación do Diario Oficial de Galicia na súa edición electrónica, que terá unha consideración de publicación única, dotándoa de validez xurídica, e que substituirá á edición impresa.

O capítulo IV trata sobre os mecanismos de identificación e autenticación, establecendo que os cidadáns poderán utilizar os seguintes instrumentos de identificación para se relacionaren coa Xunta de Galicia e as entidades incluídas no ámbito de aplicación deste Decreto:

a. En todo caso, os sistemas de sinatura electrónica incorporados ao documento nacional de identidade, para persoas físicas.

b. Sistemas de sinatura electrónica avanzada, incluíndo os baseados en certificado electrónico recoñecido, admitidos polas administracións públicas que teñan validez para a Xunta de Galicia e que se especifiquen na sede electrónica.

c. Sistemas de sinatura electrónica, como a utilización de claves concertadas nun rexistro previo como persoa usuaria inscrita no rexistro de funcionarios habilitados pola Xunta de Galicia.

d. Outros sistemas de identificación que resulten proporcionais e seguros para a identificación das persoas interesadas.



O capítulo V regula a tramitación de procedementos administrativos no ámbito da Administración electrónica, establecendo que a xestión electrónica da actividade administrativa respectará o exercicio e a titularidade do órgano ou entidade que teña atribuídas as súas competencias, así como a obrigatoriedade de impulsar a Administración electrónica.

Regula a iniciación e tramitación do procedemento por medios electrónicos, recoñecendo que calquera persoa interesada poderá iniciar e tramitar un procedemento administrativo por medios electrónicos, ante e en relación coa Xunta de Galicia ou as entidades incluídas no ámbito de aplicación deste Decreto, conforme ás previsións destas e sen outras limitacións que as establecidas nas normas e protocolos de aplicación en atención a razóns tecnolóxicas.

Os capítulos VI e VII regulan os aspectos da xestión e tramitación dos procedementos administrativos, tanto no ámbito interno, relativo a comunicacións e notificacións, onde establece que as entidades incluídas no ámbito de aplicación do presente Decreto utilizarán un sistema de notificación electrónica que acredite a data e hora da posta a disposición da persoa interesada do acto obxecto de notificación, así como a data e hora de acceso desta ao seu contido mediante sistemas de selado de tempo, como no ámbito externo relacionado, en concreto coas copias e documentos electrónicos, definindo o documento electrónico nos termos que se recollen no anexo da Lei 11/2007, de administración electrónica: “Información de calquera natureza en forma electrónica, arquivada nun soporte electrónico segundo un formato determinado e susceptible de identificación e tratamento diferenciado.”



O capítulo VIII trata sobre a interoperabilidade, ten por obxecto fomentar a cooperación interadministrativa; figura clave neste é o denominado protocolo de interoperabilidade, que é o documento que determinará o procedemento para incorporar e consumir a información en soporte electrónico das entidades incluídas no ámbito de aplicación do presente Decreto.

O capítulo IX concreta as funcións que o órgano de dirección con competencias xerais en materia de desenvolvemento da Administración electrónica leva a cabo en relación con este Decreto, en desenvolvemento das competencias e funcións que lle atribúe o Decreto 325/2009, do 18 de xuño, de estrutura orgánica dos órganos superiores dependentes da Presidencia da Xunta de Galicia para o impulso, xestión e coordinación da Administración electrónica como elemento indispensable para a modernización da Administración pública, a dirección e xestión de todas as actuacións da Xunta en materia de tecnoloxías da información e as comunicacións e o establecemento de directrices tecnolóxicas que deben seguir todos os órganos da Xunta de Galicia.

### **3. REAL DECRETO 3/2010 POLO QUE SE REGULA O ESQUEMA NACIONAL DE SEGURIDADE NO ÁMBITO DA ADMINISTRACIÓN ELECTRÓNICA.**

#### **3.1 Introducción**

Ten por obxecto regular o Esquema Nacional de Seguridade establecido no artigo 42 da Lei 11/2007, do 22 de xuño, e determinar a política de seguridade que se debe aplicar na utilización dos medios electrónicos aos que se refire a citada Lei, e recolle e regula os principios básicos e requisitos mínimos que permitan unha protección adecuada da información.



A finalidade do Esquema Nacional de Seguridade é crear as condicións necesarias para a confianza no uso dos medios electrónicos, a través de medidas para garantir a seguridade dos sistemas, dos datos, das comunicacións, e dos servizos electrónicos, que permita o exercicio de dereitos e o cumprimento de deberes a través destes medios. Persegue fundamentar a confianza en que os sistemas de información prestarán os seus servizos e custodiarán a información de acordo coas súas especificacións funcionais, sen interrupcións ou modificacións fóra de control e sen que a información poida chegar ao coñecemento de persoas non autorizadas.

Co obxecto de crear estas condicións, o Esquema Nacional de Seguridade introduce os elementos comúns que deben guiar a actuación das administracións públicas en materia de seguridade das tecnoloxías da información. En particular, introduce os seguintes elementos principais:

Os principios básicos que cómpre ter en conta nas decisións en materia de seguridade.

Os requisitos mínimos que permitan unha protección adecuada da información.

O mecanismo para lograr o cumprimento dos principios básicos e requisitos mínimos mediante a adopción de medidas de seguridade proporcionadas á natureza da información, ao sistema e aos servizos que cómpre protexer.

Ten en conta as recomendacións da Unión Europea, a situación tecnolóxica das diferentes administracións públicas, así como os servizos electrónicos xa existentes, e a utilización de estándares abertos, así como, de ser o caso, e de forma complementaria, estándares que sexan de uso xeneralizado polos cidadáns.



Na súa elaboración manexáronse, entre outros, referentes en materia de seguridade tales como directrices e guías da OCDE, recomendacións da Unión Europea, normalización nacional e internacional, normativa sobre Administración electrónica, protección de datos de carácter persoal, sinatura electrónica e documento nacional de identidade electrónico, así como referentes doutros países.

Realizouse nun proceso coordinado polo Ministerio da Presidencia co apoio do Centro Criptolóxico Nacional (CCN), coa participación de todas as administracións públicas. Ao longo dos últimos tres anos, máis dun centenar de expertos das administracións públicas colaboraron na súa elaboración, aos que hai que sumar os numerosos expertos que tamén expresaron a súa opinión a través das asociacións profesionais do sector TIC; todo iso á luz do estado da arte e dos principais referentes en materia de seguridade da información.

### 3.2 Principios básicos do Esquema Nacional de Seguridade

O obxecto último da seguridade da información é asegurar que unha organización administrativa poderá cumprir os seus obxectivos empregando sistemas de información. Nas decisións en materia de seguridade deberánse ter en conta os seguintes principios básicos:

- a) Seguridade integral.
- b) Xestión de riscos.
- c) Prevención, reacción e recuperación.
- d) Liñas de defensa.
- e) Reavaliación periódica.
- f) Función diferenciada.



Recolle o Real decreto os requisitos mínimos que deberán ter os sistemas de seguridade, e así o artigo 11 establece que “Todos os órganos superiores das administracións públicas deberán dispoñer formalmente da súa política de seguridade, que será aprobada polo titular do órgano superior correspondente. Esta política de seguridade establecerase en función dos principios básicos indicados e desenvolverase aplicando os seguintes requisitos mínimos:

- a) Organización e implantación do proceso de seguridade.
- b) Análise e xestión dos riscos.
- c) Xestión de persoal.
- d) Profesionalidade.
- e) Autorización e control dos accesos.
- f) Protección das instalacións.
- g) Adquisición de produtos.
- h) Seguridade por defecto.
- i) Integridade e actualización do sistema.
- j) Protección da información almacenada e en tránsito.
- k) Prevención ante outros sistemas de información interconectados.
- l) Rexistro de actividade.
- m) Incidentes de seguridade.
- n) Continuidade da actividade.
- o) Mellora continua do proceso de seguridade.

### 3.3 Obxectivos

Crear as condicións necesarias de confianza no uso dos medios electrónicos, a través de medidas para garantir a seguridade dos sistemas, dos datos, das comunicacións, e dos servizos electrónicos, que lles permita aos cidadáns e ás administracións públicas o exercicio de dereitos e o cumprimento de deberes a través destes medios.



Establecer a política de seguridade na utilización de medios electrónicos no ámbito da Lei 11/2007, que estará constituída polos principios básicos e os requisitos mínimos para unha protección adecuada da información.

Introducir os elementos comúns que deben guiar a actuación das administracións públicas en materia de seguridade das tecnoloxías da información.

Proporcionar unha linguaxe común para facilitar a interacción das administracións públicas, así como a comunicación dos requisitos de seguridade da información á industria.

No Esquema Nacional de Seguridade concíbese a seguridade como unha actividade integral, onde non caben actuacións puntuais ou tratamentos conxunturais, debido a que a debilidade dun sistema determínaa o seu punto máis fráxil e, a miúdo, este punto é a coordinación entre medidas individualmente adecuadas mais deficientemente ensambladas.

Dada a natureza da seguridade, a consecución destes obxectivos require un desenvolvemento que teña en conta a complexidade técnica, a obsolescencia da tecnoloxía subxacente e o importante cambio que supón na operativa da Administración a aplicación da Lei 11/2007.

### 3.4 Ámbito de aplicación

O seu ámbito de aplicación é o establecido no artigo 2 da Lei 11/2007, do 22 de xuño, é dicir, tanto as administracións públicas, entendendo por tales a Administración xeral do Estado, as administracións das comunidades autónomas e as entidades que integran a Administración local, como as entidades de dereito público vinculadas ou dependentes das mesmas, os cidadáns nas súas



relacións coas administracións públicas e as relacións entre as distintas administracións públicas.

Estarán excluídos os sistemas que tratan información clasificada regulada pola Lei 9/1968, do 5 de abril, de segredos oficiais, modificada pola Lei 48/1978, do 7 de outubro, e normas de desenvolvemento.

#### **4. REAL DECRETO 4/2010 POLO QUE SE REGULA O ESQUEMA NACIONAL DE INTEROPERABILIDADE NO ÁMBITO DA ADMINISTRACIÓN ELECTRÓNICA.**

##### **4.1 Introducción**

O Esquema Nacional de Interoperabilidade persegue a creación das condicións necesarias para garantir o adecuado nivel de interoperabilidade técnica, semántica e organizativa dos sistemas e aplicacións empregados polas administracións públicas, que permita o exercicio de dereitos e o cumprimento de deberes a través do acceso electrónico aos servizos públicos, ao mesmo tempo que redunda en beneficio da eficacia e da eficiencia.

Co obxecto de crear estas condicións, o Esquema Nacional de Interoperabilidade introduce os elementos comúns que deben guiar a actuación das administracións públicas en materia de interoperabilidade. En particular, introduce os seguintes elementos principais:

- Enúncianse os principios específicos da interoperabilidade.
- Contémplanse as dimensións da interoperabilidade organizativa, semántica e técnica ás que se refire o artigo 41 da Lei 11/2007, do 22 de xuño.



- Trátanse as infraestruturas e os servizos comúns, elementos recoñecidos de dinamización, simplificación e propagación da interoperabilidade, e tamén facilitadores da relación multilateral.

- Trátase a reutilización, aplicada ás aplicacións das administracións públicas, da documentación asociada e doutros obxectos de información, dado que a voz ‘compartir’ se atopa presente na definición de interoperabilidade recollida na Lei 11/2007, do 22 de xuño, e xunto coa voz ‘reutilizar’, as dúas son relevantes para a interoperabilidade e están entroncadas coas políticas da Unión Europea en relación coa idea de compartir, reutilizar e colaborar.

- Trátase a interoperabilidade da sinatura electrónica e dos certificados.

- Aténdese á recuperación e conservación do documento electrónico, segundo o establecido na citada Lei 11/2007, do 22 de xuño, como manifestación da interoperabilidade ao longo do tempo, e que afecta de forma singular ao documento electrónico.

- Para rematar, créanse as normas técnicas de interoperabilidade e os instrumentos para a interoperabilidade, para facilitar a aplicación do Esquema.

Ten en conta as recomendacións da Unión Europea, a situación tecnolóxica das diferentes administracións públicas, así como os servizos electrónicos xa existentes, e a utilización de estándares abertos, así como, de ser o caso, e de forma complementaria, estándares que sexan de uso xeneralizado polos cidadáns.

Na súa elaboración manexáronse, entre outros, referentes en materia de desenvolvemento da Administración electrónica e, en particular, de interoperabilidade provenientes do ámbito da Unión Europea, de actuacións semellantes noutros países, da normalización nacional e internacional, así como a normativa sobre Administración electrónica, protección de datos de carácter persoal, sinatura



electrónica e documento nacional de identidade electrónico, entre outros.

Realizouse nun proceso coordinado polo Ministerio da Presidencia, coa participación de todas as administracións públicas. Elaborouse coa participación de todas as administracións públicas. Ao longo dos últimos tres anos colaboraron na súa elaboración máis dun centenar de expertos das administracións públicas, aos que hai que sumar os numerosos expertos que tamén expresaron a súa opinión a través das asociacións profesionais do sector TIC; todo iso á luz do estado da arte e dos principais referentes en materia de interoperabilidade.

#### 4.2 Obxectivos

Os seus obxectivos son os seguintes:

- Comprender os criterios e recomendacións que deberán ser tidos en conta polas administracións públicas para a toma de decisións tecnolóxicas que garantan a interoperabilidade e que eviten a discriminación dos cidadáns por razón da súa elección tecnolóxica.

- Introducir os elementos comúns que deben guiar a actuación das administracións públicas en materia de interoperabilidade.

- Proporcionar unha linguaxe común para facilitar a interacción das administracións públicas, así como a comunicación dos requisitos de interoperabilidade á industria.

A interoperabilidade concíbese, en consecuencia, desde unha perspectiva integral, de maneira que non caben actuacións puntuais ou tratamentos conxunturais, debido a que a debilidade dun sistema determínaa o seu punto máis fráxil e, a miúdo, este punto é a coordinación entre medidas individualmente adecuadas mais deficientemente ensambladas.



Dada a natureza da interoperabilidade, a consecución destes obxectivos require un desenvolvemento que teña en conta a complexidade técnica, a obsolescencia da tecnoloxía subxacente e o importante cambio que supón na operativa da Administración a aplicación da Lei 11/2007.

#### 4.3 Ámbito de aplicación e Análise

O seu ámbito de aplicación é o establecido no artigo 42 da Lei 11/2007, do 22 de xuño, de acceso electrónico dos cidadáns aos servizos públicos, é dicir, ás administracións públicas, entendendo por tales a Administración xeral do Estado, as administracións das comunidades autónomas e as entidades que integran a Administración local, así como as entidades de dereito público vinculadas ou dependentes destas.

Os cidadáns nas súas relacións coas administracións públicas.

As relacións entre as distintas administracións públicas.

Non será de aplicación para as administracións públicas nas actividades que desenvolvan en réxime de dereito privado.

Como aspectos máis importantes que cómpre destacar atópanse:

1.- Regula de forma clara os dereitos dos cidadáns en relación coa utilización dos medios electrónicos na actividade administrativa, entre eles:

- A elixir, entre aqueles que en cada momento se atopen dispoñibles, a canle a través da cal se relacionar por medios electrónicos coas administracións públicas.



- A non proporcionar os datos e documentos que obren en poder das administracións públicas, que empregarán medios electrónicos para solicitar esta información sempre que, no caso de datos de carácter persoal, se conte co consentimento dos interesados nos termos establecidos pola Lei orgánica 15/1999, de protección de datos de carácter persoal, ou unha norma con rango de lei así o determine, salvo que existan restricións conforme á normativa de aplicación aos datos e documentos solicitados. O citado consentimento poderase emitir e solicitar por medios electrónicos.
- Á igualdade no acceso electrónico aos servizos das administracións públicas.
- A coñecer por medios electrónicos o estado de tramitación dos procedementos en que sexan interesados, salvo nos supostos en que a normativa de aplicación estableza restricións ao acceso á información sobre aqueles.
- A obter copias electrónicas dos documentos electrónicos que formen parte de procedementos en que teñan a condición de interesado.
- Á conservación en formato electrónico polas administracións públicas dos documentos electrónicos que formen parte dun expediente.
- A obter os medios de identificación electrónica necesarios, podendo as persoas físicas empregar en todo caso os sistemas de sinatura electrónica do documento nacional de



identidade para calquera trámite electrónico con calquera Administración pública.

- Á utilización doutros sistemas de sinatura electrónica admitidos no ámbito das administracións públicas.
- Á garantía da seguridade e confidencialidade dos datos que figuren nos ficheiros, sistemas e aplicacións das administracións públicas.
- Á calidade dos servizos públicos prestados por medios electrónicos.
- A elixir as aplicacións ou sistemas para se relacionaren coas administracións públicas, a condición de que empreguen estándares abertos ou, de ser o caso, aqueloutros que sexan de uso xeneralizado polos cidadáns.

Regula o réxime xurídico da Administración electrónica, define a sede electrónica como aquel enderezo electrónico dispoñible para os cidadáns a través de redes de telecomunicacións cuxa titularidade, xestión e administración lle corresponde a unha Administración pública, órgano ou entidade administrativa no exercicio das súas competencias.

Regula a identificación e autenticación, dispoñendo que as administracións públicas admitirán, nas súas relacións por medios electrónicos, sistemas de sinatura electrónica que sexan conformes ao establecido na Lei 59/2003, do 19 de decembro, de sinatura electrónica e resulten adecuados para garantir a identificación dos participantes e, se é o caso, a autenticidade e integridade dos documentos electrónicos.



Os cidadáns poderán utilizar os seguintes sistemas de sinatura electrónica para se relacionaren coas administracións públicas, de acordo co que cada Administración determine:

- En todo caso, os sistemas de sinatura electrónica incorporados ao documento nacional de identidade, para persoas físicas.
- Sistemas de sinatura electrónica avanzada, incluíndo os baseados en certificado electrónico recoñecido, admitidos polas administracións públicas.
- Outros sistemas de sinatura electrónica, como a utilización de claves concertadas nun rexistro previo como usuario, a achega de información coñecida por ambas as partes ou outros sistemas non criptográficos, nos termos e condicións que en cada caso se determinen.

Con relación aos rexistros, comunicacións e notificacións, dispón que as administracións públicas creasen rexistros electrónicos para a recepción e remisión de solicitudes, escritos e comunicacións.

Os rexistros electrónicos poderán admitir: documentos electrónicos normalizados correspondentes aos servizos, procedementos e trámites que se especifiquen conforme ao disposto na norma de creación do rexistro, cumprimentados de acordo con formatos preestablecidos, e ademais calquera solicitude, escrito ou comunicación distinta dos mencionados no apartado anterior dirixido a calquera órgano ou entidade do ámbito da Administración titular do rexistro.



Con respecto ás comunicacións, os cidadáns poderán elixir en todo momento a maneira de se comunicaren coas administracións públicas, sexa ou non por medios electrónicos, excepto naqueles casos en que dunha norma con rango de lei se estableza ou infira a utilización dun medio non electrónico. A opción de comunicarse por uns ou outros medios non vincula ao cidadán, que poderá, en calquera momento, optar por un medio distinto do inicialmente elixido.

Con respecto a documentos e copias, dispón que as administracións públicas poderán emitir validamente por medios electrónicos os documentos administrativos a que se refire o artigo 46 da Lei 30/1992, de réxime xurídico das administracións públicas e do procedemento administrativo común, sempre que incorporen unha ou varias sinaturas electrónicas.

Os documentos administrativos incluírán referencia temporal, que se garantirá a través de medios electrónicos cando a natureza do documento así o requira.

As copias realizadas por medios electrónicos de documentos electrónicos emitidos polo propio interesado ou polas administracións públicas, manténdose ou non o formato orixinal, terán inmediatamente a consideración de copias auténticas coa eficacia prevista no artigo 46 da Lei 30/1992, de réxime xurídico das administracións públicas e do procedemento administrativo común, sempre que o documento electrónico orixinal se atope en poder da Administración, e que a información de sinatura electrónica e, de ser o caso, de selado de tempo permitan comprobar a coincidencia co devandito documento.

Regula a xestión electrónica de procedementos, dispoñendo que a xestión electrónica da actividade administrativa respectará a titularidade e o exercicio da competencia pola Administración pública, órgano ou entidade que a teña atribuída e o cumprimento dos requisitos formais e materiais establecidos nas normas que regulen a



correspondente actividade. Para estes efectos, e en todo caso baixo criterios de simplificación administrativa, impulsarase a aplicación de medios electrónicos nos procesos de traballo e na xestión dos procedementos e da actuación administrativa.

Para rematar, establece o RD un marco institucional de colaboración entre administracións dispoñendo que o Comité Sectorial de Administración electrónica, dependente da Conferencia Sectorial de Administración pública, é o órgano técnico de cooperación da Administración xeral do Estado, das administracións das comunidades autónomas e das entidades que integran a Administración local en materia de Administración electrónica.

O Comité Sectorial da Administración electrónica velará polo cumprimento dos fins e principios establecidos nesta Lei, e en particular desenvolverá as seguintes funcións:

- Asegurar a compatibilidade e interoperabilidade dos sistemas e aplicacións empregados polas administracións públicas.
- Preparar plans e programas conxuntos de actuación para impulsar o desenvolvemento da Administración electrónica en España.
- Asegurar a cooperación entre as administracións públicas para proporcionarlle ao cidadán información administrativa clara, actualizada e inequívoca.

Autor:



Alfonso García Magariños

Director Asesoría Xurídica Municipal Concello da Coruña



**14. FACTURA ELECTRÓNICA.  
LEI 59/2003, DO 19 DE  
DECEMBRO, DE SINATURA  
ELECTRÓNICA. LEI DE  
SERVIZOS DA SOCIEDADE DA  
INFORMACIÓN E COMERCIO  
ELECTRÓNICO.  
ACCESIBILIDADE. DECRETO  
3/2010, DO 8 DE XANEIRO,  
POLO QUE SE REGULA O  
SISTEMA DE FACTURACIÓN  
ELECTRÓNICA DA XUNTA DE  
GALICIA.**



**Tema 14. Factura electrónica. Lei 59/2003, do 19 de decembro, de sinatura electrónica. Lei de servizos da sociedade da información e comercio electrónico. Accesibilidade. Decreto 3/2010, do 8 de xaneiro, polo que se regula o sistema de facturación electrónica da Xunta de Galicia.**

## **ÍNDICE**

- 14.1 Factura electrónica
- 14.2 Lei de sinatura electrónica
  - 14.2.1 Introducción
  - 14.2.2 Análise
  - 14.2.3 Efectos xurídicos e análise
- 14.3 Lei de servizos da sociedade da información e comercio electrónico
  - 14.3.1 Introducción
  - 14.3.2 Ámbito de aplicación
  - 14.3.3 Principios e requisitos de actuación
- 14.4 Accesibilidade
- 14.5 Decreto 3/2010, do 8 de xaneiro, polo que se regula o sistema de facturación electrónica da Xunta de Galicia
  - 14.5.1 Introducción
  - 14.5.2 Estrutura



### **14.1.- FACTURA ELECTRÓNICA**

Con carácter xeral pódese definir a factura como un documento que reflicte a entrega dun produto ou a provisión dun servizo, xunto á data de devindicación, ademais de indicar a cantidade que se ten que pagar como contraprestación.

Na factura atópanse os datos do expedidor e do destinatario, o detalle dos produtos e servizos fornecidos, os prezos unitarios, os prezos totais, os descontos e os impostos.

A facturación electrónica consiste na transmisión das facturas ou documentos análogos entre emisor e receptor por medios electrónicos (ficheiros informáticos) e telemáticos (dun ordenador a outro), asinados dixitalmente con certificados recoñecidos (ou cualificados), coa mesma validez legal que as facturas emitidas en papel.

Se buscamos unha definición específica, podemos recorrer ao artigo 1 da Lei 56/2007: "A factura electrónica é un documento electrónico que cumpre cos requisitos legal e regulamentariamente exixibles ás facturas e que, ademais, garante a autenticidade da súa orixe e a integridade do seu contido".

Inda que existen varios mecanismos para garantir a autenticidade da orixe, a integridade do contido e a lexibilidade dunha factura, ben sexa en papel ou en formato electrónico, desde o momento da súa expedición ata o final do período de conservación da factura, no caso da factura electrónica, o uso da sinatura electrónica é o máis xeneralizado en España.

Neste sentido, no texto consensuado da futura modificación da Directiva 112/2006, recóllese, no que se refire ao seu artigo 233:



*"1. Garantirase a autenticidade da orixe, a integridade do contido e a lexibilidade dunha factura, ben sexa en papel ou en formato electrónico, desde o momento da súa expedición ata o final do período de conservación da factura. Cada suxeito pasivo determinará o xeito de garantir a autenticidade da orixe, a integridade do contido e a lexibilidade das facturas. Poderase realizar mediante controis de xestión que creen un vínculo fiable de auditoría entre a factura e a entrega de bens ou a prestación de servizos.*

*Entenderase por 'autenticidade da orixe', a garantía da identidade do provedor de bens ou prestador de servizos ou do emisor da factura.*

*Por 'integridade do contido' entenderase que o contido requirido conforme ao disposto na presente Directiva non foi modificado.*

*2. Ademais dos tipos de control da xestión contemplados polo segundo parágrafo do apartado 1, outros exemplos de tecnoloxías que garanten a autenticidade da orixe e a integridade do contido dunha factura electrónica son:*

- A sinatura electrónica avanzada, no sentido do punto 2 do artigo 2 da Directiva 1999/93/CE do Parlamento Europeo e do Consello, do 13 de decembro de 1999, pola que se establece un marco comunitario para a sinatura electrónica, baseada nun certificado recoñecido e creada mediante un dispositivo seguro de creación de sinatura, no sentido dos puntos 6 e 10 da Directiva 1999/93/CE.*

- O intercambio electrónico de datos (IED), tal e como se define no artigo 2 da Recomendación 1994/820/CE da Comisión, do 19 de outubro de 1994, relativa aos aspectos xurídicos do intercambio electrónico de datos, se o acordo relativo ao intercambio contempla o uso de procedementos que garantan a autenticidade da orixe e a integridade dos datos."*

Isto significa que tralo período de transposición da Directiva (1 de xaneiro do 2013), a lexislación española contemplará a posibilidade de que se poidan enviar facturas electrónicas entre empresas sen requisito formal



ningún, inda que é probable que se manteñan os mesmos requisitos que existen na actualidade cando o destinatario sexa unha Administración pública.

En España, a adopción da sinatura electrónica como mecanismo xeneralizado para garantir a autenticidade e integridade das facturas electrónicas viuse favorecido pola extensión do DNI electrónico e a ampla dispoñibilidade de certificados electrónicos de múltiples prestadores de servizos de certificación, así como pola dispoñibilidade de software de balde que permite a xeración e sinatura electrónica das facturas electrónicas que se envían, así como a súa verificación no caso da recepción de facturas.

O proceso de facturación é un proceso importante para calquera empresa, e culmina o proceso de compra e venda. Inda que, tradicionalmente, a relación entre empresas baseouse no intercambio de documentos en papel, isto implica o emprego de grandes cantidades de recursos e a realización de moitas tarefas de forma manual. Nun contexto de universalización de Internet, cada vez máis as empresas consideran a optimización dos seus procesos para gañaren eficiencia e aforrar custos.

E por iso se avanzou na adopción da facturación electrónica, que en España está regulada no Regulamento de facturación publicado no Real decreto 1496/2003 e modificado polo Real decreto 87/2005.

As denominacións 'factura electrónica', 'factura telemática' e 'factura dixital' son equivalentes, aínda que a denominación xeralmente utilizada na normativa é 'remisión electrónica' ou 'remisión por medios electrónicos de factura'. Normalmente, distínguese coa denominación 'factura dixital' a modalidade de factura electrónica que emprega a sinatura dixital para garantir a autenticidade e integridade da factura.



As facturas electrónicas pódense emitir en diferentes formatos (EDIFACT, XML, PDF, html, doc, xls, gif, jpeg ou txt, entre outros) sempre que se respecte o contido legal exixible a calquera factura e que se cumpran os requisitos de autenticidade e integridade, por exemplo coa incorporación da sinatura electrónica recoñecida (*qualified electronic signature*, en inglés).

Con todo, trala publicación da Orde PRE/2971/2007, en definiuse o uso obrigatorio do formato XML facturae, cando o destinatario sexa unha Administración da AXE (Administración Xeral do Estado) e os seus organismos públicos.

## **14.2. LEI DE SINATURA ELECTRÓNICA**

### **14.2.1 Introducción**

Sen esquecer as referencias que sobre o tema de utilización de medios electrónicos establecía a Lei 30/92, do 20 de novembro (LRXPAC), é o Real decreto lei 14/1999, do 17 de setembro, sobre sinatura electrónica, a norma pioneira á hora de fomentar a rápida incorporación das novas tecnoloxías de seguridade das comunicacións electrónicas na actividade das empresas, os cidadáns e as administracións públicas, actualmente a Lei 59/2003 de sinatura electrónica, cuxa última modificación tivo lugar mediante a Lei 56/2007 de medidas de impulso da sociedade da información.

### **14.2.2 Estrutura e análise**

Desde o punto de vista da súa estrutura, a Lei 59/2003 de sinatura electrónica consta de 36 artigos agrupados en seis títulos, 11 disposicións adicionais, dúas disposicións transitorias, unha disposición derogatoria e tres disposicións derradeiras.



O título I contén os principios xerais que delimitan os ámbitos subxectivo e obxectivo de aplicación da Lei, os efectos da sinatura electrónica e o réxime de emprego ante as administracións públicas e de acceso á actividade de prestación de servizos de certificación.

O réxime aplicable aos certificados electrónicos contense no título II, que dedica o seu primeiro capítulo a determinar quen poden ser os seus titulares e a regular as vicisitudes que afectan a súa vixencia. O capítulo II regula os certificados recoñecidos e o terceiro o documento nacional de identidade electrónico.

O título III regula a actividade de prestación de servizos de certificación establecendo as obrigas a que están suxeitos os prestadores —distinguindo con nitidez as que só afectan aos que expiden certificados recoñecidos—, e o réxime de responsabilidade aplicable.

O título IV establece os requisitos que deben reunir os dispositivos de verificación e creación de sinatura electrónica e o procedemento que se ten que seguir para obter selos de calidade na actividade de prestación de servizos de certificación.

Os títulos V e VI dedican o seu contido, respectivamente, a fixar os réximes de supervisión e sanción dos prestadores de servizos de certificación.

Para rematar, pechan o texto as disposicións adicionais —que aluden aos réximes especiais que resultan de aplicación preferente—, as disposicións transitorias —que incorporan seguridade xurídica á actividade despregada ao amparo da normativa anterior—, a disposición derogatoria e as disposicións derradeiras relativas ao fundamento constitucional, a habilitación para o desenvolvemento regulamentario e a entrada en vigor.



Supón a incorporación ao ordenamento interno da regulación contida na Directiva 1999/93/ CE, do 13 de decembro de 1999, do Parlamento Europeo e do Consello; esta Lei semella ter presente na elaboración do seu contido a Lei modelo para as sinaturas electrónicas da Comisión das Nacións Unidas para o Dereito Mercantil Internacional (CNUDMI/UNCITRAL), aprobada, xunto á súa Guía, o 5 de xullo do 2001.

Ten como principal finalidade reforzar o marco xurídico existente, incorporando ao seu texto "algunhas novidades respecto do Real decreto 14/1999, que contribuirán a dinamizar o mercado da prestación de servizos de certificación, conferíndolle seguridade ás comunicacións a través de Internet, e configurando a sinatura electrónica como instrumento capaz de xerar confianza nas transaccións telemáticas, ademais de axilizar o comercio electrónico. Permitirase, en consecuencia, unha comprobación da procedencia e da integridade das mensaxes intercambiadas a través de redes de telecomunicacións, ofrecendo as bases para evitar o repudio, se se adoptan as medidas oportunas baseándose en datos electrónicos".

Constitúe o seu obxecto, conforme dispón o art. 1, tanto a regulación da sinatura electrónica, como elemento de seguridade das comunicacións nos seus diversos aspectos, e a súa eficacia xurídica, como a prestación de servizos de certificación nos seus diversos aspectos (obxectivo: certificados, e subxectivo: prestadores de servizo de certificación).

O apartado 1 do art. 3 da Lei de sinatura electrónica define de forma xeral a sinatura electrónica como "o conxunto de datos en forma electrónica, consignados xunto a outros ou asociados con eles, que poden ser utilizados como medio de identificación do asinante".

Trátase dunha definición ampla que pode englobar todo un conxunto de sinaturas electrónicas, desde aquelas máis complexas, como a sinatura dixital baseada en sistemas biométricos como o iris, a propia palma da



man, a pegada dactilar, etc., ata as máis simples, como un nome ou outro elemento identificador (por exemplo, a sinatura manual dixitalizada, ou un *password* ou contrasinal), incluído ao final da mensaxe electrónica, ou a existencia dunha pregunta-resposta, e un *pin* de acceso, o que se denomina tecnoloxía de segredo compartido, de tan escasa seguridade que suscita a cuestión do seu valor probatorio para os efectos de autenticación ou identificación do autor.

Así mesmo, deste concepto amplo e tecnoloxicamente indefinido de sinatura que nos ofrece o citado precepto podemos resaltar as seguintes características da sinatura electrónica:

- A sinatura electrónica é un conxunto de datos e non un símbolo, selo ou grafía electrónica que serve para identificar o asinante dunha mensaxe e para acreditar a identificación deste, así como a integridade do contido da mensaxe.
- Trátase dunha técnica para identificar o asinante dun documento electrónico.
- Os datos de sinatura electrónica poden formar parte do documento ou ir asociados funcionalmente con eles ou, o que é o mesmo, poden aparecer como un conxunto independente. O modo concreto en que en cada momento se manifeste a sinatura electrónica dependerá do sistema técnico que se elixa e das aplicacións prácticas que ofrezca cada modalidade.

Define a Lei que se considera documento electrónico a información de calquera natureza en forma electrónica, archivada nun soporte electrónico segundo un formato determinado e susceptible de identificación e tratamento diferenciado.



Para que un documento electrónico teña a natureza de documento público ou de documento administrativo deberase cumprir que, ou ben estean asinados electronicamente por funcionarios que teñan legalmente atribuída a facultade de dar fe pública, xudicial, notarial ou administrativa, sempre que actúen no ámbito das súas competencias cos requisitos esixidos pola Lei en cada caso, ou ben se trate de documentos expedidos e asinados electronicamente por funcionarios ou empregados públicos no exercicio das súas funcións públicas, conforme á súa lexislación específica.

A carón da sinatura electrónica atópase a sinatura electrónica avanzada, que, segundo a Lei, artigo 3.2: "A sinatura electrónica avanzada é a sinatura electrónica que permite identificar o asinante e detectar calquera cambio ulterior dos datos asinados, que está vinculada ao asinante de xeito único e aos datos a que se refire e que foi creada por medios que o asinante pode manter baixo o seu exclusivo control".

A Lei de sinatura electrónica, fronte á regulación contida tanto no RDL 14/99 como na Directiva, introduce un terceiro tipo de sinatura, de maior calidade e seguridade, como é a sinatura electrónica recoñecida, definida no art. 3.3 como "a sinatura electrónica avanzada baseada nun certificado recoñecido e xerada mediante un dispositivo seguro de creación de sinatura".

Supón, como sinala a exposición de motivos da Lei, "a creación dun concepto novo demandado polo sector, sen que iso implique modificación ningunha dos requisitos substantivos que tanto a Directiva 1999/93/CE como o propio Real decreto lei 14/1999 viñan esixindo".

O art. 24 da Lei de sinatura electrónica, baixo o título "Dispositivo de sinatura electrónica", define os datos de creación de sinatura como "os datos únicos, como códigos ou claves criptográficas privadas, que o asinante utiliza para crear a sinatura electrónica".



Como elemento característico destes vense establecer que estes deben ser únicos.

Pola súa banda, o art. 25 da Lei de sinatura electrónica, baixo o título "Dispositivos de verificación de sinatura electrónica", define os "datos de verificación de sinatura" como "os datos, códigos ou claves criptográficas públicas, que se utilizan para verificar a sinatura electrónica".

Xa que logo, baixo o título "Dispositivos de sinatura electrónica" regúlase como nocións previas tanto para a aplicación do dispositivo de creación como para o de verificación os datos de creación e verificación de sinatura, respectivamente, que desde un punto de vista técnico constitúen ademais elementos que posibilitan a creación dunha sinatura ou a súa verificación.

#### 14.2.3 Efectos xurídicos da sinatura electrónica

Á validez e eficacia da sinatura electrónica dedica a Lei de sinatura electrónica os apartados 4, 8, 9 e 10 do art. 3, que coincide substancialmente co disposto no art. 5 da Directiva comunitaria, e onde se equipara a sinatura electrónica recoñecida á sinatura manuscrita, se determinan as consecuencias da impugnación da autenticidade da sinatura electrónica recoñecida pola outra parte non asinante, se lle recoñece valor xurídico á autonomía da vontade das partes para dotar de eficacia a sinatura electrónica e se especifica a admisibilidade dos datos asinados electronicamente como proba documental en xuízo.

O art 3.4 da Lei establece a regra do equivalente funcional entre a sinatura electrónica recoñecida e a sinatura manuscrita, ao dispoñer que "a sinatura electrónica recoñecida terá, respecto dos datos consignados en forma electrónica, o mesmo valor que a sinatura manuscrita en relación cos consignados en papel".



En consecuencia, sobre o exposto, para a plena operatividade da regra da equivalencia funcional da sinatura electrónica recoñecida coa sinatura manuscrita, que dispón o art. 3.4 da Lei de sinatura electrónica, nunha interpretación conxunta co art. 3.3 desta mesma Lei, é necesario o cumprimento dos seguintes requisitos:

1.º Débese tratar dunha sinatura electrónica avanzada (art. 3.2 da Lei de sinatura electrónica).

2.º Esta sinatura electrónica avanzada ten que estar baseada nun certificado recoñecido, é dicir, aquel que cumpre os requisitos dos art. 11, 12 e 13 da Lei de sinatura electrónica, e que fose expedido por un prestador de servizos de certificación que cumpra cos requisitos previstos no art. 20 da Lei de sinatura electrónica.

3.º Esta sinatura electrónica avanzada, ademais, debe ser producida por un dispositivo seguro de creación de sinatura que cumpra cos requisitos do apartado 3 do art. 24 da Lei de sinatura electrónica.

Regula tamén a Lei a figura do DNI electrónico, que é o documento nacional de identidade que acredita electronicamente a identidade persoal do seu titular e permite a sinatura electrónica de documentos; imponse a obriga de que todas as persoas físicas ou xurídicas, públicas ou privadas, recoñecerán a eficacia do documento nacional de identidade electrónico para acreditar a identidade e os demais datos persoais do titular que consten nel, e para acreditar a identidade do asinante e a integridade dos documentos asinados cos dispositivos de sinatura electrónica nel incluídos.

A Lei regula a figura dos prestadores de servizos de certificación, sendo estes "a persoa física ou xurídica que expide certificados electrónicos ou presta outros servizos en relación coa sinatura electrónica".

Para a expedición de certificados electrónicos ao público, os prestadores de servizos de certificación só poderán solicitar datos persoais



directamente dos asinantes ou c o previo consentimento expreso destes, debendo cumprir, xa que logo, o disposto na Lei orgánica 15/1999, do 13 de decembro, de protección de datos de carácter persoal e nas súas normas de desenvolvemento.

Os prestadores de servizos de certificación que expidan certificados electrónicos deberán cumprir as seguintes obrigas:

- Non almacenar nin copiar os datos de creación de sinatura da persoa á que lle presten os seus servizos.
- Proporcionarlle ao solicitante, antes da expedición do certificado, a seguinte información mínima, que e deberá transmitir de forma gratuíta, por escrito ou por vía electrónica:
- As obrigas do asinante, a forma en que se teñen que custodiar os datos de creación de sinatura, o procedemento que cómpre seguir para comunicar a perda ou posible utilización indebida dos anteditos datos e determinados dispositivos de creación e de verificación de sinatura electrónica que sexan compatibles cos datos de sinatura e co certificado expedido.
- Os mecanismos para garantir a fiabilidade da sinatura electrónica dun documento ao longo do tempo.
- O método empregado polo prestador para comprobar a identidade do asinante ou outros datos que figuren no certificado.



- As condicións precisas de utilización do certificado, os seus posibles límites de uso e a forma en que o prestador garante a súa responsabilidade patrimonial.
- As certificacións que obteña, de ser o caso, o prestador de servizos de certificación e os procedementos aplicables para a resolución extraxudicial dos conflitos que puidesen xurdir polo exercicio da súa actividade.
- As demais informacións contidas na declaración de prácticas de certificación.
- A información citada anteriormente que sexa relevante para terceiros afectados polos certificados deberá estar dispoñible a instancia destes.
- Manter un directorio actualizado de certificados onde se indicarán os certificados expedidos e se están vixentes ou se a súa vixencia foi suspendida ou extinguida. A integridade do directorio protexerase mediante a utilización dos mecanismos de seguridade adecuados.
- Garantir a dispoñibilidade dun servizo de consulta sobre a vixencia dos certificados rápido e seguro.

Regula a Lei aspectos relacionados coa supervisión e control da actividade dos prestadores de servizos e, así, o Ministerio de Ciencia e Tecnoloxía controlará o cumprimento por parte dos prestadores de servizos de certificación que expidan ao público certificados electrónicos das obrigas establecidas nesta Lei e nas súas disposicións de desenvolvemento. Así mesmo, supervisará o funcionamento do sistema e



dos organismos de certificación de dispositivos seguros de creación de sinatura electrónica.

O Ministerio de Ciencia e Tecnoloxía realizará as actuacións inspectoras que sexan precisas para o exercicio da súa función de control, os funcionarios adscritos ao Ministerio de Ciencia e Tecnoloxía que realicen a inspección a que se refire o apartado anterior terán a consideración de autoridade pública no desempeño dos seus cometidos.

Os prestadores de servizos de certificación, a entidade independente de acreditación e os organismos de certificación teñen a obriga de facilitárenlle ao Ministerio de Ciencia e Tecnoloxía toda a información e colaboración precisas para o exercicio das súas funcións.

### **14.3.- LEI DE SERVIZOS DA SOCIEDADE DA INFORMACIÓN E COMERCIO ELECTRÓNICO**

#### **14.3.1 Introducción**

A Lei 34/2002 do 22 de xullo ten como obxecto a incorporación ao ordenamento xurídico español da Directiva 2000/31/CE, do Parlamento Europeo e do Consello, do 8 de xuño, relativa a determinados aspectos dos servizos da sociedade da información, en particular, o comercio electrónico no mercado interior (Directiva sobre o comercio electrónico). Así mesmo, incorpora parcialmente a Directiva 98/27/CE, do Parlamento Europeo e do Consello, do 19 de maio, relativa ás accións de cesación en materia de protección dos intereses dos consumidores, ao regular, de conformidade co establecido nela, unha acción de cesación contra as condutas que contraveñan o disposto nesta Lei.



O que a Directiva 2000/31/CE denomina sociedade da información vén determinado pola extraordinaria expansión das redes de telecomunicacións e, en especial, de Internet como vehículo de transmisión e intercambio de todo tipo de información. A súa incorporación á vida económica e social ofrece innumerables vantaxes, como a mellora da eficiencia empresarial, o incremento das posibilidades de elección dos usuarios e a aparición de novas fontes de emprego. Pero a implantación de Internet e as novas tecnoloxías tropeza con algunhas incertezas xurídicas, que é preciso aclarar co establecemento dun marco xurídico axeitado, que xere en todos os actores intervenientes a confianza necesaria para o emprego deste novo medio.

É obxecto da Lei a regulación do réxime xurídico dos servizos da sociedade da información e da contratación por vía electrónica, no referente ás obrigas dos prestadores de servizos, incluídos os que actúen como intermediarios na transmisión de contidos polas redes de telecomunicacións, as comunicacións comerciais por vía electrónica, a información previa e posterior á celebración de contratos electrónicos, as condicións relativas á súa validez e eficacia e o réxime sancionador aplicable aos prestadores de servizos da sociedade da información.

#### 14.3.2 Ámbito de aplicación

Aplicaráselles aos prestadores de servizos da sociedade da información establecidos noutro Estado membro da Unión Europea ou do Espazo Económico Europeo cando o destinatario dos servizos radique en España e os servizos afecten ás materias seguintes:

- a. Dereitos de propiedade intelectual ou industrial.
- b. Emisión de publicidade por institucións de investimento colectivo.



c. Actividade de seguro directo realizada en réxime de dereito de establecemento ou en réxime de libre prestación de servizos.

d. Obrigas nadas dos contratos celebrados por persoas físicas que teñan a condición de consumidores.

e. Réxime de elección polas partes contratantes da lexislación aplicable ao seu contrato.

f. Licitude das comunicacións comerciais por correo electrónico ou outro medio de comunicación electrónica equivalente non solicitadas.

A prestación de servizos da sociedade da información non estará suxeita a autorización previa.

No caso de que un determinado servizo da sociedade da información atente ou poida atentar contra os principios que se expresan a seguir, os órganos competentes para a súa protección, no exercicio das funcións que teñan legalmente atribuídas, poderán adoptar as medidas necesarias para que se interrompa a súa prestación ou para retirar os datos que os vulneran.

#### 14.4.3 Principios e requisitos de actuación

Os principios de aplicación son os seguintes:

a. A salvagarda da orde pública, a investigación penal, a seguridade pública e a defensa nacional.

b. A protección da saúde pública ou das persoas físicas ou xurídicas que teñan a condición de consumidores ou usuarios, mesmo cando actúen como inversores.

c. O respecto á dignidade da persoa e ao principio de non discriminación por motivos de raza, sexo, relixión, opinión,



nacionalidade, discapacidade ou calquera outra circunstancia persoal ou social.

- d. A protección da mocidade e da infancia.
- e. A salvagarda dos dereitos de propiedade intelectual.

O prestador de servizos da sociedade da información estará obrigado a dispoñer dos medios que lles permitan, tanto aos destinatarios do servizo como aos órganos competentes, accederen por medios electrónicos, de forma permanente, sinxela, directa e gratuíta, á seguinte información:

a. O seu nome ou denominación social; a súa residencia ou domicilio ou, na súa falta, o enderezo dun dos seus establecementos permanentes en España; o seu enderezo de correo electrónico e calquera outro dato que permita establecer con el unha comunicación directa e efectiva.

b. Os datos da súa inscrición no Rexistro Mercantil onde, de ser o caso, estean inscritos ou daqueloutro rexistro público onde o estivesen para a adquisición de personalidade xurídica ou para os únicos efectos de publicidade.

c. No caso de que a súa actividade estivese suxeita a un réxime de autorización administrativa previa, os datos relativos á devandita autorización e os identificadores do órgano competente encargado da súa supervisión.

d. Se exerce unha profesión regulada deberá indicar:

1. Os datos do colexio profesional ao que, no seu caso, pertenza e o número de colexiado.

2. O título académico oficial ou profesional co que conte.

3. O Estado da Unión Europea ou do Espazo Económico Europeo onde se expediu ese título e, se é o caso, a correspondente homologación ou recoñecemento.



4. As normas profesionais aplicables ao exercicio da súa profesión e os medios a través dos cales se poidan coñecer, incluídos os electrónicos.

e. O número de identificación fiscal que lle corresponda.

f. Cando o servizo da sociedade da información faga referencia a prezos, facilitarase información clara e exacta sobre o prezo do produto ou servizo, indicando se inclúe ou non os impostos aplicables e, de ser o caso, sobre os gastos de envío ou aquilo que dispoñan as normas das comunidades autónomas con competencias na materia.

g. Os códigos de conduta aos que, se é o caso, estea adherido e o xeito de consultalos electronicamente.

Os prestadores de servizos da sociedade da información están suxeitos á responsabilidade civil, penal e administrativa establecida con carácter xeral no ordenamento xurídico, sen prexuízo do disposto nesta Lei.

Os prestadores de servizos da sociedade da información que faciliten enlaces a outros contidos ou inclúan nos seus directorios ou instrumentos de procura de contidos non serán responsables pola información á que dirixan os destinatarios dos seus servizos, sempre que:

a. Non teñan coñecemento efectivo de que a actividade ou a información á que remiten ou que recomendan é ilícita ou de que lesiona bens ou dereitos dun terceiro susceptibles de indemnización, ou

b. se o teñen, actúen con dilixencia para suprimir ou inutilizar o enlace correspondente.

As administracións públicas impulsarán, a través da coordinación e o asesoramento, a elaboración e aplicación de códigos de conduta voluntarios, por parte das corporacións, asociacións ou organizacións



comerciais, profesionais e de consumidores, nas materias reguladas nesta Lei. A Administración xeral do Estado fomentará, en especial, a elaboración de códigos de conduta de ámbito comunitario ou internacional.

Os códigos de conduta poderán tratar, en particular, sobre os procedementos para a detección e retirada de contidos ilícitos e a protección dos destinatarios fronte ao envío por vía electrónica de comunicacións comerciais non solicitadas, así como sobre os procedementos extraxudiciais para a resolución dos conflitos que xurdan pola prestación dos servizos da sociedade da información.

Os contratos celebrados por vía electrónica producirán todos os efectos previstos polo ordenamento xurídico, cando concorran o consentimento e os demais requisitos necesarios para a súa validez.

A proba da celebración dun contrato por vía electrónica e a das obrigas que teñen a súa orixe nel suxeitarase ás regras xerais do ordenamento xurídico.

Cando os contratos celebrados por vía electrónica estean asinados electronicamente suxeitaranse ao establecido no artigo 3 da Lei 59/2003, do 19 de decembro, de sinatura electrónica.

Os contratos celebrados por vía electrónica en que interveña como parte un consumidor presumiranse celebrados no lugar en que este teña a súa residencia habitual.

O prestador e o destinatario de servizos da sociedade da información poderán someter os seus conflitos ás arbitraxes previstas na lexislación de arbitraje e de defensa dos consumidores e usuarios, e aos procedementos de resolución extraxudicial de conflitos que se instauren por medio de códigos de conduta ou outros instrumentos de autorregulación.



#### **14.4.-ACCESIBILIDADE**

A disposición adicional quinta da Lei de servizos da sociedade da información e do comercio electrónico (Lei 34/2002 do 11 de xullo) establece unha norma relativa á accesibilidade para as persoas con discapacidade e de idade avanzada á información proporcionada por medios electrónicos e establece:

*"As administracións públicas adoptarán as medidas necesarias para que a información dispoñible nas súas respectivas páxinas de Internet poida ser accesible para persoas con discapacidade e de idade avanzada, de acordo cos criterios de accesibilidade ao contido xeralmente recoñecidos, antes do 31 de decembro do 2005.*

*A partir do 31 de decembro do 2008, as páxinas de Internet das administracións públicas satisfarán, como mínimo, o nivel medio dos criterios de accesibilidade ao contido xeralmente recoñecidos. Excepcionalmente, esta obriga non será aplicable cando unha funcionalidade ou servizo non dispoña dunha solución tecnolóxica que permita a súa accesibilidade.*

*As administracións públicas esixirán que tanto as páxinas de Internet cuxo deseño ou mantemento financien total ou parcialmente como as páxinas de Internet de entidades e empresas que se encarguen de xestionar servizos públicos apliquen os criterios de accesibilidade antes mencionados. En particular, será obrigatorio o expresado neste apartado para as páxinas de Internet e os seus contidos dos centros públicos educativos, de formación e universitarios, así como dos centros privados que obteñan financiamento público.*

*As páxinas de Internet das administracións públicas deberán ofrecerlle ao usuario información sobre o seu nivel de accesibilidade e facilitar un sistema de contacto para que poidan transmitir as dificultades de acceso ao contido das páxinas de Internet ou formular calquera queixa, consulta ou suxestión de mellora.*



*Igualmente, promoverase a adopción de normas de accesibilidade polos prestadores de servizos e os fabricantes de equipos e software, para facilitar o acceso das persoas con discapacidade ou de idade avanzada aos contidos dixitais.*

*As administracións públicas promoverán medidas de sensibilización, educación e formación sobre accesibilidade con obxecto de promover que os titulares doutras páxinas de Internet incorporen progresivamente os criterios de accesibilidade.*

*Os incumprimentos das obrigas de accesibilidade establecidas nesta disposición adicional estarán sometidos ao réxime de infraccións e sancións vixente en materia de igualdade de oportunidades, non discriminación e accesibilidade universal das persoas con discapacidade.*

*As páxinas de Internet das empresas que presten servizos ao público en xeral de especial transcendencia económica, sometidas á obriga establecida no artigo 2 da Lei 56/2007, de medidas de impulso da sociedade da información, deberán satisfacer a partir do 31 de decembro do 2008, como mínimo, o nivel medio dos criterios de accesibilidade ao contido xeralmente recoñecidos. Excepcionalmente, esta obriga non será aplicable cando unha funcionalidade ou servizo non dispoña dunha solución tecnolóxica que permita a súa accesibilidade."*

O "Deseño Web Accesible" pretende establecer as técnicas e mecanismos para permitir que un sitio web poida ser utilizado por calquera persoa, con independencia da súa discapacidade.

En decembro do 2007 aprobouse a Lei 56/2007, de medidas de impulso da sociedade da información (BOE 29-12-2008). Por unha banda, esta Lei indica claramente que a accesibilidade das páxinas web da Administración pública está suxeita ao réxime de infraccións e sancións (Lei 49/2007). Doutra banda, a obriga de facer sitios web accesibles amplíase ao sector privado, en concreto ás empresas que presten servizos ao público en xeral de especial transcendencia económica.



En decembro do 2007 aprobouse a Lei 49/2007, pola que se establece o réxime de infraccións e sancións en materia de igualdade de oportunidades, non discriminación e accesibilidade universal das persoas con discapacidade (BOE 27-12-2007). Nesta Lei defínense sancións que poden chegar ata 1.000.000 de euros.

En novembro do 2007 aprobouse o Real decreto 1494/2007 polo que se aproba o Regulamento sobre as condicións básicas para o acceso das persoas con discapacidade ás tecnoloxías, produtos e servizos relacionados coa sociedade da información e medios de comunicación social (BOE 21-11-2007). Este Regulamento obriga as administracións públicas a que as súas páxinas web sexan accesibles de acordo co nivel 2 da norma española UNE 139803:2004 antes do 2009.

A Lei IONDAU (Lei 51/2003, do 2 de decembro, de igualdade de oportunidades, non discriminación e accesibilidade universal das persoas con discapacidade (BOE 3-12-2003) fixa varias fases: 1º- a principios do 2006 o Goberno deberá establecer os criterios básicos de accesibilidade para as tecnoloxías da sociedade da información; 2º- no 2010 todos os novos produtos e servizos da sociedade da información deberán ser accesibles; 3º- no 2014 todos os produtos e servizos da sociedade da información deberán ser accesibles.

A Lei SSICE (Lei 34/2002, do 11 de xullo, de servizos da sociedade da información e de comercio electrónico (BOE 12-7-2002), na súa disposición adicional 5ª (que fai referencia á accesibilidade para as persoas con discapacidade e de idade avanzada á información proporcionada por medios electrónicos), obrigaba as administracións públicas a teren as súas webs accesibles para todos antes do 2006.

No ámbito europeo existen numerosas normas e iniciativas neste sentido (como as iniciativas eEurope (*an Information Society for All*) 2002 e



2005 e a estratexia i2010), xa que se fala dun 20% da poboación europea afectada por algún tipo de discapacidade.

#### **14.5.- DECRETO 3/2010, DO 8 DE XANEIRO, POLO QUE SE REGULA O SISTEMA DE FACTURACIÓN ELECTRÓNICA DA XUNTA DE GALICIA**

##### **14.5.1 Introducción**

A implantación dos medios electrónicos no ámbito da facturación e a contratación pública enmárcase nas políticas corporativas comúns do Goberno galego para o desenvolvemento da Administración electrónica e integrárase harmónica e complementariamente co resto das aplicacións comúns para toda a Xunta de Galicia.

O Decreto regula, por unha banda, as liñas xerais de emprego dos medios electrónicos, informáticos e telemáticos nos procedementos de contratación e establece as condicións de utilización dos citados medios no marco do desenvolvemento da Administración electrónica da Xunta de Galicia. Así mesmo, configura as bases dun sistema que se asenta sobre varios eixes que se complementan estruturando todas as relacións telemáticas entre os actores que interveñen nos procesos.

Así, en termos de servizos aos licitadores, ordena e estrutura novas canles de información e participación do empresariado nos procesos de contratación creando a Plataforma de Contratos Públicos de Galicia e un Portal de Contratación Pública da Comunidade Autónoma que potenciará os servizos en liña existentes e incorporará prestacións relacionadas coa licitación electrónica. Doutra banda, en termos de servizos entre administracións prevénse mecanismos de interoperatividade que, en primeiro lugar, reforzan a accesibilidade ás plataformas e sistemas e aplicacións existentes ou futuras e, en segundo lugar, facilitan a



cooperación e colaboración intra- e interadministrativas para os efectos de tráfico de información.

#### 14.5.2 Estrutura

Respecto da estrutura, o Decreto consta de 18 artigos, agrupados en 5 capítulos.

O capítulo I regula o obxecto do Decreto e o seu ámbito de aplicación.

O capítulo II regula a Plataforma de Contratos Públicos de Galicia, creada ao amparo do disposto no artigo 309.5 da Lei 30/2007, do 30 de outubro, como servizo de información das licitacións do sector público galego a través de Internet.

O capítulo III regula o Sistema de Licitación Electrónica que permitirá a presentación de ofertas e proposicións por vía telemática.

O capítulo IV regula determinados sistemas de tramitación e xestión electrónica de importante incidencia na contratación e na facturación electrónica.

Para rematar, no capítulo V créase un Portal de Contratación Pública da Comunidade Autónoma como punto central de acceso e entrada para os interesados a todas as aplicacións e servizos que, en materia de contratación pública e por vía electrónica, se poidan realizar a través de Internet.

O Decreto modifica tamén, na disposición derradeira, o Decreto 262/2001, do 20 setembro, polo que se refunde a normativa reguladora do Rexistro Xeral de Contratistas da Comunidade Autónoma de Galicia ao introducir a tramitación por medios electrónicos, informáticos e telemáticos (EIT) dos procesos de alta, modificación e baixa do rexistro e un sistema de notificacións electrónicas coas empresas rexistradas.



Autor:

Alfonso García Magariños

Director Asesoría Xurídica Municipal Concello da Coruña





# **15. NORMATIVA NO ÁMBITO DA PROPIEDADE INTELECTUAL. A PROTECCIÓN XURÍDICA DOS PROGRAMAS DE ORDENADOR. TIPOS DE LICENZAS. SOFTWARE DE FONTES ABERTAS (FLOSS).**



## **Tema 15.- Normativa no ámbito da propiedade intelectual. A protección xurídica dos programas de ordenador. Tipos de licenzas: Software de fontes abertas (FLOSS)**

### **ÍNDICE**

#### 15.1 Normativa no ámbito da propiedade intelectual

##### 15.1.1 Estatal

##### 15.1.2 Comunitaria

##### 15.1.3 Internacional

#### 15.2 Protección xurídica programas ordenador

##### 15.2.1 Introducción

##### 15.2.2 Normativa Estatal

#### 15.3 Tipos de licenzas

#### 15.4 Software de fontes abertas FLOSS



## **15.1- NORMATIVA NO ÁMBITO DA PROPIEDAD INTELECTUAL**

Segundo a Organización Mundial da Propiedade Intelectual, a propiedade intelectual (P.I.) ten que ver coas creacións da mente: as invencións, as obras literarias e artísticas, os símbolos, os nomes, as imaxes e os debuxos e modelos utilizados no comercio.

A propiedade intelectual divídese en dúas categorías: a propiedade industrial, que inclúe as invencións, patentes, marcas, debuxos e modelos industriais e indicacións xeográficas de procedencia; e o dereito de autor, que abarca as obras literarias e artísticas, tales coma as novelas, os poemas e as obras de teatro; as películas, as obras musicais e as obras de arte, tales coma debuxos, pinturas, fotografías e esculturas, e os deseños arquitectónicos. Os dereitos relacionados co dereito de autor son os dereitos dos artistas intérpretes e executantes sobre as súas interpretacións e execucións; os dereitos dos produtores de fonogramas sobre as súas gravacións e os dereitos dos organismos de radiodifusión sobre os seus programas de radio e de televisión.

Polo que respecta aos dereitos que conforman a propiedade intelectual distínguense os dereitos morais e os dereitos patrimoniais:

Dereitos morais:

Fronte aos sistemas de corte anglosaxón, a lexislación española é claramente defensora dos dereitos morais, recoñecidos para os autores e para os artistas intérpretes ou executantes. Estes dereitos son irrenunciabes e inalienables, acompañan ao autor ou ao artista intérprete ou executante durante toda a súa vida e aos seus herdeiros ou titulares do



dereito ao falecemento daqueles. Entre eles destaca o dereito ao recoñecemento da condición de autor da obra ou ao recoñecemento do nome do artista sobre as súas interpretacións ou execucións, e o de esixir o respecto á integridade da obra ou actuación e a non alteración das mesmas.

Dereitos de carácter patrimonial:

Hai que distinguir entre:

Dereitos relacionados coa explotación da obra ou prestación protexida, que, pola súa vez, se subdividen en dereitos exclusivos e en dereitos de remuneración:

1.-Os dereitos exclusivos son aqueles que lle permiten ao seu titular autorizar ou prohibir os actos de explotación da súa obra ou prestación protexida polo usuario, e a esixir deste unha retribución a cambio da autorización que lle conceda.

2.-Os dereitos de remuneración, a diferenza dos dereitos exclusivos, non facultan ao seu titular a autorizar ou prohibir os actos de explotación da súa obra ou prestación protexida polo usuario, aínda que si obrigan a este ao pagamento dunha cantidade de diñeiro polos actos de explotación que realice, cantidade esta que é determinada ben pola lei ou, na súa ausencia, ben polas tarifas xerais das entidades de xestión.

3.-Dereitos compensatorios, como o dereito por copia privada, que compensa os dereitos de propiedade intelectual deixados de percibir por razón das reproducións das obras ou prestacións protexidas para uso exclusivamente privado do copista.

### **15.1.1 Normativa Estatal**



A propiedade intelectual é o conxunto de dereitos que lles corresponden aos autores e a outros titulares (artistas, produtores, organismos de radiodifusión...) respecto das obras e prestacións froito da súa creación.

A propiedade intelectual, tal e como establece o Código Civil nos seus artigos 428 e 429, forma parte das chamadas propiedades especiais e vén constituír unha forma especial de exercer o dereito de propiedade sobre determinados obxectos xurídicos que, polas súas características, especializan o dominio.

Como propiedade especial, o Código Civil remite a súa regulación a unha lei especial e declara a aplicación supletoria das regras xerais establecidas no mesmo sobre a propiedade para o non especificamente previsto na devandita lei especial. Esta lei é a Lei de propiedade intelectual (LPI), cuxo texto refundido foi aprobado polo Real decreto lexislativo 1/1996, do 12 de abril.

O citado texto foi obxecto de sucesivas modificacións, entre as que cómpre salientar a operada pola Lei 5/1998, do 6 de marzo, e a Lei 23/2006 do 7 de xullo, (responde esta á necesidade de incorporar ao dereito español unha das últimas directivas aprobadas en materia de propiedade intelectual, a Directiva 2001/29/CE do Parlamento Europeo e do Consello, do 22 de maio do 2001, relativa á harmonización de determinados aspectos dos dereitos de autor e dereitos afíns aos dereitos de autor na sociedade da información, coa que a Unión Europea quixo, pola súa vez, cumprir os tratados da Organización Mundial da Propiedade Intelectual (OMPI) de 1996 sobre Dereito de Autor e sobre Interpretación ou Execución e Fonogramas).

Cómpre tamén destacar a Lei 19/2006, do 5 de xuño, pola que se amplían os medios de tutela dos dereitos de propiedade intelectual e industrial e se



establecen normas procesuais para facilitar a aplicación de diversos regulamentos comunitarios.

### **15.1.2 Normativa comunitaria:**

1.-Directiva 92/100/CEE, do 19 de novembro de 1992, sobre dereitos de alugueiro e préstamo e outros dereitos afíns aos dereitos de autor no ámbito da propiedade intelectual. Esta Directiva, entre outros aspectos, recoñece o dereito de autorizar ou prohibir o alugueiro e préstamo de orixinais e copias de obras protexidas.

2.-Directiva 93/98/CEE, do 29 de outubro de 1993, relativa á harmonización do prazo de protección do dereito de autor e de determinados dereitos afíns. Nesta Directiva harmonízase o prazo de protección do dereito de autor fixándoo nun período de setenta anos tras a morte do autor, ou desde o momento da primeira difusión lícita entre o público, e polo que se refire aos dereitos afíns, en cincuenta anos desde que se produce o feito xerador.

3.-Directiva 96/9/CE, do 11 de marzo de 1996, sobre a protección xurídica das bases de datos. Recoñécelle un dereito de autor ao creador da estrutura da base de datos, e un dereito "sui generis" ao seu fabricante, entendendo por tal a persoa física ou xurídica que realizou un investimento substancial para a fabricación das bases de datos.

4.-Directiva 2001/29/CE, do 22 de maio do 2001, relativa á harmonización de determinados aspectos dos dereitos de autor e dereitos afíns na sociedade da información. Adecúa o sistema de dereitos de autor e conexos ao contorno dixital, asumindo ao mesmo tempo as obrigas contraídas pola Unión Europea e os seus Estados Membros no marco dos tratados dixitais OMPI (WCT e WPPT).



5.-Directiva 2000/31/CE do Parlamento Europeo e do Consello, do 8 de xuño do 2000 (DOCE do 17 de xullo) relativa a determinados aspectos xurídicos dos servizos da sociedade da información, en particular ao comercio electrónico (Directiva sobre o comercio electrónico). Nesa Directiva regúlase, entre outros aspectos e polo que interesa á materia de propiedade intelectual, a responsabilidade dos prestadores de servizos intermediarios (artigos 12 a 15 da Directiva).

### **15.1.3 Normativa internacional:**

1.-O Convenio de Berna protexe as obras literarias e artísticas. A súa acta orixinaria data de 1886 e España é socio fundador do mesmo. Entre os principios informadores do Convenio atópanse o de trato nacional (ou asimilación do estranxeiro ao nacional), o de protección automática, o de independencia da protección e o de protección mínima (para lograr un conxunto dispositivo uniformemente aplicable).

2.-Tratado OMPI sobre Dereito de Autor (Tratado WCT, 1996): como resultado da Conferencia Diplomática da OMPI sobre certas cuestións de dereitos de autor e de dereitos conexos —celebrada en Xenebra en decembro de 1996— adoptouse este tratado orientado a ofrecer a necesaria resposta lexislativa aos problemas suscitados pola tecnoloxía dixital, e particularmente por Internet.

3.-ISO 12083. Marcaxe de documentos electrónicos

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

### **15.2.1 Introducción**

Dada a súa importancia, debemos partir da Directiva do Consello do 14 de maio de 1991 sobre a protección xurídica de programas de ordenador; alí



indícasenos que para os efectos da presente Directiva, a expresión “programa de ordenador” inclúe programas en calquera forma, incluso os que están incorporados no “hardware”, e que esta expresión designa tamén o traballo preparatorio de concepción que conduce ao desenvolvemento dun programa de ordenador, sempre que a natureza do traballo preparatorio sexa tal que máis tarde poida orixinar un programa de ordenador.

Esta directiva foi obxecto de transposición ao ordenamento español pola Lei 16/1993, do 23 de decembro, que foi derogada polo Real decreto lexislativo 1/1996, do 12 de abril, polo que se aproba o texto refundido da Lei de propiedade intelectual, regularizando, aclarando e harmonizando as disposicións legais vixentes sobre a materia.

A directiva establece que os Estados membros protexerán mediante dereitos de autor os programas de ordenador como obras literarias, tal como se define no Convenio de Berna para a protección das obras literarias e artísticas.

Para os efectos fins da presente directiva, a expresión “programas de ordenador” comprenderá a súa documentación preparatoria.

Respecto da titularidade dos dereitos, considérase autor do programa de ordenador a persoa física ou grupo de persoas físicas que o crearon ou, cando a lexislación dos Estados membros o permita, a persoa xurídica que sexa considerada titular do dereito pola devandita lexislación. Cando a lexislación dun Estado membro recoñeza as obras colectivas, a persoa física ou xurídica que segundo a tal lexislación cree o programa, será considerada o seu autor.

Cando un programa de ordenador se cree conxuntamente por varias persoas físicas, os dereitos exclusivos serán propiedade común.



Cando un traballador asalariado cree un programa de ordenador no exercicio das funcións que lle foron confiadas, ou seguindo as instrucións do seu empresario, a titularidade dos dereitos económicos correspondentes ao programa de ordenador así creado corresponderán, exclusivamente, ao empresario, agás pacto en contrario.

A protección háselles conceder a todas as persoas físicas e xurídicas que cumpran os requisitos establecidos na lexislación nacional sobre dereitos de autor aplicables ás obras literarias.

De conformidade coa directiva, os dereitos exclusivos do titular incluírán o dereito de realizar ou de autorizar:

a) a reprodución total ou parcial dun programa de ordenador por calquera medio e baixo calquera forma, xa sexa permanente ou transitoria.

Cando a carga, presentación, execución, transmisión ou almacenamento dun programa necesitan a súa reprodución, estes actos estarán suxeitos á autorización do titular do dereito;

b) a tradución, adaptación, arranxo e calquera outra transformación dun programa de ordenador e a reprodución dos resultados de tales actos, sen prexuízo dos dereitos da persoa que transforme o programa de ordenador;

c) calquera forma de distribución pública, incluído o alugueiro, do programa de ordenador orixinal ou das súas copias. A primeira venda na comunidade dunha copia dun programa polo titular dos dereitos ou co seu consentimento esgotará o dereito de distribución na comunidade da devandita copia, salvo o dereito de controlar o subseguinte alugueiro do programa ou dunha copia do mesmo.



Non obstante o anterior, salvo que existan disposicións contractuais específicas, non precisarán a autorización do titular os actos indicados nas letras a) e b) anteriormente citadas cando os devanditos actos sexan necesarios para a utilización do programa de ordenador por parte do adquirente lexítimo con arranxo á súa finalidade proposta, incluída a corrección de erros.

A realización dunha copia de salvagarda por parte dunha persoa con dereito a utilizar o programa non se poderá impedir por contrato sempre que resulte necesaria para a tal utilización.

O usuario lexítimo da copia dun programa estará facultado para observar, estudar ou verificar o seu funcionamento, sen autorización previa do titular, co fin de determinar as ideas e principios implícitos en calquera elemento do programa, sempre que o faga durante calquera das operacións de carga, visualización, execución, transmisión ou almacenamento do programa que ten dereito a facer.

#### 15.2.2 Normativa Estatal

Desde o punto de vista da normativa estatal regúlase no título VII do RD 1/1996, do 12 de abril, polo que se aproba o texto refundido da Lei de propiedade intelectual, que derogou a citada Lei 16/1993, do 23 de decembro, de transposición da Directiva 91/250/CEE

O dereito de autor sobre os programas de ordenador rexerase polos preceptos do presente título VII e, no que non estea especificamente previsto nel, polas disposicións que resulten aplicables da lei.

Para os efectos da lei entenderase por programa de ordenador toda secuencia de instrucións ou indicacións destinadas a ser utilizadas, directa ou indirectamente, nun sistema informático para realizar unha función ou



unha tarefa ou para obter un resultado determinado, calquera que for a súa forma de expresión e fixación.

Para os mesmos efectos, a expresión “programas de ordenador” comprenderá tamén a súa documentación preparatoria. A documentación técnica e os manuais de uso dun programa gozarán da mesma protección que este título lles dispensa aos programas de ordenador.

O programa de ordenador será protexido unicamente se fose orixinal, no sentido de ser unha creación intelectual propia do seu autor.

A protección prevista aplicaráselle a calquera forma de expresión dun programa de ordenador. Así mesmo, esta protección esténdese a calquera versión sucesiva do programa así como aos programas derivados, salvo aquelas creadas co fin de ocasionaren efectos nocivos a un sistema informático.

Cando os programas de ordenador formen parte dunha patente ou dun modelo de utilidade gozarán da protección que lles puidese corresponder por aplicación do réxime xurídico da propiedade industrial.

Non estarán protexidos mediante os dereitos de autor as ideas e principios nos que se basean calquera dos elementos dun programa de ordenador, incluídos os que serven de fundamento ás súas interfaces.

### 1.-Titularidade

Será considerado autor do programa de ordenador a persoa ou grupo de persoas naturais que o crearon, ou a persoa xurídica que sexa considerada como titular dos dereitos de autor nos casos expresamente previstos pola lei. Cando se trate dunha obra colectiva terá a consideración de autor,



salvo pacto en contrario, a persoa natural ou xurídica que a edite e divulgue baixo o seu nome.

Os dereitos de autor sobre un programa de ordenador que sexa resultado unitario da colaboración entre varios autores serán propiedade común e corresponderán a todos estes na proporción que determinen.

Cando un traballador asalariado cree un programa de ordenador, no exercicio das funcións que lle foron confiadas ou seguindo as instrucións do seu empresario, a titularidade dos dereitos de explotación correspondentes ao programa de ordenador así creado, tanto o programa fonte coma o programa obxecto, corresponderá, exclusivamente, ao empresario, salvo pacto en contrario.

A protección háselles conceder a todas as persoas naturais e xurídicas que cumpran os requisitos establecidos na lei para a protección dos dereitos de autor.

## 2.-Duración da protección

Cando o autor sexa unha persoa natural, a duración dos dereitos de explotación dun programa de ordenador será, segundo os distintos supostos que poidan xurdir, a prevista no capítulo I do título III do RD 1/1996, do 12 de abril, polo que se aproba o texto refundido da Lei de propiedade intelectual.

Cando o autor sexa unha persoa xurídica, a duración dos dereitos a que se refire o parágrafo anterior será de setenta anos, computados desde o día 1 de xaneiro do ano seguinte ao da divulgación lícita do programa ou ao da súa creación, se non se chegou a divulgar.

## 3.-Contido dos dereitos de explotación



Os dereitos exclusivos da explotación dun programa de ordenador por parte de quen sexa o seu titular, incluírán o dereito de realizar ou de autorizar:

- a) A reprodución total ou parcial, incluso para uso persoal, dun programa de ordenador por calquera medio e baixo calquera forma, xa for permanente ou transitoria. Cando a carga, presentación, execución, transmisión ou almacenamento dun programa necesiten da súa reprodución deberá dispoñerse de autorización para ese fin, que outorgará o titular do dereito.
- b) A tradución, adaptación, arranxo ou calquera outra transformación dun programa de ordenador e a reprodución dos resultados dos tales actos, sen prexuízo dos dereitos da persoa que transforme o programa de ordenador.
- c) Calquera forma de distribución pública, incluído o alugueiro do programa de ordenador orixinal ou das súas copias.

Para os tales efectos, cando se produza cesión do dereito de uso dun programa de ordenador, entenderase, salvo proba en contrario, que a devandita cesión ten carácter non exclusivo e intransferible, presumíndose, así mesmo, que o é para satisfacer unicamente as necesidades do usuario. A primeira venda na Unión Europea dunha copia dun programa polo titular dos dereitos, ou co seu consentimento, esgotará o dereito de distribución da devandita copia, agás o dereito de controlar o subseguinte alugueiro do programa ou dunha copia do mesmo.

#### 4.-Límites aos dereitos de explotación

Non necesitarán autorización do titular, salvo disposición contractual en contrario:



a) A reprodución ou transformación dun programa de ordenador, incluída a corrección de erros, cando os devanditos actos sexan necesarios para a utilización do mesmo por parte do usuario lexítimo, con arranxo á súa finalidade proposta.

b) A realización dunha copia de seguridade por parte de quen ten dereito a utilizar o programa non se poderá impedir por contrato se resulta necesaria para a tal utilización.

c) O usuario lexítimo da copia dun programa estará facultado para observar, estudar ou verificar o seu funcionamento, sen autorización previa do titular, co fin de determinar as ideas e principios implícitos en calquera elemento do programa, sempre que o faga durante calquera das operacións de carga, visualización, execución, transmisión ou almacenamento do programa que ten dereito a facer.

O autor, salvo pacto en contrario, non se poderá opoñer a que o cesionario titular de dereitos de explotación realice ou autorice a realización de versións sucesivas do seu programa nin de programas derivados deste.

Non será necesaria a autorización do titular do dereito cando a reprodución do código e a tradución da súa forma sexa indispensable para obter a información necesaria para a interoperabilidade dun programa creado de forma independente con outros programas, sempre que se cumpran os seguintes requisitos:

1.-Que os tales actos sexan realizados polo usuario lexítimo ou por calquera outra persoa facultada para utilizar unha copia do programa, ou, no seu nome, por parte dunha persoa debidamente autorizada.

2.-Que a información necesaria para conseguir a interoperabilidade non fose posta previamente e de maneira fácil e rápida a disposición das persoas ás que se refire o parágrafo anterior.



3.-Que os devanditos actos se limiten a aquelas partes do programa orixinal que resulten necesarias para conseguir a interoperabilidade.

#### 5.-Protección rexistral

Os dereitos sobre os programas de ordenador, así como sobre as súas sucesivas versións e os programas derivados, poderán ser obxecto de inscrición no Rexistro da Propiedade Intelectual.

#### 6.-Infracción dos dereitos

De acordo coa normativa vixente terán a consideración de infractores dos dereitos de autor aqueles que, sen autorización do titular dos mesmos, realicen os actos seguintes previstos no artigo 99, ao indicar este que

*“A reprodución total ou parcial, incluso para uso persoal, dun programa de ordenador, por calquera medio e baixo calquera forma, xa for permanente ou transitoria. Cando a carga, presentación, execución, transmisión ou almacenamento dun programa necesiten a súa reprodución deberá dispoñerse de autorización para ese fin, que outorgará o titular do dereito.*

*b. A tradución, adaptación, arranxo ou calquera outra transformación dun programa de ordenador e a reprodución dos resultados dos tales actos, sen prexuízo dos dereitos da persoa que transforme o programa de ordenador.*

*c. Calquera forma de distribución pública, incluída o alugueiro do programa de ordenador orixinal ou das súas copias.”*

Ademais en particular, considéranse infractores:



- 1.- Quen poña en circulación unha ou máis copias dun programa de ordenador coñecendo ou podendo presumir a súa natureza ilexítima.
- 2.-Que teña con fins comerciais unha ou máis copias dun programa de ordenador, coñecendo ou podendo presumir a súa natureza ilexítima.
- 3.-Quen poña en circulación ou teña con fins comerciais calquera instrumento cuxo único uso sexa facilitar a supresión ou neutralización non autorizadas de calquera dispositivo técnico utilizado para protexer un programa de ordenador.

#### 7.- Medidas de protección

O titular dos dereitos recoñecidos sobre programas de ordenador que se dispoñen na lei, é dicir

- 1.-Poderá pedir o cesamento da actividade ilícita, que poderá comprender a suspensión da explotación ou actividade infractora.
- 2.-A prohibición ao infractor de renovar a explotación ou actividade infractora.
- 3.-A retirada do comercio dos exemplares ilícitos e a súa destrución.
- 4.-A retirada dos circuítos comerciais, a inutilización, e, en caso necesario, a destrución dos moldes, pranchas, matrices, negativos e demais elementos materiais, equipos ou instrumentos destinados principalmente á reprodución, á creación ou fabricación de exemplares ilícitos.
- 5.-A remoción ou o precinto dos aparatos utilizados na comunicación pública non autorizada de obras ou prestacións.
- 6.- A remoción ou o precinto dos instrumentos utilizados para facilitar a supresión ou a neutralización non autorizadas de calquera dispositivo técnico empregado para protexer obras ou prestacións, aínda que aquela non fose o seu único uso,



7.-A suspensión dos servizos prestados por intermediarios a terceiros que se valla deles para infrinxir dereitos de propiedade intelectual.

Ademais, as medidas cautelares procedentes, conforme ao disposto na Lei de axuízamento civil.

### **15.3.-TIPOS DE LICENZAS**

Software Libre ou Free Software é un software dispoñible para calquera que desexe utilizalo, copialo e distribuílo, xa sexa na súa forma orixinal ou con modificacións. A posibilidade de modificacións implica que o código fonte está dispoñible. Se un programa é libre, pode ser potencialmente incluído nun sistema operativo tamén libre. É importante non confundir software libre con software gratis, porque a liberdade asociada ao software libre de copiar, modificar e redistribuír, non significa gratuidade. Existen programas gratuítos que non poden ser modificados nin redistribuídos. E existen programas pagos.

#### Copyleft

A maioría das licenzas usadas na publicación de software libre permite que os programas sexan modificados e redistribuídos. Polo xeral, estas prácticas están prohibidas pola lexislación internacional de copyright, que tenta impedir que alteracións e copias sexan efectuadas sen a autorización do ou dos autores. As licenzas que acompañan o software libre fan uso da lexislación de copyright para impedir a utilización non autorizada, pero estas licenzas definen clara e explicitamente as condicións baixo as cales se poden realizar copias, modificacións e redistribucións, co fin de garantir as liberdades de modificar e redistribuír o software rexistrado. A esta versión de copyright, dáselle o nome de copyleft.

#### GPL



A Licenza Pública Xeral GNU (GNU General Public License GPL) é a licenza que acompaña os paquetes distribuídos polo Proxecto GNU, máis unha gran variedade de software que inclúe o núcleo do sistema operativo Linux. A formulación de GPL é tal en vez de limitar a distribución do software que protexe, chega mesmo a impedir que este software sexa integrado en software propietario. A GPL baséase na lexislación internacional de copyright, o que debe garantir cobertura legal para o software licenciado con GPL.

### Debian

A licenza Debian é parte do contrato realizado entre Debian e a comunidade de usuarios de software libre, e denomínase Debian Free Software Guidelines (DFSG). En esencia, esta licenza contén criterios para a distribución que inclúen, ademais da existencia de publicación do código fonte: (a) a redistribución libre; (b) o código fonte debe ser incluído e debe poder ser redistribuído; (c) todo traballo derivado debe poder ser redistribuído baixo a mesma licenza do orixinal; (d) pode haber restricións en canto á redistribución do código fonte se o orixinal foi modificado; (e) a licenza non pode discriminar a ningunha persoa ou grupo de persoas, así como tampouco ningunha forma de utilización do software; (f) os dereitos outorgados non dependen do sitio no que o software se atopa; e (g) a licenza non pode “contaminar” a outro software.

### BSD

A licenza BSD cobre as distribucións de software de Berkeley Software Distribution, ademais doutros programas. É unha licenza considerada “permisiva”, xa que impón poucas restricións sobre a forma de uso, alteracións e redistribución do software. O software pode ser vendido e non hai obrigas de incluír o código fonte. Esta licenza garante o crédito aos autores do software, pero non intenta garantir que as modificacións futuras sigan sendo software libre.



### X.org.

O Consorcio X distribúe X Window System baixo unha licenza que o fai software libre, aínda que sen se adherir ao copyleft. Existen distribucións baixo a licenza da X.org que son software libre, e outras distribucións que non o son. Existen algunhas versións non-libres do sistema de ventás X11 para estacións de traballo e certos dispositivos de IBM-PC que son as únicas funcións dispoñibles, sen outros similares que sexan distribuídos como software libre.

### Software con Dominio Público

O Software con dominio público é software sen copyright. Algúns tipos de copia ou versións modificadas poden non ser libres se o autor impón restricións adicionais na redistribución do orixinal ou de traballos derivados.

### Software Semi-libre

O Software semi-libre é un software que non é libre, pero permite que outros individuos o usen, o copien, o distribúan e mesmo o modifiquen. Exemplos de software semi-libre son as primeiras versións de Internet Explorer de Microsoft, ou algunhas versións de browsers de Netscape, e StarOffice.

### Freeware

O termo 'freeware' non posúe unha definición amplamente aceptada, pero é utilizado para programas que permiten a redistribución aínda que non a modificación, e que inclúen o seu código fonte. Estes programas non son software libre.

### Shareware

Shareware é o software dispoñible co permiso para que sexa redistribuído, pero a súa utilización implica o pagamento. Normalmente, o código fonte non se atopa dispoñible e daquela é imposible realizar modificacións.



### Software Propietario

O Software propietario é aquel cuxa copia, redistribución ou modificación están, nalgunha medida, prohibidos polo seu propietario. Para usar, copiar ou redistribuír, débese solicitar permiso ao propietario ou pagar.

### Software Comercial

O Software comercial é o software desenvolvido por unha empresa co obxectivo de lucrarse coa súa utilización. Nótese que "comercial" e "propietario" non son o mesmo. A maior parte do software comercial é propietario, pero existe software libre que é comercial, e existe software non-libre que non é comercial.

## **15.4 .-SOFTWARE DE FONTES ABERTAS**

O software libre e de código aberto (tamén coñecido como FOSS ou FLOSS, siglas de *free* (libre) and *open source software*, en inglés) é o software que está licenciado de tal xeito que os usuarios poden estudar, modificar e mellorar o seu deseño mediante a dispoñibilidade do seu código fonte.

A expresión "software libre e de código aberto" abrangue os conceptos de software libre e software de código aberto, que, aínda que comparten modelos de desenvolvemento semellantes, teñen diferenzas nos seus aspectos filosóficos. O software libre oríéntase ás liberdades filosóficas que lles outorga aos usuarios, namentres que o software de código aberto oríéntase ás vantaxes do seu modelo de desenvolvemento. "FOSS" é un termo imparcial respecto ás dúas filosofías.

O software gratis non ten que ser necesariamente libre ou de código aberto.

Comparación entre software libre e de código aberto:



Para que un software sexa definido como libre ou de código aberto, ou ambos, debe cumprir certas regras ou normas para merecer esta denominación:

As 4 liberdades do software libre:

- 1.- Executar o programa con calquera propósito (liberdade 0)  
(privado, educativo, público, comercial, militar, etc.).
- 2.-Estudar e modificar o programa (liberdade 1)  
(para o que é preciso poder acceder ao código fonte).
- 3.-Distribuír o programa de maneira que se poida axudar a outros (liberdade 2).
- 4.-Distribuír as versións modificadas propias (liberdade 3).

As 10 premisas do software de código aberto:

- 1.-Libre redistribución: o software debe poder ser regalado ou vendido libremente.
- 2.-Código fonte: o código fonte debe estar incluído ou obterse libremente.
- 3.-Traballos derivados: a redistribución de modificacións debe estar permitida
- 4.-Integridade do código fonte do autor: as licenzas poden requirir que as modificacións sexan redistribuídas só como parches.
- 5.- Sen discriminación de persoas ou grupos: non se pode deixar a ningún fóra.
- 6.-Sen discriminación de áreas de iniciativa: os usuarios comerciais non poden ser excluídos.
- 7.-Distribución da licenza: débese aplicar os mesmos dereitos a todo o que reciba o programa.
- 8.-A licenza non debe ser específica dun produto: o programa non se pode licenciar só como parte dunha distribución maior.



9.-A licenza non debe restrinxir outro software: a licenza non pode obrigir a que outro software que sexa distribuído co software aberto deba tamén ser de código aberto.

10.-A licenza debe ser tecnoloxicamente neutral: non se debe requirir a aceptación da licenza por medio dun acceso por clic de rato, ou doutra forma específica do medio de soporte do software.

### Organizacións e licenzas tras o FOSS

Existen organizacións detrás de cada iniciativa de distinción do software.

Por parte do software libre, existe a Free Software Foundation (FSF); apoiando o concepto de software de código aberto existe a Open Source Initiative (OSI). Ambas se centran en diferentes aspectos do uso e distribución do software, a súa dispoñibilidade e as responsabilidades que compete ao usuario ter. Por este motivo existen diferentes licenzas que as diferencian:

Licenzas de código aberto (para o software de código aberto), licenzas de software libre (para o software libre), entre outras; sen protección herdada e con protección herdada.

Autor:

Alfonso García Magariños

Director Asesoría Xurídica Municipal Concello da Coruña



**16. LEXISLACIÓN SOBRE  
PROTECCIÓN DE DATOS. LEI  
ORGÁNICA 15/1999, DE  
PROTECCIÓN DE DATOS DE  
CARÁCTER PERSOAL, REAL  
DECRETO 1720/2007, DO 21 DE  
DECEMBRO, POLO QUE SE  
APROBA O REGULAMENTO DE  
DESENVOLVEMENTO DA LEI  
ORGÁNICA 15/1999, DO 13 DE  
DECEMBRO, DE PROTECCIÓN  
DE DATOS DE CARÁCTER  
PERSOAL.**



**Tema 16.- Lexislación sobre protección de datos. Lei orgánica 15/1999, de protección de datos de carácter persoal. Real decreto 1720/2007, do 21 de decembro, polo que se aproba o Regulamento de desenvolvemento da Lei orgánica 15/1999, do 13 de decembro, de protección de datos de carácter persoal.**

## **ÍNDICE**

### 16.1.- Lexislación sobre protección de datos

#### 16.1.1 Normativa española

#### 16.1.2 Normativa comunitaria

### 16.2.- Lei orgánica 15/1999, do 13 de decembro, de protección de datos de carácter persoal

#### 16.2.1 Introducción

#### 16.2.2 Principios

#### 16.2.3 Dereitos

#### 16.2.4 Ficheiros públicos e privados

#### 16.2.5 Axencia de protección de datos

#### 16.2.6 Infraccións e sancións

### 16.3.- Real decreto 1720/2007 polo que se aproba o Regulamento de desenvolvemento da Lei orgánica 15/1999, do 13 de decembro, de protección de datos de carácter persoal

#### 16.3.1 Introducción

#### 16.3.2 Análise



## **16.1 LEXISLACIÓN SOBRE PROTECCIÓN DE DATOS.**

Sen prexuízo do desenvolvemento ao longo do tema das normas máis importantes, cabe destacar:

### **16.1.1 Normativa española**

1.-Constitución española de 1978.

2.-Lei orgánica 15/1999, do 13 de decembro, de protección de datos de carácter persoal.

3.-Lei 2/2011, do 4 de marzo, de economía sustentable. Modificación da LOPD. Disposición final quincuaxésima sexta.

4.-Real decreto 1720/2007, do 21 de decembro, polo que se aproba o Regulamento de desenvolvemento da Lei orgánica 15/1999, do 13 de decembro, de protección de datos de carácter persoal.

5.-Real decreto 3/2010, do 8 de xaneiro, polo que se regula o Esquema Nacional de Seguridade no ámbito da Administración Electrónica (D. A. 4ª ).

6.-Real decreto 1665/2008, do 17 de outubro, polo que se modifica o Estatuto da Axencia Española de Protección de Datos, aprobado polo Real decreto 428/1993, do 26 de marzo.

7.-Real decreto 156/1996, do 2 de febreiro, polo que se modifica o Estatuto da Axencia Española de Protección de Datos.

8.-Real decreto 428/1993, do 26 de marzo, polo que se aproba o Estatuto da Axencia Española de Protección de Datos.



### 16.1.2 NORMATIVA COMUNITARIA

#### Xeral

Na Carta dos dereitos fundamentais da Unión Europea, do 7 de decembro do 2000, o artigo 8 dispón:

*“Toda persoa ten dereito á protección dos datos de carácter persoal que a concirnan.*

*Estes datos trátanse de modo leal, para fins concretos e sobre a base do consentimento da persoa afectada ou en virtude doutro fundamento lexítimo previsto pola lei. Toda persoa ten dereito a acceder aos datos recollidos que a concirnan e a obter a súa rectificación.*

*O respecto destas normas estará suxeito ao control dunha autoridade independente.”*

Nas versións consolidadas do Tratado da Unión Europea e do Tratado de Funcionamento da Unión Europea, publicadas no Diario Oficial da Unión Europea o 30 de marzo do 2010, recóllense aspectos relacionados coa protección de datos. Así, no artigo 16 do Tratado de Funcionamento da UE dispónse: *“Toda persoa ten dereito á protección dos datos de carácter persoal que a concirnan.*

*2. O Parlamento Europeo e o Consello establecerán, de conformidade co procedemento lexislativo ordinario, as normas sobre protección das persoas físicas respecto do tratamento de datos de carácter persoal polas institucións, órganos e organismos da Unión, así como polos Estados membros, no exercicio das actividades comprendidas no ámbito de aplicación do dereito da Unión e sobre a libre circulación destes datos. O respecto das devanditas normas estará sometido ao control de autoridades independentes.*



*As normas que se adopten en virtude do presente artigo entenderanse sen prexuízo das normas específicas previstas no artigo 39 do Tratado da Unión Europea”.*

O artigo 39 do Tratado da Unión Europea establece: “*De conformidade co artigo 16 do Tratado de Funcionamento da Unión Europea, e malia o disposto no seu punto 2, o Consello adoptará unha decisión que fixe as normas sobre protección das persoas físicas respecto do tratamento de datos de carácter persoal polos Estados membros, no exercicio das actividades comprendidas no ámbito de aplicación do presente capítulo e sobre a libre circulación dos devanditos datos. O respecto das devanditas normas estará sometido ao control de autoridades independentes”.*

#### Regulamentos:

1.- Regulamento do Eurodac n.º 2725/2000 do Consello do 11 de decembro do 2000 relativo á creación do sistema «Eurodac» para a comparación das impresións dactilares para a aplicación efectiva do Convenio de Dublín, polo que se crea un sistema denominado «Eurodac», que ten como finalidade axudar a determinar o Estado membro responsable, consonte o Convenio de Dublín, do exame das solicitudes de asilo presentadas nos Estados membros e, ademais, facilitar a aplicación do Convenio de Dublín nas condicións establecidas no presente Regulamento. Considérase que as impresións dactilares constitúen un elemento importante para determinar a identidade exacta destas persoas. É necesario crear un sistema para comparar os seus datos dactiloscópicos, considerando estes como datos de carácter persoal.

2.- Regulamento (CE) n.º 45/2001 do Parlamento Europeo e do Consello do 18 de decembro do 2000 relativo á protección das persoas físicas no que respecta ao tratamento de datos persoais, onde se establece que as institucións e os organismos creados polos Tratados constitutivos das comunidades europeas garantirán, de conformidade co presente



Regulamento, a protección dos dereitos e das liberdades fundamentais das persoas físicas e, en particular, o seu dereito á intimidade no que respecta ao tratamento dos datos persoais, e non limitarán nin prohibirán a libre circulación de datos persoais entre eles ou entre eles e destinatarios suxeitos ao dereito nacional dos Estados membros adoptado en aplicación da Directiva 95/46/CE.

A autoridade de control independente establecida polo presente Regulamento, denominada «Supervisor Europeo de Protección de Datos», supervisará a aplicación das disposicións do presente Regulamento a todas as operacións de tratamento realizadas polas institucións e organismos comunitarios.

Enténdese por «datos persoais»: toda información sobre unha persoa física identificada ou identificable; considerarase identificable toda persoa da que se poida determinar a identidade, directa ou indirectamente, en particular mediante un número de identificación ou un ou varios elementos específicos, característicos da súa identidade física, fisiolóxica, psíquica, económica, cultural ou social.

Enténdese por «tratamento de datos persoais» calquera operación ou conxunto de operacións, efectuadas ou non mediante procedementos automatizados, aplicadas a datos persoais, como a recollida, o rexistro, a organización, a conservación, a adaptación ou modificación, a extracción, a consulta, a utilización, a comunicación por transmisión, a difusión ou calquera outra forma que permita o acceso a estes datos, así como o aliñamento ou interconexión, e o bloqueo, supresión ou destrución.

As disposicións do Regulamento aplicaranse ao tratamento de datos persoais por parte de todas as institucións e organismos comunitarios, na medida en que este tratamento se leve a cabo para o exercicio de actividades que pertencen ao ámbito de aplicación do dereito comunitario.



Os datos persoais deberán ser:

- a) tratados de forma leal e lícita;
- b) recollidos con fins determinados, explícitos e lexítimos, e non ser tratados posteriormente de xeito incompatible con estes fins; non se considerará incompatible o tratamento posterior de datos con fins históricos, estatísticos ou científicos, a condición de que o responsable do tratamento estableza as garantías oportunas, en particular para asegurar que os datos non serán tratados con outros fins e que non se van utilizar en favor de medidas ou decisións que afecten a persoas concretas;
- c) adecuados, pertinentes e non excesivos con relación aos fins para os que se soliciten e para os que se traten posteriormente;
- d) exactos e, se fose necesario, actualizados; tomaranse todas as medidas razoables para a supresión ou rectificación dos datos inexactos ou incompletos en relación cos fins para os que foron recollidos ou para os que son tratados posteriormente;
- e) conservados nunha forma que permita a identificación dos interesados durante un período non superior ao necesario para a consecución dos fins para os que foron recollidos ou para os que se traten posteriormente. A institución ou o organismo comunitario establecerá, para os datos persoais que deban ser arquivados por un período máis longo do mencionado para fins históricos, estatísticos ou científicos, que estes datos se arquiven ou ben só en forma anónima, ou, cando iso non sexa posible, só coa identidade codificada do interesado. En calquera caso, deberase impedir o uso dos datos agás para fins históricos, estatísticos ou científicos.

3.- Decisión do Consello do 13 de setembro do 2004 pola que se adoptan as normas de desenvolvemento do Regulamento (CE) n.º 45/2001 do Parlamento Europeo e do Consello relativo á protección das persoas físicas no que respecta ao tratamento de datos persoais polas institucións e os organismos comunitarios e á libre circulación destes datos.



Directivas:

1.- Directiva 2009/136/CE do Parlamento Europeo e do Consello, do 25 de novembro do 2009, pola que se modifican a Directiva 2002/22/CE relativa ao servizo universal e os dereitos dos usuarios en relación coas redes e os servizos de comunicacións electrónicas, a Directiva 2002/58/CE relativa ao tratamento dos datos persoais e á protección da intimidade no sector das comunicacións electrónicas e o Regulamento (CE) n.º 2006/2004 sobre a cooperación en materia de protección dos consumidores.

2.- Directiva 2006/24/CE do Parlamento Europeo e do Consello, do 15 de marzo do 2006, sobre a conservación de datos xerados ou tratados en relación coa prestación de servizos de comunicacións electrónicas de acceso público ou de redes públicas de comunicacións e pola que se modifica a Directiva 2002/58/CE.

3.- Directiva 2004/82/CE do Consello, do 29 de abril do 2004, sobre a obriga dos transportistas de comunicaren os datos das persoas transportadas.

4.- Directiva 2002/58/CE do Parlamento Europeo e do Consello, do 12 de xullo do 2002, relativa ao tratamento dos datos persoais e á protección da intimidade no sector das comunicacións electrónicas (Directiva sobre privacidade e as comunicacións electrónicas).

5.- Directiva 2002/22/CE do Parlamento Europeo e do Consello, do 7 de marzo do 2002, relativa ao servizo universal e os dereitos dos usuarios en relación coas redes e os servizos de comunicacións electrónicas (Directiva servizo universal).



6.- Directiva 2002/21/CE do Parlamento Europeo e do Consello, do 29 de abril do 2004, sobre a obriga dos transportistas de comunicaren os datos das persoas transportadas.

7.- Directiva 2002/20/CE do Parlamento Europeo e do Consello, do 7 de marzo do 2002, relativa á autorización de redes e servizos de comunicacións electrónicas (Directiva de autorización).

8.- Directiva 2002/19/CE do Parlamento Europeo e do Consello, do 7 de marzo do 2002, relativa ao acceso ás redes de comunicacións electrónicas e recursos asociados, e á súa interconexión (Directiva de acceso).

9.- Directiva 2000/31/CE do Parlamento Europeo e do Consello, do 8 de xuño do 2000, relativa a determinados aspectos xurídicos dos servizos da sociedade da información, en particular o comercio electrónico no mercado interior (Directiva sobre o comercio electrónico).

10.- Directiva 1999/93/CE do Parlamento Europeo e do Consello, do 13 de decembro de 1999, pola que se establece un marco comunitario para a sinatura electrónica.

11.- Directiva 97/66/CE do Parlamento Europeo e do Consello, do 15 de decembro de 1997, relativa ao tratamento dos datos persoais e á protección da intimidade no sector das telecomunicacións (derrogada pola Directiva 2002/58/CE).

12.- Directiva 95/46/CE do Parlamento Europeo e do Consello, do 24 de outubro de 1995, relativa á protección das persoas físicas no que respecta ao tratamento de datos persoais e á libre circulación destes datos.

Decisións:



1.- Decisión da Comisión do 31 de xaneiro do 2011 de conformidade coa Directiva 95/46/CE do Parlamento Europeo e do Consello, relativa á protección adecuada dos datos persoais polo Estado de Israel no que respecta ao tratamento automatizado dos datos persoais.

2.- Decisión da Comisión do 19 de outubro do 2010 de conformidade coa Directiva 95/46/CE do Parlamento Europeo e do Consello, relativa á adecuada protección dos datos persoais en Andorra.

3.- Decisión da Comisión do 5 de marzo do 2010 conforme a Directiva 95/46/CE do Parlamento Europeo e do Consello, relativa á protección adecuada dada na lei das Illas Feroe sobre o tratamento de datos persoais.

4.- Decisión da Comisión (2010/87/UE), do 5 de febreiro do 2010, relativa ás cláusulas contractuais tipo para a transferencia de datos persoais aos encargados do tratamento establecidos en terceiros países, de conformidade coa Directiva 95/46/CE do Parlamento Europeo e do Consello.

5.- Decisión da Comisión do 8 de maio do 2008 de conformidade coa Directiva 95/46/CE do Parlamento Europeo e do Consello, relativa á protección adecuada dos datos persoais en Jersey.

6.- Decisión da Comisión do 6 de setembro do 2005, relativa ao carácter adecuado da protección dos datos persoais incluídos nos rexistros de nomes dos pasaxeiros (PNR ou *Passenger Name Record* en inglés) que se lle transfiren á Axencia de Servizos de Fronteiras de Canadá.

7.- Decisión da Comisión do 27 de decembro do 2004, pola que se modifica a Decisión 2001/497/CE no relativo á introdución dun conxunto



alternativo de cláusulas contractuais tipo para a transferencia de datos persoais a terceiros países.

8.- Decisión da Comisión do 14 de maio do 2004 relativa ao carácter adecuado da protección dos datos persoais incluídos nos rexistros de nomes dos pasaxeiros que se lle transfiren ao Servizo de Aduanas e Protección de Fronteiras dos Estados Unidos (*Bureau of Customs and Border Protection*).

9.- Decisión da Comisión do 29 de abril do 2004 pola que se establece unha lista de organismos cuxos investigadores poden acceder, con fins científicos, a datos confidenciais.

10.- Decisión da Comisión do 28 de abril do 2004 relativa ao carácter adecuado da protección de datos persoais na illa de Man.

11.- Decisión da Comisión de novembro do 2003, relativa ao carácter adecuado da protección de datos persoais en Guernsey.

12.- Decisión da Comisión do 30 de xuño do 2003, conforme a Directiva 95/46/CE do Parlamento Europeo e do Consello sobre a adecuación da protección dos datos persoais en Arxentina.

13.- Decisión reguladora da Unidade de Cooperación Xudicial Eurojust.

14.- Decisión 2002/16/CE da Comisión, do 27 de decembro do 2001, relativa a cláusulas contractuais tipo para a transferencia de datos persoais aos encargados do tratamento establecidos en terceiros países, de conformidade coa Directiva 95/46/CE (queda derogada a partir do 15 de maio do 2010).



15.- Decisión da comisión do 20 de decembro do 2001 conforme a Directiva 95/46/CE do Parlamento Europeo e do Consello, sobre a adecuación da protección dos datos persoais conferida pola lei canadense *Personal Information and Electronic Documents Act*.

16.- Decisión 2001/497/CE da Comisión, do 15 de xuño do 2001, relativa a cláusulas contractuais tipo para a transferencia de datos persoais a un terceiro país previstas na Directiva 95/46/CE.

17.- Decisión da Comisión do 26 de xullo do 2000 conforme a Directiva 95/46/CE do Parlamento Europeo e do Consello relativa ao nivel de protección adecuado dos datos persoais en Suíza.

18.- Decisión da Comisión do 26 de xullo do 2000 conforme a Directiva 95/46/CE do Parlamento Europeo e do Consello relativa á protección adecuada dos datos persoais en Hungría.

19.- Decisión da Comisión do 26 de xullo do 2000 conforme a Directiva 95/46/CE do Parlamento Europeo e do Consello, sobre a adecuación conferida polos principios de porto seguro para a protección da vida privada e as correspondentes preguntas máis frecuentes, publicadas polo Departamento de Comercio dos Estados Unidos de América.

#### Convenios:

1.- Convenio de Europol.

2.- Convenio de Schengen.

3.- Convenio do Sistema de Información de Aduanas.



## **16.2 LEI ORGÁNICA 15/1999, DO 13 DE DECEMBRO, DE PROTECCIÓN DE DATOS DE CARÁCTER PERSOAL.**

### **16.2.1 Introducción**

A Lei ten por obxecto garantir e protexer, no que concirne o tratamento dos datos persoais, as liberdades públicas e os dereitos fundamentais das persoas físicas, e especialmente do seu honor e intimidade persoal e familiar.

Será de aplicación para os datos de carácter persoal rexistrados en soporte físico que os faga susceptibles de tratamento, e para toda modalidade de uso posterior destes datos polos sectores público e privado.

Para os efectos da Lei orgánica entenderase por:

- a) Datos de carácter persoal: calquera información concernente a persoas físicas identificadas ou identificables.
- b) Ficheiro: todo conxunto organizado de datos de carácter persoal, calquera que for a forma ou modalidade da súa creación, almacenamento, organización e acceso.
- c) Tratamento de datos: operacións e procedementos técnicos de carácter automatizado ou non, que permitan a recollida, gravación, conservación, elaboración, modificación, bloqueo e cancelación, así como as cesións de datos que resulten de comunicacións, consultas, interconexións e transferencias.
- d) Responsable do ficheiro ou tratamento: persoa física ou xurídica, de natureza pública ou privada, ou órgano administrativo, que decida sobre a finalidade, contido e uso do tratamento.
- e) Afectado ou interesado: persoa física titular dos datos que sexan obxecto do tratamento a que se refire o apartado c) do presente artigo.



- f) Procedemento de disociación: todo tratamento de datos persoais de xeito que a información que se obteña non se poida asociar a persoa identificada ou identificable.
- g) Encargado do tratamento: a persoa física ou xurídica, autoridade pública, servizo ou calquera outro organismo que, só ou en conxunto con outros, trate datos persoais por conta do responsable do tratamento.
- h) Consentimento do interesado: toda manifestación de vontade, libre, inequívoca, específica e informada, mediante a que o interesado consinta o tratamento de datos persoais que lle concirnen.
- i) Cesión ou comunicación de datos: toda revelación de datos realizada a unha persoa distinta do interesado.
- j) Fontes accesibles ao público: aqueles ficheiros que poden ser consultados por calquera persoa, non impedida por unha norma limitativa, ou sen máis esixencia que, se é o caso, o pagamento dunha contraprestación. Teñen a consideración de fontes de acceso público, exclusivamente, o censo promocional, os repertorios telefónicos nos termos previstos pola súa normativa específica e as listas de persoas pertencentes a grupos de profesionais que conteñan só os datos de nome, título, profesión, actividade, grao académico, enderezo e indicación da súa pertenza ao grupo. Así mesmo, teñen o carácter de fontes de acceso público os diarios e boletíns oficiais e os medios de comunicación.

### 16.2.2 Principios

- 1.- Calidade dos datos: os datos de carácter persoal só se poderán recoller para o seu tratamento, así como someter a este tratamento, cando sexan adecuados, pertinentes e non excesivos en relación co ámbito e as finalidades determinadas, explícitas e lexítimas para as que se obtiveron.
- 2.- Dereito de información na recollida de datos: os interesados aos que se lles soliciten datos persoais deberán ser previamente informados de modo expreso, preciso e inequívoco:



a) Da existencia dun ficheiro ou tratamento de datos de carácter persoal, da finalidade da recollida e dos destinatarios da información.

b) Do carácter obrigatorio ou facultativo da súa resposta ás preguntas que se lles formulen.

c) Das consecuencias da obtención dos datos ou da negativa a proporcionalos.

d) Da posibilidade de exercitaren os dereitos de acceso, rectificación, cancelación e oposición.

e) Da identidade e enderezo do responsable do tratamento ou, de ser o caso, do seu representante.

3.- Principio do consentimento do afectado: o tratamento dos datos de carácter persoal requirirá o consentimento inequívoco do afectado, salvo que a lei dispoña outra cousa.

4.- Comunicación de datos: os datos de carácter persoal obxecto do tratamento só se lle poderán comunicar a un terceiro para o cumprimento de fins directamente relacionados coas funcións lexítimas do cedente e do cesionario co consentimento previo do interesado.

5.- Deber de segredo: o responsable do ficheiro e quen interveñan en calquera fase do tratamento dos datos de carácter persoal están obrigados ao segredo profesional respecto deles e ao deber de gardalos, obrigas que subsistirán mesmo logo de finalizaren as súas relacións co titular do ficheiro ou, de ser o caso, co seu responsable.

6.- Especial protección a determinados datos: ninguén poderá ser obrigado a declarar sobre a súa ideoloxía, relixión ou crenzas.

7.- Seguridade dos datos: o responsable do ficheiro e, de ser o caso, o encargado do tratamento, deberán adoptar as medidas de índole técnica e organizativas necesarias que garantan a seguridade dos datos de carácter persoal e eviten a súa alteración, perda, tratamento ou acceso non



autorizado, tendo en conta o estado da tecnoloxía, a natureza dos datos almacenados e os riscos a que están expostos, proveñan ben da acción humana ou ben do medio físico ou natural.

### 16.2.3 Dereitos das persoas

Os cidadáns teñen dereito a non se veren sometidos a unha decisión con efectos xurídicos, sobre eles ou que lles afecte de xeito significativo, que se basee unicamente nun tratamento de datos destinados a avaliar determinados aspectos da súa personalidade. Cómpre salientar:

1.- Dereito de consulta ao Rexistro Xeral de Protección de Datos: calquera persoa poderá coñecer, solicitando con ese fin a información oportuna do Rexistro Xeral de Protección de Datos, a existencia de tratamentos de datos de carácter persoal, as súas finalidades e a identidade do responsable do tratamento. O Rexistro Xeral será de consulta pública e gratuíta.

2.- Dereito de acceso: o interesado terá dereito a solicitar e obter de balde información dos seus datos de carácter persoal sometidos a tratamento, a orixe destes datos, así como as comunicacións realizadas ou que se prevé realizar deles.

3.- Dereito de rectificación e cancelación: o responsable do tratamento terá a obriga de facer efectivo o dereito de rectificación ou cancelación do interesado no prazo de dez días.

4.- Dereito de tutela: as actuacións contrarias ao disposto na lei poden ser obxecto de reclamación por parte dos interesados perante a Axencia Española de Protección de Datos.



5.- Dereito de indemnización: os interesados que, como consecuencia do incumprimento do disposto na presente Lei por parte do responsable ou do encargado do tratamento, sufran dano ou lesión nos seus bens ou dereitos terán dereito a seren indemnizados.

#### 16.2.4 Ficheiros públicos e privados

A creación, modificación ou supresión dos ficheiros das administracións públicas só se poderán facer por medio de disposición xeral publicada no Boletín Oficial do Estado ou no diario oficial correspondente.

As disposicións de creación ou de modificación de ficheiros deberán indicar:

- a.- A finalidade do ficheiro e os usos previstos para el.
- b.- As persoas ou colectivos sobre os que se pretenda obter datos de carácter persoal ou que resulten obrigados a proporcionalos.
- c.- O procedemento de recollida dos datos de carácter persoal.
- d.- A estrutura básica do ficheiro e a descrición dos tipos de datos de carácter persoal incluídos nel.
- e.- As cesións de datos de carácter persoal e, se é o caso, as transferencias de datos que se prevé facer a terceiros países.
- f.- Os órganos das administracións responsables do ficheiro.
- g.- Os servizos ou unidades ante os que se puidesen exercitar os dereitos de acceso, rectificación, cancelación e oposición.



h.- As medidas de seguridade con indicación do nivel básico, medio ou alto esixible.

Poderanse crear ficheiros de titularidade privada que conteñan datos de carácter persoal cando resulte necesario para o logro da actividade ou obxecto lexítimos da persoa, empresa ou entidade titular e se respecten as garantías que esta Lei establece para a protección das persoas.

Toda persoa ou entidade que proceda á creación de ficheiros de datos de carácter persoal notificarallo previamente á Axencia Española de Protección de Datos.

Entre os datos que debe conter a notificación figurarán, necesariamente, o responsable do ficheiro, a súa finalidade, a súa localización, o tipo de datos de carácter persoal que contén, as medidas de seguridade, con indicación do nivel básico, medio ou alto esixible, e as cesións de datos de carácter persoal que estea previsto realizar e, de ser o caso, as transferencias de datos que estea previsto facer a terceiros países.

Deberánselle comunicar á Axencia Española de Protección de Datos os cambios que se produzan na finalidade do ficheiro automatizado, no seu responsable e no enderezo da súa localización.

O Rexistro Xeral de Protección de Datos inscribirá o ficheiro se a notificación se axusta aos requisitos esixibles.

En caso contrario, poderá pedir que se completen os datos que falten ou se proceda á súa emenda.

Transcorrido un mes desde a presentación da solicitude de inscrición sen que a Axencia Española de Protección de Datos resolverse sobre esta, entenderase inscrito o ficheiro automatizado para todos os efectos.



### 16.2.5 Axencia de protección de datos

A Axencia Española de Protección de Datos é un ente de dereito público, con personalidade xurídica propia e plena capacidade pública e privada, que actúa con plena independencia das administracións públicas no exercicio das súas funcións.

Son funcións da Axencia Española de Protección de Datos:

- 1.- Velar polo cumprimento da lexislación sobre protección de datos e controlar a súa aplicación, en especial no relativo aos dereitos de información, acceso, rectificación, oposición e cancelación de datos.
- 2.- Emitir as autorizacións previstas na lei ou nas súas disposicións regulamentarias.
- 3.- Ditar, se é o caso, e sen prexuízo das competencias doutros órganos, as instrucións precisas para adecuar os tratamentos aos principios da lei.
- 4.- Atender as peticións e reclamacións formuladas polas persoas afectadas.
- 5.- Proporcionar información ás persoas achega dos seus dereitos en materia de tratamento dos datos de carácter persoal.
- 6.- Requirir dos responsables e dos encargados dos tratamentos, logo da súa audiencia, a adopción das medidas necesarias para a adecuación do tratamento de datos ás disposicións desta Lei e, se é o caso, ordenar o cesamento dos tratamentos e a cancelación dos ficheiros cando non se axuste ás súas disposicións.



7.- Exercer a potestade sancionadora nos termos previstos pola Lei orgánica de protección de datos.

8.- Informar, con carácter preceptivo, dos proxectos de disposicións xerais que desenvolvan esta Lei.

9.- Solicitalles aos responsables dos ficheiros tanta axuda e información estime necesaria para o desempeño das súas funcións.

10.- Velar pola publicidade da existencia dos ficheiros de datos con carácter persoal, para o cal publicará periodicamente unha relación destes ficheiros coa información adicional que o director da Axencia determine.

11.- Redactar unha memoria anual e remitirla ao Ministerio de Xustiza.

12.- Exercer o control e adoptar as autorizacións que procedan en relación cos movementos internacionais de datos, así como desempeñar as funcións de cooperación internacional en materia de protección de datos persoais.

13.- Velar polo cumprimento das disposicións que a Lei da función estatística pública establece respecto da recollida de datos estatísticos e do segredo estatístico, así como ditar as instrucións precisas, ditaminar sobre as condicións de seguridade dos ficheiros constituídos con fins exclusivamente estatísticos.

As resolucións da Axencia Española de Protección de Datos faranse públicas en canto lles sexan notificadas aos interesados. A publicación realizarase preferentemente a través de medios informáticos ou telemáticos.

#### 16.2.6 Infraccións e sancións



Os responsables dos ficheiros e os encargados dos tratamentos estarán suxeitos ao réxime sancionador establecido na presente Lei.

As infraccións cualificaranse como leves, graves ou moi graves.

Son infraccións leves:

- 1.- Non lle remitir á Axencia Española de Protección de Datos as notificacións previstas nesta Lei ou nas súas disposicións de desenvolvemento.
- 2.- Non solicitar a inscrición do ficheiro de datos de carácter persoal no Rexistro Xeral de Protección de Datos.
- 3.- O incumprimento do deber de información ao afectado acerca do tratamento dos seus datos de carácter persoal cando os datos lle sexan solicitados ao propio interesado.
- 4.- A transmisión dos datos a un encargado do tratamento sen lles dar cumprimento aos deberes formais establecidos na Lei.

Son infraccións graves:

- 1.- Proceder á creación de ficheiros de titularidade pública ou iniciar a recollida de datos de carácter persoal para estes, sen autorización de disposición xeral publicada no Boletín Oficial do Estado ou diario oficial correspondente.
- 2.- Tratar datos de carácter persoal sen solicitar o consentimento das persoas afectadas, cando este sexa necesario conforme ao disposto na Lei e nas súas disposicións de desenvolvemento.



3.- Tratar datos de carácter persoal ou usalos posteriormente con conculcación dos principios e garantías establecidos na Lei e nas disposicións que a desenvolven, salvo cando sexa constitutivo de infracción moi grave.

4.- A vulneración do deber de gardar segredo acerca do tratamento dos datos de carácter persoal.

5.- O impedimento ou a obstaculización do exercicio dos dereitos de acceso, rectificación, cancelación e oposición.

6.- O incumprimento do deber de información ao afectado acerca do tratamento dos seus datos de carácter persoal cando os datos non lle sexan solicitados ao propio interesado.

7.- O incumprimento dos restantes deberes de notificación ou requirimento ao afectado.

8.- Manter os ficheiros, locais, programas ou equipos que conteñan datos de carácter persoal sen as debidas condicións de seguridade que por vía regulamentaria se determinen.

9.- Non atender os requirimentos ou apercibimentos da Axencia Española de Protección de Datos ou non lle proporcionar a aquela cantos documentos e informacións sexan solicitados por ela.

10.- A obstrución ao exercicio da función inspectora.

11.- A comunicación ou cesión dos datos de carácter persoal sen contar con lexitimación para iso, salvo que esta sexa constitutiva de infracción moi grave.



Son infraccións moi graves:

- 1.- A recollida de datos de forma enganosa ou fraudulenta.
- 2.- Tratar ou ceder os datos de carácter persoal especialmente protexidos (ideoloxía, afiliación sindical, relixión e crenzas, orixe racial, saúde e vida sexual, comisión de infraccións penais ou administrativas).
- 3.- Non cesar no tratamento ilícito de datos de carácter persoal cando existise un requirimento previo do director da Axencia Española de Protección de Datos para iso.
- 4.- A transferencia internacional de datos de carácter persoal con destino a países que non proporcionen un nivel de protección equiparable sen autorización do director da Axencia Española de Protección de Datos, agás nos supostos en que esta autorización non resulta necesaria.

Tipo de sancións.

As infraccións leves serán sancionadas con multa de 900 a 40.000 euros.

As infraccións graves serán sancionadas con multa de 40.001 a 300.000 euros.

As infraccións moi graves serán sancionadas con multa de 300.001 a 600.000 euros.

A contía das sancións graduarase atendendo os seguintes criterios:

- a.- O carácter continuado da infracción.



- b.- O volume dos tratamentos efectuados.
- c.- A vinculación da actividade do infractor coa realización de tratamentos de datos de carácter persoal.
- d.- O volume de negocio ou actividade do infractor.
- e.- Os beneficios obtidos como consecuencia da comisión da infracción.
- f.- O grao de intencionalidade.
- g.- A reincidencia por comisión de infraccións da mesma natureza.
- h.- A natureza dos prexuízos causados ás persoas interesadas ou a terceiras persoas.
- i.- A acreditación de que con anterioridade aos feitos constitutivos de infracción a entidade imputada tiña implantados procedementos adecuados de actuación na recollida e tratamento dos datos de carácter persoal, sendo a infracción consecuencia dunha anomalía no funcionamento destes procedementos non debida a unha falta de dilixencia esixible ao infractor.
- j.- Calquera outra circunstancia que sexa relevante para determinar o grao de antixuridicidade e de culpabilidade presentes na concreta actuación infractora.

### **16.3 Real decreto 1720/2007 polo que se aproba o Regulamento de desenvolvemento da Lei orgánica 15/1999, do 13 de decembro, de protección de datos de carácter persoal.**

#### **16.3.1 Introducción**



A Lei orgánica 15/1999, do 13 de decembro, de protección de datos de carácter persoal adaptou o noso ordenamento ao disposto pola Directiva 95/46/CE do Parlamento Europeo e do Consello, do 24 de outubro de 1995, relativa á protección das persoas físicas no que respecta ao tratamento de datos persoais e á libre circulación destes datos, derogando, pola súa vez, a ata entón vixente Lei orgánica 5/1992, do 29 de outubro, de regulación do tratamento automatizado de datos de carácter persoal.

O Regulamento comparte coa Lei orgánica a finalidade de facer fronte aos riscos que para os dereitos da personalidade poden supoñer a provisión e tratamento de datos persoais. Por iso, cómpre salientar que esta norma regulamentaria nace coa vocación de non reiterar os contidos da norma superior e de desenvolver, non só os mandatos contidos na Lei orgánica de acordo cos principios que emanan da Directiva, senón tamén aqueles que nestes anos de vixencia da Lei se demostrou que precisan dun maior desenvolvemento normativo.

O Regulamento parte da necesidade de dotar de coherencia a regulación regulamentaria en todo o relacionado coa transposición da Directiva e de desenvolver os aspectos novidosos da Lei orgánica 15/1999, xunto con aqueles en que a experiencia aconsellou un certo grao de precisión que dote de seguridade xurídica o sistema.

### 16.3.2 Análise

O título I contempla o obxecto e ámbito de aplicación do Regulamento. Ao longo da vixencia da Lei orgánica 15/1999, advertiuse a conveniencia de aclarar qué se entende por ficheiros e tratamentos relacionados con actividades persoais ou domésticas, aspecto moi relevante dado que está excluído da normativa sobre protección de datos de carácter persoal.



Achégase un conxunto de definicións que axudan ao correcto entendemento da norma, o que resulta particularmente necesario nun ámbito tan tecnificado como o da protección de datos persoais. Por outra banda, fixa o criterio que cómpre seguir en materia de cómputo de prazos co fin de homoxeneizar esta cuestión, evitando distincións que supoñen diferenzas de tratamento dos ficheiros públicos respecto dos privados.

O título II refírese aos principios da protección de datos. Reviste particular importancia a regulación do modo de captación do consentimento atendendo a aspectos moi específicos como o caso dos servizos de comunicacións electrónicas e, moi particularmente, a captación de datos dos menores. Así mesmo, ofrécese o que non se pode definir senón como un estatuto do encargado do tratamento, que sen dúbida contribuirá a clarificar todo o relacionado con esta figura. As previsións neste ámbito complétanse co disposto no título VIII en materia de seguridade dotando dun marco coherente a actuación do encargado.

O título III ocúpase dunha cuestión tan esencial como os dereitos das persoas neste ámbito. Estes dereitos de acceso, rectificación, cancelación e oposición ao tratamento, segundo afirmou o Tribunal Constitucional na súa sentenza número 292/2000, constitúen o feixe de facultades que emanan do dereito fundamental á protección de datos e serven a capital función que desempeña este dereito fundamental: garantirlle á persoa un poder de control sobre os seus datos persoais, o que só é posible e efectivo impoñéndolles a terceiros os mencionados deberes de facer.

A seguir, os títulos IV a VII permiten clarificar aspectos importantes para o tráfico ordinario, como a aplicación de criterios específicos a determinado tipo de ficheiros de titularidade privada que, pola súa transcendencia, o requirían —os relativos á solvencia patrimonial e crédito e os utilizados en actividades de publicidade e prospección comercial—, o conxunto de obrigas materiais e formais que deben conducir os responsables á



creación e inscrición dos ficheiros, os criterios e procedementos para a realización das transferencias internacionais de datos, e, para rematar, a regulación dun instrumento, o código tipo, chamado a cumprir un papel cada vez máis relevante como elemento dinamizador do dereito fundamental á protección de datos.

O título VIII regula un aspecto esencial para a tutela do dereito fundamental á protección de datos, a seguridade, que repercute sobre múltiples aspectos organizativos, de xestión e mesmo de investimento, en todas as organizacións que traten datos persoais. A repercusión do deber de seguridade obrigaba a un rigor particular, xa que nesta materia confluíron distintos elementos moi relevantes. Por unha banda, a experiencia dimanante da aplicación do Real decreto 994/1999 permitía coñecer as dificultades que enfrontaran os responsables e identificar os puntos débiles e fortes da regulación. Por outra, reclamábase a adaptación da regulación en distintos aspectos. Neste sentido, o regulamento trata de ser particularmente rigoroso na atribución dos niveis de seguridade, na fixación das medidas que corresponda adoptar en cada caso e na revisión destas cando iso resulte necesario. Por outra banda, ordena con maior precisión o contido e as obrigas vinculadas ao mantemento do documento de seguridade. Ademais, pretendeuse regular a materia de xeito que contemple as múltiples formas de organización material e persoal da seguridade que se dan na práctica. Para rematar, régúlase un conxunto de medidas destinadas aos ficheiros e tratamentos estruturados e non automatizados que lles ofrezca aos responsables un marco claro de actuación.

Finalmente, no título IX, dedicado aos procedementos tramitados pola Axencia Española de Protección de Datos, optouse por regular exclusivamente aquelas especialidades que diferencian os distintos procedementos tramitados pola Axencia das normas xerais previstas para os procedementos na Lei 30/1992, do 26 de novembro, de réxime xurídico



das administracións públicas e do procedemento administrativo común, cuxa aplicación se declara supletoria ao presente Regulamento.

Autor:

Alfonso García Magariños

Director Asesoría Xurídica Municipal Concello da Coruña



**17. PLAN DIRECTOR DE  
SEGURIDADE DE  
INFORMACIÓN DA XUNTA DE  
GALICIA. DECRETO 230/2008  
POLO QUE SE ESTABLECEN AS  
NORMAS DE BOAS PRÁCTICAS  
NA UTILIZACIÓN DOS  
SISTEMAS DE INFORMACIÓN  
DA ADMINISTRACIÓN DA  
COMUNIDADE AUTÓNOMA DE  
GALICIA.**



## **Tema 17.- Plan Director de Seguridade de Información da Xunta de Galicia. Decreto 230/2008 polo que se establecen as normas de boas prácticas na utilización de sistemas de información na Administración da C.A. de Galicia**

### **ÍNDICE**

#### 17.1 Plan Director de Seguridade e Información da Xunta de Galicia

- 17.1.1. Introducción
- 17.1.2 Responsabilidades
- 17.1.3 Obxectivos
- 17.1.4 Axentes involucrados

#### 17.2 Decreto 230/2008 polo que se establecen as normas de boas prácticas na utilización de sistemas de información na Administración da C.A. de Galicia.

- 17.2.1 Obxecto e ámbito de aplicación
- 17.2.2 Órganos responsables e de coordinación
- 17.2.3 Acceso á información, redes de comunicacións e Internet
- 17.2.4 Servizo de mensaxería corporativo
- 17.2.5 Deber de persoas usuarias
- 17.2.6 Inspección
- 17.2.7 Responsabilidade de persoas usuarias que teñan a condición de empregados públicos



## **17.1 PLAN DIRECTOR DE SEGURIDADE E INFORMACION DA XUNTA DE GALICIA**

### **17.1.1 Introducción**

O Plan Director de Seguridade da Información establece as actuacións que debe levar a cabo a Administración da Comunidade Autónoma de Galicia en materia de seguridade da información, así como os axentes involucrados e as súas respectivas responsabilidades. Ademais, define as directrices necesarias para xestionar de forma segura os sistemas de información da Administración e manifesta o recoñecemento da importancia que ten a seguridade da información sobre a confianza que os cidadáns depositan na Administración.

As tecnoloxías da información e as comunicacións (TIC) constitúen un instrumento de alto nivel estratéxico polo seu potencial para impulsar a modernización da Administración da Comunidade Autónoma Galega, así como pola súa capacidade para estimular e sustentar o desenvolvemento social e económico de Galicia.

Este plan abrangue todos os sistemas de información da Xunta de Galicia. Nace coa vontade de ser aplicado progresivamente ao conxunto de consellerías da Administración autonómica, aos seus organismos autónomos, sociedades públicas, fundacións do sector público autonómico e demais entidades de dereito público vinculadas ou dependentes da Comunidade Autónoma de Galicia. Sen prexuízo do anterior, está directamente destinado a garantir a seguridade dos sistemas corporativos.



Dentro deste plan, correspóndelle á Secretaría Xeral de Modernización e Innovación Tecnolóxica a análise de necesidades, planificación, deseño, xestión e implantación dos sistemas de información e elementos tecnolóxicos nos órganos da Administración de Xustiza en Galicia, en coordinación coas administracións e órganos competentes na materia de sistemas de información de xustiza.

Neste plan tamén se perfilan unha serie de iniciativas orientadas a asesorar e sensibilizar o persoal da Administración Local en materia de protección de datos, de seguridade informática e de acceso electrónico dos cidadáns aos servizos públicos.

O alcance temporal do presente plan abarca o período 2010-2014. No plan efectúase unha priorización das distintas actuacións dependendo da súa urxencia e impacto. A seguridade da información require unha visión global que recolla unha mellora pragmática e accesible a curto prazo e, á vez, un modelo obxectivo ambicioso e de longo prazo.

#### 17.1.2 Responsabilidades

Á Secretaría Xeral de Modernización e Innovación Tecnolóxica, segundo o Decreto 325/2009, do 18 de xuño, de estrutura orgánica dos órganos superiores dependentes da Presidencia da Xunta de Galicia, correspóndelle establecer a política de seguridade informática corporativa da Xunta de Galicia e a promoción de boas prácticas no relativo ao tratamento de datos de carácter persoal. A través do Centro de Seguridade da Información (CSI) elaborará os plans, medidas e directrices de seguridade informática, supervisará o cumprimento de todas as medidas de seguridade informática nos diferentes ámbitos e departamentos e deseñará e realizará as accións encamiñadas a garantir o cumprimento da normativa vixente en materia de Protección de Datos de Carácter Persoal na Administración da Comunidade Autónoma.



As consellerías da Administración da Comunidade Autónoma de Galicia, segundo o Decreto 230/2008, do 18 de setembro, polo que se establecen as normas de boas prácticas na utilización dos sistemas de información da Administración da Comunidade Autónoma de Galicia, designarán o órgano que será responsable dos sistemas de información da súa propiedade e de establecer os medios tecnolóxicos que necesitan as persoas ao seu servizo, así como de velar polo correcto funcionamento das infraestruturas e do equipamento informático e de comunicacións de que dispoñan. Cando estas atribucións non lle estean asignadas regulamentariamente a un órgano, a designación hase facer polas secretarías xerais de cada departamento. Así mesmo, cada departamento da Xunta de Galicia deberá designar unha persoa como responsable de seguridade.

O Comité de Seguridade dos Sistemas de Información (CSSI), creado no Decreto de boas prácticas, é un órgano colexiado, formado polas persoas responsables de seguridade dos distintos departamentos da Xunta de Galicia, que ten como obxectivo definir a política de seguridade corporativa. Este comité estará coordinado e asesorado polo órgano competente en materia de seguridade corporativa a través do Centro de seguridade da Información.

As persoas que lle prestan servizos á Administración da Comunidade Autónoma de Galicia, ademais de cumpriren coas medidas indicadas no Decreto de boas prácticas, teñen deber de sigilo e confidencialidade respecto da información á que poidan ter acceso por razón das súas funcións, limitándose a empregala para o estrito cumprimento das tarefas encomendadas.

A Xunta de Galicia expresa o seu compromiso coa seguridade, de forma que dará a coñecer este Plan Director de Seguridade entre todo o persoal que preste os seus servizos na Administración da Comunidade Autónoma,



velará polo seu cumprimento e impulsará a implantación e difusión da xestión da seguridade da información nas persoas e empresas de Galicia.

### 17.1.3 Obxectivos

O presente plan trata de mellorar o nivel de seguridade existente na Administración da Comunidade Autónoma de Galicia. Isto significa xestionar a seguridade da información de tal forma que as medidas de seguridade implantadas alcancen un alto grao de efectividade que reduza ao máximo o impacto das incidencias de seguridade.

Con este fin, a Xunta de Galicia fíxase, en materia de seguridade da información, os seguintes obxectivos:

- . Concienciar e implicar na xestión da seguridade a toda a dirección e persoal da Administración autonómica, os seus organismos autónomos, sociedades públicas, fundacións do sector público autonómico e demais entidades de dereito público vinculadas ou dependentes da Comunidade Autónoma de Galicia, contando coa colaboración de provedores e especialistas.
- . Promover o máximo aproveitamento das tecnoloxías da información e as comunicacións na actividade administrativa e asegurar asemade o respecto das garantías e dereitos dos cidadáns nas súas relacións coa Administración, establecendo que a seguridade nos sistemas sexa atendida, revisada e auditada por persoal cualificado e que a súa configuración e deseño garantan a seguridade por defecto.
- . Xestionar os riscos que poidan afectar aos compoñentes dos sistemas de información para poder identificalos e avalialos e tomar as medidas de seguridade informática adecuadas (físicas, lóxicas, técnicas, normativas e organizativas).



- . Garantir a dispoñibilidade dos sistemas de información de acordo cos requisitos establecidos para os servizos prestados, xestionando e controlando o acceso físico e lóxico; certificando que os produtos de seguridade usados para lle dar servizo ao cidadán cumpren cos estándares establecidos; revisando o nivel de actualización; asegurando a confidencialidade da información xestionada pola Administración e evitando accesos ou alteracións indebidas e perdas de información; implantando a seguridade perimetral necesaria ante os posibles riscos de interconexión con outros sistemas de redes externas e rexistrando toda actividade dos usuarios que accedan ao sistema.
- . Xestionar a continuidade dos servizos TIC proporcionados pola Administración establecendo os sistemas de protección a nivel organizativo, lóxico e físico que permitan reducir a probabilidade de que se produza un incidente, e, no caso de que se produza, acurtar o tempo de regreso á normalidade e minimizar o seu impacto.
- . Avaliar periodicamente o sistema de xestión de seguridade baseándose no rexistro e tratamento de incidencias, garantindo que as medidas implantadas alcancen un nivel de madurez optimizado con políticas, normas, procedementos e estruturas organizacionais que promovan a concienciación e formación continua dos usuarios en temas de seguridade da información.
- . Proporcionar un contorno de seguridade que garanta o cumprimento dos requisitos legais para a validez e eficacia dos procedementos administrativos que utilicen os medios electrónicos, informáticos e telemáticos, establecendo para cada sistema de información responsables diferenciados da información, do servizo e da seguridade.
- . A Xunta de Galicia actuará como elemento xerador de e-confianza



promovendo un tecido empresarial sólido en seguridade da información e incrementando a confianza e a protección dos dereitos da cidadanía galega na sociedade da información.

#### 17.1.4 Axentes involucrados

##### 1.- Subdirección Xeral de Calidade, Interoperabilidade e Seguridade (SXCIS).

A Subdirección Xeral de Calidade, Interoperabilidade e Seguridade (SXCIS) da SXMIT ten, entre as súas funcións, a de establecer a política de seguridade informática corporativa da Xunta de Galicia e a de promover as boas prácticas no relativo ao tratamento de datos de carácter persoal.

Con este fin creouse, dentro da SXCIS, o Servizo de Calidade e Seguridade Informática coas seguintes competencias en materia de seguridade da información:

- . Dirección das políticas corporativas de seguridade informática da Xunta de Galicia a través do Centro de Seguridade da Información.
- . Elaboración dos plans, medidas e directrices de seguridade informática para o conxunto de órganos e unidades da Xunta.
- . Supervisión do cumprimento de todas as medidas de seguridade informática nos distintos ámbitos e departamentos da Xunta.
- . Deseño e realización de accións encamiñadas a garantir o cumprimento da normativa vixente en materia de protección de datos de carácter persoal na Administración da Comunidade Autónoma de Galicia.

##### 2.- Centro de Seguridade da información (CSI)



O Centro de Seguridade da Información ten unha función transversal dentro da Xunta. Entre as súas atribucións, atópase a resolución de incidencias de seguridade e o asesoramento ás distintas consellerías en materia de seguridade da información, tanto en materia de protección de datos coma na avaliación de solucións de seguridade.

O CSI supervisará a seguridade dos sistemas corporativos da Administración da Comunidade Autónoma de Galicia e será o centro de resolución de incidentes de seguridade (ataques activos e pasivos, perda de información, falta de dispoñibilidade, etc.).

Dentro dos servizos de apoio e asesoramento ás distintas consellerías para que adecúen os seus sistemas de información ás esixencias da lexislación vixente en materia de protección de datos, o Centro de Seguridade da Información será o interlocutor coa Axencia Española de Protección de Datos.

O Centro de Seguridade da Información encárgase de avaliar, analizar e probar nun contorno controlado as distintas solucións de seguridade existentes no mercado e que sexan de aplicación e interese para a Administración da Comunidade Autónoma de Galicia. Os resultados destas avaliacións serán recollidos nunha serie de informes de análises e de estudos comparativos. Así mesmo, poderán realizarse demostracións para as consellerías, xa sexa a través do contorno de probas ou mediante charlas divulgativas. Estas avaliacións poderán efectuarse en función das necesidades detectadas polo propio Centro de Seguridade da Información ou baseándose na valoración de peticións dos distintos departamentos, e poderán referirse á seguridade perimetral, de centros de proceso de datos, de aplicacións, de posto informático, etc.



Co fin de promover o desenvolvemento do presente plan nas diferentes consellerías, inclúense dentro das funcións do CSI, as seguintes tarefas:

- . Realizar o seguimento e revisar o presente Plan Director de forma periódica.
- . Apoiar e asesorar as consellerías para alcanzar os obxectivos fixados no presente plan.
- . Impulsar e apoiar o Comité de Seguridade dos Sistemas de Información da Xunta.
- . Velar pola mellora do nivel de seguridade da información da Administración da Comunidade Autónoma de Galicia.

Para o desenvolvemento destas funcións o Centro de Seguridade da Información ten amplos coñecementos no campo da seguridade da información e poderá contar co asesoramento de expertos externos.

### 3.- Consellerías da Xunta

As Consellerías son as máximas responsables do estado de seguridade dos seus sistemas de información. Deberán velar pola súa seguridade garantindo a confidencialidade, integridade e dispoñibilidade da información. Son tamén as responsables de acometer as actuacións recomendadas no presente Plan Director e de acadar os obxectivos marcados.

Dentro de cada consellería, o persoal de soporte técnico desempeñará funcións de asesoría técnica, execución, proposta, coordinación e supervisión dos plans de informatización.



Así mesmo, os órganos responsables dos sistemas de información de cada departamento da Xunta de Galicia, co apoio do persoal de soporte técnico, serán os competentes para velar polo cumprimento dos obxectivos deste plan. Correspóndelles a eles asegurarse de que os equipos se utilizan adecuadamente e atendendo á finalidade á que están destinados.

#### 4.- Comité de Seguridade de Sistemas de Información (CSSI).

O Comité de Seguridade dos Sistemas de Información ten como obxectivo establecer o marco de traballo que impulse a implantación e difusión da xestión da seguridade da información no ámbito da Administración da Comunidade Autónoma de Galicia.

Os membros do Comité reúnense para revisar o estado da seguridade da información na Administración da Comunidade Autónoma de Galicia, aprobar as políticas de seguridade, revisar e aprobar os proxectos de seguridade, revisar os procesos de monitorización das incidencias de seguridade e realizar outras tarefas de xestión da seguridade de alto nivel que sexan necesarias.

As principais funcións asumidas por este Comité son as que se presentan deseguido:

- . Identificar, revisar e propoñer obxectivos estratéxicos en materia de seguridade da información.
- . Establecer roles e responsabilidades en materia de seguridade da información.
- . Propoñer e aprobar políticas, normas e directrices de seguridade da información para a Xunta de Galicia e velar polo seu cumprimento.
- . Proporcionar un soporte ao esforzo de seguridade da información, dando unha visión máis transversal para a análise e toma de decisións a fin de lograr a mellor relación custo-efectividade na súa xestión.



- . Constituír a canle primaria de discusión sobre os aspectos da seguridade da información que se deban abordar na Administración da Comunidade Autónoma de Galicia.
- . Apoiar os coordinadores de seguridade no desenvolvemento de estratexias de mitigación de riscos baseándose no coñecemento que os seus integrantes teñen das súas respectivas áreas.
- . Impulsar a implantación e difusión da xestión da seguridade da información.
- . Revisar e aprobar anualmente a Política de Seguridade.
- . Realizar outras tarefas de xestión da seguridade de alto nivel que sexan necesarias.
- . Revisar periodicamente o presente Plan Director.

Para o desenvolvemento destas funcións, o CSSI poderá contar co apoio, en materia de actualización normativa, da Asesoría Xurídica Xeral. Puntualmente, poderá solicitar a colaboración da Xunta Consultiva de Contratación Administrativa da Xunta de Galicia e do equipo de auditores e analistas de xestión do rendemento e calidade forma parte da Dirección Xeral de Avaliación e Reforma Administrativa da Consellería de Presidencia, Administracións Públicas e Xustiza.

## 5.- Asesoría Xurídica Xeral

O papel do dereito nas tecnoloxías da información é amplo e abrangue varios campos, como poden ser os que enunciámos a continuación:

- . Protección de datos de carácter persoal.
- . Propiedade Intelectual.
- . Servizos da sociedade da información.
- . Sinatura electrónica.
- . Constitución de proba por medios informáticos.
- . Uso de ferramentas informáticas polo persoal.



A Asesoría Xurídica Xeral facilitará ao Comité de Seguridade dos Sistemas de Información o seu coñecemento no ámbito legal como apoio á actualización normativa mediante a emisión de ditames ou informes en dereito, a formulación de criterios xerais de asesoramento xurídico e o estudo dos proxectos de regulamentos con exame da súa adecuación ao ordenamento constitucional, estatutario e legal.

Polo seu coñecemento en materias xurídicas vinculadas ao dereito das tecnoloxías da información, velará por que as iniciativas do Comité de Seguridade dos Sistemas de Información e da Secretaría Xeral competente sexan acordes á lexislación vixente aplicable. Por outra banda, a Asesoría Xurídica de cada Consellería ha prestarlle un servizo de apoio nas materias anteriormente mencionadas.

## 6.- Xunta Consultiva de Contratación Administrativa

A Xunta Consultiva de Contratación Administrativa da Comunidade Autónoma de Galicia está adscrita á Consellería responsable dos asuntos tributarios como un órgano consultivo específico en materia de contratación administrativa.

Dentro das funcións e competencias da Xunta , atópanse:

∴ Elaborar e propoñer as normas, instrucións e medidas que considere precisas para a mellora e eficacia da contratación da Administración autonómica, os seus organismos e sociedades, fundacións do sector público e demais entidades de dereito público dela dependentes.

. Realizar estudos e investigacións sobre contratación administrativa, trasladándolles aos órganos de contratación as recomendacións que daquela se deriven.



A Xunta de Galicia recorre a provedores para a prestación de servizos como poden ser o desenvolvemento de novas aplicacións, infraestruturas, externalización dunha actividade.

Para garantir a seguridade da información é importante xestionar a actuación dos provedores. Por iso cómpre identificar os requisitos de seguridade vinculados á prestación do servizo e incluílos nos contratos.

Xa que logo, a Xunta Consultiva de Contratación Administrativa trasladará aos órganos de contratación todas as recomendacións que considere oportunas co propósito de que se inclúan cláusulas sobre dispoñibilidade, confidencialidade, integridade e autenticidade da información manexada polos sistemas de información.

Coa finalidade de asegurar este aspecto, exercerá unha función de análise e proposta de inclusión nos pregos de contratación de cláusulas ambientais, sociais, de comercio xusto e de protección de datos que os adxudicatarios deberán cumprir en función do servizo contratado.

## 7. Equipo de auditores e analistas de xestión do rendemento e calidade

O equipo de auditores e analistas de xestión do rendemento e calidade forma parte da Dirección Xeral de Avaliación e Reforma Administrativa da Consellería de Presidencia, Administracións Públicas e Xustiza.

Esta Dirección ten, entre outras, as seguintes competencias:

. En materia de avaliación do rendemento e xestión da calidade:

1.-O desenvolvemento e a xestión das medidas para a implantación de sistemas de mellora da calidade tendendo a promover a mellora continua dos servizos da Administración autonómica, tanto dos que se lle prestan directamente ao cidadán como os servizos internos.



. En materia de racionalización e simplificación de procedementos administrativos:

1.-Coordinar a aplicación da normativa europea e estatal sobre simplificación e mellora da xestión administrativa.

. En materia de información administrativa e atención ao cidadán:

1.-Avaliar periodicamente a calidade do sistema de información administrativa, propoñendo as medidas de melloras convenientes co fin de facilitarlles aos cidadáns e usuarios os servizos que solicitan.

2.- Tramitar, sen prexuízo das competencias que lles corresponden ás secretarías xerais e en colaboración con estas, as queixas e as propostas que formulen os cidadáns e usuarios sobre o funcionamento dos servizos prestados pola Administración autonómica, de acordo co establecido nos artigos 25 e seguintes do Decreto 164/2005, do 16 de xuño.

Entre os principios de protección de datos atópanse o principio de información e o principio de consentimento. O principio de información, que se regula no artigo 5 da LOPD, establece que o interesado debe estar informado do momento da recollida de datos. En canto ao principio de consentimento, esixe que se obteña o consentimento do afectado para o tratamento dos datos, salvo excepcións recollidas na LOPD. Estes dereitos fundamentais deben garantirse non só para cumprir coa lexislación vixente senón tamén para garantir a calidade do servizo prestado ao cidadán.

Os procedementos administrativos son uns dos puntos aos que é necesario prestar atención tendo en conta que recollen, a miúdo, datos de carácter persoal.

Cando a recollida de datos se realiza mediante formulario, ben sexa en formato papel, ben en formato electrónico, débese introducir unha cláusula informativa no formulario que atenda ás esixencias do artigo 5 da Lei



orgánica de protección datos. Tocante ao consentimento, recollerase mediante a sinatura do formulario en papel e a aprobación do formulario electrónico.

O equipo revisa os procedementos administrativos antes da súa publicación no DOG.

## 17.2 DECRETO 230/2008 DO 18 DE SETEMBRO POLO QUE SE ESTALBECEN BOAS PRÁCTICAS

### 17.2.1 Obxecto e ámbito de aplicación

Este decreto ten por obxecto regular as normas de utilización dos sistemas de información e de comunicacións, fixos e móbiles, dos que dispón a Administración da Comunidade Autónoma de Galicia, establecendo os dereitos e os deberes das persoas usuarias destes sistemas no relativo á súa seguridade e bo uso.

A finalidade da presente norma é conseguir o mellor aproveitamento das tecnoloxías da información e as comunicacións na actividade administrativa, así como garantir a protección da información das persoas e das empresas nas súas relacións coa Administración da Comunidade Autónoma de Galicia.

Será de aplicación a todas as persoas que presten servizos para a Administración da Comunidade Autónoma de Galicia e utilicen para o desempeño das súas funcións os sistemas de información ou as redes de comunicacións propiedade da Administración autonómica.

O contido deste Decreto será de aplicación na utilización do equipamento informático e de comunicacións, fixo e móbil, incluíndo calquera dispositivo



posto á disposición das persoas que prestan servizos para a Administración autonómica.

### 17.2.2 Órganos responsables e de coordinación

#### 1.- Órganos responsables

As secretarías xerais designarán, dentro de cada departamento da Xunta de Galicia, o órgano que será responsable dos sistemas da súa propiedade e de establecer os medios tecnolóxicos que precisan as persoas ao seu servizo, así como de velar polo correcto funcionamento das infraestruturas e do equipamento informático e de comunicacións de que dispoñan. Naqueles casos en que as devanditas atribucións xa lle estean asignadas regulamentariamente a un órgano, non será precisa esta designación.

Estes órganos, co apoio do persoal de soporte técnico, son os competentes para velar polo cumprimento das normas contidas neste decreto. Corresponderalles a eles asegurarse de que os equipos se utilizan adecuadamente e atendendo á finalidade para a que están destinados.

Para o mellor cumprimento destas atribucións sobre os sistemas, cada departamento da Xunta de Galicia deberá designar unha persoa como responsable de seguridade. As persoas designadas deberán comunicar, dentro do seu ámbito, as normas, procedementos e políticas de seguridade para o seu coñecemento polo persoal, así como impulsar a súa implantación.

#### 2.- Órganos de coordinación

Son órganos de coordinación:



a) A Comisión de Informática da Xunta de Galicia, regulada polo Decreto 290/1992, do 8 de outubro .

b) A Dirección Xeral de Calidade e Avaliación das Políticas Públicas da Consellería de Presidencia, Administracións Públicas e Xustiza.

c) O Comité de Seguridade dos Sistemas de Información da Xunta de Galicia. É un órgano colexiado, adscrito á Consellería de Presidencia, Administracións Públicas e Xustiza, formado polas persoas responsables de seguridade dos distintos departamentos da Xunta de Galicia, que ten como obxectivo definir a política de seguridade corporativa. Este comité estará coordinado e asesorado pola dirección xeral competente a través do Centro de Seguridade Informática. O seu réxime básico de funcionamento regúlase por orde da Consellería de Presidencia, Administracións Públicas e Xustiza.

### 17.2.3 Acceso á información, redes de comunicacións e Internet.

#### 1.- Acceso á información

As persoas usuarias terán autorizado o acceso unicamente a aquela información e recursos que precisen para o desenvolvemento das súas funcións. O acceso á información contida nos sistemas da Administración da Comunidade Autónoma de Galicia estará restrinxido a aquelas persoas posuidoras da correspondente autorización, que será persoal e intransferible e composta polo menos dun identificador e dun contrasinal.

Os órganos responsables dos sistemas establecerán os mecanismos adecuados para evitar que as persoas poidan acceder a datos ou modificalos sen autorización. O persoal de soporte técnico exclusivamente, conforme aos criterios establecidos polo responsable de cada un dos sistemas de información, poderá conceder, alterar ou anular a autorización de acceso aos datos e recursos.



Non se poderán obter dereitos de acceso á información distintos aos autorizados, nin se utilizará o identificador doutra persoa, aínda que se dispoña de permiso desta, salvo indicación expresa e puntual do órgano responsable da devandita información ou recurso. Con este fin, as unidades de persoal dos distintos departamentos da Xunta de Galicia comunicarán ao servizo de informática todos os cambios que se produzan nos postos de traballo.

As persoas ao servizo da Administración da Comunidade Autónoma de Galicia deberán velar pola seguridade dos datos aos que teñan acceso debido ás tarefas do seu posto de traballo, especialmente os confidenciais ou de carácter persoal.

Por motivos de seguridade, a Administración da Comunidade Autónoma de Galicia poderá monitorar os accesos á información contida nos seus sistemas, cumprindo os requisitos que para este fin estableza a normativa vixente.

## 2 Redes de comunicacións

. A conexión á rede corporativa da Xunta de Galicia será facilitada pola Dirección Xeral de Calidade e Avaliación das Políticas Públicas en uso das competencias atribuídas no decreto de estrutura orgánica da Consellería de Presidencia, Administracións Públicas e Xustiza.

Non se poderá conectar a esta rede de comunicacións ningún dispositivo por medios distintos aos definidos e autorizados polo Centro de Xestión de Rede da devandita dirección xeral.

No caso daquelas redes de comunicacións da Administración da Comunidade Autónoma de Galicia xa xestionadas por outras consellerías, a



conexión ás mesmas será facilitada polo órgano responsable de cada unha delas.

### 3 Internet

A Administración da Comunidade Autónoma de Galicia proverá de conexión a Internet ás persoas ao seu servizo cunha finalidade exclusivamente profesional.

O equipo que teña acceso a Internet, a través das redes de comunicación xestionadas pola Administración da Comunidade Autónoma de Galicia, deberá dispoñer de «software» de protección fronte a virus e demais códigos maliciosos.

Os datos de conexión e tráfico serán monitorados e gardarase un rexistro durante o tempo que establece a normativa vixente en cada suposto. En ningún caso esta retención de datos afectará ao segredo das comunicacións.

As conexións a sitios Web que conteñan material ofensivo ou software malicioso serán bloqueadas, agás excepcións debidamente autorizadas.

#### 17.2.4 Servizo de mensaxería corporativo

A Administración da Comunidade Autónoma de Galicia proverá de servizo de mensaxería ás persoas ao seu servizo cunha finalidade exclusivamente profesional.

Por razóns de seguridade e rendemento, os órganos responsables do servizo poderán monitorar o servizo de mensaxería corporativa. Esta monitorización non será nunca selectiva ou discriminatoria senón que será



realizada de forma sistemática ou aleatoria e sen vulneración da intimidade persoal nin do segredo das comunicacións.

Aquelas contas nas que se detecte un uso inadecuado, que se definirá no documento de política de seguridade corporativa, poderán ser bloqueadas ou suspendidas temporalmente. En ningún caso, poderá utilizarse o servizo de mensaxería para:

- a) A difusión de mensaxes ofensivas ou discriminatorias.
- b) O uso da conta de correo corporativo para expresar opinións persoais en foros temáticos fóra do ámbito das administracións.
- c) A difusión masiva non autorizada; subscrición indiscriminada a listas de correo ou calquera ataque co obxecto de impedir ou dificultar o servizo de correo.

#### 17.2.5 Deber das persoas usuarias

As persoas que prestan servizos á Administración da Comunidade Autónoma de Galicia, ademais de cumpriren coas medidas indicadas neste decreto relativas ao equipamento informático e de comunicacións, ás aplicacións informáticas, á información e ao uso dos servizos corporativos, son responsables do bo uso dos medios electrónicos, informáticos, telemáticos e de comunicacións, fixos e móbiles, postos á súa disposición para as actividades propias das funcións que desenvolven.

Non se poderá acceder aos recursos informáticos e telemáticos para desenvolver actividades que persigan ou teñan como consecuencia:

- a) A degradación dos servizos.



b) A destrución ou modificación non autorizada da información de maneira premeditada.

c) A violación da intimidade, do segredo das comunicacións e do dereito á protección de datos persoais.

d) A deterioración intencionada do traballo doutras persoas.

e) O uso dos sistemas de información para fins alleos aos da Administración.

f) Incorrer en actividades ilícitas de calquera tipo.

g) Danar intencionadamente os recursos informáticos da Administración da Comunidade Autónoma de Galicia ou doutras institucións.

h) Instalar ou utilizar «software» que non dispoña da licenza correspondente.

3. Para garantir uns mínimos de seguridade no equipamento asignado, deberase:

a) Utilizar e gardar en segredo o contrasinal que protexe a conta de acceso, responsabilidade directa da persoa usuaria. Esta debe pechar a súa conta ao final de cada sesión ou cando deixa desatendido o equipo, co fin de que non poida ser utilizado por terceiras persoas.

b) Revisar de forma periódica os seus ordenadores, eliminando calquera virus, programa ou ficheiro que lles poida causar danos a outros equipos da rede ou impedindo outras actuacións que contraveñan a lexislación vixente.



c) No caso de que o seu equipo conteña información importante que non estea gardada nun servidor, realizar copias de seguridade periódicas para garantir a súa dispoñibilidade.

As persoas usuarias, no exercicio das súas funcións, deberán colaborar co órgano competente en materia de seguridade dos sistemas de información e seguir as súas recomendacións e, en particular, as do Centro de Seguridade Informática, en aplicación da política de seguridade corporativa definida polo Comité de Seguridade dos Sistemas de Información da Xunta de Galicia.

Tamén estarán obrigadas ao cumprimento daquelas outras medidas adicionais que especifiquen os órganos responsables dos sistemas.

#### 17.2.6 Inspección

A Administración da Comunidade Autónoma de Galicia, mediante os medios tecnolóxicos e persoais que estime oportunos, revisará periódica e puntualmente, por razóns de seguridade e de calidade do servizo, o estado dos equipos, dispositivos e redes de comunicacións da súa responsabilidade, así como a súa correcta utilización, co obxecto de verificar o seu correcto funcionamento, eficiencia e o cumprimento das medidas e protocolos de seguridade establecidos na lexislación vixente.

A dirección xeral competente en materia de seguridade corporativa velará polo cumprimento da presente normativa e informará o Comité de Seguridade dos Sistemas de Información da Xunta de Galicia sobre os incumprimentos ou deficiencias de seguridade observados co obxecto de que se tomen as medidas oportunas.

Os servizos para os que se detecte un uso inadecuado, ou que non cumpran os requisitos de seguridade, que se definirán no documento de



política de seguridade corporativa, poderán ser bloqueados ou suspendidos temporalmente para aquelas contas nas que se detecte un dano para os dos sistemas de información e de comunicacións. O servizo ha restablecerse cando a causa da súa degradación desapareza.

#### 17.2.7 Responsabilidade das persoas usuarias que teñan a condición de empregados públicos

A Administración da Comunidade Autónoma de Galicia esixirá dos empregados públicos a responsabilidade na que incorresen por dolo, culpa ou descoído graves dos que se deriven danos e prexuízos nos seus bens ou dereitos ou indemnizacións para particulares, tras instrución do procedemento correspondente nos termos previstos na normativa de aplicación.

O incumprimento dos deberes e obrigas impostos polo presente decreto, que sexan constitutivos de infracción disciplinaria, segundo a tipificación efectuada na normativa aplicable, dará lugar á incoación do correspondente procedemento disciplinario, que se tramitará conforme ao establecido na normativa aplicable aos empregados públicos en función da natureza xurídica do seu vínculo coa Administración. Non obstante o anterior, a incoación dos expedientes e a imposición das sancións requirirá informe previo da Dirección Xeral de Calidade e Avaliación das Políticas Públicas.

Autor:

Alfonso García Magariños

Director Asesoría Xurídica Municipal Concello da Coruña



**18. IMPLANTACIÓN DA  
ADMINISTRACIÓN ELECTRÓNICA.  
SEDE ELECTRÓNICA E SERVIZOS  
DE SEDE. REXISTRO  
ELECTRÓNICO. EXPEDIENTE  
ELECTRÓNICO. ARQUIVO  
ELECTRÓNICO DE DOCUMENTOS.  
DIXITALIZACIÓN, COMPULSA  
ELECTRÓNICA. FACTURA E  
LICITACIÓN ELECTRÓNICAS.  
INICIATIVAS DO GOBERNO  
GALEGO: OSIMGA, REDE CEMIT,  
PLAN DE BANDA LARGA 2013,  
AXENDA DIXITAL 2014.GAL,  
ESTRATEGIA DE IMPULSO AO  
SECTOR TIC.**



**TEMA 18. IMPLANTACIÓN DA ADMINISTRACIÓN ELECTRÓNICA. SEDE ELECTRÓNICA E SERVIZOS DE SEDE. REXISTRO ELECTRÓNICO. EXPEDIENTE ELECTRÓNICO. ARQUIVO ELECTRÓNICO DE DOCUMENTOS. DIXITALIZACIÓN, COMPULSA ELECTRÓNICA. FACTURA E 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 DA ADMINISTRACIÓN ELECTRÓNICA**

**18.2 SEDE ELECTRÓNICA E SERVIZOS DA SEDE**

**18.3 REXISTRO ELECTRÓNICO**

**18.4 EXPEDIENTE ELECTRÓNICO**

**18.5 ARQUIVO ELECTRÓNICO DE DOCUMENTOS**

**18.6 DIXITALIZACIÓN, COMPULSA ELECTRÓNICA**

**18.7 FACTURA E 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 DA ADMINISTRACIÓN ELECTRÓNICA**

Segundo establece a Comisión Europea, a Administración electrónica defínese como o uso das Tecnoloxías da Información e as Comunicacions nas administracións públicas, combinada con cambios de organización e novas aptitudes, co fin de mellorar os servizos públicos e os procesos democráticos e reforzar o apoio ás políticas públicas.



A e-Administración ou Administración electrónica fai referencia á incorporación das tecnoloxías da información e as comunicacións en dúas vertentes:

- Desde un punto de vista organizativo, transformando as oficinas tradicionais, convertendo os procesos en papel en procesos electrónicos.
- Desde unha perspectiva das relacións externas, habilitando a vía electrónica como un novo medio para a relación co cidadán, empresas e outras institucións.

A Administración electrónica ten o seu maior impulso na última década, motivado en parte por un marco legal que permitiu levar as garantías xurídicas que existen no mundo real ao mundo virtual, e tamén pola evolución das tecnoloxías relacionadas e o desenvolvemento de proxectos emblemáticos, como o DNI electrónico.

Pero sobre todo é necesario facer mención da Lei 11/2007, do 22 de xuño, de acceso electrónico dos cidadáns aos servizos públicos, moitas veces denominada simplemente "Lei de Administración electrónica" ou mencionada polas siglas LAECSPC<sup>1</sup>, que consagra o concepto de Administración electrónica no marco xurídico español e a eleva á categoría de dereito dos cidadáns. É, xa que logo, a norma legal de referencia nesta materia e establece un marco homoxéneo para as tres administracións na materia.

O seu principal obxectivo é recoñecer e garantir o dereito do cidadán a se relacionar por medios electrónicos coas administracións públicas. Por outra banda, preténdese impulsar o uso dos servizos electrónicos na Administración creando as condicións necesarias e, de xeito indirecto, exercer con iso un efecto arrastre sobre a sociedade da información en xeral.

---

<sup>1</sup> Dadas as múltiples referencias que se fan neste texto á devandita lei, optaremos habitualmente por utilizar esta expresión ou ben simplemente Lei 11/2007.



A LAECSPC, ademais, supuxo o punto de partida para un desenvolvemento normativo que permite avanzar en aspectos concretos, e que tamén se mencionarán nestes capítulos e aparecen no apartado de referencias.

En particular, e con respecto aos aspectos técnicos, destacan o Esquema Nacional de Seguridade e o Esquema Nacional de Interoperabilidade, e con respecto a este último, as recentes Normas Técnicas de Interoperabilidade.

Ademais de recoñecer o dereito dos cidadáns a se relacionar coas administracións públicas por medios electrónicos, regula os aspectos básicos da utilización das tecnoloxías da información na actividade administrativa, nas relacións entre as administracións públicas, así como nas relacións dos cidadáns con estas.

A LAECSPC obriga as administracións a asegurar a dispoñibilidade, o acceso, a integridade, a autenticidade, a confidencialidade e a conservación dos datos, informacións e servizos que xestionen no exercicio das súas competencias.

Tamén establece unha serie de fins, que definen con claridade qué debe perseguir todo proxecto de Administración electrónica:

1. Facilitar o exercicio de dereitos e o cumprimento de deberes por medios electrónicos.
2. Facilitar o acceso por medios electrónicos dos cidadáns á información e ao procedemento administrativo, con especial atención á eliminación das barreiras que limiten este acceso.
3. Crear as condicións de confianza no uso dos medios electrónicos.
4. Promover a proximidade co cidadán e a transparencia administrativa, así como a mellora continua na consecución do interese xeral.



5. Contribuír á mellora do funcionamento interno das administracións públicas.
6. Simplificar os procedementos administrativos e proporcionar oportunidades de participación e maior transparencia, coas debidas garantías legais.
7. Contribuír ao desenvolvemento da sociedade da información no ámbito das administracións públicas e na sociedade en xeral.

Tamén establece uns principios: o respecto ao dereito á protección de datos de carácter persoal, de igualdade, de accesibilidade á información e aos servizos, de legalidade, de cooperación, de seguridade, de proporcionalidade, de responsabilidade, de neutralidade tecnolóxica e de adaptabilidade, de simplificación administrativa e de transparencia e publicidade.

Establece unha serie de dereitos para os cidadáns:

- a. A elixir, entre aquelas que en cada momento se encontren dispoñibles, a canle a través da cal relacionarse por medios electrónicos coas administracións públicas.
- b. A non achegar os datos e documentos que obren en poder das administracións públicas.
- c. Á igualdade no acceso electrónico aos servizos das administracións públicas (entendida como non discriminación de persoas que non teña un acceso fácil aos medios electrónicos).
- d. A coñecer por medios electrónicos o estado de tramitación dos procedementos en que sexan interesados.
- e. A obter copias electrónicas dos documentos electrónicos que formen parte de procedementos en que teñan a condición de interesado.
- f. Á conservación en formato electrónico polas administracións públicas dos documentos electrónicos que formen parte dun expediente.



- g. A obter os medios de identificación electrónica necesarios, podendo as persoas físicas utilizar en todo caso os sistemas de sinatura electrónica do documento nacional de identidade para calquera trámite electrónico con calquera Administración pública.
- h. Á utilización doutros sistemas de sinatura electrónica admitidos no ámbito das administracións públicas.
- i. Á garantía da seguridade e confidencialidade dos datos que figuren nos ficheiros, sistemas e aplicacións das administracións públicas.
- j. Á calidade dos servizos públicos prestados por medios electrónicos.
- k. A elixir as aplicacións ou sistemas para relacionarse coas administracións públicas sempre e cando utilicen estándares abertos ou, de ser o caso, aqueloutros que sexan de uso xeneralizado polos cidadáns.

En particular, nos procedementos relativos ao acceso a unha actividade de servizos e o seu exercicio, os cidadáns teñen dereito á realización da tramitación a través dun portelo único, por vía electrónica e a distancia, e á obtención da seguinte información a través de medios electrónicos, que deberá ser clara e inequívoca:

- a. Os requisitos aplicables aos prestadores establecidos en territorio español, en especial os relativos aos procedementos e trámites necesarios para acceder ás actividades de servizo e para o seu exercicio.
- b. Os datos das autoridades competentes nas materias relacionadas coas actividades de servizos, así como os datos das asociacións e organizacións distintas das autoridades competentes ás que os prestadores ou destinatarios se poidan dirixir para obteren asistencia ou axuda.
- c. Os medios e condicións de acceso aos rexistros e bases de datos públicos relativos a prestadores de actividades de servizos.



- d. As vías de reclamación e recurso en caso de litixio entre as autoridades competentes e o prestador ou o destinatario, ou entre un prestador e un destinatario, ou entre prestadores.

Co fin de que os cidadáns poidan exercer o seu dereito a non achegar datos que xa obren en poder da Administración pública, cada Administración deberá facilitar o acceso das restantes administracións públicas aos datos relativos aos interesados que obren no seu poder e estean en soporte electrónico.

## **18.2 SEDE ELECTRÓNICA E SERVIZOS DA SEDE**

A LAECSPC define a sede electrónica como aquel enderezo electrónico dispoñible para os cidadáns a través de redes de telecomunicacións cuxa titularidade, xestión e administración lle corresponde a unha Administración pública, órgano ou entidade administrativa no exercicio das súas competencias.

Cada Administración pública determinará as condicións e instrumentos de creación das sedes electrónicas, con suxeición aos principios de publicidade oficial, responsabilidade, calidade, seguridade, dispoñibilidade, accesibilidade, neutralidade e interoperabilidade. En todo caso deberase garantir a identificación do titular da sede, así como os medios dispoñibles para a formulación de suxestións e queixas.

As sedes electrónicas dispoñerán de sistemas que permitan o establecemento de comunicacións seguras sempre que sexan necesarias. Este apartado cobra especial importancia cando se ofrecen servizos de tramitación.

A publicación nas sedes electrónicas de informacións, servizos e transaccións respectará os principios de accesibilidade e usabilidade de



acordo coas normas establecidas respecto diso, estándares abertos e, se é o caso, aqueloutros que sexan de uso xeneralizado polos cidadáns.

A sede electrónica diferénciase de calquera sede institucional tradicional en contorno telemático pola responsabilidade do titular, o feito de que a publicación de diarios ou boletíns oficiais reviste os mesmos efectos que a publicación impresa e que pode actuar como substituta ou complemento do taboleiro de anuncios ou edictos. Ademais, cómpre ter en conta as condicións que deben cumprir as sedes electrónicas da Administración xeral do Estado no marco do Real decreto 1671/2009. Para o resto das administracións públicas, inda que están fóra do seu ámbito, é indubidable a súa validez como elemento de referencia.

En todo caso, a través dos seus artigos, a LAECSP establece os seguintes requisitos para as sedes electrónicas:

- Debe permitir o acceso dos cidadáns para a realización de calquera tipo de trámite ou interacción coa Administración.
- Debe permitirlles aos cidadáns realizar consultas sobre o estado de tramitación de expedientes en que teñan a condición de interesado.
- Tanto a sede como os elementos e contidos dela débense basear en aplicacións e sistemas que utilicen estándares abertos ou sexan de uso xeneralizado polos cidadáns.
- Debe conter a información sobre os pasos que cómpre seguir para cada un dos trámites e procedementos das administracións públicas.
- Debe conter a información sobre as autoridades competentes para cada actividade dos servizos ofrecidos polas administracións públicas.
- A AXE deberá dispoñer dunha sede electrónica que sirva como punto de acceso xeral único aos servizos que presta a AXE e os seus organismos.



- O punto de acceso xeral creado pola AXE deberá estar integrado co resto de sedes da AXE e organismos públicos para a prestación dos distintos servizos.
- Debe garantir a identificación do seu titular.
- Debe permitir establecer as conexións seguras cando sexan necesarias.
- Debe cumprir os principios de accesibilidade e usabilidade de acordo coas normas establecidas respecto diso (segundo o BOE n.º 141 do 13/6/2003 na disposición 7, artigo 2: débense cumprir os requisitos AA).
- Permitirá a publicación de actos e comunicacións que, por disposición legal ou regulamentaria, se deban publicar do taboleiro de anuncios ou edictos.
- Debe conter a lista de sistemas de sinatura electrónica avanzada admitidos.
- Debe conter a lista de selos electrónicos utilizados por cada Administración.
- Debe conter as disposicións de creación de rexistros electrónicos.
- A sede permitirá a publicación electrónica do boletín oficial da Administración, órgano ou entidade competente.
- Debe conter os distintos tipos de escritos, comunicacións, solicitudes, etc. que se poden presentar.
- Deberá publicar os medios electrónicos dispoñibles para que o cidadán se relacione coas administracións públicas.
- Deberá mostrar de xeito visible a data e hora garantindo a súa integridade.
- Deberá publicar unha lista cos días considerados inhábiles.
- Naquelas administracións que teñan linguas cooficiais, débese garantir o acceso en ambas as linguas.



Desde o punto de vista técnico, a sede electrónica non presenta características tecnolóxicas distintas ás de calquera sitio web tradicional, inda que é necesario establecer as medidas de seguridade que permitan garantir a responsabilidade establecida pola LAECSP.

En particular, é importante a identificación segura da sede electrónica. Neste sentido, a lei establece a posibilidade de creación de certificados de sede electrónica con este propósito.

A Fábrica Nacional da Moeda e Timbre – Real Casa da Moeda (FNMT-RCM), que xa ofrecía certificados de identificación segura, dispón xa da versión actualizada para o cumprimento da LAECSP (son os chamados xenericamente certificados APE).

### **18.3 REXISTRO ELECTRÓNICO**

As tarefas fundamentais do rexistro electrónico son tomar unha referencia de tempo, anotar o asento da entrada/saída, gardar os datos da presentación de información, e devolver un acuse de recibo co número de rexistro e momento da presentación. O rexistro poderá, así mesmo, incluír funcionalidades adicionais, por exemplo, o selado de tempo para obter a referencia temporal, o cotexo/compulsa electrónica de documentos presentados fisicamente ou o funcionamento como rexistro único para toda a Administración.

Con respecto aos rexistros electrónicos, a LAECSP establece que as administracións públicas crearán rexistros electrónicos para a recepción e remisión de solicitudes, escritos e comunicacións. Os rexistros electrónicos poderán admitir:

- a. Documentos electrónicos normalizados correspondentes aos servizos, procedementos e trámites que se especifiquen conforme o disposto



na norma de creación do rexistro, cubertos de acordo con formatos preestablecidos.

- b. Calquera solicitude, escrito ou comunicación distintos dos mencionados no apartado anterior dirixidos a calquera órgano ou entidade do ámbito da Administración titular do rexistro.

En cada Administración pública existirá, polo menos, un sistema de rexistros electrónicos abondo para recibir todo tipo de solicitudes, escritos e comunicacións dirixidos a esa Administración pública.

Os rexistros electrónicos emitirán automaticamente un recibo consistente nunha copia autenticada do escrito, solicitude ou comunicación de que se trate, incluíndo a data e hora de presentación e o número de entrada de rexistro.

Poderanse achegar documentos que acompañen a correspondente solicitude, escrito ou comunicación, sempre que cumpran os estándares de formato e requisitos de seguridade que se determinen nos Esquemas Nacionais de Interoperabilidade e de Seguridade.

Os rexistros electrónicos rexeranse para os efectos de cómputo dos prazos imputables tanto aos interesados como ás administracións públicas pola data e hora oficial da sede electrónica de acceso, que deberá contar coas medidas de seguridade necesarias para garantir a súa integridade e figurar visible. Permitirán a presentación de solicitudes, escritos e comunicacións todos os días do ano durante as vinte e catro horas.

Para os efectos do cómputo de prazo fixado en días hábiles ou naturais, e no que se refire ao cumprimento de prazos polos interesados, a presentación nun día inhábil entenderase realizada na primeira hora do primeiro día hábil seguinte, salvo que unha norma permita expresamente a recepción en día inhábil.



Desde o punto de vista técnico, o rexistro electrónico debe permitir a presentación de documentación en formato electrónico e a xeración do correspondente asento no rexistro de entrada e saída da Administración. Iso implica probablemente a necesidade de integración cun sistema de xestión de rexistro de entrada e saída xeral, que permita a introdución de asentos manual desde os distintos departamentos.

Ademais, para poder realizar esta función coas suficientes garantías, debe permitir o seguinte:

- a) Presentar a documentación en formato electrónico e asinada polo cidadán mediante o correspondente certificado electrónico que garanta a súa autenticidade, integridade e non repudio.
- b) Rexistrar de modo fidedigno o proceso de rexistro da devandita documentación. Para ese efecto pódese utilizar algún tipo de resgardo asinado mediante o uso dun certificado da Administración pública (previsiblemente un certificado de selo electrónico) que inclúa selado de tempo para garantir a data e hora de entrega.
- c) Entregarlle ao cidadán o resgardo do antedito rexistro en formato electrónico, convenientemente asinado pola Administración, onde constará a data e hora de entrada e os documentos entregados, e podendo incorporar un *checksum* ou sistema de equivalente que permita verificar que os contidos presentados son os que constan no antedito resgardo.

#### **18.4 EXPEDIENTE ELECTRÓNICO**

A LAECSP define expediente electrónico como o conxunto de documentos electrónicos correspondentes a un procedemento administrativo, calquera que sexa o tipo de información que conteñan.



O foliado dos expedientes electrónicos levarase a cabo mediante un índice electrónico, asinado pola Administración, órgano ou entidade actuante, segundo proceda. Este índice garantirá a integridade do expediente electrónico e permitirá a súa recuperación sempre que sexa preciso, sendo admisible que un mesmo documento forme parte de distintos expedientes electrónicos.

A remisión de expedientes poderá ser substituída para todos os efectos legais pola posta a disposición do expediente electrónico, tendo o interesado dereito a obter copia del.

### **18.5 ARQUIVO ELECTRÓNICO DE DOCUMENTOS**

As administracións públicas poderán emitir validamente por medios electrónicos os documentos administrativos a que se refire o artigo 46 da Lei 30/1992, de réxime xurídico das administracións públicas e do procedemento administrativo común, sempre que incorporen unha ou varias sinaturas electrónicas conforme o establecido na sección III do capítulo II da LAECSP.

Os documentos administrativos incluírán referencia temporal, que se garantirá a través de medios electrónicos cando a natureza do documento así o requira.

Poderanse almacenar por medios electrónicos todos os documentos utilizados nas actuacións administrativas. Os documentos electrónicos que conteñan actos administrativos que afecten a dereitos ou intereses dos particulares deberanse conservar en soportes desta natureza, ben sexa no mesmo formato a partir do que se orixinou o documento ou noutro calquera que asegure a identidade e integridade da información necesaria para reproducilo. Asegurarase en todo caso a posibilidade de trasladar os



datos a outros formatos e soportes que garantan o acceso desde diferentes aplicacións.

Os medios ou soportes en que se almacenen documentos deberán contar con medidas de seguridade que garantan a integridade, autenticidade, confidencialidade, calidade, protección e conservación dos documentos almacenados. En particular, asegurarán a identificación dos usuarios e o control de accesos, así como o cumprimento das garantías previstas na lexislación de protección de datos.

Desde o punto de vista técnico, o almacenamento e tratamento de documentos en formato electrónico implica a utilización dunha sinatura avanzada que permita garantir a súa integridade e o non repudio. En caso de tratarse de documentos noutro formato, coma o papel, pode ser necesaria a súa conversión previa a algún formato electrónico mediante técnicas como a do escaneamento.

Ademais, o sistema informático de soporte debe garantir a seguridade dos documentos tanto desde o punto de vista da dispoñibilidade coma do control de acceso. Para rematar, implica tamén a implantación dun sistema de custodia de documentos electrónicos con mecanismos que permitan garantir a súa validez ao longo do tempo, mediante o uso, por exemplo, de sinaturas lonxevas ou da actualización periódica de formatos para evitar a súa obsolescencia.

#### **18.6 DIXITALIZACIÓN, COMPULSA ELECTRÓNICA.**

En primeiro lugar, debemos distinguir entre cotexo ou copia compulsada e copia auténtica:

- O cotexo e a compulsa de documentos é a técnica consistente na comprobación de que unha copia coincide co seu orixinal, que leva a



poder afirmar que esta é exacta. A copia cotexada ou compulsada en ningún caso acredita a autenticidade do documento orixinal.

- A copia auténtica dun documento acredita a autenticidade dos datos contidos nela, non só desde a perspectiva da súa identidade co documento orixinal, senón polos seus efectos de certificación, no sentido de que garante, igualmente, a autenticidade dos datos contidos neste último.

Xa que logo, a copia auténtica goza da mesma validez e eficacia que o documento orixinal, non limitando os seus efectos a un procedemento administrativo concreto.

As copias realizadas por medios electrónicos de documentos electrónicos emitidos polo propio interesado ou polas administracións públicas, manténdose ou non o formato orixinal, terán inmediatamente a consideración de copias auténticas coa eficacia prevista no artigo 46 da Lei 30/1992, de réxime xurídico das administracións públicas e do procedemento administrativo común, sempre que o documento electrónico orixinal se atope en poder da Administración, e que a información de sinatura electrónica e, de ser o caso, de selado de tempo permitan comprobar a coincidencia co devandito documento.

As copias realizadas polas administracións públicas, utilizando medios electrónicos, de documentos emitidos orixinalmente polas administracións públicas en soporte papel terán a consideración de copias auténticas sempre que se cumpran os requirimentos e actuacións previstas no artigo 46 da Lei 30/1992, de réxime xurídico das administracións públicas e do procedemento administrativo común.

As administracións públicas poderán obter imaxes electrónicas dos documentos privados achegados polos cidadáns, coa súa mesma validez e



eficacia, a través de procesos de dixitalización que garantan a súa autenticidade, integridade e a conservación do documento imaxe, do que se deixará constancia. Esta obtención poderase facer de forma automatizada, mediante o correspondente selo electrónico.

As copias realizadas en soporte papel de documentos públicos administrativos emitidos por medios electrónicos e asinados electronicamente terán a consideración de copias auténticas sempre que inclúan a impresión dun código xerado electronicamente ou outros sistemas de verificación que permitan contrastar a súa autenticidade mediante o acceso aos arquivos electrónicos da Administración pública, órgano ou entidade emisora.

Desde o punto de vista técnico, a compulsa e creación de copias auténticas implica a sinatura dun documento electrónico mediante o uso dun certificado electrónico. No caso de que este documento estea en formato papel, debe ser previamente dixitalizado mediante un procedemento establecido e seguro.

O proceso tecnolóxico de conversión a formato electrónico denomínase “dixitalización certificada”, entendendo como tal a definición dada compatible co termo “dixitalización” que se pode atopar no anexo do Esquema Nacional de Interoperabilidade (RD 4/2010).

Ao ser necesario que a copia sexa fiel e íntegra, o proceso de dixitalización debe cumprir unha serie de características:

- a) Debe ser completamente automático e realizarse de forma atómica, sen intervención humana, obtendo como entrada o documento orixinal e devolvendo como resultado a copia electrónica.



- b) O proceso tecnolóxico debe ser deseñado de xeito que non produza alteración con respecto ao documento orixinal. Neste aspecto hai que ter en conta que a obtención da copia implica a realización de distintas operacións. Así, por exemplo, será necesario nun primeiro momento obter, a partir do contido en papel, unha representación en formato electrónico, posiblemente mediante unha operación de escaneamento ou similar. Nesta fase será importante definir as características técnicas dos dispositivos que se van utilizar, e parámetros do proceso como pode ser o nivel de resolución (termo definido no anexo do RD 4/2010) mínima. A partir desta información, xa en formato electrónico, é moi probable que sexa necesario, ademais, realizar conversións entre formatos ou aplicar algoritmos de compresión con ou sen perda de información, para os que será necesario establecer uns límites de tolerancia.

Para a sinatura, a LAECSP establece a posibilidade de utilizar certificados de persoal adscrito á Administración ou funcionario. A Fábrica Nacional da Moeda e Timbre – Real Casa da Moeda (FNMT-RCM), que xa ofrecía certificados de identificación segura, dispón xa da versión actualizada para o cumprimento da LAE SCP.

O certificado para o persoal da Administración pública é a certificación electrónica emitida pola FNMT-RCM que vincula o seu titular cuns datos de verificación de sinatura e confirma, de forma conxunta:

- A identidade do seu titular, número de identificación persoal, cargo, posto de traballo e/ou condición de autorizado.
- O órgano, organismo ou entidade da Administración pública, ben sexa xeral, autonómica, local ou institucional, onde exerce as súas competencias, presta os seus servizos, ou desenvolve a súa actividade.



## **18.7 FACTURA E LICITACIÓN ELECTRÓNICAS**

### **18.7.1 FACTURA ELECTRÓNICA**

A facturación electrónica é un equivalente funcional da factura en papel e consiste na transmisión das facturas ou documentos análogos entre emisor e receptor por medios electrónicos (ficheiros informáticos) e telemáticos (dun ordenador a outro), asinados dixitalmente con certificados recoñecidos.

A Lei 56/2007, de medidas de impulso da sociedade da información, define a factura electrónica como “un documento electrónico que cumpre cos requisitos legal e regulamentariamente esixibles ás facturas e que, ademais, garante a autenticidade da súa orixe e a integridade do seu contido, o que permite atribuírlle a factura ao seu obrigado tributario emisor”.

Desta definición estendida en todo o mercado transmítense tres condicionantes para a realización de e-Factura:

- Necesítase un formato electrónico de factura de maior ou menor complexidade (EDIFACT, XML, PDF, html, doc, xls, gif, jpeg ou txt, entre outros).
- É necesaria unha transmisión telemática (ten que partir dun ordenador, e ser recollida por outro ordenador).
- Este formato electrónico e transmisión telemática deben garantir a súa integridade e autenticidade a través dunha sinatura electrónica recoñecida. O artigo 3.3 da Lei 59/2003, do 19 de decembro, define a sinatura electrónica recoñecida como: “a sinatura electrónica avanzada baseada nun certificado recoñecido e xerada mediante un dispositivo seguro de creación de sinatura”.



O certificado que se usa é o do expedidor real da factura, ben sexa este o obrigado tributario, un terceiro que actúe no seu nome ou o destinatario da factura, se se acordou a auto-facturación.

Para cumprir coa norma e que unha factura electrónica teña a mesma validez legal ca unha emitida en papel, o documento electrónico que a representa debe conter os campos obrigatorios esixibles a toda factura, estar asinado mediante unha sinatura electrónica avanzada baseada nun certificado recoñecido e ser transmitido dun ordenador a outro recollendo o consentimento de ambas as partes.

Para homoxeneizar estes aspectos técnicos desenvolveuse a Orde PRE/2971/2007, sobre a expedición de facturas por medios electrónicos cando o seu destinatario sexa a Administración xeral do Estado ou organismos públicos vinculados ou dependentes dela, e sobre a presentación perante a Administración xeral do Estado, ou os organismos públicos vinculados ou dependentes dela, de facturas expedidas entre particulares.

Nesta orde créase o formato de factura electrónica Facturae, xunto coa previsión de compatibilidade en futuras versións con normas como UBL (*Universal Business Language*). Facturae define fundamentalmente as tecnoloxías de sinatura que cómpre utilizar nas facturas e unha estrutura en XML que estas deben cumprir. Pódese atopar ampla información sobre este formato no sitio web <http://www.facturae.es/>.

Obrigas legais do expedidor:

1. Creación da factura. Mediante unha aplicación informática, cos contidos obrigatorios mínimos requiridos.
2. Uso de sinatura electrónica recoñecida.
3. Remisión telemática.



4. Conservación de copia ou matriz da factura. Esta obriga regúlase no artigo 1 do RD 1496/2003, onde se especifica a obriga de expedir, entregar e conservar facturas.
5. Contabilización e anotación en rexistros de IVE.
6. Conservación durante o período de prescrición.
7. Garantía de accesibilidade completa. Deber xestionar as facturas de xeito que se garanta unha accesibilidade completa: visualización, busca selectiva, copia ou descarga en liña e impresión. Esta é unha obriga inherente á conservación das facturas por medios electrónicos que o lexislador denomina acceso completo a datos, tratando de facilitar a auditoría e inspección das facturas electrónicas (artigo 9 do RD 1496/2003).
8. Subcontratación a un terceiro. Todas as fases anteriores poden ser subcontratadas a un terceiro, sen perder a súa responsabilidade. Regulado no artigo 5.1 do RD 1496/2003, o lexislador deixa claro nese mesmo parágrafo que, aínda que se permite a subfacturación a terceiros, é o obrigado tributario o responsable de cumprir todas estas obrigas.

#### Obrigas legais do destinatario:

1. Recepción da factura por medio electrónico
  - Verificación dos contidos mínimos esixibles
  - Verificación segura da sinatura electrónica. Regulado no artigo 21 e inherente ás obrigas da conservación das facturas electrónicas, indícase que: “o destinatario débese asegurar da lexibilidade no formato orixinal en que se recibiu, así como, de ser o caso, dos datos asociados e do mecanismo de verificación de sinatura”.

A diferenza do emisor, ao que se lle permite construír a factura desde a matriz, o destinatario debe conservar os orixinais asinados.



2. Contabilización e anotación en rexistros de IVE
3. Conservación durante o período de prescrición. Deber xestionar as facturas de xeito que se garanta unha accesibilidade completa.
4. Todas as fases anteriores pódellas subcontratar a un terceiro, sen perder a súa responsabilidade.

A obriga do uso de facturas electrónicas nace das previsións da Lei 30/2007, de contratos do sector público, que regula, entre outras moitas materias, o establecemento dunha plataforma de contratación electrónica do Estado e a utilización de medios electrónicos, informáticos ou telemáticos por parte das empresas do sector privado para a contratación con administracións públicas.

#### 18.7.2 LICITACIÓN ELECTRÓNICA

A Lei 30/2007, de contratos do sector público, na súa disposición derradeira novena, dedicada á habilitación normativa en materia de uso de medios electrónicos, informáticos ou telemáticos, e uso de factura electrónica, autoriza o ministro de Economía e Facenda para aprobar, logo do ditame do Consello de Estado, as normas de desenvolvemento que poidan ser necesarias para facer plenamente efectivo o uso de medios electrónicos, informáticos ou telemáticos nos procedementos regulados nesta lei, as especificacións técnicas das comunicacións de datos e os modelos que se deban utilizar.

Na práctica, un sistema de licitación electrónica debe permitir:

- Consultar en Internet as convocatorias dos contratos e obter os pregos. Para iso créase a figura do perfil do contratante, onde é posible consultar toda a información relativa a expedientes de contratación. O perfil do contratante debe garantir tecnicamente a data e hora da publicación, así como a integridade do contido. Isto



lévase á práctica mediante un sistema polo cal o contido que vai ser publicado é primeiro asinado electronicamente, incluíndo a sinatura un selado de tempo.

- Presentar por medios electrónicos solicitudes de participación, ofertas e documentos. Os licitantes deben poder presentar durante o prazo previsto ofertas de modo telemático. Para iso poderán facer uso do correspondente rexistro electrónico. Tal e como se explicou no devandito apartado, os licitantes poderán obter o correspondente resgardo.
- Obter información sobre o desenvolvemento do procedemento mediante a consulta dun taboleiro de anuncios electrónico, novamente a través do perfil do contratante.
- Recibir notificacións telematicamente. Para a emisión de notificacións telemáticas de modo fidedigno (de non ser así, en ocasións denomínanse simplemente comunicacións), dispónse de servizos como o Sistema de Notificacións Telemáticas Seguras creado pola Sociedade Estatal de Correos e Telégrafos. Neste sistema, o cidadán dispón dunha caixa do correo onde son enviadas as notificacións. O cidadán ten a posibilidade de ignorar, aceptar ou rexeitar as notificacións, coas mesmas garantías que pola vía tradicional, e o servizo informa a Administración da situación en cada caso.

Un sistema de licitación electrónica debe ademais garantir que, en función do procedemento de contratación establecido, só se poderá acceder ás ofertas na fase de tramitación prevista. Permitirá, polo tanto, definir as mesas de contratación, se é o caso, e establecerá os mecanismos necesarios para que as ofertas non poidan ser abertas ata que se constitúan formalmente.

Tecnicamente, isto resólvese mediante a creación de sobres electrónicos, seguindo os pasos que se detallan a seguir:

PREPARACIÓN DA LICITACIÓN:



- Identifícanse os membros da mesa. O sistema debe ter acceso á clave pública de cada un dos membros.

#### PRESENTACIÓN DE OFERTAS:

- No momento da presentación das ofertas, xéranse un par de claves pública e privada para cada unha, e a oferta é cifrada coa clave pública, creando o sobre electrónico.
- Este sobre só pode ser aberto mediante a correspondente clave privada. Con todo, a clave privada non se almacena no sistema. No seu lugar, aplícaselle un algoritmo que a divide en varias partes.
- Cada parte é asignada a un membro da mesa e cifrada coa súa clave pública, de xeito que é a única persoa que pode acceder a ela coa súa clave privada.

#### APERTURA DE OFERTAS:

- No día e hora de constitución da mesa de contratación, o sistema considera aberta a mesa de contratación e os membros poden acceder ao sistema.
- Cada un dos membros pode acceder á parte da clave privada do sobre que lle corresponde. Para iso, xa que está cifrada coa súa clave pública, deben utilizar a súa clave privada, identificándoos de xeito fidedigno.
- Unha vez o sistema dispón de suficientes partes da clave privada (non é necesario que participen todos os membros da mesa, senón que é posible establecer previamente un quórum), recompón a clave privada do sobre.
- Coa clave privada do sobre, o sistema xa pode extraer e mostrar a oferta.

Para rematar, o sistema de licitación electrónica debe contar cun compoñente de xestión de expedientes que permita levar a cabo todos os trámites na secuencia correcta, así como garantir o acceso á documentación xerada (actas, informes, etc.) e 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 a 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 facilítalles 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 a aplicación do modelo de eGoberno nas administracións públicas galegas.
2. Facilitar análises comparativas e aliñamentos 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 reflictan o estado da 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 da 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 onde 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 á 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.

#### 18.10 **PLAN DE BANDA LARGA 2013.**

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 planificaren os proxectos de banda larga.

A Xunta de Galicia aprobou o 18 de febreiro do 2010 o Plan Banda Larga de Galicia 2010-2013, que define a estratexia que se debe 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 socioeconómicos, 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, que 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 o 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 Procesamento de Datos Integral, e a aposta decidida pola Administración electrónica, impulsando o Decreto 198/2010, do 2 de decembro, polo que se regula o desenvolvemento da Administración electrónica na Xunta de Galicia e nas entidades dependentes dela, desenvolvendo proxectos para a mellora de servizos públicos dixitais e fomentando e divulgando o uso de 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 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.

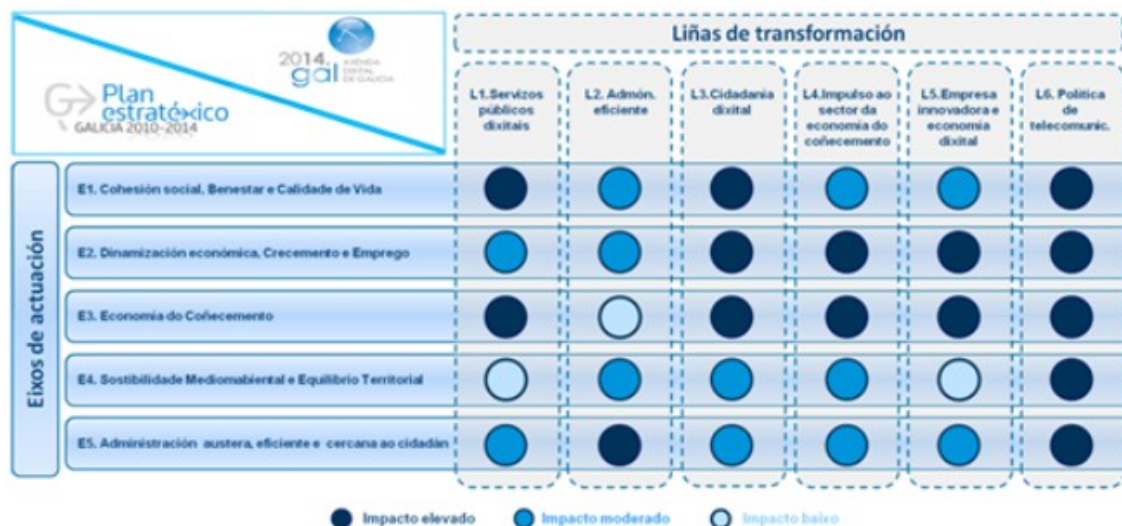
Este enfoque marcouno a Comisión Europea, que deseñou unha estratexia para axudarnos a saír fortalecidos da crise e a converter a UE nunha economía que goce de altos niveis de emprego, de produtividade e de cohesión social. Este é o labor de "Europa 2020: Unha estratexia para un crecemento intelixente, 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 clave para que Europa consiga as súas ambicións para o 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 inclúa a todos. As novas tecnoloxías deben servir como catalizadoras dos eixes 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 pódese ver como se alia a Axenda Dixital de Galicia co Plan Estratéxico Galicia 2010-2014.



Ademais do **Plan Estratéxico 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 propón Galicia establécense sete liñas estratéxicas dentro da Axenda Dixital de Galicia:

### L1. SERVIZOS PÚBLICOS DIXITAIS:

#### Obxectivos:

- Incluir 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.
- Facilitarlle 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:

- Proporcionarlles 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.
- Transferencia e compartimento 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 sustentable 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:

- Garantirle 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 último, una sétima liña, que lle dá 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 da Xunta de Galicia.

O Consello da Xunta de Galicia que tivo lugar o 21 de xullo do 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 estase a converter nunha das áreas produtivas clave 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 ca no 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 as L4 e L5, explican máis polo miúdo 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 persecue a Axenda Dixital a través da definición desta liña son os seguintes:

- Contribuír á creación de emprego e ao crecemento económico sustentable 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 á 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 facilitarlles ás empresas TIC os medios para achegaren a súa oferta de produtos ás empresas doutros sectores produtivos, de xeito que poidan desenvolver produtos adaptados ás necesidades do 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**

- Lei 30/1992, do 26 de novembro, de réxime xurídico das administracións públicas e do procedemento administrativo común.
- Lei 15/1999, do 13 de decembro, de protección de datos de carácter persoal.
- Lei 53/1999, do 19 de decembro, de sinatura electrónica.
- Lei 34/2002, do 11 de xullo, de servizos da sociedade da información e de comercio electrónico.
- Lei 11/2007, do 22 de xuño, de acceso electrónico dos cidadáns aos servizos públicos.
- Lei 30/2007, do 30 de outubro, de contratos do sector público.
- Lei 37/2007, do 16 de novembro, sobre reutilización da información do sector público.
- Real decreto 1720/2007, do 21 de decembro, polo que se aproba o Regulamento de desenvolvemento da Lei 15/1999, do 13 de decembro, de protección de datos de carácter persoal.
- Lei 56/2007, do 28 de decembro, de medidas de impulso da sociedade da información.
- Real decreto 1494/2007, do 12 de novembro, polo que se aproba o Regulamento sobre as condicións básicas para o acceso das persoas con discapacidade ás tecnoloxías, produtos e servizos relacionados coa sociedade da información e medios de comunicación social.
- Lei 17/2009, do 23 de novembro, sobre o libre acceso ás actividades de servizos e o seu exercicio.
- Lei 25/2009, do 22 de decembro, de modificación de diversas leis para a súa adaptación á Lei 17/2009, do 23 de novembro, sobre o libre acceso ás actividades de servizos e o seu exercicio.
- Real decreto 3/2010, do 8 de xaneiro, polo que se regula o Esquema Nacional de Seguridade no ámbito da Administración electrónica.



- Real decreto 4/2010, do 8 de xaneiro, polo que se regula o Esquema Nacional de Interoperabilidade no ámbito da Administración electrónica.
- Decreto 198/2010, do 2 de decembro, polo que se regula o desenvolvemento da Administración electrónica na Xunta de Galicia e nas entidades dependentes.
- Resolucións da Secretaría de Estado para la función pública polas que se aproban distintas normas técnicas de interoperabilidade.
- 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 baixo licenza 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

Xefe do Servizo de Informática do Concello de Santiago de  
Compostela

Colexiado do CPEIG



**19. DEFINICIÓN E ESTRUCTURAS  
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 TRABAJO.  
DISPOSITIVOS PERSOAIS.**



**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 e estruturas dos sistemas de información. 2**

*19.1.1 Definición dun sistema de información 2*

*19.1.2 Compoñentes e características 2*

### **19.2 Subsistema físico e lóxico. 4**

### **19.3 Principios e fundamentos dos ordenadores. Arquitectura e compoñentes dos ordenadores. Unidade central e periféricos. 5**

### **19.4 Servidores 9**

*19.4.1 Características dun servidor 9*

*19.4.2 Clúster 10*

*19.4.2.1 Clases de clúster 10*

*19.4.2.2 Compoñentes dun clúster 11*

*19.4.3 Servidores Blade 12*

### **19.5 Posto de traballo. 14**

### **19.6 Dispositivos persoais 15**

*19.6.1 PDA 15*

*19.6.2 TABLET 16*

*19.6.3 Smartphones 16*

### **19.7 BIBLIOGRAFÍA 19**



## *1. Definición e estruturas dos sistemas de información.*

Os sistemas de información foron considerados inicialmente como un elemento que podía proporcionar aforros de custo nas organizacións, na medida que estes poderían dar soporte a actividades de índole operativa nas que a información constituía o principal elemento para procesar.

Hoxe o sistema de información constitúe a base para o desenvolvemento de novos produtos ou servizos na empresa actual. É o soporte principal do traballo da empresa e das actividades da dirección xa que permite coordinar o traballo dentro e entre as organizacións e, sobre todo, permite mellorar o funcionamento delas, desenvolvendo novos modelos organizativos cunha clara orientación á información.

### *1. Definición dun sistema de información*

Non existe unha definición aceptada universalmente para o sistema de información da empresa. Podemos dicir que "sistema" é, nun sentido amplo, un conxunto de compoñentes que interactúan para cumprir algún obxectivo.

As definicións máis aceptadas na literatura son:

"O sistema de información pode definirse tecnicamente como un conxunto de compoñentes interrelacionados que recollen, procesan, almacenan e distribúen a información para dar soporte á toma de decisións e ao control nunha organización. Ademais do apoio á toma de decisións, coordinación e control, os sistemas de información poden axudar aos xestores e traballadores en xeral a analizar problemas ou situacións e a desenvolver novos produtos".

"Un sistema de información é un conxunto de compoñentes interrelacionados que recollen, procesan e distribúen datos e información, incluíndo así mesmo os mecanismos de retroalimentación implicados".

As definicións de sistema de información dunha organización implican a todas as persoas da organización, xa sexa como usuario, técnico, directivo, ou especialista; todas as persoas da organización están involucradas nalgunha das actividades de entrada, proceso ou saída do sistema de información.

Velaquí dúas ideas fundamentais relacionadas coa definición de sistema de información no ámbito da empresa:



A primeira é que o sistema de información da organización non é un departamento novo nin moito menos unha parte dependente dalgún dos departamentos funcionais clásicos, aínda que por motivos históricos os primeiros tratamentos automáticos de datos caeron dentro das competencias do departamento financeiro.

Un proxecto de sistemas de información non se debe considerar como unha tarefa funcional ou departamental, senón como un proxecto que compromete representantes de todos os departamentos sexa cal sexa o seu nivel na estrutura xerárquica.

A segunda, que o sistema de información non é só o centro de proceso de datos (CPD). O centro de proceso de datos só sería unha parte dos recursos de información, isto é, unha parte das actividades e compoñentes do sistema de información.

Como a información é a base da maioría das actividades realizadas nunha compañía, é necesario desenvolver sistemas que produzan e administren esta información. Un sistema de información debe prover información que apoie as operacións, a administración e as funcións de toma de decisións nunha empresa. En termos moi xerais, a tarefa do deseño e desenvolvemento dun sistema de información consiste en analizar un problema de administración, determinar os requirimentos de información por parte do usuario, deseñar un novo sistema e conseguir a súa implantación á vista dunhas restricións de tempo, recursos e orzamentos determinadas pola dirección.

Un sistema de información, entendido como un conxunto de persoas, datos e procedementos que funcionan de xeito unitario, executa tres actividades xerais. En primeiro lugar, recibe datos de fontes internas ou externas da empresa, en segundo lugar actúa sobre os datos para producir información e por último o sistema produce a información para os usuarios. Ou sexa, o sistema é un xerador de información que utiliza procedementos que determinan como se elabora a información. O obxectivo principal destes sistemas é asegurar que a información sexa exacta, estea dispoñible no momento preciso e que se presente de forma axeitada.

Os sistemas de información non necesitan estar baseados na informática, aínda que a maioría das veces o estean. O factor determinante consiste en saber se un sistema pode ser mellorado incluíndo nel a capacidade de procesamento con ordenadores. Se se ten un sistema manual que executa o traballo eficientemente e sen erro, existen poucos motivos para utilizar ordenadores. De todas as maneiras, cando crece o volume de traballo, os procedementos aumentan a súa complexidade e as actividades chegan a estar máis interrelacionadas, polo que se obteñen grandes melloras ao introducir a axuda de sistemas informáticos.

## ***2. Compoñentes e características***



Como xa se comentou anteriormente, é importante matizar a función de "soporte" que as tecnoloxías da información xogan para a implantación de sistemas de información.

Os sistemas de información precisan os seguintes elementos:

**Hardware:** consiste no equipamento informático para levar a cabo a entrada, proceso e saída da información. Os dispositivos de entrada inclúen teclados, lectores de cinta magnética, etc. Os dispositivos de proceso inclúen a unidade central de proceso, memoria e almacenamento. Existen múltiples dispositivos de saída como impresoras e pantallas de ordenador. Sen ánimo de afondar neste tipo de recursos, podemos clasificar os recursos de hardware en tres grandes categorías:

Ordenadores, que ademais dos miniordenadores e *mainframes* utilizados dende hai varias décadas, cómpre destacar a presenza habitual nos postos de traballo do ordenador persoal.

Periféricos, os cales inclúen toda unha variada gama que abrangue dende os elementos máis comúns de entrada, visualización, almacenamento e distribución de datos e información como poden ser os teclados, modems, pantallas de raios catódicos, cintas magnéticas e impresoras de impacto ata os elementos máis sofisticados, tales como un escáner, pantallas de vídeo xigantes, discos compactos e impresoras láser e de inxección de tinta, etc.

Outro hardware, en especial o que permite a interconexión entre equipos informáticos.

**Software:** consiste nos programas e instrucións que se dan ao ordenador. Estes programas e instrucións permiten ao ordenador procesar a información. Podemos clasificar o software nos tres seguintes grupos:

**Software de sistemas:** é un compoñente estreitamente asociado ao propio hardware, que lle permite operar e soportar software específico para as aplicacións.

**Software de aplicación:** que é o conxunto de programas que permitirán realizar as funcións previstas para o sistema de información.

Outro software: como poden ser as ferramentas de desenvolvemento, compiladores, software de comunicacións, etc.

**Bases de datos:** onde se encontra organizada a información da empresa. A base de datos da organización pode recoller feitos e información sobre clientes, empregados, vendas de competencia, etc. Supón un dos elementos claves para o sistema de información baseado en T.I.



Telecomunicacións: permítenlles ás organizacións enlazar sistemas informáticos constituíndo redes. As redes de área local interconectan equipos nun ámbito reducido, tipicamente nunha oficina ou edificio. As redes de área ampla (WAN) poden interconectar sistemas afastados mediante redes de comunicación exteriores, habitualmente públicas.

Persoas: é o elemento máis importante na maior parte dos sistemas de información baseados en ordenador. Podemos distinguir dous tipos de persoas en relación co sistema de información:

Persoal específico da área de sistemas: inclúe aquelas persoas que xestionan, desenvolven programas, e manteñen o sistema informático.

Usuarios: directivos e usuarios en xeral que interactúan dalgún modo co sistema.

Procedementos: inclúe as estratexias, políticas, métodos e regras que en xeral se aplican no uso e xestión do sistema de información.



## *2. Subsistema físico e lóxico.*

Un computador é un conxunto de dispositivos físicos integrados e de elementos lóxicos (programas) destinado ao tratamento automatizado da información. O sistema informático, a partir duns datos de entrada, realiza funcións de almacenamento, proceso e control proporcionando datos de saída, por iso consideramos que o usuario utiliza os datos que lle proporciona o sistema informático, pero non forma parte del. O usuario é un compoñente máis do sistema de información, que interactúa, no caso dun sistema de información automatizado, con outro compoñente do mesmo denominado sistema informático.

A arquitectura dun sistema informático defínese como o subconxunto de regras, normas e procedementos que especifican as interrelacións que deben existir entre os compoñentes e elementos, físicos e lóxicos, dun sistema informático e as características que deben cumprir cada un destes compoñentes.

O sistema informático, como soporte do sistema de información, evolucionou dende unha primeira situación en que todos os compoñentes do sistema —físicos, lóxicos e persoais— se encontraban centralizados nunha sala de ordenadores ata a situación actual en que os compoñentes do sistema están amplamente distribuídos en diferentes lugares físicos.

Este camiño cara á implantación progresiva de sistemas distribuídos pasou por diferentes fases e non se pode dar por finalizado pensando na evolución futura cara a sistemas repartidos. As fases son as seguintes:

- 1) Nunha primeira fase coincidente cos inicios dos procesos de informatización nas organizacións, os recursos están totalmente centralizados.
- 2) Unha segunda fase iníciase coa distribución dos compoñentes físicos e, nalgúns casos, humanos do sistema. Esta fase caracterízase pola introdución dos terminais non intelixentes asociados ás primeiras redes de teleproceso. Nesta segunda fase toda a capacidade de proceso e almacenamento está centralizada, pero a entrada e saída de datos distribuíuse fisicamente. Os inconvenientes deste tipo de sistemas eran os custos das comunicacións e a gran complexidade dos sistemas centralizados que soportaban as redes de teleproceso. Estes sistemas consumían gran cantidade de recursos na súa xestión interna.
- 3) Na terceira fase distribúense ademais os elementos lóxicos, introducindo certa intelixencia nos terminais. En realidade esta é unha fase de transición porque o abaratamento de custos dos equipos e o desenvolvemento tecnolóxico permitiron pasar rapidamente á informática distribuída de hoxe en día. Non obstante, este tipo de organización do sistema informática no que existen terminais con certa capacidade de proceso conectados a un equipo central, séguese mantendo en moitas



organizacións, xa que é netamente preferible ao modelo anterior sobre todo polo menor custo das comunicacións.



### ***3. Principios e fundamentos dos ordenadores. Arquitectura e compoñentes dos ordenadores. Unidade central e periféricos.***

O ordenador pódese ver como un dispositivo electrónico destinado ao tratamento automatizado da información. Para que un ordenador trate a información é necesario un sistema de información que, ante unha entrada, execute unha serie de instrucións e devolva un resultado.

Unha arquitectura de ordenador consiste no deseño, estudo da estrutura, e funcionamento dun ordenador. Especifica as interrelacións que deben existir entre os compoñentes e elementos físicos e lóxicos.

#### **Modelos de arquitecturas de ordenadores:**

##### **Arquitectura Von Newmann:**

Consiste nunha unidade central de proceso que se comunica a través dun único bus cun banco de memoria onde se almacenan tanto as instrucións do programa como os datos que serán procesados por el. Esta arquitectura é a máis empregada na actualidade.

Na memoria almacénanse tanto os datos coma as instrucións que forman o programa, co cal o cambio dun programa a outro só implica un cambio no valor de posicións de memoria.

Na arquitectura de von Newman prodúcese na CPU certa ralentización debido a que tanto as instrucións como os datos deben pasar da memoria á CPU por unha única canle (bus). Para este efecto coñéceselle como "o colo de botella de Von Newmann". Isto limita o grao de paralelismo (accións que se poden realizar ao mesmo tempo) e, xa que logo, o desempeño da computadora.

Nesta arquitectura asígnase un código numérico a cada instrución. Eses códigos almacénanse na mesma unidade de memoria que os datos que se van procesar para ser executados na orde en que se encontran almacenados en memoria. Isto permite cambiar rapidamente a aplicación da computadora e deu orixe ás computadoras de propósito xeral.

##### **Arquitectura Harvard**

Esta arquitectura xurdiu na universidade do mesmo nome pouco despois da arquitectura Von Newman. Do mesmo xeito que na arquitectura Von Newman, o programa almacénase como un código numérico na memoria, pero non no mesmo



espazo de memoria nin no mesmo formato que os datos. Por exemplo, pódense almacenar as instrucións en doce bits na memoria de programa, mentres os datos de almacenan en 8 bits nunha memoria á parte.

O feito de ter un bus separado para o programa e outro para os datos permite que se lea o código de operación dunha instrución ao mesmo tempo que se len da memoria de datos os operandos da instrución previa. Así evítase o problema do colo de botella de Von Newman e obtense máis rendemento.

A complexidade desta arquitectura só compensa cando o fluxo de instrucións e de datos é máis ou menos o mesmo. Por iso **non** é amplamente utilizada en ordenadores de propósito xeral. Non obstante, si que se utiliza nalgúns casos para construír procesadores de sinal (DSP).

### **Arquitecturas segmentadas**

Buscan mellorar o rendemento realizando paralelamente varias etapas do ciclo de instrución ao mesmo tempo. O procesador divídese en varias unidades funcionais independentes e divídense entre elas o procesamento das instrucións.

Se un procesador ten un ciclo de instrución sinxelo consistente só nunha etapa de busca do código de instrución e noutra etapa de execución da instrución, nun procesador sen segmentación as dúas etapas realizaríanse de xeito secuencial para cada unha das instrucións, pola contra, nun procesador con segmentación, cada unha destas etapas estaría asignada a unha unidade funcional diferente, a busca á unidade de busca e a execución á unidade de execución. Estas unidades poden traballar de forma paralela en instrucións diferentes. Estas unidades comunícanse por medio dunha cola de instrucións na que a unidade de busca coloca os códigos de instrución que leu para que a unidade de execución os tome da cola e os execute.

A mellora no rendemento non é proporcional ao número de segmentos debido a que cada etapa non toma o mesmo tempo en realizarse, ademais de que se pode presentar competencia polo uso dalgúns recursos como a memoria principal. Outra razón pola que as vantaxes deste esquema se perden é cando se encontra un salto no programa e todas as instrucións que xa se buscaron e están na cola se deben descartar e cómpre comezar a buscar as instrucións dende cero a partir da dirección á que se saltou. Isto reduce o desempeño do procesador e aínda se investigan maneiras de predicir os saltos para evitar este problema.

### **Arquitectura multiprocesamento.**

Cando se desexa incrementar o rendemento é necesario utilizar máis dun procesador para a execución do programa.



Para facer unha clasificación deste tipo de arquitecturas utilízase a taxonomía de Flynn que se basea no número de instrucións concorrentes e nos fluxos de datos sobre os que operar:

- *SISD (Simple Instruction Simple Data)*. Computador secuencial que non explota o paralelismo nin nas instrucións nin nos fluxos de datos, por exemplo, as máquinas con monoprocesador.
- *MISD (Multiple Instruction Simple Data)*. Pouco común debido ao feito de que a efectividade dos múltiples fluxos de instrucións adoita precisar de múltiples fluxos de datos. Utilízanse en situacións de paralelismo redundante como por exemplo en navegación aérea.
- *SIMD (Simple Instruction Multiple Data)*. Un computador que explota varios fluxos de datos dentro dun único fluxo de instrucións para realizar operacións que poden ser paralelizadas de forma natural. Nesta clasificación entrarían os procesadores matriciais e os procesadores vectoriais (aplican un mesmo algoritmo numérico a unha serie de datos matriciais).
- *MIMD (Multiple Instruction Multiple Data)*. Téñense múltiples procesadores que de forma sincronizada executan instrucións sobre diferentes datos. O tipo de memoria que estes sistemas utilizan é distribuída. Nesta arquitectura englobanse os sistemas distribuídos, distinguindo aqueles que explotan un único espazo compartido de memoria (procesadores superescalares, multiprocesador simétrico (SMP) e acceso non uniforme a memoria (NUMA)) daqueles que traballan con espazos de memoria distribuída, como os clúster.
  - Nos sistemas SMP (Simetric Multiprocessors), varios procesadores comparten a mesma memoria principal e periféricos de E/S, normalmente conectados por un bus común. Coñécense como simétricos, xa que ningún procesador toma o papel de mestre e os demais de escravos, senón que todos teñen dereitos similares en canto ao acceso á memoria e periféricos e ambos os dous son administrados polo sistema operativo.
  - Os clúster son conxuntos de computadoras independentes conectadas nunha rede de área local ou por un bus de interconexión e que traballan de maneira cooperativa para resolver un problema. É clave no seu funcionamento contar cun sistema operativo e programas de aplicación capaces de distribuír o traballo entre as computadoras da rede.

As partes físicas (hardware) que compoñen un ordenador pódense esquematizar nas seguintes:

**PROCESADOR** tamén coñecido como **CPU** (Central Process Unit). Encárgase de interpretar e executar as instrucións dos programas, realizando cálculos aritméticos e lóxicos cos datos. Tamén é o encargado de comunicarse coas demais partes do sistema.



Internamente está constituído por unha colección complexa de circuítos electrónicos. Cando se incorporan todos estes circuítos nun chip de silicio, este chip denomínase microprocesador.

A CPU está composta pola unidade aritmética lóxica, a unidade de control e os rexistros do sistema:

- a. Unidade de control (UC): A función da unidade de control consiste en ler as instrucións que residen na memoria principal, interpretalas e executalas dando as oportunas ordes á unidade aritmética lóxica e aos restantes elementos do sistema.
- b. Unidade aritmética lóxica (ALU): Executa as operacións aritméticas lóxicas que lle sinala a instrución residente na unidade de control.
- c. Rexistros do sistema: Son circuítos que serven como área interna de traballo. Almacenan unha palabra de bits. Estes circuitos son moi rápidos e forman parte do propio procesador.

Hai que facer mención especial aos **microprocesadores multinúcleo** que combinan dous ou máis procesadores independentes nun único circuítro integrado. Un dispositivo de dobre núcleo contén só dous microprocesadores independentes. En xeral, os microprocesadores multinúcleo permiten que un dispositivo computacional exhiba certa forma do paralelismo no ámbito de subproceso, tamén chamado fío ou thread (thread level parallelism, TLP) sen incluír múltiples microprocesadores en paquetes físicos separados. Esta forma de TLP coñécese a miúdo como multiprocesamento no chip (chip-level multiprocessing) ou CMP.

2. **MEMORIA PRINCIPAL.** Lugar onde se almacenan os datos e as instrucións dos programas en execución, onde se poden recuperar e gravar nela datos a través das dúas operacións básicas definidas sobre ela: lectura ou escritura.

Está constituída por celas ou elementos capaces de almacenar 1 bit de información. A memoria organízase en conxuntos de elementos dun tamaño determinado chamados *palabras de memoria*. A cada palabra correspóndelle unha dirección única.

Cada palabra é unha unidade direccionable na memoria. O mapa de memoria correspóndese co espazo de memoria direccionable. Este espazo vén determinado polo tamaño das direccións.

3. **BUSES.** Para funcionar o hardware necesita unhas conexións que lles permitan aos compoñentes comunicarse entre si e interactuar. Estas conexións denomínanse buses ou canles. Un bus constitúe un sistema común interconectado que coordina e transporta información entre as partes do ordenador.

Un bus caracterízase por dúas propiedades:



- A cantidade de información que pode manipular simultaneamente, chamada "largo de bus".
- A rapidez con que pode transferir eses datos.

Existen tres tipos de buses nun ordenador, en función do tipo de datos que transporten:

- *Bus de control*: Encárgase de transmitir datos que serán utilizados como ordes de control.
- *Bus de enderezos*: Encárgase de transmitir datos que serán utilizados como enderezos de memoria.
- *Bus de datos*: Encárgase de transportar datos como tales.

O conxunto destes tres buses forma o **bus do sistema**.

**PERIFÉRICOS.** Unha das funcións básicas do computador é enviar e recibir datos dende dispositivos externos á CPU. Estes dispositivos coñécense co nome xenérico de periféricos, podendo ser de lectura, de escritura e de lectura e escritura. A pesar de que o termo periférico implica a miúdo o concepto de "adicional pero non esencial", moitos deles son elementos fundamentais para un sistema informático.

Os periféricos clasifícanse segundo o uso que fagan da información en:

- *Dispositivos periféricos de entrada*. Introducen datos e instrucións na CPU, por exemplo: un rato, un teclado.
- *Dispositivos periféricos de saída*. Permiten ver os resultados, por exemplo: un monitor, unha impresora.
- *Dispositivos periféricos de entrada/saída (E/S)*. Teñen comunicación bidireccional coa CPU, por exemplo, un dispositivo de almacenamento.

Segundo a súa función temos as seguintes clases principais de periféricos:

## PERIFÉRICOS DE ALMACENAMENTO

Estes periféricos encárganse de gardar ou salvar os datos dos que fai uso a CPU para que ela poida facer uso deles despois de seren eliminados da memoria principal, a cal se borra cada vez que se apaga o ordenador. Poden ser internos, como un disco duro, ou extraíbles, como un CD. Os máis comúns son: disco duro, gravadora/lector de CD ou de DVD, memorias flash (usb, sd, etc.), cintas magnéticas.

## PERIFÉRICOS DE SAÍDA



Son os que reciben información procesada pola CPU e reproducena para que sexa perceptible para o usuario. Os principais son: monitor, impresoras, altofalantes, auriculares.

## PERIFÉRICOS DE COMUNICACIÓN

A súa función é permitir ou facilitar a interacción entre dous ou máis ordenadores, ou entre un ordenador e outro periférico externo a el. Entre eles encóntranse os seguintes: modem, tarxeta de rede (Ethernet ou Wireless), tarxeta Bluetooth, controladores de portos (serie, paralelo, infravermello, etc.).



#### 4. Servidores

Como se observa en apartados anteriores, na clasificación dos ordenadores non aparece o termo servidor. Este termo xorde orixinalmente do mundo software debido á arquitectura cliente/servidor, na que uns programas denominados *clientes* realizan peticións a outros programas denominados *servidores* que atenden esas peticións realizando as accións necesarias.

- Un **servidor** defínese daquela como un programa que acepta conexións co obxecto de atender peticións mediante o envío de respostas.
- Un **cliente** defínese como un programa que establece conexións co propósito de realizar peticións.

Este uso dual pode levar a confusión. Por exemplo, no caso dun servidor web, este termo podería referirse á máquina que almacena e manexa os sitios web, e neste sentido é utilizada polas compañías que ofrecen *hosting* ou hospedaxe. Alternativamente, o servidor web podería referirse ao software, como o servidor de http de Apache, que funciona na máquina e manexa a entrega das páxinas web como resposta a peticións dos navegadores dos clientes.

Debido á especialización e criticidade de moitos tipos de servidores, o termo "servidor" utilízase para referirse ao ordenador (hardware) onde está instalado o programa que atende ás peticións.

Así, un servidor no ámbito profesional é un ordenador especificamente deseñado para optimizar a execución dun determinado programa servidor ou un conxunto deles.

Loxicamente este hardware específico necesita un sistema operativo (SO) personalizado para executar programas servidores, sendo habitual que as compañías proporcionen SO para usuario final (Windows 7, Ubuntu Desktop) e SO para servidores (Windows 2008 Server R2, Ubuntu Server...).

##### 1. Características dun servidor

Existen factores como a fiabilidade, o rendemento ou o custo que determinan o tipo de servidor (hardware) que se require para albergar un software servidor. Así podemos ter nun mesmo servidor hardware varios programas servidores, ou ben pódese ter un servidor hardware por cada programa servidor.

Dende este punto de vista, calquera computador que albergue un determinado software servidor podería ser considerado un servidor. Non obstante, as máquinas que se deseñan co propósito de albergar programas servidores teñen unha serie de características particulares que fai necesario empregar hardware especializado, orientado a unha alta fiabilidade e rendemento.



- Teñen que procesar numerosas peticións de clientes nun tempo curto, por iso necesitan unhas CPU con velocidades altas de procesamento. Se por características da aplicación se require unha gran cantidade de procesamento, é máis recomendable engadir máis CPU para traballar en paralelo, en lugar de aumentar a velocidade dunha única CPU, por cuestións de redundancia e fiabilidade.
- Se o servidor recibe peticións concorrentemente é necesario que conte cunha cantidade de memoria principal ou RAM elevada que lle permita abrir *threads* e atender de forma axeitada aos clientes.
- Os buses polos que circula a información dentro do servidor teñen que ser de alto rendemento para non provocar colos de botella.
- Algúns tipos de servidores (ficheiros e bases de datos sobre todo) necesitan unha tecnoloxía de almacenamento altamente eficiente sendo normal encontrar dous tipos de tecnoloxías distintas:
  - SAN (Storage Area Network). É unha rede especializada que permite un acceso rápido e fiable entre servidores e recursos de almacenamento independentes ou externo. Desta forma un dispositivo de almacenamento non é propiedade exclusiva dun servidor, senón que os dispositivos de almacenamento son compartidos entre todos os servidores da rede como recursos individuais. Esta arquitectura implica dispoñer dunha infraestrutura de rede de alta velocidade dedicada só para almacenamento e *backup*, optimizada para mover grandes cantidades de datos, e consistente en múltiples recursos de almacenamento xeograficamente distribuídos.
  - NAS (Network Attached Storage). Os dispositivos NAS son dispositivos de almacenamento aos que se accede a través de protocolos de rede.

Os dispositivos NAS utilizan usualmente máis dun dispositivo de almacenamento, na maioría dos casos están compostos por RAID (Redundant Arrays of Independent Disks) de discos o que aumenta a capacidade de almacenamento, a seguridade, e a velocidade de acceso á información.

- Os servidores están preparados para ofrecer servizos cun grao de dispoñibilidade de máis do 99%. Isto implica:
  - Que está acendido as 24 horas do día, co que é necesario un sistema de refrixeración axeitado. Para iso sitúanse en centros de procesamento de datos onde existe a temperatura e humidade óptimas de funcionamento.
  - Que teñen que contar con sistemas de alimentación ininterrompida para evitar que un corte eléctrico os deixe indispoñibles.
  - Que é necesario utilizar compoñentes **hot swap**, que son compoñentes que se poden substituír "en quente sen parar o servidor". Isto ten especial importancia con servidores críticos que non poden estar



parados por unha acción planificada. Os compoñentes **hot swap** máis comúns son:

- Os discos duros configurados en RAID
- As fontes de alimentación
- Os servidores poden estar situados en armarios RACK ou non. A configuración de servidores en RACK é modular permitindo agregar ou quitar compoñentes con máis facilidade (engadir unha cabina de cintas de *backup*, unha nova fonte de alimentación ou un novo servidor).

## 2. Clúster

Un clúster é un tipo de computador distribuído ou paralelo que consiste nun grupo de computadoras interconectadas que traballan conxuntamente na solución dun problema. Estes sistemas constitúen unha solución flexible, de baixo custo e de grande escalabilidade para aplicacións que requiren unha elevada capacidade de cómputo e memoria.

A tecnoloxía dos clúster evolucionou en apoio de actividades que van dende aplicacións de supercómputo e software de misións críticas, servidores web e comercio electrónico, ata bases de datos de alto rendemento, entre outros usos.

Os clúster ofrecen as seguintes características:

- *Alto rendemento*: deseñados para dar altas prestacións en canto a capacidade de cálculo e velocidade de proceso.
- *Alta dispoñibilidade*: deseñados para garantir a total e absoluta dispoñibilidade do servizo no tempo ofrecendo un funcionamento ininterrompido. Todas as máquinas deste clúster están sincronizadas e monitoradas entre si. Se se produce un fallo nalguna das máquinas do clúster, detéctase ese fallo automaticamente e as outras máquinas asumen as funcións e seguen funcionando mantendo así o software a dispoñibilidade do sistema. Son tolerantes a fallos.
- *Balanceo de carga*: un clúster estará composto por un ou máis nodos que actúan como *frontend* do clúster, e que se ocupan de repartir as peticións de servizo que reciba o clúster a outros ordenadores do clúster que forman o *backend* deste, evitando así os colos de botella.
- *Escalabilidade*: É relativamente alcanzable aumentar un nodo nun sistema clúster.

Un clúster de servidores ten principalmente dúas vantaxes considerables sobre as solucións de servidores estándar:

- Garanten a alta dispoñibilidade de servizos e datos.



- Permite aproveitar ao 100% a capacidade dos nodos introducidos (non hai nodos en *stand-by*).

### **1. Clases de clúster**

A forma en que operará o clúster está determinada pola función que este deberá desempeñar:

- Clúster de alto rendemento: deseñado para dar altas prestacións en canto a capacidade de cálculo. Existen distintas aplicacións que se lles pode dar a este tipo de clúster, entre as que encontramos: cálculos matemáticos, renderizacións de gráficos, compilación de programas, descifrado de códigos.
- Clúster de alta dispoñibilidade: están deseñados para garantir o funcionamento ininterrompido de certas aplicacións. A idea principal deste tipo de clúster é proporcionar un servizo ininterrompido as 24 horas do día, os 7 días da semana.

Están formados por un conxunto de dous ou máis máquinas que comparten os discos de almacenamento de datos, e que se monitoran mutuamente. De se producir un fallo do hardware ou das aplicacións dalgunha das máquinas do clúster, o software de alta dispoñibilidade é capaz de volver arrancar automaticamente os servizos que fallaron en calquera das outras máquinas do clúster. E cando a máquina que fallou se recupera, os servizos son novamente migrados á máquina orixinal.

- Clúster de alta eficiencia: Son clúster co obxectivo de deseño de executar a maior cantidade de tarefas no menor tempo posible. Existe independencia de datos entre as tarefas individuais.

Os clúster de alta eficiencia e alta dispoñibilidade adoitan utilizarse para ámbitos empresariais e esta funcionalidade só pode ser efectuada por hardware especializado, mentres que os clúster de alto rendemento son propios de universidades e centros de cálculo.

### **2. Compoñentes dun clúster**

Para que un clúster funcione como tal non abonda só con conectar entre si os ordenadores, senón que é necesario proverlos dun sistema de manexo do clúster que se encargue de interactuar co usuario e os procesos que corren nel para optimizar o funcionamento. É dicir que, para poder funcionar, require tantos compoñentes hardware como software.

- **Nodos.** Son os ordenadores en si mesmos, existindo ordenadores persoais, sistemas multiprocesador ou estacións de traballo (*workstations*). Poden ser:
  - *Dedicados:* o seu uso está exclusivamente dedicado a realizar tarefas relacionadas co clúster.



- *Non dedicados:* o seu uso non está exclusivamente dedicado a realizar tarefas relacionadas co clúster, utilizándose os ciclos de reloxo do computador cando este non se utiliza.
- **Almacenamento.** Pode consistir nunha NAS, unha SAN, ou almacenamento interno no servidor. O protocolo máis comunmente utilizado é NFS (Network File System), sistema de ficheiros compartido entre servidor e os nodos. Non obstante existen sistemas de ficheiros específicos para os clúster como Lustre (CFS) e PVFS2.
- **Rede de interconexión.** Utilízanse redes de alta velocidade como solución de alto rendemento para que as comunicacións non sexan o colo de botella do rendemento do sistema.

As redes de interconexión son un compoñente fundamental dos clúster que proporcionan gran largo de banda, baixa latencia, fiabilidade e escalabilidade.

As redes de interconexión comúns en clúster son:

- Ethernet: Estándar de redes de computadoras de área local con acceso ao medio por contenda CSMA/CD.
- Fast Ethernet: Serie de estándares de IEEE de redes Ethernet de 100 Mbps (megabits por segundo).
- Gigabit Ethernet: Ampliación do estándar Ethernet que consegue unha capacidade de transmisión de 1 xigabit por segundo.
- SCI (Scalable Coherent Interface): Estándar de interconexión de redes de alta velocidade utilizado para multiprocesamento con memoria compartida e paso de mensaxes.
- ATM (Asynchronous Transfer Mode): Tecnoloxía de telecomunicación desenvolvida para facer fronte á gran demanda de capacidade de transmisión para servizos e aplicacións.
- Myrinet: Rede de interconexión dos clúster de altas prestacións. O procesamento das comunicacións de rede faise a través de chips integrados nas tarxetas de rede de Myrinet (Lanai chips), descargando a CPU de gran parte do procesamento das comunicacións.
- HIPPI (High Performance Parallel Interface): Bus para conexións de alta velocidade para dispositivos de almacenamento en supercomputadores. Foi substituído progresivamente por outras tecnoloxías máis rápidas.
- FiberChannel: Tecnoloxía de rede utilizada principalmente para redes de almacenamento, dispoñible primeiro á velocidade de 1 Gbps e posteriormente a 2, 4 e 8 Gbps.
- Infiniband: É unha rede xurdida dun estándar desenvolvido especificamente para realizar a comunicación nos clúster. A conexión básica é de 2 Gbps efectivos e poderíanse alcanzar os 96Gbps.



- **Sistema Operativo.** Ten que ser multiproceso e multiusuario.
- **Middleware.** Actúa entre o sistema operativo e as aplicacións, recibindo os traballos entrantes ao clúster e redistribuíndoos de maneira que o proceso se execute máis rápido e o sistema non sufra sobrecargas nun servidor determinado. Está composto de dous subniveis de software:
  - *SSI (Single System Image):* ofrécelles aos usuarios un acceso unificado a todos os recursos do sistema.
  - *Disponibilidade do sistema:* que permite servizos como puntos de revisión, recuperación de fallos, soporte para tolerancia a fallos.

### 3. Servidores Blade

Blade Server é unha arquitectura que conseguiu integrar en tarxetas todos os elementos típicos dun servidor. Cada servidor blade é unha tarxeta (chamada *blades*) que contén a memoria RAM, o disco duro e a CPU. Os servidores blade en tarxetas insírense nun chasis que se coloca nun rack estándar ocupando entre 4 U e 6 U dentro do rack, permitindo albergar un máximo de 16 servidores blade nun chasis. Este chasis, á súa vez, integra e permite compartir os elementos comúns como son:

- A ventilación e a refrixeración.
- Os *switches* de rede redundante co cableado.
- As fontes de alimentación e o SAI tipo *hot swap*.
- Interfaces de almacenamento.

Ao estar todo integrado no chasis conséguese reducir o consumo eléctrico, cableado, sistemas de arrefriamento e o espazo dentro do rack.

As empresas que requiren da actualización dos seus sistemas enfróntanse ao problema de consumo eléctrico, espazo, control de temperatura e situación dos novos equipos. Tradicionalmente, ata a chegada dos servidores Blade, o método para incrementar novos requirimentos era agregar máis servidores en rack, o que ocupa máis espazo, complica o cableado, fai máis complexa a xestión de administración dos sistemas, consome máis recursos, etc.

A tecnoloxía blade supón un deseño máis eficiente en canto a custo e espazo. Para iso reduciuse o chasis, baixouse o consumo, simplificouse o cableado e o mantemento, mentres se incrementan as funcionalidades.

Estes son os principios básicos nos que se fundamenta a arquitectura blade e que ao final proporciona unha redución do custo total.



## **Diferenzas entre un sistema de servidores montados en rack e blade server**

A principal diferenza é que nun sistema montado en rack, o servidor é unha unidade completa en si mesma; isto é que contén a CPU, memoria, fonte de alimentación, ventiladores e disipadores. Estes servidores son aparafusados no rack, e cada un é conectado á rede corporativa usando un cable separado.

Os blade servers son unha versión compacta de sistemas montados en rack. O blade inclúe unha CPU, memoria e dispositivos para almacenar datos. Pero non ten fonte de alimentación eléctrica nin ventiladores. Os blades son inseridos en slots e enlazados entre si grazas a un bus de alta velocidade dentro do chasis.

### **Vantaxes**

- Reduce a xestión grazas á súa infraestrutura simplificada.
- Comparte fontes de alimentación e ventiladores e unha xestión do sistema centralizada diminuindo custos porque requiren menos electrónica e consumen menos enerxía.
- O chasis elimina a maioría do cableado que se encontra nos sistemas montados en rack.
- Intercambio en quente (hot-swap): se un blade falla pode ser substituído sen ningún impacto nos outros blades.
- Facilitan a xestión e reducen tempo e custo administrativo ao estar todos os servidores nun único equipo.
- Redúcese o espazo ao integrar nun único chasis moitos servidores, sen reducir poder de cómputo.
- Escalabilidade horizontal: porque nos ofrece ampliar o número de servidores doadamente a medida que vai crescendo a demanda.
- Alta dispoñibilidade, pois a maioría dos equipos posúen elementos redundantes que garanten o funcionamento continuado dos servidores sen interrupcións.



## 5. *Posto de traballo.*

O máis habitual en calquera empresa é que no posto de traballo exista un microcomputador, é dicir, ou ben un PC ou unha estación de traballo (en inglés *workstation*). O concepto de PC ou ordenador persoal é aceptado amplamente.

Unha *workstation* é un microordenador de altas prestacións especialmente deseñado para niveis de alto rendemento en certas tarefas, como poden ser deseño gráfico, edición de vídeo, xestión de redes da internet, aplicacións de alto consumo, etc. Estes potentes ordenadores encontraron o seu sitio na enxeñaría e desenvolvemento de software entre outras cousas, debido á súa habilidade multitarefa.

Na actualidade os PC son bastantes potentes en canto á memoria e capacidade de procesamento. Non obstante, o hardware das estacións de traballo está optimizado para situacións que requiren un alto rendemento e fiabilidade, moita cantidade de memoria, computación de multitarefa, etc. onde xeralmente se manteñen operativas en situacións nas que calquera computadora persoal tradicional deixaría rapidamente de responder.

Os profesionais cando escoitan a palabra estación de traballo pensan que é unha máquina que non necesitan e que ten un custo moi superior ás expectativas. A realidade é que iso cambiou, especialmente en todo o relativo ao factor prezo, e agora, cun investimento mínimo, un profesional pode, grazas a unha estación de traballo, obter ata un 50 por cento máis de rendemento nas súas tarefas diarias.

Existen profesionais que compran un PC potente, con máis de 2 Gb de memoria, cunha tarxeta gráfica de alto nivel, con alta capacidade de memoria interna, etc. porque necesitan traballar con aplicacións de software. Fano porque non coñecen a existencia das estacións de traballo pero, sobre todo, porque non saben as diferenzas que teñen cun PC e as funcionalidades e vantaxes que lle poden ofrecer. Un PC, por exemplo, no apartado de memoria, chega ata onde chega e aí xorden os problemas. Existen estacións de traballo que pode alcanzar os 128 Gb de memoria, cinco discos duros, bipoceadores, etc.

As principais aplicacións dunha *workstation* son:

- CAD (Computer Aided Design, Deseño asistido por ordenador): destinadas ao deseño e análise de sistemas de enxeñaría e arquitectura.
- AEC (Architecture Engineering Construction): aplicables á edición de planos de construción e arquitectura, elaboración de orzamentos e seguimentos de obras.



- CAM (Computer Aided Manufacturing): aplicables no deseño, análise e proba de circuitos integrados, tarxetas e outros sistemas electrónicos.
- CASE (Computer Aided Software Engineering): axuda á xestión completa dos ciclos de vida dos desenvolvementos de aplicacións lóxicas.
- GIS (Geographic Information System): para edición, captura e análise de información sobre determinadas zonas xeográficas, con base en referencias de mapas dixitalizados.
- Sistemas expertos: baseados en técnicas de programación de intelixencia artificial, para aplicacións tales como detección electrónica de erros, funcións de diagnóstico ou configuración de ordenadores.
- Aplicacións empresariais: investigación cuantitativa, seguridade, simulación de análises reais...
- Edición electrónica: creación para a súa posterior publicación de xornais, revistas, presentacións e documentación en xeral.
- Telecomunicacións: xestión de redes, desenvolvemento de aplicacións de telecomunicacións baseadas en intelixencia artificial, aplicacións de apoio á investigación e desenvolvemento (I+D), edición electrónica e procesado de imaxes.
- As estacións de traballo tamén poden ser utilizadas como pasarelas (*gateways*), para acceder a grandes ordenadores, e para executar remotamente utilizando protocolos de comunicacións.



## 6. Dispositivos persoais

Por dispositivos persoais, entendemos aqueles dispositivos cunha relativa capacidade de cálculo e que polo seu tamaño poden ser transportados de forma sinxela polo usuario e, o que é máis importante, ser utilizado fóra dun ámbito regulado de traballo como pode ser a oficina. Os tres dispositivos máis habituais hoxe en día son: as PDA, as *tablets* e os *smartphones*.

### 1. PDA

Unha PDA (Persoal Digital Assistant) é un ordenador de peto deseñado como unha axenda electrónica, pero que actualmente posúen unha potencia razoable e son capaces de realizar numerosas funcións máis alá das de simple axenda electrónica constituíndose como unha extensión mesma do ordenador persoal, que poderemos sincronizar con el.

Outros termos asociados son palmtop e handhelds. Un palmtop é un ordenador pequeno que literalmente colle na palma da man. Un handheld é un ordenador sumamente pequeno que se pode soste coa man.

Os termos PDA, palmtop e handhelds xurdiron para cubrir necesidades diferentes. Actualmente a división entre ambas as dúas é moi difusa; ambos os dous termos utilízanse indistintamente.

As tecnoloxías de comunicacións sen fíos (Bluetooth, Wi-Fi, IrDA (infravermellos), GPS...) permiten que cunha PDA se poida consultar o correo electrónico, usalos como navegador GPS ou para temas relativos á domótica.

Pero máis alá das funcións e software coas que vén equipada a PDA, o que a fai verdadeiramente potente é a posibilidade de personalización case ilimitada ao permitir cargar as aplicacións "baixo demanda".

#### **Características:**

- Teñen un tamaño físico moi reducido para que caiba na man.
- Son bastantes lixeiras para que sexa doado o seu transporte nun peto.
- A pantalla é táctil ocupando gran parte do dispositivo e deixando pouco espazo para situar botóns hardware. Non adoitan dispoñer dun teclado con botóns (agás algúns dispositivos) polo que para agregar texto se utiliza un teclado virtual ou se lle engade un teclado externo por USB.
- Teñen capacidade multimedia, xa que integran altofalante, micrófono e gravadora de voz.



- Dispoñen de conexión de periféricos: para dispositivos de almacenamento externo e para módulos de expansión.
- Amplo soporte de conexións sen fíos: Bluetooth, infravermellos, Wi-fi.
- Funcionamento con baterías de Litio-ión.
- Capacidade de almacenamento por enriba dos 64 MB que se pode ampliar mediante o uso de tarxetas de memoria Flash.
- A sincronización cos ordenadores persoais permite a actualización do directorio, facendo que a información do computador e da PDA sexa a mesma. A sincronización tamén evita a perda da información almacenada en caso de que o accesorio se perda, sexa roubado ou destruído.
- Utilizan sistemas operativos específicos como son Windows Mobile, HP webOS, Linux.

### **Limitacións:**

- Potencia de computación reducida, debido a que os microprocesadores teñen que ter en conta a duración das baterías, o sobrequecemento, etc.
- Capacidade de almacenamento, aínda que hoxe en día con tarxetas de memoria de varios Gb é unha limitación menor.
- Baixa duración das baterías.
- Comunicacións.
- Software específico.

## **2. TABLET**

A tablet é un ordenador portátil de tamaño reducido, con pantalla sobre a cal o usuario pode escribir usando un lapis especial (stylus). O texto manuscrito é dixitalizado mediante recoñecemento de escritura. O lapis tamén se utiliza para moverse dentro do sistema e utilizar as ferramentas e funcións das tablet

A tablet combina a potencia dun ordenador portátil coa comodidade dun PDA.

En función de se dispoñen ou non de teclado se distinguen:

- Tablet "Slate": non dispón de teclado e é necesario utilizar un lapis ou os dedos para manipulalo.
- Tablet "Convertible": posúe un teclado. Pode ser deslizable para poder deslizarse debaixo da pantalla ou de modo que a pantalla poida xirar.

### **Características:**



- Os microprocesadores empregados nestes dispositivos están baseados en solucións para móbil.
- Para o almacenamento adóitanse utilizar discos EIDE convencionais pero de 2,5" (máis finos).
- A memoria que adoitan utilizar é SODIMM (small online DIMM), especiais para portátiles e impresoras.
- Pantalla táctil.
- Novas formas de control mediante voz e escritura manual.

### **3. Smartphones**

Estes dispositivos fan as funcións dun teléfono móbil convencional, pero están dotados dunha maior versatilidade, xa que tamén inclúen algunhas das funcións dun ordenador persoal. Na actualidade todos eles teñen en común un conxunto amplo de características, como unha pantalla táctil de gran formato, conectividade WiFi, Bluetooth, 3 G... Non obstante, existen outros importantes parámetros que convén ter en consideración:

#### **Pantalla**

É un compoñente de extrema importancia debido a que as pantallas táctiles dos smartphones son a interface directa de comunicación entre o usuario e o propio dispositivo.

Existen dúas tecnoloxías aplicables a estas superficies táctiles:

- As capacitivas: é a máis adecuada para facilitar a interacción directa co dedo en lugar dos habituais lapis, xa que para que respondan ao instante abonda con deslízalo, polo que o usuario non necesita exercer ningún tipo de presión sobre a superficie. Ademais, poden detectar varias pulsacións de xeito simultáneo, polo que a experiencia para o usuario é máis atractiva que no caso das resistivas.
- As resistivas: están formadas por varias capas, polo que cando as prememos entran en contacto. Isto produce un cambio de corrente, facilitando, deste modo, a detección da pulsación. Por esta razón, a experiencia de usuario neste caso parece ser menos atractiva que no anterior, xa que a resposta do dispositivo é algo máis lenta, ou polo menos é a sensación que pode brindarnos.

#### **Sistema operativo**

O funcionamento dun S.O. afecta directamente ao rendemento do dispositivo, a usabilidade da súa interface e as funcionalidades que poñen a disposición dos usuarios.



Actualmente o S.O. que se implanta nun *smartphone* adoptou tanta transcendencia como o equipo mesmo. A tal punto que se fala de "Smartphones Android", para referirse aos teléfonos que funcionan a través deste desenvolvemento de Google. Polo tanto, a elección do sistema é case tan importante como a dun smartphone en si. Hai que ter en conta que ademais estes S.O. tamén se utilizan nas tablets, por exemplo o iOS de Apple encóntrase no seu smartphone iPhone e no seu tablet iPad, o HP webOS implántase nos smartphones PalmPre e no seu tablet TouchPad, etc. A continuación expóñense os S.O. máis relevantes no mercado:

- **HP webOS** é un sistema operativo multitarefa baseado en Linux, desenvolvido por Palm, Inc., agora propiedade de Hewlett-Packard Company. Cabe destacar que usa tecnoloxías web como HTML5, JavaScript e CSS e soporta Flash.

webOS inclúe unha característica chamada "Synergy" que permite conectar o sistema con numerosos servizos de redes sociais e integrar información de varias fontes.

webOS fai uso da *cloud computing* para a sincronización de datos

- **Android** é un sistema operativo multitarefa baseado en Linux non só no seu núcleo, senón tamén no seu concepto: de código aberto e gratuito. Isto significa que calquera fabricante que desexe poderá instalar Android nos seus equipos posibilitando que o sistema estea dispoñible nunha ampla gama de smartphones. Foi deseñado orixinalmente para dispositivos móbiles, tales como teléfonos intelixentes, tablets, pero que actualmente se encontra en desenvolvemento para usarse en netbooks e PC.

O *Android Market* é un catálogo de aplicacións que pode ser descargado e instalado en dispositivos Android sen a necesidade dun PC

- **BlackBerry OS** un sistema operativo móbil desenvolvido por Research In Motion para os seus dispositivos BlackBerry. Ao comezo da súa andaina os BlackBerry estiveron orientados ao público corporativo, pero tras a aparición do iPhone abriuse ao uso persoal (ao igual que moitos smartphones). A interface máis cómoda para usar un BlackBerry é o teclado físico, que non é só un accesorio como noutros smartphones senón que é a chave para acceder a todas as funcionalidades.
- **Windows Phone** é un sistema operativo móbil desenvolvido por Microsoft, como sucesor da plataforma Windows Mobile.[2] Está pensado para o mercado de consumo xeralista en lugar do mercado empresarial[3] polo que carece de moitas funcionalidades que proporciona a versión anterior

O Hub Marketplace é o lugar no que se poden comprar e descargar todo tipo de contido como aplicacións, música, películas, programas de TV, podcast.

- **iOS** é o sistema operativo móbil de Apple desenvolvido orixinalmente para o iPhone, sendo usado despois no iPod Touch e iPad. Dise que é un SO que marca tendencias. Na última versión do SO (iOS 4) sopórtase a multitarefa. Un dos aspectos máis criticados é a súa falta de soporte para Flash.



A interface de usuario de iOS baséase no concepto de manipulación mediante xestos multitáctil. Os elementos da interface compóñense por deslizador, interruptores e botóns. A resposta é inmediata e provese dunha interface fluída. A interacción co sistema operativo realízase mediante xestos como deslizar, tocar e beliscar

O App Store de Apple é onde se poden comprar e descargar contidos. Foi pioneira nese aspecto.

A carga das aplicacións realízase case instantaneamente, brindando fluidez ao desempeño xeral do teléfono.

- Outros sistemas operativos: Bada, Meego, Symbian, etc.



## **7. 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-Vestíbulo. Laudon, K.C. e Laudon, J.P. (2002)
- PCWORLD Marzo 2010.
- <http://es.wikipedia.com>

**Autor:** Francisco Javier Rodríguez Martínez

Subdirector de Sistemas Escola Superior Enxeñaría Informática Ourense

Colexiado do CPEIG



# **20. ESTRUTURAS DA INFORMACIÓN. MODELO ENTIDADE-RELACIÓN. ENTIDADES E ATRIBUTOS. RELACIÓNS. DIAGRAMAS DE ENTIDADE-RELACIÓN.**



Tema 20. Estruturas da información. Modelo entidade-relación. Entidades e atributos. Relacións. Diagramas de entidade-relación.

---

## **ÍNDICE**

### **20.1 Estruturas da información**

### **20.2 Modelo entidade - relación (E-R)**

#### *20.2.1 Modelo entidade relación estendido*

##### *20.2.1.1 Cardinalidade*

##### *20.2.1.2 Xerarquía subconxunto*

##### *20.2.1.3 Xeneralización*

##### *20.2.1.4 Tipos de relacións*

##### *20.2.1.5 Control de redundancias*

##### *20.2.1.6 Dimensión temporal*

### **20.3 Entidades e atributos**

#### *20.3.1 Entidade*

#### *20.3.2 Relación ou interrelación*

#### *20.3.3 Dominio*

#### *20.3.4 Atributos*

### **20.4 Relacións**

#### *20.4.1 Elementos*

#### *20.4.2 Restricións*

##### *20.4.2.1 Restricións inherentes*

##### *20.4.2.2 Restricións semánticas*

### **20.5 Diagramas entidade-relación**

#### *20.5.1 Tipo de correspondencia*

#### *20.5.2 Entidades débiles*

#### *20.5.3 Papel ou rol*

#### *20.5.4 Atributos multivaluados e compostos*

#### *20.5.5 Atributos derivados*

### **20.6 Bibliografía**







### **20.1 ESTRUCTURAS DA INFORMACIÓN**

Chámase datos ao conxunto de propiedades que caracterizan un fenómeno, e información ao conxunto de valores que poden tomar estas propiedades xunto coas relacións ou dependencias entre elas.

Os modelos de datos son ferramentas de abstracción que permiten representar a realidade captando as restricións semánticas que nela se poidan dar.

Cando no mundo real se dá información de calquera suceso ou obxecto sempre os datos subministrados van acompañados dunha semántica ou dun significado. Do mesmo xeito estes datos están suxeitos a unhas restricións e nós entendemos os datos subministrados só se entendemos o dominio e as restricións de significados que acompañan á información. Non obstante, cando aparecen os ordenadores e as bases de datos se empezan a informatizar ocorre que se tende a almacenar datos separando a estes da súa interpretación, isto é, da súa semántica. Como consecuencia foi necesaria a aparición dos modelos de datos como unha ferramenta que axudase a lles incorporar significado aos datos almacenados.

Deste xeito, os modelos de datos proporcionan mecanismos de abstracción que permiten a representación da parte do mundo real que nos interesa rexistrar. Esta representación concíbese en dous niveis: o das estruturas que fan posible a representación da información, e o da información en si mesma.

Isto lévanos a diferenciar entre o que se denomina o esquema da base de datos (descrición específica en termos dun modelo de datos) e a colección de datos en si mesma que é o que denominamos base de datos.

O primeiro paso na representación dun problema do mundo real é a súa caracterización, ou o que é o mesmo, a determinación mediante un proceso de simplificación dos datos de interese (de entre todos os que interveñen no problema) e os seus límites (universo do discurso).



Os modelos de datos ofrecen distintos niveis de abstracción que facilitan a representación dos datos:

1. Clasificación. É a acción de crear unha categoría a partir das características comúns a un conxunto de exemplares. Por exemplo, a partir dos elementos Pedro, Xoán e Cristina podemos crear a categoría profesor de instituto.

O seu proceso inverso é a particularización.

2. Agregación. É a capacidade de considerar un obxecto sobre a base dos elementos que o constitúen. Por exemplo, podemos crear a clase coche a partir das clases volante, rodas, motor e carrozaría.

O seu proceso inverso é a desagregación.

3. Xeneralización. É similar á clasificación, pero creando unha categoría a partir das características comúns a un conxunto doutras categorías. Por exemplo, a partir das categorías profesor de matemáticas, profesor de física e profesor de informática, podemos crear a categoría profesor de instituto.

O seu proceso inverso é a especialización.

Segundo o grao de abstracción que apliquemos, podemos falar de tres tipos de modelos de datos:

- **Modelo conceptual.** Describe os tipos ou clases de obxectos dende un punto de vista estrutural. Para cada un destes tipos de obxectos describe as súas propiedades e o dominio e restricións de cada unha, así como as relacións entre eles. (Modelo Entidade/Relación).
- **Modelo lóxico.** Representa o problema baixo as restricións específicas do tipo de sistema xestor de base de datos (SXBD) que se aplique en cada caso específico. (Modelo relacional para o caso dos SXBD relacionais).
- **Modelo físico.** Representa o problema dende o punto de vista da súa implantación no sistema de tratamento utilizado e os métodos e mecanismos que se van usar no seu almacenamento.



Un modelo de datos define as regras mediante as cales se deben estruturar os datos do mundo real.

A representación dun mundo real mediante un determinado modelo dá lugar a un esquema que describe as categorías existentes. Non obstante, a realidade contempla ademais dos aspectos estáticos, os aspectos dinámicos. Polo tanto as propiedades do mundo real son de dous tipos:

- **Estáticas:** relativamente invariantes no tempo, que é o que se adoita coñecer como estruturas. Este tipo de propiedades está composto por:
  - Elementos permitidos como: os obxectos (entidades, relacións, rexistros...), asociacións entre obxectos, propiedades dos obxectos e asociacións (atributos, elementos de datos...), dominios (conxuntos de valores que poden tomar as propiedades).
  - Elementos non permitidos ou restricións, posto que non todos os valores, cambios de valor ou estruturas están permitidos no mundo real. Cada modelo ten por si mesmo limitacións en canto ás estruturas que permite:
    - As restricións impostas polo modelo coñécense como restricións inherentes
    - As restricións que permiten capturar a semántica do universo de discurso que se quere modelar e verificar a corrección dos datos almacenados na base de datos coñécense como restricións de integridade ou semánticas.
    - As restricións de integridade son impostas polo usuario, mentres que as restricións inherentes ao modelo son impostas directamente polo modelo.
- **Dinámicas:** Son as operacións que se aplican aos datos ou valores almacenados nas estruturas, os cales varían ao longo do tempo ao lles aplicar esas operacións. A aplicación de calquera operación sobre os valores dos elementos debe deixar estes cun estado válido, é dicir, os valores dos elementos deben pertencer a



algunha das categorías definidas no esquema e deben cumprir as restricións de integridade.

A compoñente estática dun modelo de datos defínese a través da linguaxe de definición de datos (DDL) e a compoñente dinámica defínese a través da linguaxe de manipulación de datos (DML), constituíndo ambas as dúas compoñentes a linguaxe de datos. Tamén se pode mencionar a linguaxe de control de datos (DCL) que engade unha capa de seguridade.



## **20.2 MODELO ENTIDADE - RELACIÓN (E-R)**

Proposto por Peter Chen en dous artigos (1976 e 1977). É un modelo moi estendido que experimentou unha serie de ampliacións ao longo dos anos.

O modelo apóiase en dous conceptos, o concepto de entidade e o concepto de relación. Este modelo de datos permite representar case calquera restrición do deseño de datos.

O modelo E/R percibe o mundo real como unha serie de obxectos relacionados entre si e pretende representalos graficamente mediante un mecanismo de abstracción. Este mecanismo de abstracción está baseado nunha serie de símbolos, regras e métodos que permitirán representar os datos de interese do mundo real, ofrecéndolle ao deseñador unha ferramenta para illar o modelo de consideracións relativas á máquina e aos usuarios.

### **20.2.1 *Modelo entidade relación estendido***

O modelo E/R co paso do tempo sufriu unha serie de modificacións tanto no seu simbolismo gráfico, como na ampliación dos seus elementos.

#### **20.2.1.1 Cardinalidade**

Este primeiro concepto en certo modo estaba tratado de forma implícita no modelo E/R orixinal. Non obstante, foi posteriormente cando se lle deu certa relevancia e mesmo unha forma de representación.

O concepto cardinalidade, tamén denominado "clase de pertenza", permite especificar se todas as ocorrencias dunha entidade participan ou non na interrelación establecida con outra(s) entidade(s):

- Se toda ocorrencia da entidade A debe estar asociada con polo menos unha ocorrencia da entidade B á que está asociada por unha determinada



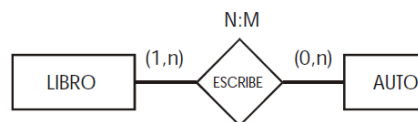
interrelación, dise que a clase de pertenza é obrigatoria, é dicir, a cardinalidade mínima é 1.

- Pola contra, se non toda ocorrencia da entidade A necesita estar asociada con algunha ocorrencia da entidade B asociada, dise que a clase de pertenza é opcional, é dicir, a cardinalidade mínima é 0.

Podemos definir a **cardinalidade dun tipo de entidade** como o número mínimo e máximo de ocorrencias dun tipo de entidade que poden estar relacionadas cunha ocorrencia do outro tipo de entidade que participan no tipo de interrelación.

A súa representación gráfica é unha etiqueta do tipo (0,1), (1,1), (0,n) ou (1,n) segundo corresponda, ao lado das entidades asociadas pola relación tal como se pode observar no seguinte exemplo, onde o primeiro elemento da tupla é a cardinalidade mínima, e o segundo elemento da tupla é a cardinalidade máxima, que coincide co tipo de correspondencia

Exemplo: 'Un libro pode estar escrito por ningún, un ou varios autores. Un autor escribe polo menos un libro e pode escribir varios'.



#### 20.2.1.2 Xerarquía subconxunto

A descomposición de tipos de entidade en varios subtipos é unha necesidade moi habitual no modelado conceptual. No mundo real pódense identificar varias xerarquías de entidades. A interrelación que se establece entre un supertipo e os seus subtipos corresponde á noción de " É-UN" ( IS-A) ou máis exactamente "é un tipo de".

Para a súa representación utilízase un triángulo invertido, coa base paralela ao rectángulo que representa o supertipo.



O concepto xerarquía subconxunto establece que unha entidade A é un subconxunto doutra entidade B cando toda ocorrencia da primeira tamén é unha ocorrencia da segunda, e o contrario non ten por que ser certo.

Polo tanto, teremos unha xerarquía subconxunto cando cada ocorrencia dunha entidade xenérica poida ser tamén unha ocorrencia doutras entidades que, potencialmente, son subconxuntos non disxuntos (solapados). É dicir, nas entidades subconxunto poden aparecer ocorrencias repetidas.

#### Características:

- Toda ocorrencia dun subtipo é unha ocorrencia do supertipo, as cardinalidades serán sempre (1,1) no supertipo e (0,1) ou (1,1) nos subtipos.
- Todo atributo do supertipo pasa a ser un atributo dos subtipos.

A entidade subconxunto pode ter atributos, ademais de ter os atributos da entidade xenérica, pero sempre as entidades subconxunto están identificadas pola clave da entidade xenérica. Ademais todos os atributos comúns das entidades subconxunto deberían aparecer na entidade xenérica para evitar repetir os atributos en cada unha das entidades subconxunto.

#### 20.2.1.3 Xeneralización

O concepto de xerarquía de xeneralización ou xeneralización establece que unha entidade xenérica X é unha xeneralización doutras entidades especializadas se cada ocorrencia da primeira é unha ocorrencia e soamente unha das outras entidades. Ás veces este concepto coñécese tamén como xerarquía de especialización.

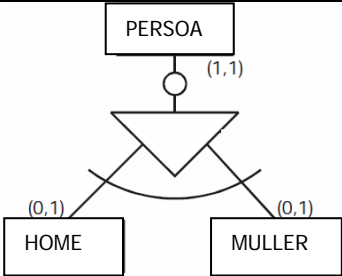
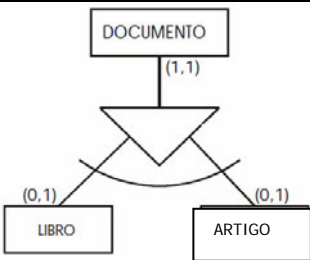
Terase unha xerarquía de xeneralización cando a entidade xenérica se divida nunha serie de entidades en función do valor que tome un determinado atributo da entidade xenérica.

A xeneralización ten dúas restricións semánticas asociadas:

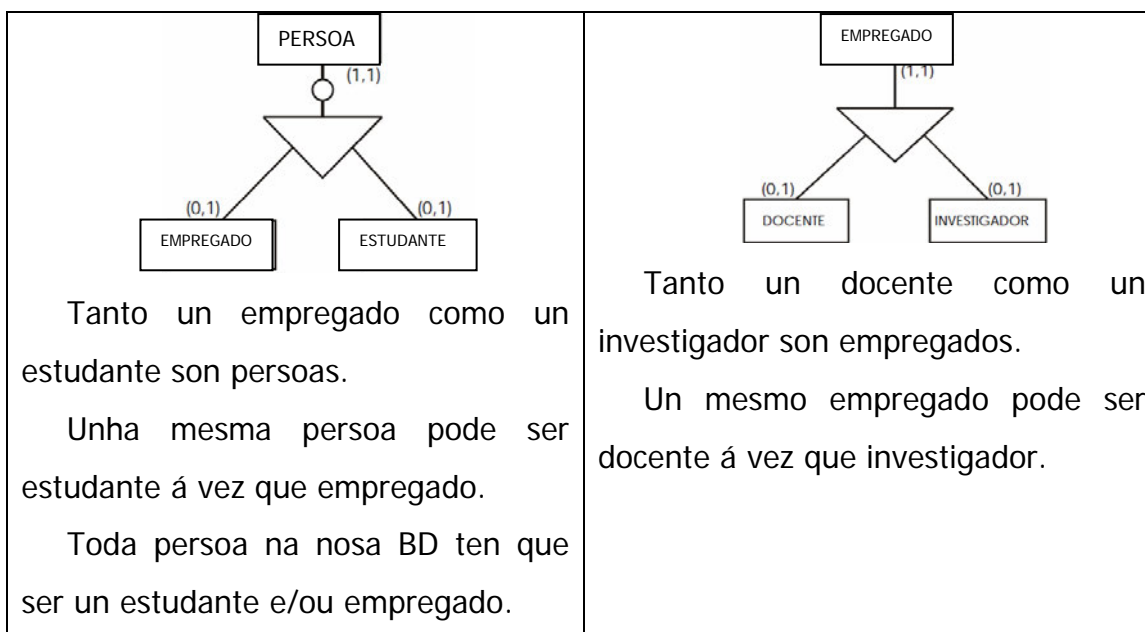


- **Totalidade** se todo exemplar do supertipo ten que pertencer a algún subtipo. O caso contrario chámase **parcialidade**.
- **Solapamento** se un mesmo exemplar do supertipo pode pertencer a máis dun subtipo. O caso contrario chámase **exclusividade**.

Poden existir interrelacións de cada unha das catro combinacións posibles, e representárianse da seguinte forma:

TOTAL SEN SOLAPAMENTO	PARCIAL SEN SOLAPAMENTO
 <p>Tanto un home como unha muller son persoa.</p> <p>Unha persoa non pode ser á vez home e muller.</p> <p>Toda persoa ten que ser un home ou unha muller.</p>	 <p>Tanto un artigo como un libro son documentos.</p> <p>Un mesmo documento non pode ser á vez un artigo e un libro.</p> <p>Pode haber documentos que non sexan nin artigos nin libros.</p>
TOTAL CON SOLAPAMENTO	PARCIAL CON SOLAPAMENTO

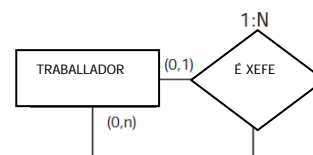




#### 20.2.1.4 Tipos de relacións

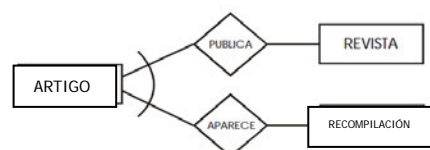
A. RELACIÓNS REFLEXIVAS Son interrelacións nas que intervén un único tipo de entidade (unarias).

Exemplo: Un traballador pode ser xefe de ningún traballador ou pode selo de varios traballadores, mentres que un traballador só é dirixido por ningún ou un traballador



B. INTERRELACIÓNS CON RESTRICIÓNS DE EXCLUSIVIDADE. Dúas interrelacións que implican un mesmo tipo de entidade participan dunha restrición de exclusividade se os exemplares desa entidade poden participar dunha ou outra interrelación, pero non de ambas as dúas.

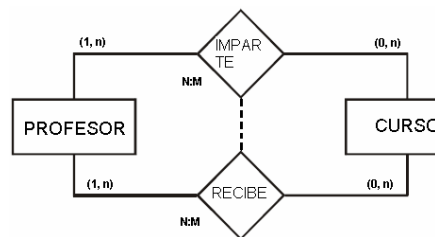
Recolleuse no esquema que nunha determinada biblioteca os artigos están publicados en revistas ou aparecen en recompilacións, pero non en ambos os dous





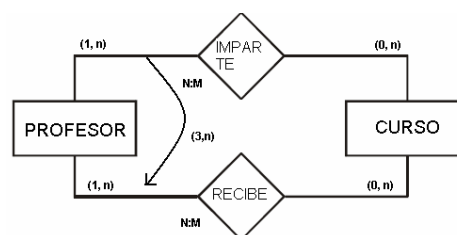
C. INTERRELACIÓNS CON RESTRICIÓNS DE EXCLUSIÓN. Dúas interrelacións entre os mesmos dous tipos de entidade son exclusivas se un exemplar do primeiro tipo de entidade e outro exemplar do segundo tipo de entidade só poden estar relacionados por unha das dúas interrelacións, nunca por ambas as dúas simultaneamente.

Un profesor non pode recibir e impartir o mesmo curso, aínda que ao contrario que na restrición anterior pode impartilo ou recibilo.



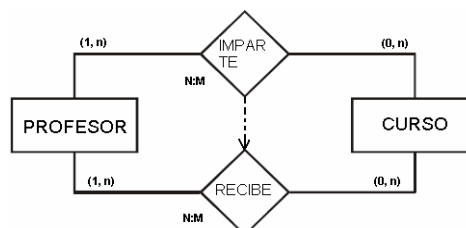
D. INTERRELACIÓNS CON RESTRICIÓNS DE INCLUSIVIDADE. Son dúas interrelacións que implican un mesmo tipo de entidade, nas que os exemplares da entidade tiveron que participar dunha interrelación cunha cardinalidade determinada para poder participar da outra.

Para que un profesor poida impartir un curso, ten que ter recibido o curso un mínimo de 3 veces.



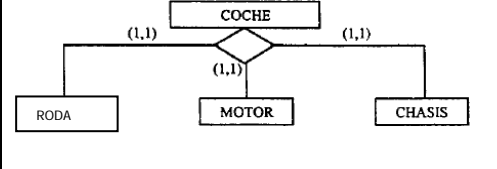
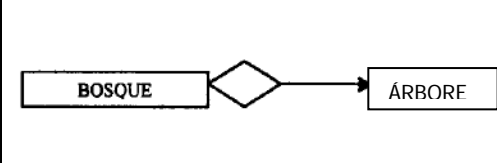
E. INTERRELACIÓNS CON RESTRICIÓNS DE INCLUSIÓN. Son aquelas que se establecen entre os mesmos dous tipos de entidade e que restrinxen unha interrelación entre dous exemplares de cada unha das entidades á vinculación deses dous mesmos exemplares a través da outra interrelación.

Todo exemplar de profesor que estea unido a un exemplar de curso mediante a interrelación imparte ten necesariamente que estar unido ao mesmo exemplar de curso mediante a interrelación recibe.





F. AGREGACIÓN. É un tipo especial de interrelación que permite representar tipos de entidade compostos que se forman a partir doutros máis simples. Existen dúas clases de agregacións

Composto/Compoñente: O supertipo de entidade obtense pola unión dos subtipos. Representase da seguinte forma	
Membro/Colección: O supertipo de entidade é unha colección de elementos dun mesmo subtipo. Representase como:	

### 20.2.1.5 Control de redundancias

No modelo E/R cómpre evitar as redundancias para non ter problemas de inconsistencias da representación. Un elemento dun esquema é redundante se pode ser eliminado sen perda de semántica.

Existen dúas formas principais de redundancia:

- Nos atributos (atributos derivados ou calculados): Aínda que son redundantes, non dan lugar a inconsistencias sempre que no esquema se indique a súa condición de derivados e a fórmula mediante a que se deben calculados.
- Nas interrelacións (tamén chamadas interrelacións derivadas): Unha interrelación é redundante se a súa eliminación non implica perda de semántica porque existe a posibilidade de realizar a mesma asociación de exemplares por medio doutras interrelacións. Para iso é condición necesaria pero non suficiente que forme parte dun ciclo.

A existencia dun ciclo non implica a existencia de interrelacións redundantes.

Para que unha interrelación poida ser eliminada por redundante tense que cumprir:

- Que exista un ciclo.



- Que as interrelacións que compoñen o ciclo sexan equivalentes semanticamente,
- Que despois de eliminar a interrelación se poidan seguir asociando os exemplares das dúas entidades que estaban interrelacionadas
- Que a interrelación non teña atributos ou que estes poidan ser transferidos a outro elemento do esquema co fin de non perder a súa semántica.

#### *20.2.1.6 Dimensión temporal*

É necesario establecer un método semántico e gráfico que recolla dalgún modo, no esquema conceptual, o transcurso do tempo e a súa influencia na forma en que cambian os datos. Existen varias aproximacións:

- A máis simple constitúena os atributos de tipo “data” asociados a algunhas entidades ou interrelacións:
  - Para sucesos instantáneos, é dicir, sen duración, abondará cun único atributo deste tipo.
  - Para poder almacenar feitos que transcorren nun intervalo de tempo determinado necesitaremos unha data\_inicio e unha data\_fin.
  - Nas bases de datos históricas, nas que unha interrelación entre dous exemplares concretos se poida repetir no tempo, o atributo data será multivaluado.
- Cando é necesario representar a evolución dun tipo de entidade ao longo do tempo utilízase un atributo de estado, que indicará en que estado concreto se encontra a entidade.

En moitos casos leva asociado outro atributo, que é a data na que se produciu o cambio de estado ou o intervalo de tempo en que permaneceu nese estado.



### 20.3 ENTIDADES E ATRIBUTOS

Chen distingue no modelo E/R os seguintes elementos: entidade, relación, atributo e dominio

#### 20.3.1 Entidade

Unha entidade é un obxecto real ou abstracto de interese nunha organización e acerca do cal se pode e quere obter unha determinada información; persoas, cousas, lugares, etc., son exemplos de entidades

A estrutura xenérica que describe un conxunto de entidades aplicando a abstracción denomínase tipo de entidade, mentres que entidade se refire a cada unha das ocorrencias ou exemplares dese tipo de entidade. Así pois, asociado ao concepto de entidade xorde o concepto de ocorrencia de entidade. Unha ocorrencia de entidade é unha realización concreta dunha entidade. Deste xeito, "hospital" é un tipo de entidade mentres que "CHOU" é unha ocorrencia ou exemplar.

Unha entidade debe cumprir as seguintes regras:

- Debe ter existencia propia (veremos que hai un tipo de entidades que en puro rigor non cumpre esta restrición como son as entidades débiles)
- Cada ocorrencia dun tipo de entidade ten que poder distinguirse das demais
- Todas as ocorrencias dun mesmo tipo de entidade deben ter as mesmas propiedades ou atributos

Unha entidade represéntase graficamente no modelo E/R mediante un rectángulo e no interior deste escríbese en maiúsculas o nome do tipo de entidade.

ENTIDADE
----------

Existen dous tipos de entidades:



- **Regulares:** os seus exemplares teñen existencia por si mesmos, p. ex. LIBRO.
- **Débiles** nas que a existencia dun exemplar depende de que exista certo exemplar doutro tipo de entidade. Veranse no apartado 2.2.2

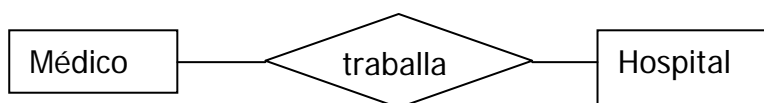
### 20.3.2 *Relación ou interrelación*

Unha interrelación é unha asociación entre entidades e caracterízase por unhas determinadas restricións que determinarán as entidades que poden ou non participar desa relación.

A interrelación represéntase graficamente por un rombo etiquetado co nome da interrelación en maiúsculas unido mediante arcos ás relacións que vincula.

Asociado ao concepto de interrelación xorde o concepto de ocorrencia de interrelación.

Unha ocorrencia de interrelación é a asociación concreta de ocorrencias de entidade de diferentes entidades. Por exemplo, se temos as entidades MEDICO e HOSPITAL, e a interrelación "traballa en", unha ocorrencia de interrelación será: MARTA GARCÍA traballa no CHOU.



Unha interrelación queda caracterizada por tres propiedades:

- Nome: as interrelacións deben ter un nome que as identifique univocamente.
- Grao: número de tipos de entidade sobre as que se realiza a asociación. A interrelación do exemplo anterior será binaria, é dicir, o seu grao sería dous.
- Tipo de correspondencia: Número máximo de ocorrencias de cada tipo de entidade que poden intervir nunha ocorrencia do tipo de interrelación.

As relacións poden ter atributos propios.



### 20.3.3 *Dominio*

Representa o conxunto de valores posibles dunha determinada propiedade ou atributo dun tipo de entidade ou dun tipo de interrelación. En termos de abstracción, é unha especialización dun conxunto. Co que se pode dicir que o dominio é un conxunto de valores homoxéneos cun nome.

Represéntase por un círculo pequeno acompañado do seu nome en minúsculas.

É importante resaltar neste punto que os dominios teñen existencia propia e é o que realmente captura unha semántica do mundo real. O que ocorre moi a miúdo é que se tende a confundir dominio con atributo.

### 20.3.4 *Atributos*

É cada unha das posibles propiedades ou características dun tipo de entidade ou tipo de interrelación. Os atributos toman valor nun dominio polo que un atributo é unha determinada interpretación dun dominio e varios atributos poden tomar valores no mesmo dominio. Por exemplo, se temos o atributo COR o dominio sobre o que se define podería ser: (LARANXA, BRANCO, AZUL e NEGRO).

Pódese representar graficamente de 2 formas:

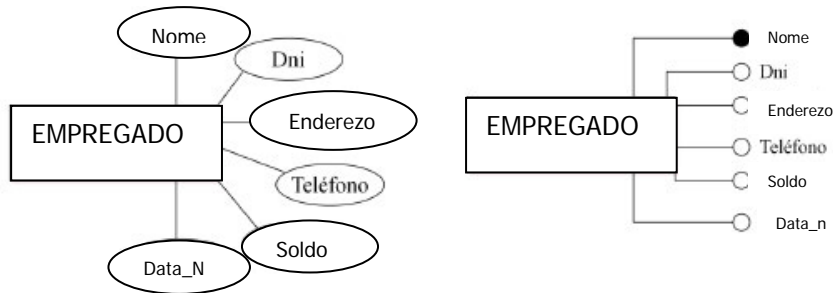
- a) Por un círculo pequeno unido por un arco ao tipo de entidade e acompañado do nome do atributo.
- b) Pechando nun ovalo o nome do atributo unido por un arco ao tipo de entidade.

En función das características do atributo respecto da entidade distínguense dous tipos de atributos:

- **Atributo identificador clave:** distingue de xeito único unha ocorrencia de entidade do resto de ocorrencias de entidade. Normalmente, o atributo identificador é único, pero pode haber casos nos que haxa varios atributos identificadores, polo que denominaremos a cada un deles **atributo identificador candidato**. Elixiremos un como identificador clave e o resto serán atributos identificadores. Represéntase graficamente:



•



•

- **Atributo descriptor:** caracteriza unha ocorrencia de entidade pero non a distingue do resto de ocorrencias de entidade.

Unha relación pode ter atributos ao igual que as entidades.



## 20.4 RELACIÓNS

O modelo relacional é un modelo lóxico de datos, desenvolvido por Codd, que introduciu a teoría matemática das relacións no campo das BD e supuxo un importante paso na investigación dos SXBD. O documento de Codd propón un modelo de datos baseado na "teoría das relacións", onde os datos se estruturan lóxicamente en forma de relacións —táboas—, e un obxectivo fundamental do modelo é manter a independencia desta estrutura lóxica respecto ao modo de almacenamento e a outras características de tipo físico (independencia de ordenación, indexación e dos camiños de acceso).

Este novo modelo de datos perseguía os seguintes obxectivos:

- Independencia lóxica: engadir, eliminar ou modificar calquera elemento da BD non debe repercutir nos programas e/ou usuarios que accedan a vistas deles.
- Independencia física: o modo en que se almacenan os datos non debe influír na súa manipulación lóxica e, polo tanto, os usuarios que acceden a eses datos non deben modificar os seus programas por cambios no almacenamento físico.
- Flexibilidade: poder ofrecer a cada usuario os datos da forma máis axeitada á súa aplicación.
- Uniformidade: As estruturas lóxicas dos datos presentan un aspecto uniforme (táboas), o que facilita a concepción e manipulación da BD por parte dos usuarios.
- Sinxeleza: As características anteriores, así como unhas linguaxes de usuario moi sinxelas, producen como resultado que o modelo de datos relacional sexa doado de comprender e utilizar por parte do usuario final.

### 20.4.1 Elementos

O modelo relacional introduce a súa propia terminoloxía para denominar os obxectos e elementos utilizados:



1. **Relación.** É o elemento central do modelo relacional. Son matrices bidimensionais (táboas) caracterizadas por un nome, un conxunto de atributos (dimensión vertical da táboa = columnas) e un conxunto de tuplas (dimensión horizontal = filas).

Cada tupla está formada polo conxunto de valores que toma cada un dos atributos para un elemento da relación.

Nas relacións podemos falar de dúas compoñentes:

- Intensión é a parte definitoria e estática da relación. Define a estrutura abstracta de datos e as restricións de integridade desta. É o que chamaremos *esquema de relación*.
- Extensión é o conxunto de tuplas que satisfai o esquema de relación nun instante dado e están almacenadas na base de datos. Varía co transcurso do tempo.

O número de tuplas dunha relación nun instante dado denomínase **cardinalidade** da relación, e normalmente varía co transcurso do tempo. O número de columnas ou atributos denomínase **grao** da relación.

2. **Dominio.** É o conxunto definido, finito e homoxéneo dos valores atómicos posibles dun determinado atributo.

Cada atributo está ligado a un determinado dominio e representa o uso dun dominio para unha determinada relación.

Os dominios poden estar definidos por intención (conxunto definido mediante unha serie de regras abstractas) ou por extensión (conxunto finito de valores posibles).



3. **Claves dunha relación.** Unha clave é unha(s) columna(s) cuns valores que identifican unha única fila dunha táboa. Hai varias clases de claves

- *Crave candidata:* Cada un dos conxuntos mínimos de atributos que identifiquen sen ambigüidade e de forma única cada unha das tuplas dunha relación.
- *Clave primaria ou principal:* De entre todas as claves candidatas dunha relación, na definición do esquema deberase especificar cal delas se considera como identificador primario. O resto das claves candidatas denominaranse *claves alternativas*.
- *Claves foráneas* ou *claves alleas* son o conxunto de atributos dunha relación que se corresponden coa clave primaria doutra relación do modelo. Proporcionanlle ao modelo relacional os mecanismos axeitados para representar as (inter)relacións existentes entre os obxectos do problema.
  - Pode referenciar a clave primaria da mesma táboa (relacións reflexivas)
  - Debe ter sempre un valor correspondente na táboa onde é clave primaria
  - Debe estar formada por toda a clave primaria e non só por unha parte dela
  - Pode ter nulos
  - Pode ter valores duplicados
  - Unha táboa pode conter múltiples claves foráneas, onde cada unha representa a relación con outra táboa

#### **20.4.2      *Restricións***

No modelo relacional existen restricións, é dicir, estruturas ou ocorrencias non permitidas, sendo preciso distinguir entre restricións inherentes (propias do modelo) e restricións semánticas (de usuario).

##### **20.4.2.1      *Restricións inherentes***

1. Non se define ningunha orde nos elementos que forman unha relación, nin no sentido horizontal (tuplas) nin no vertical (atributos). A orde é sempre irrelevante.



2. En toda relación é obrigatoria a existencia da clave primaria, e polo tanto non pode haber dúas tuplas iguais.
3. Cada atributo dunha tupla só pode tomar un único valor do dominio sobre o cal está definido.
4. Regra de integridade de clave ou entidade: ningún dos atributos que forman parte dunha clave primaria dunha relación pode tomar un valor nulo para ningunha tupla desa relación.

#### 20.4.2.2 Restricións semánticas

1. Declaración de clave primaria (PRIMARY KEY): Permite declarar un atributo ou un conxunto de atributos como clave primaria dunha relación, polo que os seus valores non poderán repetir nin admitirán nulos.
2. Unicidade (UNIQUE): indica que os valores dun atributo (ou conxunto) non se poden repetir nunha relación. Esta restrición permite definir claves candidatas.
3. Obrigatoriedade (NOT NULL), indica que un atributo (ou conxunto) non admite nulos
4. Integridade referencial (restrición de clave allea): permiten que as claves foráneas dunha relación referencien unha tupla válida da relación pai. O usuario pode especificar, na definición do esquema relacional, as operacións que deben levarse a cabo cando se produce o borrado ou modificación dunha tupla na relación pai. As posibilidades son:
  - Borrado/modificación en fervenza (CASCADE). O borrado ou modificación dunha tupla na relación pai, provoca o borrado ou modificación de todas as tuplas relacionadas na relación filla.
  - Borrado / modificación restrinxido (NON ACTION). Se existen tuplas relacionadas na relación filla, non se permite o borrado ou modificación das tuplas da relación pai.



- Borrado / modificación con posta a nulos (SET NULL). Pon a nulo os valores de todos os atributos que conforman a clave allea na relación filla. Só está permitido cando eses valores se poidan poñer a nulo.
  - Borrado / modificación con posta a un valor por defecto (SET DEFAULT). Similar ao anterior, pero os atributos que conforman a clave allea na relación filla póñense a un valor especificado previamente na definición do esquema.
5. Restricións de rexeitamento. Na definición do esquema relacional poden impoñerse outra serie de restricións que garantan a integridade do modelo e, polo tanto, da información almacenada na base de datos. Estas restricións deben ser verificadas en toda operación de actualización para que o novo estado constitúa unha ocorrencia válida do esquema; en caso de que a operación intente violar a condición impídese que a operación se leve a cabo, como son:
- Restricións de verificación (CHECK). Especifican condicións que deben cumprir os valores de determinados atributos dunha relación, como poden ser os atributos de existencia obrigatoria (NOT NULL).
  - Asercións (ASSERTION). Permiten especificar condicións entre os elementos de distintas relacións do esquema.
  - Disparadores (TRIGGER). Permiten especificar condicións e accións que se leven a cabo cando se efectúe unha acción determinada sobre algunha relación do esquema



## 20.5 DIAGRAMAS ENTIDADE-RELACIÓN

Diciamos que o interese dos modelos de datos é captar tanta semántica como sexa posible do mundo real. Co que vimos ata o momento comprobamos que se permite establecer calquera número de relacións diferentes entre tipos de entidade pero non podemos establecer restricións do tipo:

- Un médico só traballa nun hospital
- Nun hospital traballan n médicos
- Todos os socios do videoclub alugaron polo menos unha película

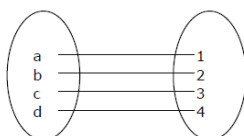
A continuación analízanse en detalle todos os aspectos das relacións que nos permitirán captar toda a semántica desexada.

### 20.5.1 Tipo de correspondencia

Denomínase tipo de correspondencia ao tipo de asociación que se establece entre as entidades relacionadas. Concretamente, pódese definir o tipo de correspondencia como o número máximo de ocorrencias dunha entidade asociada a unha ocorrencia doutra ou da mesma entidade a través dunha relación.

Para unha relación binaria, é dicir, de grao dous, entre as entidades A e B, existen tres tipos posibles de correspondencias:

- **Correspondencia 1:1** Una ocorrencia da entidade A asóciase como máximo cunha ocorrencia da entidade B e viceversa, como se pode observar na figura

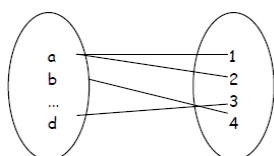


Entidade A      Entidade B



Un exemplo deste tipo de correspondencia pode ser que un cliente ten unha única conta bancaria nunha sucursal determinada e unha conta determinada dunha sucursal pertence a un único cliente.

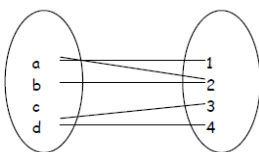
- **Correspondencia 1:N** Unha ocorrencia da entidade A asóciase cun número indeterminado de ocorrencias da entidade B, pero unha ocorrencia da entidade B asóciase como máximo cunha ocorrencia da entidade A. Se fose ao revés a correspondencia sería N:1.



Entidade A      Entidade B

Un exemplo deste tipo de correspondencia pode ser que unha persoa vive nunha cidade e nunha cidade viven moitas persoas.

- **Correspondencia N:M** Unha ocorrencia da entidade A asóciase cun número indeterminado de ocorrencias da entidade B e viceversa.



Entidade A      Entidade B

Un exemplo deste tipo de cardinalidade pode ser que un provedor subministra varios produtos e cada produto pode ser subministrado por varios provedores.



### 20.5.2 Entidades débiles

O concepto de entidade débil está directamente relacionado coas restricións de tipo semántico do modelo E/R e, máis concretamente, coa denominada restrición de existencia.

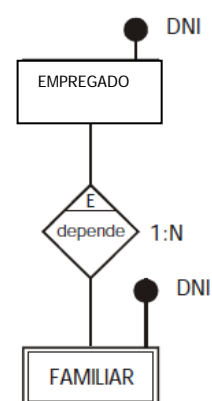
Esta restrición establece o feito de que a existencia dunha entidade non ten sentido sen a existencia doutra, é dicir, unha entidade ten dependencia de existencia doutra cando sen a primeira a segunda carecería de sentido.

Isto leva consigo que a desaparición das ocorrencias da entidade da cal depende a súa existencia leve á desaparición das ocorrencias da entidade débil que dependan delas. Por exemplo, un exemplar da entidade EDICIÓN non existiría se non houbo un exemplar correspondente na entidade LIBRO. As entidades débiles represéntanse graficamente por dous rectángulos concéntricos e no interior o nome da entidade.

ENTIDADE DÉBIL

Hai dous tipos de dependencias das entidades débiles respecto ás entidades regulares:

*Dependencia en existencia* os exemplares da entidade débil non poden existir se desaparece o exemplar da entidade regular co que están relacionados, pero a entidade débil pode ser identificada sen necesidade de identificar a entidade forte relacionada, é dicir, a entidade débil ten un atributo identificador clave.



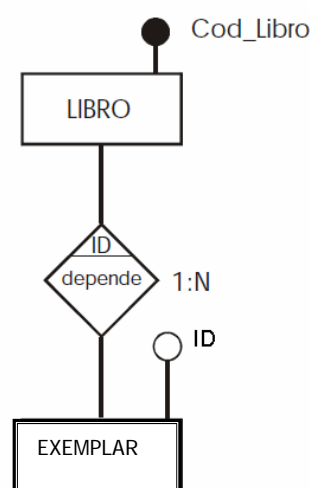
No exemplo é evidente que se desaparece un empregado da base de datos a existencia dos seus familiares carece de sentido, é dicir, a entidade FAMILIAR ten dependencia de existencia respecto



da entidade EMPREGADO. Non obstante, cada unha das ocorrencias da entidade familiar pode identificarse por si mesma.

*Por identificación* a entidade débil non ten sentido en si mesma e non pode ser identificada sen a entidade forte relacionada, é dicir non ten un atributo identificador clave senón tan só un descriptor discriminador e necesita o atributo clave da entidade forte para poder identificar de xeito único as súas ocorrencias de entidade.

No exemplo, o atributo identificador crave será Cod\_Libro (como clave da entidade forte LIBRO) máis ID como discriminador da entidade EXEMPLAR.



Como conclusión ao concepto de entidade débil convén resaltar as circunstancias seguintes:

1. A dependencia en existencia non implica unha dependencia en identificación, feito que se sucede no caso inverso pois unha entidade que depende doutra polo seu atributo clave non terá sentido sen a existencia desta última.
2. Nunha interrelación con cardinalidade N:M nunca haberá entidades débiles. A razón é que a suposta ocorrencia da entidade débil que se tivese que borrar podería estar asociada a máis dunha ocorrencia da suposta entidade forte, o que implicaría a imposibilidade do seu borrado, feito este en clara contraposición coa definición de entidade débil.

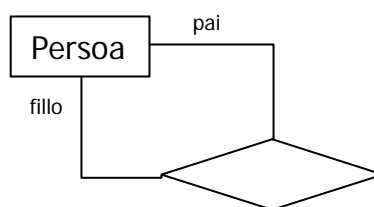


### 20.5.3 *Papel ou rol*

É a función que cada unha das entidades realiza nunha interrelación concreta. Graficamente represéntase indicando o nome do rol na liña que une as entidades coas relacións.

Os roles xogan un papel especialmente importante en relacións reflexivas onde é necesario coñecer os dous roles que o mesmo tipo de entidade xoga nunha determinada relación.

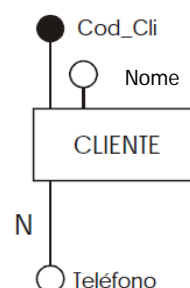
A razón está en que estamos a asociar entre si ocorrencias dunha mesma entidade de forma que cada unha delas ten un significado diferente. No exemplo, unha ocorrencia de PERSOAS fará papel de 'pai' e a outra papel de 'fillo'.



### 20.5.4 *Atributos multivaluados e compostos*

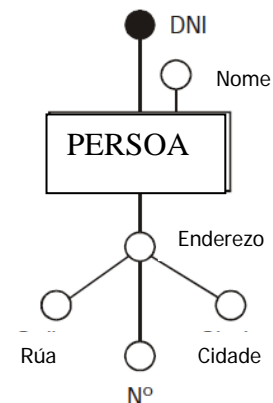
Un último tipo de restricións que se deben ter en conta á hora de realizar o deseño conceptual dunha base de datos co modelo E/R son as que afectan á tipoloxía dos diferentes atributos. Dende este punto de vista podemos definir dous tipos diferentes de atributos respecto aos manexados ata o momento, que son os seguintes:

**Atributos multivaluados.** Son aqueles atributos que para unha mesma ocorrencia da entidade toman máis dun valor. Por exemplo, se cada cliente pode ter máis dun teléfono e é de interese gardar todos os seus posibles valores, o atributo teléfono sería multivaluado. Represéntase etiquetando o seu arco cun valor de cardinalidade N.





**Atributos Compostos.** Son aqueles que agrupan en si mesmos, por afinidade ou por forma de uso, máis dun atributo. Por exemplo o atributo "endereço" engloba os atributos rúa, número, cidade, provincia e código postal.



Represéntase especificando os seus atributos compoñentes rodeando a este e enlazándoos ao símbolo do atributo composto mediante arcos.

#### **20.5.5** *Atributos derivados*

Son aqueles que se poden calcular a partir doutros. Por exemplo, se temos a entidade PERSOA cos atributos DNI, Nome, Data\_nacemento e Idade, o último atributo (idade) pode obterse a partir doutro atributo (a data de nacemento) e é, polo tanto, redundante. Este tipo de atributos deben eliminarse do 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 Actas do congreso, 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

Colexiado do CPEIG





# **21. SISTEMAS DE XESTIÓN DE BASES DE DATOS. MODELO RELACIONAL. NORMALIZACIÓN. SQL. LINGUAXE DE DEFINICIÓN DE DATOS (DDL), LINGUAXE DE MANIPULACIÓN DE DATOS (DML) E LINGUAXE DE CONTROL DE DATOS (DCL).**



**Tema 21: Sistemas de xestión de bases de datos. Modelo relacional. Normalización. SQL. Linguaxe de definición de datos (DDL), linguaxe de manipulación de datos (DML) e linguaxe de control de datos (DCL).**

---

## **ÍNDICE**

### **21.1 Sistemas de xestión de bases de datos (SXBD)**

#### *21.1.1 Introducción*

#### *21.1.2 SXBD obxectivo e características*

#### *21.1.3 Evolución*

##### *21.1.3.1 Arquitectura en 2 capas*

##### *21.1.3.2 Arquitectura en 3 capas*

#### *21.1.4 Modelo de referencia ANSI*

##### *21.1.4.1 Obxectivos e beneficio*

##### *21.1.4.2 Niveis de descrición de datos*

##### *21.1.4.3 Contorno*

##### *21.1.4.4 Compoñentes dun SXBD*

##### *21.1.4.5 Modelos de datos*

#### *21.1.5 Estrutura xeral dun SXBD*

#### *21.1.6 SXBD relacionais (SXBD-R)*

##### *21.1.6.1 Características dos SXBD-R*

### **21.2 MODELO RELACIONAL**

#### *21.2.1 Elementos*

#### *21.2.2 Restricións*

### **21.3 NORMALIZACIÓN**

### **21.4 SQL**

#### *21.4.1 Partes da linguaxe SQL*

#### *21.4.2 Modos de traballo con SQL*

### **21.5 Linguaxe e definición de datos (DDL)**

#### *21.5.1 Obxectos da base de datos*

#### *21.5.2 Xestión de táboas*





*21.5.3 Xestión de vistas*

*21.5.4 Xestión de índices*

## **21.6 Linguaxe de manipulación de datos (DML)**

*21.6.1 Inserción de valores nunha táboa*

*21.6.2 Borrado de valores dunha táboa*

*21.6.3 Modificación de valores dunha táboa*

*21.6.4 Consulta de datos*

*21.6.5 Consultas sobre múltiples táboas.*

*21.6.6 Subconsultas*

*21.6.7 Operacións con consultas*

## **21.7 . Linguaxe de control de datos DCL**

*21.7.1 Seguridade*

*21.7.2 Transaccións*

## **21.8 Bibliografía**



## **21.1 SISTEMAS DE XESTIÓN DE BASES DE DATOS (SXBD)**

### **21.1.1 Introducción**

Para que unha base de datos funcione correctamente precisa dun software que xestione todas as súas operacións e que ademais proporcione unha interface de comunicación para os usuarios facilitando o acceso aos datos contidos nela. Este tipo de software son os denominados *sistemas de xestión de bases de datos* ou *SXBD*.

Segundo unha definición formal, un sistema de xestión de bases de datos *"é un conxunto coordinado de programas, procedementos, linguaxes, etc... que lles subministra, tanto aos usuarios non informáticos, como aos analistas programadores, ou ao administrador, os medios necesarios para describir e manipular os datos integrados na base, mantendo a súa integridade, confidencialidade e seguridade"*.

Este grupo definido de software encargado de ofrecer soporte ás bases de datos proporciona unha serie de facilidades para o manexo das bases de datos. Estas facilidades tradúcense nunha serie de linguaxes que permiten operar contra as distintas bases de datos que soporta o SXBD, sendo as principais:

- **Linguaxe de definición de datos ou DDL** (Data Definition Language): permite definir os esquemas conceptuais dunha base de datos.
- **Linguaxe de manipulación de datos ou DML** (Data Manipulation Language): subministra operacións para a realización de consultas e actualizacións da información contida nas bases de datos.
- **Linguaxe de control de datos ou DCL** (Data Control Language): permite administrar e controlar o acceso aos datos existentes nunha base de datos.

Dende os anos 70, o grupo ANSI/X3/SPARC é o encargado de ocuparse da normalización dos SXBD, e en 1975 publicou un informe provisional onde propón unha arquitectura de 3 capas para os SXBD, informe que posteriormente se revisaría e



detallaría en 1977. Non obstante, ata 1985 non se presentou o *Modelo de referencia* para a estandarización dos SXBD (Modelo ANSI).

### **21.1.2      *SXBD obxectivo e características***

Para un sistema xestor de bases de datos o seu obxectivo principal é o de ofrecer un ámbito idóneo para a extracción, manipulación e almacenamento da información das bases de datos. Os SXBD realizan unha xestión centralizada de todas as peticións de acceso funcionando de interface entre os usuarios e a base de datos. Tamén é o xestor da estrutura física dos datos e do seu almacenamento, polo que, en definitiva, libera os usuarios de coñecer a organización física exacta dos datos, así como de crear os algoritmos de almacenamento, actualización ou consulta da información contida na base de datos.

Non todos os SXBD son iguais nin teñen as mesmas funcionalidades, posto que dependen de cada produto e do modelo de base de datos que xestionen. Independentemente disto, existen unha serie de características que se poderían identificar como comúns a todos eles, e que foron definidas por Codd e revisadas con posterioridade, a medida que as novas necesidades se foron integrando. As características necesarias para que un SXBD poida cubrir as necesidades dun usuario son:

- Co fin de simplificar o mantemento das aplicacións que fan uso das bases de datos, un SXBD debe manter a independencia entre as solucións software e a estrutura da base de datos. Esta independencia non é completa pero cada vez aproxímanse máis a esta esixencia.
- Na medida do posible, non debe existir redundancia de datos, é dicir, estes non deben de estar almacenados varias veces. Con iso conséguese asegurar a coherencia dos datos.
- Un SXBD ofrece as ferramentas necesarias a un usuario para:
  - Almacenar datos.



- Acceder á información.
- Actualizar os datos.

Estas ferramentas deben proporcionar estes servizos de tal maneira que lle resulte transparente ao usuario.

- Permite o acceso de múltiples usuarios á mesma base de datos e no mesmo momento. Cando isto se produce, se algún dos usuarios está a realizar operacións de actualización dos datos, o SXBD ha de asegurarse de realizar unha correcta xestión da concorrencia evitando a corrupción dos datos ou que estes se volvan inconsistentes. Para realizar esta xestión o SXBD fai un correcto uso dos bloqueos das bases de datos.
- Ofrécelles aos usuarios un catálogo ao cal poden acceder onde se almacenan dun xeito centralizado as descrições dos datos. Este servizo é o que permite a eliminación e detección das redundancias de datos e denomínase *dicionario de datos*.
- Realiza as transaccións garantindo que as actualizacións correspondentes a unha transacción se realizen, ou en caso de que non sexan posible realizar algunha, ningunha terá efecto. Isto débese a que, durante unha transacción, se producen accións que cambian o contido da base de datos. Se por algún motivo a transacción falla, daquela a base de datos pasa a un estado de inconsistencia, posto que non todos os cambios da transacción se produciron, obrigando o SXBD a desfacer os cambios para que a base de datos volva a un estado consistente.
- Garante a recuperación das bases de datos. Se acontece algún problema que provoque que a información se vexa afectada, un fallo de hardware ou software que fagan abortar o SXBD, ou que un usuario interrompa unha operación antes de que se finalice a transacción, o SXBD debe proporcionar os mecanismos necesarios para solucionar este tipo de situacións e recuperar a base de datos a un estado consistente.



- Proporcionálles seguridade ás bases de datos, é dicir, restrinxe mediante diferentes niveis, que o acceso ás bases de datos só o realizen usuarios autorizados, protexendo as bases de accesos non autorizados, xa sexan accidentais ou non.
- Un SXBD garante a integridade das bases de datos. Proporciona un conxunto de regras que a base de datos non pode violar conseguindo así a validez e consistencia dos datos.
- Proporciona ferramentas para a administración das bases de datos. Este conxunto de ferramentas permiten unha serie de funcionalidades:
  - Importación e extracción de datos
  - Monitorización do funcionamento e obtención de datos estatísticos sobre o uso das bases de datos.
  - Reorganización de índices
  - Optimización do espazo liberado para a súa reutilización.
- Mantén unha dispoñibilidade continua, garantindo que en todo momento as bases de datos están accesibles. Proporciona que as tarefas de administración, xestión e mantemento se poidan levar a cabo sen interromper o correcto funcionamento das bases de datos.
- Todo SXBD intégrase cun xestor de comunicacións, software encargado de xestionar o acceso dos usuarios que realicen a conexión coa máquina que serve de soporte ao SXBD dun xeito remoto a través dunha rede de datos. O xestor de comunicacións non forma parte dun SXBD pero si cómpre que o SXBD se integre con el.
- Posúe un DDL, linguaxe de definición de datos, para a creación e modificación das bases de datos.



- Posúe un DML, linguaxe de manipulación de datos, para a inserción, manipulación e consulta da información contida nas bases de datos.
- Posúe un DCL, linguaxe de control de datos, para controlar o acceso á información das bases de datos.
- Garante a escalabilidade e elevada capacidade de proceso, é dicir, é capaz de aproveitar os recursos da máquina dispoñibles, aumentando a súa capacidade de procesado a medida que dispoña de recurso.
- É capaz de almacenar enormes cantidades de datos sen que o usuario perciba unha degradación no rendemento do sistema.

### **21.1.3      *Evolución***

Nos primeiros inicios da informática os datos formaban parte dos programas, integrados como constantes. Posteriormente, coa aparición dos ficheiros como colección de datos homoxénea, empézase a diferenciar a estrutura lóxica que representa o punto de vista do usuario e a estrutura física dos datos.

Esta diferenciación faise máis evidente coa aparición nos sistemas operativos de subsistemas de xestión de datos, pero non resulta suficiente para romper a dependencia entre os datos e os programas e viceversa, e de ambos os dous con respecto á máquina.

Para limar estas dependencias existentes entre os datos e as aplicacións, comézanse a utilizar arquitecturas que diferencian a estrutura lóxica dos datos, representación dos datos orientados cara ao problema, da estrutura física, representación orientada cara á máquina. Esta diferenciación funciona a través dunha transformación ou *mapeamento* que fai as tarefas de conversión entre unha estrutura e outra.

#### **21.1.3.1      Arquitectura en 2 capas**

Coa aparición dos primeiros SXBD nos anos 60, os sistemas pasan dunha orientación centrada no proceso a unha orientación enfocada cara ás bases de datos.



Isto produce que os datos e as relacións entre estes se sitúen na bases de datos, conseguindo illalos das aplicacións. Esta evolución provoca un cambio nas tendencias das estruturas de datos facendo que a estrutura lóxica sexa máis flexible e sinxela, mentres que a estrutura física se volve máis complexa co fin de mellorar o rendemento.

Os primeiros SXBD facilitan en boa medida a descrición e o almacenamento das relacións permitidas, caracterizando ao mesmo tempo os distintos modelos de datos: xerárquico, rede, relacional.

A arquitectura seguida por estes SXBD estaba definida en dous niveis:

- *Estrutura global*, coa características lóxicas e físicas: esquema
- *Vistas lóxicas externas* dos usuarios: Subesquemas

#### 21.1.3.2      *Arquitectura en 3 capas*

A organización ANSI (American National Standards Institute) publica no ano 1975 un informe que resultaría clave para o desenvolvemento dos SXBD. Neste informe indícase a necesidade de evolucionar os SXBD co fin de conseguir unha total independencia entre os datos e as aplicacións. Para tal propósito propón un modelo arquitectónico en 3 capas e define á súa vez o modelo conceptual para conseguilo.

Neste informe a estrutura global do modelo de 2 niveis divídese dando lugar a dúas estruturas, nunha quedan os aspectos lóxicos e o esquema conceptual, mentres que a outra queda cos aspectos físicos ou esquema interno.

Mediante esta definición, os SXBD que seguen esta normativa mostran internamente as tres capas perfectamente diferenciados:

- **Nivel interno ou físico:** representa o nivel máis baixo de abstracción e neste nivel é onde se describe en detalle a estrutura física da base de datos, dispositivos de almacenamento físico, estratexias de acceso, índices, etc.



Para iso interactúa co sistema operativo e co xestor de ficheiros. En definitiva o esquema interno especifica que datos son almacenados e como, ademais de describir a estrutura da base de datos en forma de modelo conceptual de almacenamento.

- **Nivel conceptual:** correspóndese co nivel intermedio de abstracción e describe os datos que son almacenados na base de datos e as relacións existentes entre eles. Tamén describe a base de datos segundo a súa estrutura de deseño. Neste nivel a base de datos resulta unha colección de rexistros lóxicos sen descritores de almacenamento. Mediante este nivel conséguese o illamento da representación da información dos requirimentos da máquina e das esixencias dos usuarios.
- **Nivel externo ou lóxico:** supón o de maior grao de abstracción e contén as vistas externas da base de datos asociadas a un esquema externo. Proporcionalle a cada tipo de usuario unicamente a parte do esquema que resulta relevante para el e para a cal ten acceso. Cada base de datos pode ter tantas vistas como necesite.

Con esta arquitectura preténdese conseguir que o esquema conceptual sexa unha descrición estable e independente do nivel superior e do inferior, é dicir, independente tanto das vistas coma do almacenamento dos datos. Coa consecución desta independencia as bases de datos convértense en sistemas máis flexibles e adaptables.

#### **21.1.4 Estrutura Xeral dun SXBD**

Os principais módulos do SXBD son:

- **O compilador da DDL.** Comproba a sintaxe das sentenzas da DDL e actualiza as táboas do dicionario de datos ou catálogo que conteñen os metadatos.
- **O precompilador da DML.** Converte as sentenzas da DML embebidas na linguaxe anfitrión en sentenzas listas para o seu procesamento por parte do





compilador de linguaxe anfitrión e ademais extrae esas sentenzas DML para que poidan ser procesadas de forma independente polo compilador da DML.

- ***O compilador da DML.*** Comproba a sintaxe das sentenzas da DML e pásallas ao procesador de consultas.
- ***O procesador de consultas.*** Realiza a transformación das consultas nun conxunto de instrucións de baixo nivel que se dirixen ao xestor da base de datos.
- ***O xestor da base de datos.*** Serve de interface para os programas de aplicación e as consultas dos usuarios. O xestor da base de datos acepta consultas e examina os esquemas externo e conceptual para determinar que rexistros se requiren para satisfacer a petición. Daquela o xestor da base de datos realiza unha chamada ao xestor de ficheiros para executar a petición.

Os principais compoñentes do xestor da base de datos son os seguintes:

- ***O xestor de transaccións.*** Realiza o procesamento das transaccións.
- ***O xestor de buffers.*** Transfire os datos entre memoria principal e os dispositivos de almacenamento secundario.
- ***O xestor de ficheiros.*** Xestiona os ficheiros en disco onde se almacena a base de datos. Este xestor establece e mantén a lista de estruturas e índices definidos no esquema interno. Para acceder aos datos pasa a petición aos métodos de acceso do sistema operativo que se encargan de ler ou escribir nos ficheiros físicos que almacenan a información da base de datos.

No esquema proposto reflíctense distintos bloques nos que se indican:

- Tipos de usuarios que poden acceder ao SXBD
- Métodos utilizados polos usuarios para acceder á información.



- SXBD que se divide en:
  - O primeiro subsistema é o encargado de recibir as peticións e dirixilas ao xestor da base de datos ou ao dicionario de datos.
  - O segundo é o xestor da base de datos, que posúe un xestor de transaccións, un xestor de *búfer* e o xestor de ficheiros.
  - A base de datos cos seus índices e o dicionario de datos.

### **21.1.5      *SXBD relacionais (SXBD-R)***

Os sistemas xestores de base de datos relacionais están baseados no ***modelo relacional*** o cal intenta representar o universo do discurso mediante a álgebra relacional e as súas principais características son:

Está baseado nun modelo matemático cun conxunto de regras e algoritmos establecidos, permitindo que se desenvolvan linguaxes de acceso e manipulación moi potentes e fiables.

A estruturación dos datos realízase mediante relacións que son modeladas utilizando táboas bidimensionais que representan as entidades como as súas relacións.

Establece regras de integridade que posibilitan a incorporación de aspectos semánticos e o traslado de restricións ou comportamentos dos datos ao esquema conceptual, que, doutra forma, non se podería modelar só coas táboas.

#### **21.1.5.1      Características dos SXBD-R**

As súas tres principais características son as estruturas de datos, os operadores asociados e os aspectos semánticos.

##### **21.1.5.1.1      Estruturas de datos: relacións e claves**

Elementos:



- *Relación*: subconxunto dun produto cartesiano entre conxuntos de atributos que no modelo relacional se mostra como unha táboa con  $m$  filas e  $n$  columnas.
- *Atributo*: representan as columnas dunha táboa e correspóndense coas propiedades das entidades. Estes atributos encóntranse limitados por un dominio que especifica o rango de valores que poden tomar podendo ser compartido por varios atributos.
- *Dominio*: rango de valores que un atributo pode adoptar. Este rango é dependente do tipo de atributo e os valores do dominio han de ser homoxéneos.
- *Tuplas*: nome que se lle asocia a cada unha das filas dunha táboa que se corresponden con cada unha das ocorrencias da relación que se representa na táboa. A súa orde non é relevante.
- *Cardinalidade da relación*: número de tuplas dunha relación.
- *Grao da relación*: número de atributos dunha relación.

Dentro dos elementos que conforman a estrutura de datos, os máis importantes son as relacións, das que as características máis importantes son:

- Todas as tuplas dunha relación están formadas polo mesmo número, tipo de atributos e na mesma orde.
- A orde das tuplas carece de relevancia.
- En cada atributo dunha tupla só pode aparecer un valor que ademais debe pertencer ao dominio correspondente.
- Non poden existir dúas tuplas iguais na mesma relación. Isto provoca que exista un ou varios atributos que sirvan para distinguir unhas tuplas doutras denominados *claves candidatas*.



Algunha destas claves candidatas son seleccionadas polo administrador ou deseñador da base de datos para a identificación de tuplas; daquela, a clave denomínase *clave primaria* e non pode adoptar nunca o valor nulo. O resto de claves candidatas que non son seleccionadas como primarias denomínanse *claves alternativas ou secundarias*.

Ademais unha relación pode incluír dentro dos seus atributos a clave primaria doutra relación, pasando esta a ser *clave foránea* da primeira relación.

#### **21.1.5.1.2 Operadores asociados**

A álgebra coa que se move o modelo relacional está formada por un conxunto de operadores asociados e é completa, é dicir, garante matematicamente que con ela se pode realizar calquera acceso á base de datos.

Os operadores utilizan as relacións do modelo como operandos. Os operadores máis importantes móstranse a continuación:

- **Unión.** A unión de dúas relacións "A" e "B" produce o conxunto de tuplas formado polas tuplas de "A" e as tuplas de "B". Só é aplicable a relacións co mesmo grao e cos mesmos atributos.
- **Diferenza.** A diferenza entre dúas relacións "A" e "B" é o conxunto de tuplas da relación A que non están en "B". Só é aplicable a relacións co mesmo grao e cos mesmos atributos.
- **Produto cartesiano.** O produto cartesiano de dúas relacións "A" de grao m e "B" de grao n, está formado polo conxunto de todas as posibles tuplas de m+n atributos cos m primeiros valores de "A" e os n restantes de "B".
- **Proxección.** Considerando "x" un subconxunto de atributos da relación "A", a *proxección* do atributo "x" sobre a relación "A" é a relación formada polos atributos de "A" correspondentes cos do subconxunto "x".
- **Selección.** Se "F" resulta unha fórmula que está composta por operadores lóxicos, aritméticos e de comparación e se os operandos se corresponden



con valores dos atributos dunha relación "A", daquela a selección de "F" sobre "A" é o conxunto resultante formado polas tuplas de "A" que cumpren a condición establecida por "F".

Partindo deste conxunto de operadores pódense xerar outros derivados como a intersección, o cociente ou a unión natural.

#### **21.1.5.1.3 Aspectos semánticos**

Cando unha característica do ámbito ou do universo do discurso non se pode modelar mediante a definición dunha relación, esta debe definirse mediante un nivel de descrición superior pasando a formar parte dos aspectos semánticos. Estes aspectos, dende un punto de vista práctico, son restricións que se engaden ás propias do modelo relacional e que o seu propósito é o de garantir a integridade e validez dos datos. Á súa vez tamén proporcionan un maior grao de información ao esquema lóxico de datos.

Dentro deste conxunto de restricións pódense identificar dous grupos:

- *Restricións de usuario.* Son restricións que se aplican aos valores pertencentes ao domino dos atributos, como por exemplo limitar os meses a 12 e os días a 31 nun atributo data.
- *Integridade referencial.* As restricións pertencentes á integridade referencial ocúpanse do mantemento das referencias existentes entre as propias relacións.

Para manter a integridade referencial, cando se realiza algunha tarefa de borrado ou modificación das tuplas débese realizar algunha das seguintes accións:

- Impedir a operación, para asegurarse de que unha vez establecida a relación entre dúas tuplas de distintas táboas non se poida desfacer.



- Transmitir en fervenza, é dicir, de se borrar ou modificar unha tupla, todas aquelas que fan referencia a ela teñen tamén que se borrar ou modificar.
- Poner a nulo, manter a integridade asignando o valor nulo ao atributo que realiza as tarefas de clave foránea.
- Establecer valor por omisión ou lanzar un procedemento de usuario que o estableza.

## **21.2    *MODELO RELACIONAL***

É un modelo lóxico de datos, desenvolvido por Codd, que introduciu a teoría matemática das relacións no campo das BD e supuxo un importante paso na investigación dos SXBD. O documento de Codd propón un modelo de datos baseado na "teoría das relacións", onde os datos se estruturan lóxicamente en forma de relacións —táboas—, e onde un obxectivo fundamental do modelo é manter a independencia desta estrutura lóxica respecto ao modo de almacenamento e a outras características de tipo físico (independencia de ordenación, de indexación e dos camiños de acceso).

Este novo modelo de datos perseguía os seguintes obxectivos:

- ☐ Independencia lóxica: engadir, eliminar ou modificar calquera elemento da BD non debe repercutir nos programas e/ou usuarios que accedan ás vistas deles.
- ☐ Independencia física: o modo en que se almacenan os datos non debe influír na súa manipulación lóxica e, polo tanto, os usuarios que acceden a eses datos non deben modificar os seus programas por cambios no almacenamento físico.
- ☐ Flexibilidade: poder ofrecer a cada usuario os datos da forma máis axeitada á súa aplicación.



- Uniformidade: As estruturas lóxicas dos datos presentan un aspecto uniforme (táboas), o que facilita a concepción e manipulación da BD por parte dos usuarios.
- Sinxeleza: As características anteriores, así como unhas linguaxes de usuario moi sinxelas, producen como resultado que o modelo de datos relacional sexa doado de comprender e utilizar por parte do usuario final.

### 21.2.1 *Elementos*

O modelo relacional introduce a súa propia terminoloxía para denominar os obxectos e elementos utilizados:

1. **Relación.** É o elemento central do modelo relacional. Son matrices bidimensionais (táboas) caracterizadas por un nome, un conxunto de atributos (dimensión vertical da táboa = columnas) e un conxunto de tuplas (dimensión horizontal = filas).

Cada tupla está formada polo conxunto de valores que toma cada un dos atributos para un elemento da relación.

Nas relacións podemos falar de dous compoñentes:

- Intensión é a parte definitoria e estática da relación. Define a estrutura abstracta de datos e as restricións de integridade dela. É o que chamaremos *esquema de relación*.
- Extensión é o conxunto de tuplas que satisfai o esquema de relación nun instante dado e que se encontran almacenadas na base de datos. Varía co transcurso do tempo.

O número de tuplas dunha relación nun instante dado denomínase **cardinalidade** da relación, e normalmente varía co transcurso do tempo. O número de columnas ou atributos denomínase **grao** da relación.



2. **Dominio.** É o conxunto definido, finito e homoxéneo dos valores atómicos posibles dun determinado atributo.

Cada atributo está ligado a un determinado dominio e representa o uso dun dominio para unha determinada relación.

Os dominios poden estar definidos por intensión (conxunto definido mediante unha serie de regras abstractas) ou por extensión (conxunto finito de valores posibles).

3. **Claves dunha relación.** Unha clave é unha(s) columna(s) cuns valores que identifican unha única fila dunha táboa. Hai varias clases de claves

- *Crave candidata:* Cada un dos conxuntos mínimos de atributos que identifiquen sen ambigüidade e de forma única cada unha das tuplas dunha relación.
- *Clave primaria ou principal:* De entre todas as claves candidatas dunha relación, na definición do esquema deberase especificar cal delas se considera como identificador primario. O resto das claves candidatas denominarase *claves alternativas*.
- *Claves foráneas ou claves alleas:* son o conxunto de atributos dunha relación que se corresponden coa clave primaria doutra relación do modelo. Proporcionanlle ao modelo relacional os mecanismos axeitados para representar as (inter)relacións existentes entre os obxectos do problema.
  - Pode referenciar a clave primaria da mesma táboa (relacións reflexivas)
  - Debe ter sempre un valor correspondente na táboa onde é clave primaria
  - Debe estar formada por toda a clave primaria e non só por unha parte dela
  - Pode ter nulos
  - Pode ter valores duplicados



- Unha táboa pode conter múltiples claves foráneas, onde cada unha representa a relación con outra táboa

### 21.2.2 *Restricións*

No modelo relacional existen restricións, é dicir, estruturas ou ocorrencias non permitidas, sendo preciso distinguir entre restricións inherentes (propias do modelo) e restricións semánticas (de usuario).

#### Restricións inherentes

1. Non se define ningunha orde nos elementos que forman unha relación, nin no sentido horizontal (tuplas) nin no vertical (atributos). A orde é sempre irrelevante.
2. En toda relación é obrigatoria a existencia da clave primaria e, polo tanto, non pode haber dúas tuplas iguais.
3. Cada atributo dunha tupla só pode tomar un único valor do dominio sobre o cal está definido.
4. Regra de integridade de clave ou entidade: ningún dos atributos que forman parte dunha clave primaria dunha relación pode tomar un valor nulo para ningunha tupla desa relación.

#### Restricións semánticas

Declaración de clave primaria (PRIMARY KEY): Permite declarar un atributo ou un conxunto de atributos como clave primaria dunha relación, polo que os seus valores non se poderán repetir nin admitirán nulos.

Unicidade (UNIQUE): indica que os valores dun atributo (ou conxunto) non poden repetirse nunha relación. Esta restrición permite definir claves candidatas.

Obrigatoriedade (NOT NULL), indica que un atributo (ou conxunto) non admite nulos

Integridade referencial (restrición de clave allea): permiten que as claves foráneas dunha relación referencien unha tupla válida da relación pai. O



usuario pode especificar, na definición do esquema relacional, as operacións que se deben levar a cabo cando se produce o borrado ou modificación dunha tupla na relación pai. As posibilidades son:

- Borrado/modificación en fervenza (CASCADE). O borrado ou modificación dunha tupla na relación pai, provoca o borrado ou modificación de todas as tuplas relacionadas na relación filla.
- Borrado/modificación restrinxido (NON ACTION). Se existen tuplas relacionadas na relación filla, non se permite o borrado ou modificación das tuplas da relación pai.
- Borrado/modificación con posta a nulos (SET NULL). Pon a nulo os valores de todos os atributos que conforman a clave allea na relación filla. Só está permitido cando eses valores se poidan poñer a nulo.
- Borrado/modificación con posta a un valor por defecto (SET DEFAULT). Similar ao anterior, pero os atributos que conforman a clave allea na relación filla póñense a un valor especificado previamente na definición do esquema.

Restricións de rexeitamento. Na definición do esquema relacional poden impoñerse outra serie de restricións que garantan a integridade do modelo, e polo tanto, da información almacenada na base de datos. Estas restricións deben ser verificadas en toda operación de actualización para que o novo estado constitúa unha ocorrencia válida do esquema; en caso de que a operación intente violar a condición, impídese que a operación se leve a cabo, como son:

- Restricións de verificación (CHECK). Especifican condicións que deben cumprir os valores de determinados atributos dunha relación, como poden ser os atributos de existencia obrigatoria (NOT NULL).
- Asercións (ASSERTION). Permiten especificar condicións entre os elementos de distintas relacións do esquema.



- Disparadores (TRIGGER). Permiten especificar condicións e accións que se leven a cabo cando se efectúe unha acción determinada sobre algunha relación do esquema

### **21.3 NORMALIZACIÓN**

Ao estudar a estrutura do modelo relacional, dedúcese que a información da nosa base de datos se pode representar por medio dun conxunto de obxectos (relacións e dominios) e dun conxunto de regras de integridade.

No modelo relacional, como nos demais modelos de datos, o deseño dunha base de datos pódese abordar de dúas formas distintas:

- Obtendo o esquema relacional directamente a partir da observación do noso universo do discurso, de forma que plasmemos a nosa percepción deste nun conxunto de esquemas de relación que conterán os atributos e as restricións de integridade que representan os obxectos e as regras que podemos captar na nosa análise do mundo real.
- Realizando o proceso de deseño en dúas fases: na primeira lévase a cabo o deseño conceptual, por exemplo no modelo E/R, co que se obtén o correspondente esquema conceptual; na segunda, este transfórmase nun esquema relacional seguindo unhas determinadas regras de transformación.

Estas relacións que resultan da observación do mundo real ou da transformación ao modelo relacional do esquema E/R elaborado na etapa de modelado conceptual, poden presentar algúns problemas derivados de fallos na percepción do universo do discurso, no deseño do esquema E/R, ou no paso ao modelo relacional;

Entre estes problemas cabe destacar os seguintes:

- Incapacidade para almacenar certos feitos.



- Redundancias e, polo tanto, posibilidade de inconsistencias,
- Ambigüidades.
- Perda de información (aparición de tuplas repetidas).
- Perda de dependencias funcionais, é dicir, de certas restricións de integridade que dan lugar a interdependencias entre os datos.
- Existencia de valores nulos.
- Aparición, na base de datos, de estados que non son válidos no mundo real (anomalías de inserción, borrado e modificación).

En definitiva, o esquema relacional debe ser sempre analizado para comprobar que non presenta os problemas anteriormente citados, evitando a perda de información e a aparición de estados que non son válidos no mundo real.

Para evitar que se poidan dar estes problemas, existen unha serie de regras ou formas normais. Estas formas normais serán aplicadas normalmente a bases de datos xa implantadas en forma de relacións (táboas), o que nos permitirá pasar a outras relacións (táboas) que non dean os problemas anteriormente descritos.

Existen seis formas normais. As tres primeiras na maior parte dos casos son suficientes para normalizar os esquemas de relación.

### **1. Primeira forma normal (1FN).**

Dise que unha relación está en 1FN cando cada atributo só toma un valor do dominio simple subxacente. É dicir, cada atributo ten asociado un dominio do cal só toma un valor en cada tupla.

É unha restrición inherente ao modelo relacional, polo que o seu cumprimento é obrigatorio e afecta ao número de valores que poden tomar os atributos dunha relación.

### **2. Segunda forma normal (2FN).**

Dise que unha relación está en 2FN se:



- Está en 1FN.
- Cada atributo non principal da relación ten dependencia funcional completa respecto da clave primaria desa relación, isto é, o valor dos atributos non principais da relación vén determinado polo valor de todos os atributos da clave.

Por exemplo, a relación:

<b>Matrícula</b> ( <b>DNI</b> , <b>materia</b> , nome, apelidos, curso, nota, aula, lugar)
--

Non está en 2FN, posto que nome e apelidos dependen unicamente de DNI e non do valor de materia. Igualmente, curso depende unicamente de materia, e non do valor de DNI.

Para pasar a 2FN descomponse a relación noutras tres, da forma:

<b>Matrícula2</b> ( <b><u>DNI</u></b> , <b><u>materia</u></b> , nota, aula, lugar)
--

<b>Alumno</b> ( <b><u>DNI</u></b> , nome, apelidos)
---

<b>Materia</b> ( <b><u>materia</u></b> , curso)
---

### 3. Terceira forma normal (3FN).

Unha relación R satisfai a terceira forma normal (3FN), se e soamente se está en 2FN, e cada atributo non principal (atributo que non forma parte da clave) da relación non depende funcionalmente doutros atributos non principais desa relación. É dicir, non poden existir dependencias entre os atributos que non forman parte da clave primaria da relación R.

Se se considera, como é lóxico, que cada aula está situada fisicamente nun único lugar, pódese observar que a relación Matrícula2, a cal se encontra en 2FN, segue presentando problemas debidos a que existe unha dependencia entre os atributos aula e lugar (non se encontra, polo tanto, en 3FN),



Para eliminar os problemas que ocasiona na relación Matricula2 a existencia desta dependencia funcional, esta relación debe descompoñerse en dúas relacións, quedando o esquema da forma:

**Matrícula3** (DNI, materia, nota, aula)

**Situación** (aula, lugar)

**Alumno** (DNI, nome, apelidos)

**Materia** (materia, curso)

#### 4. Forma normal de Boyce-Codd (FNBC).

É unha redefinición máis estrita da 3FN, xa que esta presentaba certos problemas en relacións con varias claves candidatas compostas que se solapaban. Por iso en 1974, Boyce e Codd definiron a chamada forma normal que leva o seu nome (FNBC). Baséase no concepto de determinante funcional.

Chámase determinante funcional a un atributo ou a un conxunto de atributos dunha relación R do cal depende funcionalmente de forma completa algún outro atributo da mesma relación.

Unha relación R satisfai a forma normal de Boyce-Codd (FNBC) se e soamente se está en 1FN, e cada determinante funcional é unha clave candidata da relación R, isto é, ningún atributo facilita información doutro atributo que non é clave candidata.

Por exemplo, no esquema:

**Matrícula4** (DNI, materia, apelidos, nome, nota, aula)

**Situación** (aula, lugar)

**Materia** (materia, curso)

(onde *materia*, *apelidos*, *nome* é unha clave candidata e *DNI*, *materia* é outra clave candidata da relación Matricula4).

Neste caso, o esquema está en 3FN pero non está en FNBC, posto que DNI é un determinante funcional (apelidos e nome dependen de DNI) e non é clave candidata



da relación. Para poñer o esquema en FNBC hai que descompoñer Matricula4 en 2 relacións, de tal forma que queda un esquema similar ao obtido no apartado anterior:

**Matrícula3** (DNI, materia, nota, aula)

**Alumno** (DNI, nome, apelidos)

**Situación** (aula, lugar)

**Materia** (materia, curso)

### 5. Cuarta forma normal (4FN)

Está baseada na eliminación das dependencias multivaluadas. Dise que nunha relación existe unha dependencia multivaluada ( $\alpha \twoheadrightarrow \beta$ ), se os valores dun conxunto de atributos  $\beta$  depende unicamente do valor que tome outro conxunto de atributos  $\alpha$ , de forma independente ao resto de atributos da relación.

Por exemplo, cando temos unha relación para un concesionario con todos os modelos de coches que vende, coa súa cor e equipamento respectivo:

**Concesionario** (modelo, cor, equipamento)

e sabemos que pode vender dous modelos, *utilitario* e *berlina*. Se o modelo utilitario se pode vender en cor azul ou verde con dous tipos de equipamento (base ou normal) e o modelo berlina se pode vender en cor prata ou azul con equipamentos normal ou luxo.

Neste caso, se sabemos que o modelo é utilitario, sabemos os posibles valores para a cor e o equipamento, e polo tanto existen dúas dependencias multivaluadas:  $\text{modelo} \twoheadrightarrow \text{cor}$  e  $\text{modelo} \twoheadrightarrow \text{equipamento}$ , e a relación encóntrase en 4FN pero non en FNBC

Para poñer o esquema en 4FN débese descompoñer a relación noutras dúas da forma:

**Concesionario1** (modelo, cor)



<b>Concesionario 2 (<u>modelo</u>, <u>equipamento</u>)</b>
--

## **6. Quinta forma normal (5FN).**

Está baseada na eliminación das dependencias de reunión. Dise que existe unha dependencia de reunión se a relación pode ser construída sobre a base da reunión natural das proxeccións desa relación sobre os atributos que a forman.

Unha relación está en 5FN se e soamente se toda dependencia de reunión nesa relación está implicada polas claves candidatas entre si, e non por calquera outro atributo desa relación.

Por exemplo, na relación: **Docencia (DNI, materia, aula)**

que define as aulas que se asignan a cada materia e os alumnos matriculados nela (cada alumno matriculado nunha materia recibe clase en todas as aulas asignadas a esa materia), existe dependencia de reunión entre os atributos DNI, materia e aula.

Para eliminar as dependencias de reunión e poñelo en 5FN, descomponse a relación noutras tres:

<b>Docencia1 (<u>DNI</u>, <u>materia</u>)</b>
---

<b>Docencia2 (<u>materia</u>, <u>aula</u>)</b>
--

<b>Docencia3 (<u>DNI</u>, <u>aula</u>)</b>
--

## **21.4 SQL**

O SQL (Structured Query Language) é unha linguaxe estandarizada de peticións (query) a bases de datos relacionais. É a linguaxe de manipulación de bases de datos relacionais máis estendida, despois de se converter nun estándar de facto.



Permite realizar consultas utilizando os recursos da álgebra relacional combinados co cálculo relacional de tuplas.

**SQL sexa unha linguaxe declarativa** no que o importante é definir que se desexa facer por riba de como facelo. Con esta linguaxe pretendíase que as instrucións se puidesen escribir coma se fosen ordes humanas; é dicir, utilizar unha linguaxe o máis natural posible. De aí que estea considerada unha linguaxe de cuarta xeración.

A base teórica de SQL é bastante forte. As operacións funcionan en termos de conxuntos e non de rexistros individuais. Ademais non inclúe ningunha especificación de localización dos datos ou ruta de acceso deixando esta tarefa ao intérprete da linguaxe.

A primeira definición do modelo relacional de bases de datos foi publicada por Codd en 1970. O traballo de Codd foi desenvolvido inmediatamente por empresas e universidades. A SQL foi desenvolvida no centro de investigación de IBM baixo o nome SEQUEL (Structured English Query Language) en 1974 e 1975. A versión SEQUEL/2 cambiou de nome a SQL por motivos legais. IBM comezou a traballar nunha versión de SEQLTL/2 (SQL) chamada System R que estivo operativa en 1977.

En 1986 publicouse o estándar ANSI da linguaxe, que posteriormente foi adoptado por ISO en 1987, o que converte a SQL en estándar mundial como linguaxe de bases de datos relacionais.

En 1989 aparece o estándar ISO (e ANSI) chamado SQL89 ou SQL1. En 1992 aparece a nova versión estándar de SQL (a día de hoxe segue sendo a máis coñecida) chamada SQL92. En 1999 apróbase unha nova SQL estándar que incorpora melloras que inclúen disparadores, procedementos, funcións,... e outras características das bases de datos obxecto-relacionais. Ese estándar coñécese como SQL99.

O último estándar é o do ano 2008 (SQL2008). ISO/IEC 9075-1:2008



#### *21.4.1 Partes da linguaxe SQL*

Podemos considerar dúas fases na vida da base de datos: a etapa de preparación e posta en marcha e a etapa de explotación. Esta última é o obxectivo final de todo o sistema e as tarefas de preparación realizaranse antes de entrar nesa fase de utilidade para a organización.

A linguaxe SQL componse dun conxunto de instrucións ao igual que calquera linguaxe tradicional. Debido á existencia das dúas fases antes descritas adóitanse agrupar estas instrucións en tres "sublinguaxes" que en particular en bases de datos relacionais son as seguintes:

- **Linguaxe de definición de datos, DDL** (Data Description Language): É a linguaxe utilizada para a creación e mantemento da estrutura da base de datos. Utilízase para definir e modificar os esquemas das relacións, crear ou destruír índices e eliminar relacións. Permite tamén a definición de vistas e permisos de acceso para os usuarios. É a linguaxe que utiliza o administrador das bases de datos para realizar as súas tarefas.
- **Linguaxe de manipulación de datos, DML** (Data Manipulation Language). Inclúe todas as instrucións para realizar consultas ás bases de datos, inserir, modificar ou eliminar datos. Este é o que utilizan os usuarios finais na fase de explotación da base de datos. A parte da DML que permite realizar consultas sobre os datos da BD chámase DQL (Data Query Language), pero é unha parte da DML.
- **Linguaxe de control de datos, DCL** (Data Control Language). Existen unha serie de tarefas relacionadas coas bases de datos que non están incluídas en ningún dos dous grupos antes descritos xa que non son propiamente de descrición nin de manipulación de datos. Mediante esta linguaxe establécense as restricións adicionais necesarias para controlar a privacidade e a integridade da información almacenada na base de datos.



Ademais, as linguaxes comerciais, e a SQL en particular, inclúen outras facilidades como son control de principio e final das operacións ou o bloqueo de datos mentres dura unha consulta.

#### **21.4.2**      *Modos de traballo con SQL*

Os modos en que a linguaxe SQL actúa sobre unha base de datos son os seguintes.

- Modo interactivo: O usuario da base de datos establece un diálogo co sistema xestor de base de datos (SXBD) a través do intérprete de SQL. Desta forma pode realizar operacións interactivamente sobre a base de datos introducindo calquera sentenza SQL e sen restrición na orde destas.

As sentenzas SQL así introducidas son traducidas polo intérprete de SQL que producirá a solicitude correspondente para o xestor da base de datos, que a xestionará e xerará unha resposta para o usuario.

- Dende un programa: O usuario executa unha aplicación sobre o sistema operativo. A aplicación pode ser de dous tipos:
  - Programa escrito integramente en SQL ou mellor en extensións dela que inclúen as estruturas de programación habituais (bucles e selección). Un exemplo disto é PL/SQL de Oracle que é unha linguaxe procesual que permite desenvolver programas que acceden á base de datos vía SQL. Estes programas ou módulos de sentenzas SQL son en realidade guións que o intérprete SQL vai seguindo.
  - Programa escrito nunha linguaxe convencional de programación con partes escritas en SQL. Isto é o que se chama SQL embebida e a linguaxe que contén o texto SQL chámase linguaxe anfitrión (host). Neste caso as instrucións da linguaxe de programación execútanse polo procedemento habitual mentres que as sentenzas SQL pásanas a un módulo especial de execución do SXBD. Unha implantación de SQL embebida establece as relacións que deben manter os obxectos da base



de datos cos obxectos do programa anfitrión e restricións de funcionamento. Este modo de traballo admite dúas variantes:

- SQL estática: o programa non admite cambios durante a execución. Este é o método utilizado na maioría das aplicacións.
- SQL dinámica: se durante a execución se debe modificar algún parámetro tense que utilizar SQL dinámica. Isto resulta menos eficiente que un programa SQL estático e utiliza técnicas dinámicas de manexo de variables, o que dificulta a tarefa de programación.

### ***21.5 LINGUAXE E DEFINICIÓN DE DATOS (DDL)***

Os datos en bases de datos relacionais almacénanse en táboas. Unha táboa componse de filas e columnas ou rexistros e campos correspondentes a entidades e atributos dunha relación. A todas as estruturas lóxicas de datos chámanlles dunha forma xenérica obxectos das bases de datos.

Normalmente nun SXBD existen varios usuarios que deberán dispoñer en primeiro lugar de acceso ao sistema operativo da máquina. Cada usuario para o SXBD estará identificado por un nome, unha palabra clave e unhas propiedades. Todo iso é fixado e administrado polo administrador da base de datos (ABD) que é un usuario especial con dereitos sobre todos os obxectos.

O normal é que as estruturas de datos que van ser compartidas sexan creadas polo administrador pero isto non é un requisito imprescindible dado que pode haber usuarios para os que sexa de interese manter as súas propias bases de datos que compartirán con parte ou toda a organización.



O creador ou propietario dunha táboa pode permitir o acceso á súa táboa a outros usuarios do sistema pero pode que interéselle tamén permitir o acceso a unha parte da táboa. Para estes casos creárase unha vista (view). Nunha vista selecciónanse algúns atributos ou algunhas tuplas da táboa e permítese que usuarios seleccionados as poidan manexar.

A busca dun rexistro nunha táboa é unha busca secuencial. Unha forma de axilizar a busca consiste en ordenar os rexistros segundo algún criterio e preguntar polos rexistros segundo esa orde. Se existe unha táboa na que se anota o valor característico de cada rexistro e o seu enderezo, a busca chámase indexada e esta táboa auxiliar chámase índice.

Nun sistema multiusuario cada usuario ten un conxunto de obxectos que lle pertencen de diferentes tipos como táboas, índices ou vistas. Este conxunto adóitase chamar esquema.

Todos os obxectos citados ata agora deben crearse e configurarse antes de pasar ao uso da base de datos. Estas tarefas realízanse con DDL.

Cada obxecto dunha base de datos ten un nome que serve para localizalo ao longo da súa vida. Estes nomes teñen un ámbito onde son recoñecidos. Cada nome debe ser único no seu ámbito. Os obxectos que pertencen a un esquema (táboas, índices, etc.) teñen ese esquema como ámbito máximo. Por iso un nome de táboa, por exemplo, pode repetirse en distintos esquemas de usuarios diferentes pero non dentro do mesmo esquema.

Ademais destas normas básicas as diferentes implantacións teñen as súas propias regras sobre ámbitos.

Resumindo, cunha DDL pódese facer:

- Xestión de táboas
- Xestión de vistas



- Xestión de índices

### 21.5.1 *Obxectos da base de datos*

Segundo os estándares, unha base de datos é un conxunto de obxectos pensados para xestionar datos. Estes obxectos están contidos en esquemas. Un **esquema** representa a estrutura da base de datos. Os elementos que inclúe un esquema son táboas, dominios, vistas, restricións, disparadores e outros construtores.

No estándar SQL existe o concepto de **catálogo** que serve para almacenar esquemas. Así o nome completo dun obxecto viría dado por:

catálogo.esquema.obxecto
--------------------------

Se non se indica o catálogo toma o catálogo por defecto. Se non se indica o esquema enténdese que o obxecto está no esquema actual.

#### **Tipos de datos e dominios**

Un dominio é un conxunto do cal toma os seus valores unha columna ou atributo dunha relación. Segundo este concepto os tipos de datos predefinidos son dominios.

Algúns dos tipos de datos predefinidos no estándar son:

- Integer (4 Bytes) e SmallInt (2 Bytes)
- Decimal (precisión, (escala)). Representa un decimal de coma fixa. De se omitir a escala suponse 0.
- Float. Representa un decimal de coma variable.
- Char (n) Cadea de caracteres de lonxitude fixa (de n caracteres).
- Varchar (n) Cadea de caracteres de lonxitude variable (de máximo n caracteres).
- Date (data), Time (hora), TimeStamp (data e hora).
- Boolean, bit



- CLOB. Representa textos de gran lonxitude.
- BLOB. Representa binarios de gran lonxitude.

### Definición de dominios

Unha definición de dominios é un tipo de datos especializado que se pode utilizar na definición de columnas. A súa sintaxe é a seguinte.

```
CREATE DOMAIN nome dominio tipo de datos  
    [ DEFAULT valor defecto ]  
    [ definición de restricións de dominio ];
```

Onde:

- tipo de datos: un dos proporcionados por SQL
- valor defecto: especifica o valor por omisión para columnas definidas deste dominio. Será asignado a cada columna con ese dominio, se non ten xa a súa propia cláusula DEFAULT
- definición de restricións de dominio: implica unha restrición que se aplica a toda columna definida sobre o dominio. Defínese coa cláusula

```
[CONSTRAINT nome_restrición] CHECK (expresión condicional)
```

Exemplo: Enumeración de posibles valores das cores para un dominio particular

```
CREATE DOMAIN Cor VARCHAR(8) DEFAULT 'senCor'  
    CONSTRAINT cor_valida  
    CHECK (VALUE IN ( 'vermello', 'amarelo', 'azul', 'verde', 'senCor' ) );
```

## **21.5.2 Xestión de táboas**

Unha táboa é un obxecto da base de datos que almacena datos.



Para describir unha táboa utilízase a sentenza CREATE, especificando o nome e as características da táboa.

As características básicas que debe de ter unha táboa son:

- Definición dos atributos ou columnas
- Restricións de integridade

Para definir unha columna haberá que dicir o nome e tipo de datos que se van almacenar nese campo da táboa

### **Restricións de integridade**

Denomínase restrición de integridade a unha propiedade que debe cumprirse para manter nas táboas os criterios de deseño.

As restricións teñen un nome para poder ser manipuladas posteriormente e poden afectar a unha táboa enteira ou a unha columna ou atributo (restricións de táboa ou restricións de columna).

A restrición de táboa é algunha característica de mantemento da integridade que se asocia á táboa ao creala. Esta restrición aplicarase aos valores que conteña a táboa, e despois supoñerá unha validación dos datos ao introducilos ou modificalos.

As restricións de columnas indican certas características que deben asociarse á columna que se está a describir.

En SQL as restricións de integridade fíxanse mediante unha sentenza CONSTRAINT.

A sintaxe de restricións para táboas é a seguinte:

```
[CONSTRAINT nome_restrición] (  
    [UNIQUE | PRIMARY KEY (atrib1[,atrib2]...)]  
    [FOREIGN KEY (atrib1 [,atrib2]...)  
        REFERENCES táboa(atrib1[,atrib2]...)]
```



```
[ON UPDATE [CASCADE | NON ACTION | RESTRICT | SET
            NULL | SET DEFAULT]
[ON DELETE [CASCADE | NON ACTION | RESTRICT | SET
            NULL | SET DEFAULT] ]
[CHECK condición]
)
```

A sintaxe de restricións para atributos é a seguinte:

```
[CONSTRAINT nome_restrición] (
    [NOT NULL]
    [UNIQUE | PRIMARY KEY]
    REFERENCES táboa(atrib1[,atrib2]...)
        [ON UPDATE [CASCADE | NON ACTION | RESTRICT | SET
                    NULL | SET DEFAULT]
        [ON DELETE [CASCADE | NON ACTION | RESTRICT | SET
                    NULL | SET DEFAULT] ]
    [CHECK condición]
)
```

As restricións que se poden establecer son:

- NOT NULL: aplícase a un campo e indica que non pode conter valores nulos.
- UNIQUE: a(s) columna(s) non pode conter valores duplicados. Debe declararse como NOT NULL e non poden formar parte da clave primaria.
- PRIMARY KEY: denota a(s) columna(s) que son clave principal da táboa. Teñen valor único e non nulo.



- CHECK: pódese aplicar a un atributo ou a toda a táboa. Indica unha condición que debe satisfacer cada atributo ou cada fila da táboa antes de ser inserida borrada ou actualizada.
- FOREIGN KEY / REFERENCES: cando entre dúas táboas se establece unha relación as tuplas dunha relaciónanse coas da outra mediante certos campos clave en cada táboa. A táboa da que parte a relación chámase táboa primaria e a outra chámase táboa secundaria.

Chámase **clave allea** (foreign key) o conxunto de atributos da táboa secundaria que é clave principal na táboa primaria. Esta última debe ser clave principal ou única.

Unha relación mantén a integridade referencial se cumpre as seguintes dúas condicións:

- Toda tupla da táboa filla está asociada cunha tupla da táboa pai.
- Se unha tupla da táboa filla non cumpre o anterior, o valor que ten a columna da clave allea é nulo.

A restrición de integridade referencial debe establecerse na táboa filla. Úsanse certos parámetros para fixala

- FOREIGN KEY: indica que columna ou columnas constitúen a clave allea nunha restrición de integridade referencial. Aplícase na restrición de táboa.
- REFERENCES: especifica a táboa pai. Se non indica a que clave primaria ou única se refire a clave allea enténdese que a clave referenciada é a clave primaria da táboa indicada.

As seguintes opcións fan que o xestor manteña a integridade referencial



- a) ON [DELETE | UPDATE] CASCADE: Borra ou actualiza o rexistro na táboa pai e automaticamente borra ou actualiza os rexistros coincidentes na táboa filla. Sen esta cláusula non se permite borrar ou actualizar un rexistro principal que teña rexistros secundarios asociados.
- b) ON [DELETE | UPDATE] RESTRICT: Non se pode borrar ou actualizar un rexistro na táboa pai mentres non se borre ou actualice o rexistro secundario asociado na táboa filla.
- c) ON [DELETE | UPDATE] SET NULL: Borra ou actualiza o rexistro na táboa pai e establece en NULL a ou as columnas de clave foránea na táboa filla.
- d) ON [DELETE | UPDATE] NON ACTION: Significa ningunha acción no sentido de que un intento de borrar ou actualizar un valor de clave primaria non será permitido se na táboa referenciada hai un valor de clave foránea relacionado.
- e) ON [DELETE | UPDATE] SET DEFAULT: Borra ou actualiza o rexistro na táboa pai e establece o valor por defecto da ou as columnas de clave foránea na táboa filla.

## 1) Creación dunha táboa

A sintaxe básica da instrución é a seguinte.

```
CREATE TABLE nometáboa (  
    atributo1 tipo1 [restricións de atributo]  
    [,atributo2 tipo2 [NOT NULL] [UNIQUE] [DEFAULT valor]  
    [Restricións De Táboa])
```

A creación dunha táboa engloba as definicións de atributos e/ou restricións:



a) **A DEFINICIÓN DE ATRIBUTOS** realízase dando o nome do atributo (axústase ás mesmas regras que os nomes de táboas) e o seu tipo.

Opcionalmente pódese indicar unhas restricións de atributos:

- a. Not null: restrición de valor non nulo.
- b. Definicións de restricións de clave primaria, valor UNIQUE, clave allea.
- c. Definición de restricións xerais coa cláusula check.

b) **A DEFINICIÓN DE RESTRICIÓN DE INTEGRIDADE / SEMÁNTICAS:**

Permítenlle ao deseñador restrinxir o rango de valores dunha táboa. As restricións poden ser de columna se afectan a unha única columna, ou de táboa se afectan a unha ou máis columnas, tal como se viu no apartado anterior.

Exemplo:

```
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_empr PRIMARY KEY (EMPNO),
FOREIGN KEY (DEPTNO) REFERENCES DEPT (DEPTNO)
          ON UPDATE CASCADE ON DELETE RESTRICT,
CHECK SAL >600,04
)
```

## 2) Modificación dunha táboa

Para modificar unha táboa xa creada utilízase o comando ALTER TABLE co que poden especificarse novas columnas, novas restricións (ADD) ou ben modificarse unha columna (MODIFY). A sintaxe é:

```
ALTER TABLE táboa (
  [ADD (col1|restric1) (,col2|restric2)...]
  [ALTER (col1 tipo1(,col2 tipo2)...)]
  [DROP CONSTRAINT restrición]
  [DROP COLUMN columna [CASCADE | RESTRICT] ]
)
```

- Ao engadir unha columna a unha táboa xa existente hai que ter en conta que **non** está permitido NOT NULL na definición dunha nova columna, e se se desexa introducir un valor para a columna, en cada fila existente, hai que especificalo coa cláusula DEFAULT ao engadir a columna.
- Só se pode cambiar o tipo ou diminuír o tamaño dunha columna se ten valores nulos en todas as columnas.
- Só se poden borrar restricións que teñan nome.
- Ao eliminar unha columna dunha táboa podemos indicar a opción:



- CASCADE: elimina a columna e toda restrición ou vista que lle fai referencia.
- RESTRICT: só elimina a columna se ningunha vista nin restrición lle referencia.

Por exemplo, a sentenza:

```
ALTER TABLE EMP (  
  ADD ADDRESS VARCHAR(12) DEFAULT "Unknow";  
  ALTER ENAME VARCHAR(30)  
)
```

Modifica a lonxitude do atributo ENAME e engade o campo ADDRESS.

### 3) Destrución dunha táboa

A sintaxe para eliminar unha táboa é:

```
DROP TABLE nome_da_táboa [CASCADE | RESTRICT]
```

Para executar esta instrución débense ter suficientes privilexios no sistema ou ser o propietario da táboa.

O parámetro opcional:

- RESTRICT: Destrúe a táboa só se non se lle fai referencia dende ningunha outra táboa (clave allea), nin é táboa base dunha vista.
- CASCADE: Elimina a táboa xunto coas restricións e vistas que a referencian.

Despois da execución da sentenza DROP calquera referencia á táboa dará un erro. A táboa borrarase independentemente de que conteña datos ou non.

#### 21.5.3 *Xestión de vistas*

Unha vista é unha táboa virtual, é dicir, unha táboa que non existe fisicamente na base de datos pero que lle aparece ao usuario coma se existise. As vistas non teñen



datos almacenados propios, distinguibles e fisicamente almacenados. No seu lugar, o sistema almacena a definición da vista (é dicir, as regras para acceder ás táboas base fisicamente almacenadas para materializar a vista).

As vistas teñen varias utilidades:

- Mostrar aos usuarios os datos que lles interesan.
- Protexer os datos.
- Reestruturar datos que poden estar distribuídos en diferentes soportes de maneira que aparezan como unha táboa.
- Crear interfaces para aplicacións que esperan unha estrutura de táboas distinta á de creación. Mediante as vistas as aplicacións independízanse da estruturación real dos datos.

### **Creación dunha vista**

Úsase a sentenza CREATE VIEW coa sintaxe seguinte:

```
CREATE [OR REPLACE] VIEW nome_de_vista [lista_de_campos]
AS consulta
```

Cando se executa unha sentenza de creación de vista realízase unha consulta que selecciona tuplas e atributos dunha ou varias táboas. Exemplo: se se lle quere dar unha lista de empregados á empresa de seguridade que controla o acceso ao edificio, utilizarase unha vista sobre a táboa orixinal:

```
CREATE VIEW EMP_SECURITY
AS SELECT EMPNO, ENAME
FROM EMP
```

Por defecto, a vista toma os nomes dos atributos seleccionados dende as táboas base, sempre que ningún atributo sexa o resultado dunha operación aritmética ou



función de agregado. Se se desexa cambiar os nomes dos atributos utilízase a `lista_de_campos`.

### **Modificación da estrutura dunha vista**

A estrutura dunha vista non pode ser modificada como tal. O que se pode facer é utilizar unha sentenza de creación coa cláusula `OR REPLACE` que substituirá unha vista por outras.

O comando de SQL `ALTER VIEW` utilízase para recompilar unha vista. Isto débese facer cando se modificaron as táboas bases da vista para actualizar a vista sobre as novas estruturas. A súa sintaxe é:

```
ALTER VIEW vista COMPILE
```

Despois desta instrución toda referencia á vista dende outro obxecto destrúese.

### **Destrucción dunha vista**

A instrución `DROP VIEW` permite eliminar unha vista que fose creada anteriormente

```
DROP VIEW vista
```

## **21.5.4 Xestión de índices**

Nunha base de datos un índice é un medio de acceder aos rexistros dunha forma máis rápida que co simple percorrido secuencial dunha táboa. O índice é un obxecto da base de datos que conterá unha entrada para cada valor das columnas indexadas, co enderezo do rexistro onde debe buscarse ese valor.

Un dos usos máis comúns dos índices é o mantemento dunha táboa ordenada por distintos criterios.

Só é recomendable crear índices para aqueles campos que teñan moitas buscas pois o índice ocupa espazo e ten que actualizarse cada vez que se borra, actualiza ou insire un elemento nunha táboa.



### Creación dun índice

A instrución para crear un índice é CREATE INDEX

```
CREATE INDEX nome índice  
ON nome_táboa (campo{, campo)  
[NOSORT]
```

Especificase o nome da táboa sobre a que crea o índice, así como o campo ou campos sobre o que se indexa. É posible crear índices concatenados, que se forman con máis dunha columna. Emprégase en caso de columnas que sempre se consultan xuntas.

A opción NOSORT que aparece nalgúns sistemas SQL fai que aforre tempo e espazo na creación dun índice facendo que se a táboa se encheu con rexistros que están fisicamente ordenados co mesmo criterio que o índice se poida evitar a ordenación que se produce ao crear o índice.

```
CREATE INDEX IND_EMPRE ON EMPRESA (Abre_emp)
```

### Eliminar un índice

Para destruír un índice utilízase a cláusula DROP INDEX

```
DROP INDEX índice
```

## 21.6 LINGUAXE DE MANIPULACIÓN DE DATOS (DML)

Chámanse manipulacións a aquelas operacións sobre unha base de datos que non afectan á súa estrutura senón ao seu contido. Estas operacións realízanse con DML



(Data Manipulation Language). As manipulacións posibles sobre unha base de datos son as seguintes:

- Inserir valores en tuplas (INSERT)
- Eliminar unha tupla (DELETE)
- Actualizar o valor dun campo nunha ou varias tuplas (UPDATE)
- Consultar ou listar todos ou algúns campos dun grupo de tuplas (SELECT)

As operacións de manipulación pódense facer tanto en táboas coma en vistas xa que estas non son senón táboas lóxicas.

#### **21.6.1**      *Inserción de valores nunha táboa*

Para inserir unha fila ou tupla nunha táboa xa creada utilízase o comando INSERT cuxa sintaxe é:

```
INSERT INTO nome_táboa [columna (,columna)*]  
VALUES (valor (,valor)*)}
```

Se non se especifican nomes de columnas, os valores inseridos deben corresponder en cantidade e tipo de datos cos atributos da táboa e teñen que estar na mesma orde coa que se creou a táboa.

Pódense especificar unhas columnas e outras non, tendo en conta que as columnas non especificadas tomarán o valor por defecto, se se definiu, ou NULL.

```
INSERT INTO DEPT (DEPTNO, DNAME, LOC)  
VALUES (90, 'CONTABILIDADE', 'OURENSE')
```

Para inserir varias tuplas cunha única instrución pódese utilizar unha subconsulta que devolva tuplas de estrutura compatible coa da táboa:



```
INSERT INTO nome_táboa [columna (,columna)*]  
consulta
```

### 21.6.2 *Borrado de valores dunha táboa*

Para borrar unha fila utilízase o comando DELETE coa sintaxe seguinte:

```
DELETE FROM nome_táboa  
[WHERE condición]
```

Con esta instrución bórranse todas as tuplas que cumpran a condición WHERE. Se non se inclúe esta, bórranse todos os elementos da táboa. Por exemplo, para borrar o departamento "CONTABILIDADE":

```
DELETE FROM DEPT WHERE DNAME = "CONTABILIDADE"
```

### 21.6.3 *Modificación de valores dunha táboa*

Para modificar os valores de determinadas tuplas dunha táboa utilízase a sentenza UPDATE coa seguinte sintaxe:

```
UPDATE nome_táboa  
SET columna1=valor1{,columna2=valor2)*}  
[WHERE condición]
```

Coa cláusula SET especificase as columnas que se deben modificar cos seus novos valores, e coa cláusula WHERE selecciónanse as filas que cómpre actualizar. Se non hai WHERE, aplícase a modificación a todas as filas. Por exemplo, se se quere conceder a todo empregado do departamento de Informática un aumento salarial do 18%.

```
UPDATE EMP  
SET SAL = SAL*1.18  
WHERE DEPTNO IN (SELECT DEPTNO
```



```
FROM DEPT
WHERE DNAME='INFORMÁTICA')
```

#### 21.6.4 Consulta de datos

Unha consulta serve para extraer os datos almacenados nunha base de datos. A consulta en SQL consta de tres partes:

- Cláusula SELECT: para indicar que atributos se desexan consultar
- Cláusula FROM: indica sobre que relación ou relacións se quere facer a consulta
- Cláusula WHERE: indica as condicións que deben cumprir as tuplas para ser seleccionadas

A súa sintaxe abreviada é a seguinte:

```
SELECT * | {[DISTINCT] columna | expresión [[AS] alcume],...}
FROM táboas
[WHERE condicións_where]
[GROUP BY columnas_group]
[HAVING condicións_having]
[ORDER BY columnas_orde]
```

O efecto dunha consulta como esta é o seguinte:

- Realízase o produto cartesiano das relacións citadas na cláusula FROM
- Aplícase o operador selección da álgebra relacional para seleccionar aquelas tuplas do produto cartesiano que fagan verdadeiro o predicado WHERE
- Proxéctase o resultado obtido sobre os atributos especificados en SELECT

#### Selección de atributos



A consulta máis sinxela é seleccionar todas as tuplas dunha táboa. Por exemplo, seleccionar todos os datos de todos os empregados.

```
SELECT EMPNO, ENAME, JOB, MGR, HIREDATE, SAL, COMM, DEPTNO  
FROM EMP
```

Utilízase o asterisco (\*) como comodín para seleccionar todos os campos. A sentenza anterior é equivalente a:

```
SELECT * FROM EMP;
```

Pódense seleccionar columnas individuais: "Lista todos os salarios"

```
SELECT SAL FROM EMP
```

Nunha táboa, algunhas columnas terán valores repetidos. De querer só mostrar os valores diferentes dunha columna nunha táboa hai que usar a palabra clave **DISTINCT** anteposta ao nome da columna

```
SELECT DISTINCT SAL FROM EMP
```

Pódese cambiar o nome que se lle dá á cabeceira da columna no resultado da instrución SELECT. Para iso utilízase un alcume co comando AS despois do nome da columna.

```
SELECT SAL AS Salario FROM EMP
```

## Orde

Por defecto SQL non ordena os resultados; para ordenalos, utilízase a cláusula ORDER BY:



```
ORDER BY campo1 [ASC|DESC], campo2 [ASC|DESC],...
```

O modo de ordenación indícase para cada campo:

- ASC: orde ascendente
- DESC: orde descendente

### Consulta con expresións

Tamén se poden realizar consultas nas que se avalíe unha expresión.

Por exemplo, se se dispón dunha táboa co salario bruto anual dos empregados dunha empresa, é posible consultar o seu soldo neto mensual. Supoñendo que se paguen 14 pagas anuais e que se coñeza o tipo de IRPF a instrución para realizar esta consulta sería:

```
SELECT ENAME, (SAL/14)(1-IRPF/100) FROM EMP
```

### Condicións para restrinxir a consulta

Para restrinxir as tuplas que se obteñen pódense impoñer condicións. Para iso úsase a cláusula WHERE que admite os seguintes operadores

Operadores relacionais (comparación):  $>$ ,  $>=$ ,  $<$ ,  $<=$ ,  $=$ ,  $<>$ : Estes operadores utilízanse para comparar datos.

Operadores lóxicos: AND, OR, NOT: Utilízanse para unir condicións.

```
SELECT * FROM EMP
      WHERE DEPTNO =99
            AND SAL >1250;
```

Ademais, as condicións unidas por AND, OR e NOT admiten parénteses. Se non se poñen parénteses a prioridade, de maior a menor, é NOT, AND e OR.



### Consulta de pertenza a un rango

Pódese comprobar se unha expresión entra ou non dentro dun rango marcado. Utilízase o operador BETWEEN. A sintaxe é:

```
expresión [NOT] BETWEEN expresión [AND expresión]
```

```
SELECT * FROM EMP  
WHERE EMPNO BETWEEN 9 AND 54
```

### Consultas de pertenzas a unha lista

Co operador IN compróbase se un elemento pertence a unha lista de valores. A sintaxe da condición é:

```
elemento[NOT] IN lista_expresións | subconsulta
```

Exemplo:

```
SELECT * FROM EMP  
WHERE ENAME IN('Pepe Martínez' e 'Xosefa Martín')
```

Selecciona da táboa de empregados os datos de 'Pepe Martínez' e 'Xosefa Martín'.

### Consulta con patróns

Coa utilización do operador LIKE pódese buscar unha cadea de caracteres dentro doutra. A sintaxe é a seguinte:

```
<cadea>[NOT] LIKE <cadea>
```

É unha comparación de igualdade pero admite comodíns:

- %: pódese substituír por calquera número de caracteres (0 ou máis).
- Os SXBDR tamén admiten substituír un único carácter.



Exemplo: Para seleccionar aqueles elementos da táboa ALUMNO nos que o campo Nome empece pola cadea "Ma"

```
SELECT * FROM EMP WHERE ENAME LIKE 'Ma%'
```

### Funcións de agregación

Existen funcións que permiten calcular, dende unha sentenza SQL, sumas, medias aritméticas, etc. de datos. Moitas destas funcións aceptan o parámetro DISTINCT|ALL. Se toma o valor ALL (valor por defecto) indica que se deben considerar todas as aparicións aínda que sexan repetidas e se é DISTINCT deben ignorarse as repeticións. Algunhas das máis importantes funcións son:

- AVG (Atributo): media aritmética dos valores de atributo.
- MIN (Atributo): valor mínimo dos valores de atributo.
- MAX (Atributo): valor máximo dos valores de atributo.
- SUM (Atributo): suma os valores de atributo.
- COUNT (Atributo): conta o número de filas onde atributo non é nulo.
- COUNT (\*): conta o número de filas incluíndo aquelas con nulos.
- LCASE (Atributo): transforma Atributo a maiúsculas.
- UCASE (Atributo): transforma Atributo a minúsculas.
- MID (Atributo, m [, n]): devolve unha porción de Atributo comezando no carácter m e con n caracteres de lonxitude.
- LEN (Atributo): devolve a lonxitude de Atributo.

```
SELECT AVG (SAL) FROM EMP  
SELECT COUNT (*) FROM EMP
```



## Consulta con agrupamento de filas

As funcións de agregación adóitanse utilizar combinadas coa cláusula de agrupamento GROUP BY, que agrupa o resultado por unha serie de atributos.

Unha instrución SELECT con este parámetro devolve grupos de tuplas en lugar de tuplas individuais. Como resultado da consulta aparecerá un resumo da información por cada grupo en lugar de todas as filas.

Por exemplo para listar os números de departamento e a suma dos salarios de cada un deles utilizaríase:

```
SELECT DEPTNO, SUM(SAL)
FROM EMP
GROUP BY DEPTNO
```

A expresión dun GROUP BY pode conter referencias a calquera campo das táboas nomeadas en FROM. Non obstante a lista de expresións que seguen ao SELECT non pode conter máis que:

- Constantes
- Funcións de grupo (AVG, MAX, MIN, COUNT, SUM)
- Expresións idénticas ás da cláusula GROUP BY
- Expresións que devolvan o mesmo valor para todas as tuplas que formen parte dun grupo.

Non se poden seleccionar atributos que non se poidan agrupar polos atributos indicados no GROUP BY.

Se nunha consulta aparece unha cláusula WHERE e unha GROUP BY, primeiro seleccionaranse as tuplas que cumpran a condición do WHERE e despois aplícase o agrupamento.



### Restricións nos agrupamentos

Cando se selecciona un conxunto de atributos agrupados por un ou máis atributos, pódense impoñer condicións aos grupos (é dicir, condicións aos atributos que se están a seleccionar). É a cláusula HAVING, que sería o equivalente á cláusula WHERE pero aplicada aos grupos. Por exemplo,

"Lista a suma dos soldos agrupada por departamentos, pero só aqueles nos que a suma sexa maior que 7.000"

```
SELECT SUM(SAL), DEPTNO
      FROM EMP
      GROUP BY DEPTNO
      HAVING SUM(SAL)>7.000
```

#### **21.6.5**      *Consultas sobre múltiples táboas.*

É máis que habitual necesitar nunha consulta datos que se encontran distribuídos en varias táboas. As bases de datos relacionais baséanse en que os datos se distribúen en táboas que se poden relacionar mediante un campo.

Para realizar consultas a máis dunha táboa, abonda con indicar na cláusula FROM as táboas separadas por comas e engadir as condicións necesarias na cláusula WHERE.

```
SELECT ENAME, DNAME
      FROM EMP, DEPT
```

Este exemplo realiza o produto cartesiano da táboa EMP e DEPT e devolvería para cada rexistro da táboa EMP, todos os rexistros da táboa DEPT.

Se se quere facer correctamente, asociando nome de traballador co departamento no que traballa, utilízase un criterio de comparación pola clave allea, por exemplo:

```
SELECT ENAME, DNAME
```



```
FROM EMP, DEPT
WHERE EMP.DEPTNO = DEPT.DEPTNO
```

Os nomes dos atributos, en caso de que as táboas teñan atributos co mesmo nome, irán precedidos polo nome da táboa e un punto, como en EMP.DEPTNO. Se non existe confusión posible poden indicarse sen o nome da táboa, como por exemplo DNAME, que só existe na táboa DEPT.

Para evitar repetir continuamente o nome das táboas, pódese especificar un alcume, engadindo ao nome da táboa na cláusula FROM o alcume.

A partir da versión SQL 1999 ideouse unha nova sintaxe para consultar varias táboas. A razón foi separar as condicións de asociación respecto das condicións de selección de rexistros. A sintaxe completa é:

```
SELECT táboa1.column1, táboa1.column2... táboa2.column1, táboa2.column2...
FROM táboa1
    [CROSS JOIN táboa2] |
    [NATURAL JOIN táboa2] |
    [JOIN táboa2 USING (columna)] |
    [JOIN táboa2 ON (táboa1.columna=táboa2.columna)] |
    [LEFT|RIGHT|FULL OUTER JOIN táboa2 ON (tb1.column=tb2.column)]
```

- **CROSS JOIN.** Realiza un produto cruzado entre as táboas indicadas. Iso significa que cada tupla da primeira táboa se combina con cada tupla da segunda táboa. É dicir se a primeira táboa ten 10 filas e a segunda outras 10, como resultado obtéñense 100 filas, resultado de combinar todas entre si.

```
SELECT ENAME, DNAME
FROM EMP CROSS JOIN DEPT
```



- **NATURAL JOIN.** Establece unha relación de igualdade entre as táboas a través dos campos que teñan o mesmo nome en ambas as dúas táboas:

```
SELECT ENAME, DNAME  
FROM EMP NATURAL JOIN DEPT
```

Nese exemplo obtéñense a lista dos empregados e os nomes dos departamentos aos que pertencen a través dos campos que teñan o mesmo nome en ambas as dúas táboas. Hai que asegurarse de que só son as claves principais e secundarias das táboas relacionadas, as columnas nas que o nome coincide, doutro modo fallaría a asociación e a consulta non funcionaría.

- **JOIN USING.** Permite establecer relacións indicando que columna (ou columnas) común ás dúas táboas hai que utilizar. As columnas deben de ter exactamente o mesmo nome en ambas as dúas táboas:

```
SELECT ENAME, DNAME  
FROM EMP JOIN DEPT USING (DEPTNO)
```

- **JOIN ON** Permite establecer relacións cunha condición que se establece manualmente, o cal é útil para asociacións nas que os campos nas táboas non teñen o mesmo nome:

```
SELECT ENAME, DNAME  
FROM EMP e JOIN DEPT d ON (e.DEPTNO=d.DEPTNO)
```

- **OUTER JOIN** Utilizando as formas vistas ata agora de relacionar táboas só aparecen no resultado da consulta filas presentes nas táboas relacionadas. É dicir na consulta anterior só aparecen empregados relacionados coa táboa de departamentos. Se hai empregados que non están en departamentos, estes non aparecen (e se hai departamentos que non están na táboa de empregados, tampouco saen).
- Para solucionar isto, utilízanse relacións laterais ou externas (outer join):



- o táboa1 LEFT OUTER JOIN táboa2 ON. Obtén os datos da táboa 1 estean ou non relacionados cos datos da táboa 2.
- o táboa1 RIGHT OUTER JOIN táboa2 ON. Obtén os datos da táboa 2 estean ou non relacionados con datos da táboa 1.
- o táboa1 FULL OUTER JOIN táboa2 ON. Obtén os rexistros non relacionados de ambas as dúas táboas.

#### 21.6.6 Subconsultas

Son sentenzas SELECT que se encontra aniñadas dentro doutras SELECT. Estas sentenzas escríbense entre paréntese para advertir o xestor de que se debe executar primeiro. Permite solucionar consultas que requiren para funcionar o resultado previo doutra consulta.

```
SELECT ENAME  
FROM EMP  
WHERE SAL >= (SELECT AVG(SAL) FROM EMP)
```

Dá como resultado o listado dos empregados que superan a media de soldo. Primeiro realízase a operación do SELECT da cláusula WHERE e seguidamente execútase o SELECT do principio.

Unha subconsulta que utilice os valores >, <, >=... ten que devolver un único valor, doutro modo ocorre un erro. Ademais teñen que ter o mesmo tipo de columna para relacionar a subconsulta coa consulta que a utiliza (non pode suceder que a subconsulta teña dúas columnas e ese resultado se compare usando unha única columna na consulta xeral).

#### A cláusula (NOT) EXISTS

A cláusula EXISTS (ou NOT EXISTS) comproba se unha subconsulta devolve algún valor (EXISTS) ou non devolve ningún (NOT EXISTS). Por exemplo:



"Lista os departamentos que non contratasen a ninguén o 28 de decembro 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 " ' )
```

A consulta de primeiro nivel busca na táboa de departamentos os nomes e, para cada fila, comproba —mediante a subconsulta— que para ese número de departamento non existan empregados que fosen contratados o 28 de decembro de 2010.

### **Consulta con cuantificadores**

Denomínanse cualificadores certos predicados que permiten utilizar subconsultas que devolven varias filas na columna correspondente a un atributo.

Por exemplo se se quere mostrar o soldo e o nome dos empregados cun soldo superior ao de calquera empregado do departamento de vendas, a subconsulta necesaria para ese resultado mostraría todos os soldos do departamento de vendas. Pero non poderemos utilizar un operador de comparación directamente xa que esa subconsulta devolve máis dunha fila. A solución a isto é utilizar os cuantificadores entre o operador e a consulta, que permiten o uso de subconsultas de varias filas.

A sintaxe de uso dentro de condicións é a seguinte:

expresión operador\_relacional cuantificador {lista\_exps | subconsulta}

Os cuantificadores son:



- ❑ **ANY ou SOME:** A comparación cun ANY (ou SOME, equivalente) é verdadeira se o é para algún valor dos obtidos cunha subconsulta.
- ❑ **ALL.** Neste caso a comparación é verdadeira se o é con todos os valores devoltos pola consulta subordinada e falsa no caso contrario. Por exemplo: para saber o empregado co soldo máis alto de toda a empresa.

```
SELECT ENAME  
FROM EMP  
WHERE SAL > = ALL(SELECT SAL FROM EMP)
```

### 21.6.7 Operacións con consultas

Existen certos operadores que permiten combinar os conxuntos de tuplas que se obteñen de dúas consultas SELECT e obter un novo conxunto de tuplas. Estas operacións corresponden con operadores da álgebra relacional e son os seguintes:

- UNION e UNION ALL realizan a unión das tuplas obtidas por dúas consultas que se especifican como operandos. UNION non inclúe as tuplas repetidas mentres que UNION ALL si as inclúe. UNION corresponde á unión de relacións da álgebra relacional.

Para iso ambas as dúas instrucións teñen que utilizar o mesmo número e tipo de columnas

```
SELECT nome FROM empregados  
UNION  
SELECT nome FROM visitantes
```

Isto crea unha táboa que inclúe os nomes dos empregados e dos visitantes

- INTERSECT permite unir dúas consultas SELECT de modo que o resultado serán as filas que estean presentes en ambas as dúas consultas. Equivale ao operador intersección da álgebra relacional.



```
SELECT tipo,modelo FROM produtos
```

```
WHERE chip="QWER-21"
```

#### **INTERSECT**

```
SELECT tipo,modelo FROM produtos
```

```
WHERE chip="WDFV-23"
```

- MINUS combina dúas consultas SELECT de forma que aparecerán os rexistros do primeiro SELECT que non estean presentes no segundo. Corresponde á diferenza da álgebra relacional.

```
SELECT tipo,modelo FROM produtos
```

```
WHERE chip="QWER-21"
```

#### **MINUS**

```
SELECT tipo,modelo FROM produtos
```

```
WHERE chip="WDFV-23"
```

As dúas consultas sobre as que se aplique calquera destes operadores deben devolver tuplas coa mesma estrutura.

## **21.7 . LINGUAXE DE CONTROL DE DATOS DCL**

Co nome de linguaxe de control de datos (DCL Data Control Language) faise referencia á parte da linguaxe SQL que se ocupa dos apartados de seguridade e da integridade no procesamento concorrente.

### **21.7.1      Seguridade**

A linguaxe SQL supón un nivel xeral de seguridade do software xestor da base de datos e as súas sentenzas utilízanse para especificar restricións de seguridade. O esquema de seguridade SQL baséase en tres conceptos:

- Os usuarios son os actores da base de datos. Cada vez que o xestor da base de datos recupera, insire, suprime ou actualiza datos, faino a conta dalgún usuario.



- Os obxectos da base de datos son os elementos aos que se pode aplicar a protección de seguridade SQL. A seguridade aplícase xeralmente a táboas, vistas e columnas
- Os privilexios son as accións que un usuario ten permitido efectuar para un determinado obxecto da base de datos.

A creación e eliminación de usuarios en SQL non é estándar, dependendo de cada produto comercial. Non obstante, a concesión e revogación de privilexios se é estándar e está recollida nas sentenzas GRANT e REVOKE.

### **Concesión de privilexios: GRANT**

A sentenza GRANT utilízase para conceder privilexios de seguridade sobre obxectos da base de datos a usuarios específicos. Normalmente a sentenza GRANT é utilizada polo propietario da táboa ou vista para proporcionar a outros usuarios acceso aos datos. A sentenza GRANT inclúe unha lista específica dos privilexios que se van conceder, o nome do obxecto ao que se aplican os privilexios e a lista de usuarios aos que lles conceden os privilexios. A sintaxe é a seguinte:

```
GRANT listaPrivilexios  
ON listaObxectos  
TO listaUsuarios  
[WITH GRANT OPTION]
```

- ListaPrivilexios: Concédense todos (ALL) ou un subconxunto de privilexios (separados por comas) que permiten borrar, inserir, consultar, actualizar ou modificar (DELETE, INSERT, SELECT, UPDATE, ALTER) unha táboa, vista ou un conxunto delas.
- ListaObxectos: Obxectos aos que se lle aplican os privilexios.
- ListaUsuarios: Concédense a todos os usuarios (PUBLIC) ou a unha lista de usuarios.



- **WITH GRANT OPTION:** Indica que aqueles usuarios aos que se concedeu estes privilexios poden á súa vez concedelos (nunca máis dos que teñen actualmente) a outros usuarios por medio de sentenzas GRANT.

Por norma xeral os privilexios de acceso aplícanse sobre todas as columnas na táboa ou vista, pero tamén se pode especificar unha lista de columnas co privilexio UPDATE.

```
GRANT UPDATE (SAL) ON EMP TO grupoNóminas
```

Só o propietario dun obxecto pode conceder os privilexios del. O propietario é sempre o seu creador.

As operacións co esquema da base de datos (CREATE, DROP, etc.) só poden ser realizadas polo propietario do esquema.

### **Revogación de privilexios: REVOKE**

Os privilexios que se concederon coa sentenza GRANT poden ser retirados coa sentenza REVOKE. A sentenza REVOKE ten unha estrutura que se asemella estreitamente á sentenza GRANT, especificando un conxunto específico de privilexios que van ser revogados, para un obxecto da base de datos específico, para un ou máis usuarios. Unha sentenza REVOKE pode retirar todos ou parte dos privilexios que previamente se concederon a un usuario. É necesario especificar que un usuario só pode retirar os privilexios que el mesmo concedeu a outro usuario.

```
REVOKE ListaPrivilexios  
ON ListaObxectos  
FROM ListaUsuarios
```

A utilización de vistas combinada cunha definición de usuarios e a concesión xuízosa de privilexios constitúe o mecanismo de seguridade que o administrador da base de datos SQL utiliza para levar a cabo as políticas de seguridade do sistema.



### *21.7.2 Transaccións*

Enténdese por transacción o efecto producido por un grupo de instrucións DML executadas unha tras outra, é dicir, unha transacción é un conxunto de accións que ou ben se realizan todas, ou ben non se realiza ningunha.

En SQL unha transacción comeza implicitamente na primeira instrución que altera o estado da información almacenada na base de datos. Para preservar as propiedades ACID (Atomic, Consistent, Isolate, Durable) dunha transacción, SQL dispón de dúas sentenzas que permiten que os cambios realizados por unha transacción queden reflectidos permanentemente na base de datos (comprometer)

- COMMIT [WORK]. Remata a transacción actual gravando permanentemente as modificacións.
- ROLLBACK [WORK]. Obriga o sistema a volver ao estado anterior ao inicio da transacción.

Tamén é posible que cada contorno de programación e/ou SXBD dispoña de elementos adicionais para o control de concorrencia que poidan ser utilizados polo usuario, como por exemplo bloqueos.

Dende o momento en que a unha base de datos poden acceder diferentes usuarios ao mesmo tempo, en cada instante poderemos ter distintas transaccións que manipulen a base de datos ao mesmo tempo.

As transaccións especifican un nivel de illamento que define o grao en que se debe illar unha transacción das modificacións de recursos ou datos realizadas por outras transaccións. En teoría, toda transacción debe estar completamente illada doutras transaccións, pero na realidade, por razóns prácticas, isto pode non ser certo sempre. Os niveis de illamento descríbense en canto aos efectos secundarios da simultaneidade que se permiten, como as lecturas desfasadas ou ficticias.



O estándar SQL define catro niveis de illamento transaccional en función de tres eventos que son permitidos ou non dependendo do nivel de illamento. Estes eventos son:

- *Lectura sucia*. As sentenzas SELECT son executadas sen realizar bloqueos, pero podería usarse unha versión anterior dun rexistro. Polo tanto, as lecturas non son consistentes ao usar este nivel de illamento.
- *Lectura non repetible*. Unha transacción volve ler datos que previamente lera e encontra que foron modificados ou eliminados por unha transacción cursada.
- *Lectura fantasma*. Unha transacción volve executar unha consulta, devolvendo un conxunto de rexistros que satisfán unha condición de busca e encontra que outros rexistros que satisfán a condición foron inseridos por outra transacción cursada.

Os niveis de illamento SQL defínense sobre a base de se permiten cada un dos eventos definidos anteriormente.

Niveis de illamento:

Nivel de illamento	Comportamento Permitido		
	Lect. sucia	Lect. non repetible	Lect. fantasma
Lectura non confirmada	SI	SI	SI
Lectura confirmada	NON	SI	SI
Lectura repetible	NON	NON	SI
Serializable	NON	NON	NON

A sentenza para controlar o nivel de illamento en SQL é:

SET TRANSACTION ISOLATION LEVEL {READ UNCOMMITTED   READ
--



COMMITTED  REPEATABLE READ SERIALIZABLE }
---

- READ UNCOMMITTED. Especifica que as instrucións poden ler filas que foron modificadas por outras transaccións pero aínda non se confirmaron.
- READ COMMITTED. Especifica que as instrucións non poden ler datos que fosen modificados.
- REPEATABLE READ. Especifica que as instrucións non poden ler datos que foron modificados pero aínda non confirmados por outras transaccións e que ningunha outra transacción pode modificar os datos lidos pola transacción actual ata que esta finalice.
- SERIALIZABLE. Especifica que as instrucións non poden ler datos que fosen modificados, pero aínda non confirmados, por outras transaccións. Ningunha outra transacción pode modificar os datos lidos pola transacción actual ata que a transacción actual finalice.



## **21.8 BIBLIOGRAFÍA**

- Codd, E.F. *"A Relational Model of Data for Large Shared Data Banks"*. En: 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, marzo 1986)
- de Miguel e M. Piattini. *"Fundamentos y Modelos de Bases de Datos"*. Ed. RA-MA 1999. ISBN 978-84-78-97361-3
- M. Piattini, E. Marcos, C. Caleiro e 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 Actas do congreso, 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. Oitava Edición. Pearson. Prentice Hall.



- Piattiani, Esparza Marcos, Caleiro 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

Colexiado do CPEIG



## **22. SISTEMAS DE XESTIÓN DE CONTIDOS. SISTEMAS DE XESTIÓN DOCUMENTAL. XESTIÓN DO COÑECEMENTO. OS SISTEMAS DE INFORMACIÓN XEOGRÁFICA (SIX).**



## **Tema 22: Sistemas de xestión de contidos. Sistemas de xestión documental. Xestión do coñecemento. Os sistemas de información xeográfica (SIX)**

---

### **ÍNDICE**

#### **22.1. Sistemas de xestión de contidos**

- 22.1.1. Introducción aos sistemas de xestión de contidos.*
- 22.1.2. Funcionalidade dos SXC*
- 22.1.3. Arquitectura xeral dos sistemas de xestión de contidos.*
- 22.1.4. Categorías*
- 22.1.5. Criterios de valoración*
- 22.1.6. JOOMLA!*
- 22.1.7. WORDPRESS*
- 22.1.8. DRUPAL*

#### **22.2. Sistemas de xestión documental**

- 22.2.1. Definición de sistemas de xestión documental*
- 22.2.2. Funcións da xestión documental*
- 22.2.3. Ciclo de vida dos documentos*
- 22.2.4. Beneficios da xestión documental*

#### **22.3. Xestión do coñecemento**

- 22.3.1. Definición de xestión do coñecemento*
- 22.3.2. Cuestións sobre xestión do coñecemento*

#### **22.4. Sistemas de información xeográfica SIX**

- 22.4.1. Arquitectura dun SIX*
- 22.4.2. Clasificación dos SIX*
- 22.4.3. Ámbitos de aplicación*

#### **22.5. Bibliografía**



## **22.1. Sistemas de xestión de contidos**

### **22.1.1. Introducción aos sistemas de xestión de contidos.**

Os sistemas de xestión de contidos, en diante SXC (CMS en inglés), son un tipo especial de software orientado á creación, administración e distribución de contidos dixitais. Os SXC proporcionan unha estrutura ou *framework* para dar soporte a tarefas básicas e complexas de xestión de contido. Están principalmente orientados para servir como marco de publicación de contidos na rede a través de web. O éxito deste tipo de sistemas radica fundamentalmente na facilidade de uso, establecendo mecanismos sinxelos para a creación de contidos, a súa actualización, a súa administración e categorización, e a súa publicación. Proporcionar facilidade no manexo de contidos implica outorgar un maior dinamismo no fluxo da información.

Unha das súas principais características é que permiten separar o contido da presentación, cuestión que proporciona versatilidade á hora de realizar cambios no deseño. Ademais, achegan ferramentas que permiten descentralizar a publicación de contidos na web.

Un aspecto clave na xestión de contidos é a categorización da información. A capacidade de establecer mecanismos que permitan localizar a información útil é outra das características propias dos sistemas de xestión de contidos. Esta capacidade baséase no uso de metadatos que serve para proporcionar información engadida aos contidos publicados, e que é utilizado polos buscadores e clasificadores de información.

### **22.1.2. Funcionalidade dos SXC**

A funcionalidade xeral ofrecida polos SXC pode agruparse en cinco bloques



fundamentalmente:

- **Creación de contido:** Realízase de forma sinxela. Os SXC achegan ferramentas para que os creadores sen coñecementos técnicos en páxinas web poidan concentrarse no contido. A forma habitual consiste en proporcionar un editor de texto WYSIWYG, no que o usuario ve o resultado final mentres escribe. O uso deste tipo de editores é moi sinxelo. O acceso a estes é moi cómodo, xa que só se require para iso un equipo con acceso á internet e un navegador web.
- **Xestión de contido:** Todo o contido creado almacénase na base de datos que utiliza o sistema. Na propia base de datos é onde tamén se gardan datos relacionados coa estrutura da web ou os usuarios autorizados.
- **Xestión de usuarios:** A maioría dos SXC presentan unha xestión de usuarios na que cada un conta con diferentes permisos para xestionar o contido. Dependendo dos permisos de usuarios, pódense encontrar distintos roles que van dende o administrador xeral da plataforma ata o usuario final que consulta a información.
- **Publicación de contido:** Unha vez creado o contido, os SXC proporcionan diferentes mecanismos para proceder á súa publicación. Pódesele asignar unha data de publicación ou ben pódese publicar directamente. Na publicación do contido, o aspecto que terá vén marcado polo patrón marcado para a sección onde se encontre a información, que habitualmente se corresponde cun conxunto de estilos predefinidos. Esta separación entre contido e forma representa unha característica moi importante dos SXC dado que permite que se poida modificar o estilo dun portal web sen necesidade de modificar o contido.
- **Presentación de contido:** Os SXC xestionan automaticamente a accesibilidade do web, proporcionando mecanismos de adaptación ás necesidades de cada usuario e ademais son perfectamente compatibles coa maioría dos navegadores web existentes. O sistema encárgase de xestionar outros aspectos



como os menús de navegación, engadindo ligazóns de forma automática. Tamén xestionan todos os módulos, internos ou externos, que sexan incorporados ao sistema.

Dentro do ciclo de actividades que se corresponde coa funcionalidade dos SXC, é necesario definir un conxunto de roles ou usuarios aos que se asocian unha serie de tarefas:

1. Autor: que pode ser calquera membro usuario do sistema que desexe publicar contido.
2. Publicador: que revisa esa información e autoriza a súa publicación en tempo e forma axeitadas.
3. Administrador do sistema: desempeña funcións técnicas que consisten en optimizar o rendemento e arquitectura do sistema. Poden ademais propoñer os modelos de deseño e os sistemas de categorización máis adecuados, de acordo cos provedores de información e de manter o sistema en constante mellora e actualización.

A definición de roles ou usuarios é dependente da plataforma final que se pode escoller como SXC, e van dende os presentados anteriormente ata unha definición de gran fino, onde se especifican roles intermedios e se diversifican máis as tarefas. O sistema de xestión de contidos controla e axuda a manexar cada paso deste proceso, incluídos os labores técnicos de publicar os documentos a un ou máis sitios.

### **22.1.3.      Arquitectura xeral dos sistemas de xestión de contidos.**

A arquitectura destes sistemas é modular, o que proporciona un marco de desenvolvemento que facilita a implantación de novas funcionalidades. Neste sentido, os SXC incorporan unha gran variedade de módulos que permiten estender o funcionamento do sistema. Existen módulos para a xestión integral dun sitio web,



para a xestión de páxinas xeradas dinamicamente, e outros módulos que posibilitan a personalización do sistema por parte do usuario.

Os SXC, como software, poden definirse como un *framework* que habitualmente consta de dúas partes diferenciadas:

- *Backend* ou parte administrativa: A través do *backend* pódense controlar todos os aspectos relativos á configuración do *framework*, a administración do contido (creación, categorización, edición, publicación, eliminación), a personalización do contorno de consulta (*frontend*), actualización e configuración de novas funcionalidades.
- *Frontend* ou parte pública: A través do *frontend* pódense consultar os contidos publicados, acceder ás funcionalidades proporcionadas para os usuarios configuradas dende o *backend*, e tamén serve para reunir certos datos de entrada.

A separación do sistema en *frontends* e *backends* é un tipo de abstracción que axuda a manter as dúas principais partes do SXC separadas. Dentro da arquitectura de SXC considérase a existencia dunha ou varias bases de datos, responsables fundamentalmente da persistencia do contido publicado a través do SXC e de todos os datos relativos á configuración do sistema.

O fluxo básico é que xestor responde ás solicitudes de páxinas que se formulan dende os lectores e recupéraas a base de datos, compoñendo os modelos definidos e devolvendo ao servidor web o contido final que este lle ofrece o lector.

Os SXC, ao seren aplicacións web, execútanse no servidor web onde estean aloxados. Dependendo das tecnoloxías utilizadas para o desenvolvemento do SXC, a complexidade do servizo web que soporta a plataforma será maior. O acceso aos SXC realízase a través dos navegadores web. Cando un usuario realiza unha petición dunha páxina, o xestor de contidos é o encargado de interactuar co servidor para xerar unha páxina dinámica, cun formato definido, e cun contido que se extrae da base de datos.



#### 22.1.4. Categorías

En canto á categorización dos SXC, non existe unha clasificación estrita, senón máis ben, categorizacións en función de determinadas características propias dos sistemas. Así pois, podemos clasificalos en función da linguaxe de programación na que se desenvolvan, segundo a súa licenza (código aberto ou software privativo), e mesmo pola súa funcionalidade (Blogs, Wikis, Foros...)

#### 22.1.5. Criterios de valoración

Á hora de proceder á implantación dun SXC, é necesario ter en conta unha serie de criterios que nos servirán para establecer, en función da situación, cal é o SXC máis axeitado. Estes criterios son os seguintes:

- **Código aberto ou código propietario:** No caso dos SXC de tipo privativo, é dicir, os comercializados por empresas baixo licenzas restritivas, non se permite o acceso ao código fonte por parte de terceiros. Non obstante cos SXC de código fonte aberto, esta limitación non existe, dado que os desenvolvedores si que permiten o acceso libre e a modificación do código. Esta característica é moi importante posto que poder dispoñer do código fonte proporciona poder modificar o produto, achegándolle novas funcionalidades ou mesmo corrixindo posibles erros. Esta é unha faceta moi importante relacionada coa evolución do produto. Outra vantaxe dos SXC de código libre é o custo, posto que este tipo de xestores de contido son gratuítos, sen ningún custo de licenzas. No caso dos SXC comerciais, o custo pode chegar a ser moi elevado, sobre todo para un particular. Ademais de todo isto, arredor dos xestores de contido de código libre adoitan existir comunidades de usuarios que comentan as súas experiencias co uso destes sistemas, achegan novidades e desenvolven novas



funcionalidades.

- **Arquitectura técnica:** O SXC ten que ser fiable, robusto e adaptable a futuras necesidades. Para iso, é preciso ter en conta cal é a arquitectura do sistema, que tecnoloxías se utilizaron, e analizar o deseño da plataforma, co obxectivo de poder emprender ampliacións nas funcionalidades ofrecidas en caso de ser necesario. Tamén é conveniente que permita separar contido, presentación e estrutura, de acordo cos estándares establecidos para o web. Para iso, é altamente recomendable decantarse polo uso de sistemas que fagan uso de motores de modelos, así como uso de definicións de estilos baseadas en follas de estilo (CSS).
- **Grao de desenvolvemento:** É moi importante que a ferramenta seleccionada teña un grao de madureza axeitado para poder desenvolver a funcionalidade requirida, e que se dispoña de módulos ou compoñentes para poder engadirlle funcionalidade.
- **Soporte, posición no mercado e opinións:** A ferramenta ten que ter soporte tanto polos creadores coma polos desenvolvedores. É fundamental que unha ferramenta sexa coñecida por moitos usuarios e expertos; este feito pode axudar a posibles usuarios a decidirse polo SXC en cuestión. Habitualmente as grandes comunidades de usuarios e desenvolvedores encóntranse arredor dos SXC libres, proporcionando un marco ideal para o rápido desenvolvemento destes sistemas así como do seu mantemento.
- **Usabilidade:** Partindo da premisa de que existen diferentes roles con diferenciación clara de tarefas, debemos de ter en conta que determinados perfís de usuarios non teñen por que ter coñecementos técnicos. Iso implica que o SXC ten que ser doado de aprender e utilizar.
- **Accesibilidade:** Temos que ter en conta que no momento en que traballamos con SXC, o sistema debe estar preparado para o uso por parte da maior cantidade de usuarios posible. Polo tanto, é sempre recomendable que o portal



web cumpra un estándar de accesibilidade.

- **Velocidade de descarga:** É importante que as páxinas solicitadas polos usuarios se carguen rápido. A natureza das páxinas dinámicas e a separación de estrutura, presentación e contido contribúen a que as páxinas sexan máis lixeiras.

## 22.1.6. JOOMLA!

### 22.1.6.1 Introducción

Joomla! é un dos SXC con maior impacto e distribución. Isto foi proporcionado polo feito de ser un sistema de código aberto desenvolvido nunha das linguaxes maioritarias para a internet como é PHP. Está recollido baixo licenza GPL e actualmente conta cunha das maiores comunidades de usuarios e desenvolvedores. Este administrador de contidos pode traballar na internet ou nas intranets e require dunha base de datos MySQL, así como dun servidor web, preferiblemente HTTP Apache.

### 22.1.6.2 Arquitectura

En canto ao seu deseño, dende o punto de vista de desenvolvemento, Joomla! está programado en PHP baixo un patrón Modelo-Vista-Controlador, integrando un motor de modelos, e permitindo separar totalmente a capa de presentación da lóxica dos datos. Esta modularidade proporciona unha gran facilidade para estender o sistema. As funcionalidades en Joomla! engádense nos módulos ou compoñentes. Estes módulos ou compoñentes son partes do sistema que se executan de forma independente, baixo o patrón MVC, e intégranse perfectamente dentro do SXC principal. Existen repositorios libres da comunidade de usuarios e desenvolvedores onde se poden encontrar centos de módulos gratuítos para estender as funcionalidades de Joomla!. Non obstante, esta cota tamén representa un modelo de negocio para moitas empresas que proporcionan os seus produtos en forma de



módulos para Joomla!. Así pois, o deseño patronizado mediante MVC e o uso de tecnoloxías maduras como PHP e MySQL fai que resulte relativamente sinxelo ampliar as funcionalidades deste SXC a partir da execución propia de módulos que satisfagan algunha funcionalidade concreta.

O SXC Joomla! presenta unha arquitectura en tres niveis: nivel de extensións, nivel de aplicación e de desenvolvemento.

O nivel superior, de extensións, componse de extensións do marco de desenvolvemento de Joomla! e das súas aplicacións. Nesta capa sitúanse os módulos, compoñentes e modelos (templates). O nivel do medio, de aplicación, consiste nunha serie de aplicacións que se estenden do core para dar soporte aos módulos e compoñentes. Executa tamén as aplicacións necesarias para a administración (*backend*) así como a arquitectura principal do *frontend*. O nivel inferior, correspondente ao de desenvolvemento, consta do conxunto de clases PHP que o forman, as bibliotecas que son utilizadas polo marco de desenvolvemento ou que se instalan para o uso polos desenvolvedores e finalmente os *plugins*, que estenden a funcionalidade.

Algunhas características básicas que se inclúen en Joomla! son: sistema adaptado para mellorar o rendemento web, versións imprimibles de páxinas e xeración directa en PDF, módulos de flash con noticias, integración con blogs e foros, módulos nativos para a xestión de enquisas, calendarios, busca no sitio web e internacionalización da linguaxe. O nome de Joomla! provén dunha pronunciación fonética para anglófonos da palabra suahili jumla, que significa "todos xuntos" ou "como un todo". Foi escollido como unha reflexión do compromiso do grupo de desenvolvedores e a comunidade do proxecto.

#### **22.1.6.3 Comunidade de desenvolvemento**

A comunidade de Joomla!, para o desenvolvemento das súas múltiples fronteiras, usa diferentes formas de comunicación como son o uso de salas de chat a través de IRC, participación en foros especializados, listas de correo, "wikis" e blogs. A xestión de administración principal do proxecto esta delegada ao grupo principal, coñecido



como "Core Team". Este grupo de desenvolvedores representa a columna vertebral do proxecto, xa que son os encargados de guiar a Joomla! dentro do movemento de código aberto. Este grupo esta composto por diferentes perfís, con variadas experiencias e totalmente multidisciplinar. Leva activo dende o ano 2005, aproximadamente co nacemento oficial de Joomla!. A súa responsabilidade principal radica na organización con respecto a Joomla! na súa estrutura funcional como organización e non unicamente na programación do sistema de xestión de contidos.

Ademais do grupo principal ou core team, existen tamén outros grupos que se crearon para enriquecer o coñecemento que a comunidade Joomla! proporciona. Cada un dos grupos especialízase nun aspecto específico de Joomla! que é importante para a expansión e desenvolvemento. O grupo principal non pode estar en cada discusión destes temas, por iso existe unha estrutura xerarquizada onde un responsable de cada grupo de desenvolvemento se encarga de comunicarse de forma directa co grupo principal.

Ademais do traballo da comunidade de usuarios e desenvolvedores, existe unha organización que proporciona soporte para moitos aspectos do proxecto. É a Open Source Matters Inc (OSM), que é unha organización sen ánimo de lucro de orixe estadounidense. O obxectivo fundamental desta organización é dar soporte á parte legal e financeira do proxecto de código aberto Joomla! Recentemente a OSM incorporouse como unha organización sen ánimo de lucro de Nova York, proporcionando unha garantía de continuidade para o proxecto e actividades futuras, proporcionando o soporte necesario para que as comunidades de usuarios e desenvolvedores poidan seguir participando.

#### **22.1.6.4 Principais características**

As principais características que fixeron de Joomla! un dos mellores SXC do momento son as seguintes:

- **Usabilidade da súa interface:** Esta característica faise principalmente notoria na interface de administración. O obxectivo fundamental é que calquera



persoa sen coñecementos técnicos poida ter control do sistema para acurtar a curva de aprendizaxe das tarefas administrativas.

- **Xestión de contido:** O sistema presenta unha estrutura xerárquica para xestionar o contido baseada en agrupacións de artigos (a unidade fundamental de contido) que se organizan en seccións e categorías. Permite crear menús e submenús, subir imaxes e ficheiros, así como syndicar de forma nativa noticias mediante RSS.
- **Xestión de usuarios:** Existen dous tipos de usuarios básicos: os usuarios invitados, que son aqueles que acceden ao portal navegando, que non posúen ningunha conta no sistema e que habitualmente están capacitados para consultar os artigos, e os usuarios rexistrados que son aqueles que dispoñen dunha conta (nome de usuario/contrasinal) para autenticarse no sitio e acceder a funcionalidades específicas. Dentro dos usuarios rexistrados existen distintos roles cada un cunha serie de privilexios. A xestión das contas e permisos dos usuarios en Joomla! pódese facer de forma nativa, ou ben facendo uso dun sistema externo como LDAP.
- **Personalizable:** Grazas á combinación do uso de estándares, e ao deseño desadaptado proporcionado polo patrón MVC, a presentación do contido pódese personalizar de forma moi sinxela. A aparencia do *frontend* é perfectamente modificable grazas ao uso de modelos. Os modelos poden modificarse de xeito sinxelo permitindo que se adapten ás necesidades do sistema.
- **Extensibilidade:** Como xa se comentou con anterioridade, unha das principais características que definen a este software é a modularidade da plataforma, o que permite o desenvolvemento e integración dunha gran cantidade de módulos e compoñentes que permiten estender as funcionalidades do sistema. A facilidade no desenvolvemento destas pezas software supuxo que actualmente exista un gran número de extensións e módulos existentes, programados pola comunidade de usuarios, que aumentan as posibilidades da



aplicación con novas características e que se integran doadamente no sistema. Como exemplo de extensións dispoñibles, cítanse xestores de documentos, galerías de imaxes multimedia, motores de comercio e venda electrónica, calendarios, etc.

- **Multiplataforma:** Debido á utilización de tecnoloxías libres estandarizadas, este SXC pode correr sobre calquera sistema operativo, xa sexa GNU/Linux, en Windows ou en Mac OSX. Os únicos requisitos son dispoñer na máquina dun servidor web, e dunha base de datos MySQL.

## 22.1.7. WORDPRESS

### 23.1.7.1 Introducción

A popularidade crecente dos blogs ou bitácoras como medio popular para difundir contido, tivo tamén cabida dentro do desenvolvemento dos sistemas de xestión de contidos. WordPress é un SXC enfocado precisamente á creación de blogs, especialmente orientado a ofrecer comodidade para a ardua tarefa de manter os sitios web periodicamente actualizados.

WordPress está desenvolvido en PHP e MySQL, baixo licenza GPL, o que tamén implica que é software libre e, polo tanto, o seu código é modificable e adaptable. Neste sentido, comparte moitas das vantaxes que esta filosofía outorga a outros SXC como Joomla!.

O fundador do proxecto de WordPress é Matt Mullenweg. WordPress foi creado a partir do desaparecido b2/cafelog e actualmente é o SXC máis popular orientado á creación de blogs. As causas do seu enorme crecemento están relacionadas coa súa licenza libre, a facilidade de uso e as características que proporcionan en xeral os sistemas de xestión de contido.

Ao igual que a maioría dos SXC máis populares, WordPress está implantado baixo un patrón MVC. Sumado a isto, ao presentarse como produto libre, posibilitase o labor da enorme comunidade de desenvolvedores para revisións e a implantación de



módulos que engadan novas funcionalidades. Este é outro dos factores que favoreceu a súa crecente expansión.

Como acontece con Joomla!, sumado ao traballo da comunidade libre de desenvolvedores, o liderado do proxecto recae sobre unha entidade chamada Automattic.

#### **23.1.7.2 Características**

Algunhas características básicas que definen WordPress son as seguintes:

- Proporciona un sistema de publicación web baseado en entradas ordenadas por data.
- A estrutura e deseño visual do sitio depende dun sistema de modelos, que é independente do contido en si. Separación da capa de presentación.
- Apóstase decididamente polas recomendacións do W3C, pero é dependente sempre do modelo que se usa.
- A xestión e execución corre a cargo do sistema de administración cos plugins e os widgets que usan os modelos.
- Como noutros SXC, existe unha xerarquía de usuarios/roles, e WordPress permite múltiples autores ou usuarios.
- Aínda que o sistema está orientado a configurar un único blog ou bitácora por sistema instalado, permite múltiples blogs ou bitácoras.
- Dispón de múltiples ferramentas para organizar o contido (artigos) en categorías.
- Dispón de compoñentes visuais para a edición dos artigos (compoñentes WYSIWYG "What You See Is What You Get")
- Permite comentarios e ferramentas de comunicación entre blogs.



- Dispón de funcionalidades necesarias para a sindicación de contidos nos principais formatos estándar (RSS 2.0 e ATOM 1.0).
- Suba e xestión de adxuntos e arquivos multimedia.
- Sistema de busca integrada dentro da plataforma.

## 22.1.8. DRUPAL

### 23.1.8.1 Introducción

Outro dos máis coñecidos no mundo dos SXC é Drupal. Drupal é un sistema de xestión de contido, similar en canto a súa arquitectura e orientación a Joomla!. É un sistema modular multipropósito e moi configurable. Permite xestionar e publicar artigos, imaxes, ou outros arquivos. O seu deseño modular permite integrar unha gran cantidade de servizos diferentes como foros, enquisas, votacións, blogs e administración de usuarios e permisos.

Drupal é un sistema dinámico. Isto implica que, como en todos os anteriores, o contido se almacena de forma persistente nunha base de datos, e as páxinas que se demandan dende o *frontend* de consulta son xeradas dinamicamente. O sistema encárgase de acceder ao contido da base de datos e montar a páxina que subministrará ao servidor web.

É un programa libre, con licenza GNU/GPL, escrito en PHP baixo un patrón de deseño MVC, o que de novo facilita a súa modificación e adaptabilidade, potenciando o traballo da extensa comunidade de usuarios.

Algunhas características propias de Drupal no que atinxe ao desenvolvemento son a calidade do seu código e das páxinas xeradas. Fai especial fincapé no respecto dos estándares da web, e unha énfase particular na usabilidade e consistencia de todo o sistema.

O deseño de Drupal faino especialmente idóneo para construír e xestionar



comunidades na internet. Non obstante, grazas ás súas características de flexibilidade e adaptabilidade, así como a gran cantidade de módulos adicionais dispoñibles, Drupal convértese nun SXC de propósito xeral, capaz de adecuarse a moitos tipos diferentes de sitio web.

### 23.1.8.2 Características

As súas características principais son as seguintes:

- **Extensibilidade:** Grazas á extensa comunidade de usuarios e desenvolvedores, dispón dunha gran cantidade de módulos con distintas funcionalidades: foro, galería, enquisas, boletín de noticias, correo electrónico, chat, etc.
- **Código aberto:** Ao estar dispoñible o código fonte baixo os termos da licenza GNU/GPL, é posible estender ou adaptar Drupal segundo as necesidades.
- **Personalización:** A capa de presentación está perfectamente illada do resto do sistema, facendo a aparencia totalmente configurable en función das preferencias dos usuarios.
- **Xestión de usuarios:** Como todo SXC, dispón dunha xerarquía de usuarios/roles e dun sistema interno para xestionalos e permitir a autenticación. Esta última pode facerse ben de forma local ou utilizando un sistema de autenticación externo.
- **Xestión de contidos:** Proporciona un sistema de control de versións, que permite seguir e examinar todas as actualizacións do contido. Dispón dun sistema de temas ou modelos que permite separar o contido do sitio da presentación. Tamén conta coa posibilidade de exportar o contido en formato RDF/RSS para ser utilizado por outros sitios web.
- **Multiplataforma:** Pode funcionar con calquera servidor web (Apache, Microsoft IIS) e en sistemas como Linux, Windows, Solaris, BSD e Mac OS X.



Ao estar implantado en PHP, é portable.

## **22.2. Sistemas de xestión documental**

### **22.2.1. Definición de sistemas de xestión documental**

Un sistema de xestión documental defínese como un conxunto de elementos e relacións entre eles, que ten o propósito de normalizar, controlar e coordinar todas as actividades e procesos que afectan en calquera medida aos documentos xerados no transcurso da actividade dunha organización. As operacións máis habituais que se realizan sobre estes documentos abranguen todo o seu ciclo de vida, dende a súa creación ata o seu almacenamento e posta a disposición dos usuarios.

Ademais, un sistema de xestión documental ten que satisfacer o seguinte:

- Conservar os atributos básicos dos documentos que lles confiren o seu valor informativo, legal e probatorio.
  - Orixinalidade
  - Autenticidade
  - Integridade
  - Veracidade
- Manter a organización dos documentos integrados nun contexto. Isto implica conservar unha interrelación cos outros documentos que xorden da mesma función, actividade, que son producidos polo mesmo departamento ou organismo, que forman parte da mesma serie, etc.

### *Software de xestión documental*

O software de xestión documental abrangue todos aqueles programas software



deseñados para xestionar grandes cantidades de documentos. Nestes documentos non necesariamente debe existir organización dentro dos seus contidos, de feito, o máis común é que o contido destes documentos non garde unha organización clara.

Existen diversos métodos que usados en combinación coas bibliotecas de documentos e unha serie de índices, permiten un acceso rápido á información almacenada neses documentos, os cales, habitualmente están comprimidos e adoitan almacenar, ademais do texto plano, outros contidos multimedia como imaxes, vídeos, etc.

Entre os obxectivos que se perseguen á hora de implantar un sistema de xestión documental cabe mencionar:

- Resaltar a importancia que teñen os documentos dentro de calquera tipo de organización, pública ou privada.
- Facilitar a recuperación de información de forma rápida, exacta e efectiva.
- Analizar a produción documental, para evitar documentos innecesarios ou que non paga a pena almacenar pasado certo tempo.
- Conseguir que os arquivos sexan útiles e significativos como unidades de información non só dentro da empresa senón tamén externamente.

Antes de montar un sistema de xestión documental é necesario realizar unha serie de consideracións previas que podemos agrupar nas seguintes categorías:

- Administrativas: céntrase todo o que pode influír na administración da empresa.
- Económicas: refírese á avaliación do aforro que xera a xestión de documentos.

Para a implantación deste tipo de sistema, é necesario tamén realizar un diagnóstico e unha avaliación dos requisitos tanto técnicos coma administrativos.

### **22.2.2. Funcións da xestión documental**



As principais funcións da xestión documental son:

- Almacenamento
- Captura
- Conservación
- Consulta
- Creación
- Difusión
- Eliminación
- Ingreso
- Uso

### 22.2.3. Ciclo de vida dos documentos

O ciclo de vida dun documento abrangue todas as fases polas que un documento pasa, dende que se crea ata que se archiva ou elimina.

Os documentos poden ter distintos valores que son:

- *Valor primario (administrativo)*: o seu propósito é deixar constancia dunha actividade.
  - Valor fiscal ou contable: acreditar o cumprimento das obrigas contables ou tributarias.
  - Valor legal ou xurídico: a súa finalidade é, entre outras, servir de proba ante a lei.
- *Valor secundario*:
  - Valor informativo: o seu propósito é servir de base para a reconstrución de calquera actividade realizada.
  - Valor histórico: serve de fonte para a investigación histórica.

As distintas fases que atravesará un documento son:



- Arquivo de oficina (documentación activa): fase na cal os documentos son creados ou recibidos por algún departamento, sobre os cales se pode realizar unha serie de operacións de edición.
- Arquivo xeral (documentación semiactiva): nesta etapa a principal función é a consulta da documentación e a actividade que recibe este tipo de documentación é menor que no arquivo de oficina.
- Arquivo histórico (documentación inactiva): nesta etapa a documentación só ten utilidade como fonte de información histórica. As consultas que recibe son menores.

#### **22.2.4. Beneficios da xestión documental**

Se se realiza unha boa xestión dos documentos, repercute na empresa cunha serie de beneficios, como son:

- Obter información precisa das actividades da empresa que sirva de apoio para actividades futuras, toma de decisións, etc.
- Facilitar a realización das actividades da empresa.
- Documentar as políticas e o proceso de toma de decisións.
- Garantir a continuidade da empresa en caso de fallo masivo nos sistemas, catástrofe, etc.
- Cumprir cos requisitos legais que existen con algún tipo de ficheiros de datos.
- Almacenamento de evidencias das actividades relacionadas coa empresa e entidades externas.
- Manter un histórico da evolución da entidade.
- Centralizar o almacenamento de documentos.
- Facilitar a prestación de servizos aos usuarios da empresa.



## **22.3. Xestión do coñecemento**

### **22.3.1. Definición de xestión do coñecemento**

Non existe unha definición universalmente aceptada da xestión do coñecemento. Non obstante, existen numerosas definicións de diversos expertos. En xeral, a xestión do coñecemento é a conversión do coñecemento tácito en coñecemento explícito e o seu intercambio dentro da organización. A xestión do coñecemento é o proceso mediante o cal as organizacións xeran valor dos seus activos intelectuais. Definida deste xeito, faise evidente que a xestión do coñecemento ten que ver co proceso de identificación, adquisición, distribución e mantemento dos coñecementos que son esenciais para a organización.

Se se considera a xestión do coñecemento nun contexto máis amplo, daquela existen múltiples definicións, non obstante, todas elas apuntan á mesma idea, aínda que cada unha se centre nun aspecto particular da xestión do coñecemento.

- Unha definición orientada aos resultados pode afirmar que a xestión do coñecemento é "ter o coñecemento axeitado no lugar correcto, no momento axeitado e no formato correcto".
- Unha definición orientada ao proceso pode afirmar que a xestión do coñecemento consiste "na xestión sistemática dos procesos polos cales o coñecemento se identifica, se crea, se une, se comparte e se aplica".
- Unha definición orientada á tecnoloxía pode presentar unha fórmula de xestión do coñecemento como "Business Intelligence + motores de busca + axentes intelixentes".



### 22.3.2. Cuestións sobre xestión do coñecemento

Existen dous aspectos principais na xestión do coñecemento, que son a *xestión da información* e a *xestión das persoas*. Visto dende esta perspectiva, a xestión do coñecemento é, por un lado, a información e, por outro, a xente.

A maioría de empresarios e directivos están familiarizados co manexo de información a longo prazo. Este termo asóciase coa xestión do coñecemento en relación cos obxectos, que son identificados e controlados polos sistemas de información. A práctica da xestión da información foi aceptada amplamente cando os executivos se decataron de que a información era un recurso importante e de que debía ser manexado correctamente para que as empresas poidan mellorar a súa competitividade.

Como consecuencia do crecemento da práctica da xestión da información, os conceptos de "*análise da información*" e "*planificación da información*", desenvolvéronse, proporcionando ferramentas adicionais para os profesionais.

Na vertente teórica a xestión da información evolucionou converténdose en xestión do coñecemento. Na práctica, a xestión do coñecemento implica, entre outros, a identificación e mapeamento dos activos intelectuais dunha organización. Isto significa, basicamente, a identificación de quen sabe que dentro da empresa. Cando se mira dende esta perspectiva, a xestión do coñecemento pode ser considerada como un proceso de realización dunha auditoría dos activos intelectuais. Non obstante, a xestión do coñecemento vai máis alá deste nivel da cartografía e tamén implica a creación de coñecemento para obter vantaxes competitivas e a conversión de grandes cantidades de datos da organización en información de doado acceso.



Demostrouse unha e outra vez que cando o coñecemento se xestiona ben, hai unha redución significativa no tempo necesario para completar as tarefas e evítase a duplicación innecesaria.

Como xa se comentou anteriormente, un aspecto da xestión do coñecemento é a xestión de persoas. Basicamente, trátase da xestión do coñecemento tácito que reside dentro das cabezas das persoas. Na práctica implica a xestión do coñecemento que existe xunto aos procesos organizativos que implica unha serie complexa de capacidades dinámicas, coñecementos técnicos e outras capacidades relacionadas co coñecemento.

Co fin de xestionar de forma eficaz as persoas que posúen o coñecemento tácito que se desexa, é esencial ter en conta a súa diversidade cultural e os valores sociais, actitudes, aspiracións e gustos. Se isto se pode facer con éxito, pode conducir á creación de novos coñecementos que doutro xeito non se pode lograr mediante a xestión de información por si soa.

A pesar da importancia dos dous aspectos da xestión do coñecemento, a cal está ben recoñecida por moitas organizacións, o verdadeiro potencial da xestión do coñecemento aínda queda por acadarse. De feito, non todas as organizacións con algún sistema de xestión do coñecemento son conscientes de que teñen estes sistemas.

A maioría das organizacións teñen algún tipo de sistema para a xestión do coñecemento explícito, xa sexa simple ou complexa, aínda que non necesariamente se refiran a el como un sistema de xestión do coñecemento. Por outro lado, a xestión do coñecemento tácito non é común e a tecnoloxía actual baseada na xestión do coñecemento non se desenvolveu de forma plenamente eficaz para a extracción de coñecemento tácito. Aínda que o coñecemento tácito é a base de coñecemento organizacional, é algo tan persoal que é difícil de formalizar e comunicar.



Ambos os dous aspectos da xestión do coñecemento presentan dúas cuestións inmediatas:

- Facer que o coñecemento da organización sexa máis produtivo.
- Producir beneficios significativamente maiores que os previstos.

A xestión do coñecemento ofrece unha excelente oportunidade para adoptar estratexias de negocio que antes eran imposibles. Por exemplo, pódese abrir a porta á creación dunha rede case ilimitada que mellore as relacións con clientes e provedores. Na mellora de relacións cos clientes, a xestión do coñecemento fai posible o descubrimento de novos problemas e oportunidades a través do uso óptimo dos activos de coñecemento tales como o contrato de venda, os rexistros, os datos demográficos dos clientes, etc. É precisamente deste xeito que a xestión do coñecemento se pode complementar e mellorar o impacto doutras iniciativas da organización como a xestión da calidade total, o proceso de reenxeñaría de negocios, e a aprendizaxe organizacional.

É evidente, a partir desta discusión, que as iniciativas de xestión do coñecemento se poden aplicar nunha variedade de ámbitos para lograr resultados superiores en case calquera tipo de organización. E é posible alcanzar estes resultados, independentemente do nivel de dispoñibilidade tecnolóxica ou o sector do mercado en cuestión.

#### **22.4. Sistemas de información xeográfica SIX**

Como información espacial (xeográfica, xeorreferenciada ou xeodatos) referímonos a todo tipo de información relativa a sucesos ou elementos para a cal se inclúe unha referencia á súa localización, a cal está situada sobre ou nas inmediacións da superficie da Terra. A forma de referenciar a posición destes elementos ou estes



sucesos pode realizarse de distintas formas, mediante unha simple dirección postal, con coordenadas xeográficas (lonxitude e latitude) ou con coordenadas cartesianas nalgún sistema de referencia cartográfico.

A maior parte da información en formato electrónico almacenada actualmente en sistemas de todo tipo é información espacial ou que podería sêlo. O porque deste auxe da información espacial encontrámolo nunha serie de características que xustifican o interese de asociar a unha información a referencia da súa localización.

Por unha parte temos a calidade da información espacial para a súa representación en forma gráfica e simbólica mediante mapas. Os mapas son un sistema de comunicación que foi utilizado dende as primeiras civilizacións e co que está familiarizado practicamente todo o mundo. Ademais os mapas tiveron grande importancia ao longo da historia militar, económica e política das nacións, polo que foron considerados sempre como un recurso clave a cuxo desenvolvemento se dedicaron importantes esforzos.

Por outra parte, temos a capacidade que posúe a información espacial para integrar conxuntos de información que doutra forma serían inconexos mediante a aplicación das relacións espaciais de coincidencia, proximidade ou inmediatez inherentes a esa localización espacial. Esta característica é probablemente a que maior potencial lle outorga á información espacial, constituíndo a base da análise espacial.

A primeira manifestación dos sistemas de información xeográfica podémolos encontrar, como se comentaba anteriormente, nos mapas, non obstante, xa en épocas máis recentes, as contribucións das tecnoloxías da información no ámbito da cartografía foron moi importantes. Pódense destacar aqueles avances destinados á mellora dos procesos de produción cartográfica ou os orientados á explotación e análise da información cartográfica.

No que atinxe á produción cartográfica, actualmente cóntase con técnicas moi depuradas para a produción de mapas en todas as súas fases, dende a captura de



datos (fotogrametría aérea, imaxes de satélite, teledetección, telemetría láser, GPS, etc), ata os diferentes procesos que compoñen a fase de elaboración da cartografía. Estas técnicas permitiron non só notables melloras na calidade, diversidade e flexibilidade dos produtos cartográficos, senón que fixo posible dispoñer de información cartográfica moi actualizada

No tocante á análise da información xeográfica é necesario destacar en primeiro lugar as importantes limitacións prácticas que presentan os mapas tradicionais para a súa utilización en análise mediante técnicas manuais. A superación destas limitacións foi a motivación inicial para o desenvolvemento dos sistemas de información xeográfica, SIX (ou GIS de acordo coa terminoloxía anglosaxona), que se converteu na outra gran rama de contribucións das tecnoloxías da información no ámbito da cartografía.

O desenvolvemento dos primeiros SIX datan de finais dos anos 60 e supuxo un gran cambio na utilización da información espacial que se facía ata ese momento. De feito as técnicas e metodoloxías de análise espacial da información, que ata o momento foran pouco examinadas pola excesiva complexidade asociada aos tratamentos manuais, víronse paulatinamente melloradas e, en moito casos, empezou a ser posible a súa utilización co procesamento automatizado da información espacial en formato dixital.

Un sistema de información xeográfica está orientado á captura, manipulación, recuperación, análise, representación, etc, de información xeorreferenciada, aquela na que a posición espacial ocupada polos obxectos do mundo real que se modelizan forma parte inherente a esa información.

Os SIX gozan de grande aceptación dende as súas primeiras implantacións o cal se debe en boa medida á súa capacidade para construír modelos orientados á resolución de problemas dos que o universo de discurso se caracteriza por ter un compoñente espacial.



Estas primeiras realizacións foron impulsadas principalmente por organizacións con responsabilidades na xestión de recursos con implantación territorial como son ordenación do territorio, recursos naturais, censo, defensa, etc.

Dende estas primeiras implantacións, nos anos 60, ata a década dos 80, o desenvolvemento dos SIX produciuse dunha forma relativamente lenta debido sobre todo á capacidade e custo da tecnoloxía dixital dispoñibles naquel momento. Dende a segunda metade dos 80 prodúcese un grande auxe, tanto en diversificación dos ámbitos de aplicación desta tecnoloxía como na oferta de produtos comerciais, o que outorgou gran popularidade e difusión aos SIX en todo tipo de organizacións.

Actualmente a evolución caracterízase por unha serie de factores que impiden unha plena estabilidade do sector:

- Evolución das tecnoloxías nas que se apoian os SIX, como son, xestores de bases de datos, procesamento paralelo, visualización, etc.
- Alto grao de relación entre os SIX e a internet, que está en permanente evolución.
- A tecnoloxía dispoñible oríéntase a aplicacións que non aproveitan o seu potencial.
- Restricións institucionais que aínda impiden o acceso e utilización da información cartográfica de que se dispón a todos os niveis.

#### **22.4.1.      Arquitectura dun SIX**

Nos primeiros desenvolvementos dos SIX encontramos principalmente sistemas software pechados de gran tamaño e complexidade (SIX monolítico), que eran utilizados principalmente por grupos de usuarios reducidos cun nivel de especialización bastante elevado e, ademais, orientábanse a tarefas moi concretas e xeralmente presentaban pouca ou ningunha integración con outros sistemas. Actualmente os sistemas diferéncianse moito destes iniciais, son cada vez máis



habituais e sinxelos, non requiren de usuarios expertos para o seu manexo e permiten a integración da información con outros sistemas.

Nun sistema SIX podemos falar de arquitecturas de 3 capas, así temos:

- Capa de presentación: incorpora todas as funcionalidades que permiten a interacción entre o usuario e o sistema, para o acceso á información e presentación de resultados. Habitualmente isto tradúcese nunha GUI que facilita o acceso ás ferramentas da seguinte capa ou tamén nunha aplicación externa que poida acceder a determinadas funcións de xeoproceso.
- Capa de proceso: abrangue unha serie de ferramentas diferentes que integran o núcleo do SIX.
- Capa de xestión de datos: centraliza o acceso aos datos que se poden localizar en distintos almacéns. Ademais, integra tamén moitas das funcións que se encargan de proporcionar transparencia sobre os detalles dos datos: sistemas de proxección, formatos, transformación de coordenadas, etc.

Para implantar estas capas funcionais podemos ter os mesmos ou diferentes sistemas físicos, tendo unha gran cantidade de posibilidades. Se se fai unha desagregación completa, cada unha destas capas residirá nun ou máis servidores diferentes, adaptado ás necesidades específicas do ámbito de implantación do SIX.

No mercado dos SIX comerciais, estanse a asumir cada vez mais unha serie de estándares de facto que foron xurdindo nos últimos anos debido ás tecnoloxías de compoñentes (Java Beans, .NET...) e de plataformas interoperables de obxectos distribuídos (SOAP, CORBA). Estas tecnoloxías permiten construír novos SIX de forma extensible e integrando funcionalidades proporcionadas por diversos provedores.

Dende a perspectiva dos almacéns de datos, téndese cada vez mais ao uso dos sistemas post relacionais, que permiten integrar en sistemas relacionais tradicionais



algunhas características das BDOO e mesmo inclúen extensións espaciais do modelo multimedia do estándar SQL3.

Por outro lado, o avance no campo das telecomunicacións, e máis en concreto da internet, cun gran potencial tanto para a transmisión de grandes cantidades de información e acceso a datos, favoreceu a expansión das arquitecturas de xeoproceso baseadas en servizos web.

Os servizos web permiten concibir e desenvolver sistemas que integran, cun mínimo nivel de acoplamento, información e servizos de xeoproceso interoperables de múltiples fontes e en distintos formatos aos que se accede nun contorno de rede distribuído.

Estas novas arquitecturas pretenden satisfacer o desexo da comunidade SIX de dispoñer dun acceso ilimitado e en calquera momento a información actualizada e interoperable. A dispoñibilidade deste tipo de servizos está a facilitar unha expansión do intercambio e da difusión electrónica da información espacial.

Por outra parte, este crecemento na implantación de produtos e servizos de información cartográfica na rede, establece os principios para o asentamento real dun ámbito no que sexa posible o intercambio de información xeográfica e servizos de xeoproceso. Cabe destacar tamén a acción tan importante que nesta liña está a desenvolver o consorcio OpenGIS, no que se aglutinan os principais entes involucrados no sector da información espacial e todos os sistemas e tecnoloxías que a soportan (usuarios, universidades, administracións, industrias software...). O propósito destes colectivos é elaborar de forma consensuada, especificacións de interfaces interoperables no campo das tecnoloxías da información espacial.

Dende a creación de OpenGIS, a mediados dos anos 90, foron xa moitas as realizacións prácticas nas que tomou parte e son as súas especificacións estándares de facto no ámbito das tecnoloxías da información espacial. Ademais, en moitos casos estes estándares son a base para a formulación de estándares internacionais.



Por exemplo, os dous seguintes son servizos web que xa foron enteiraamente especificados:

- Servizo de entidades vectoriais: facilita información relativa á entidade ou entidades que se encontran almacenadas nunha capa vectorial e que reúnen as características especificadas durante a consulta.
- Servizo de mapas na web: xera mapas no formato desexado para ser visualizados nun navegador ou outro tipo de cliente sinxelo. Estes mapas serán a resposta a algunha consulta con certos parámetros realizada previamente.

Podemos concluír que as arquitecturas dos SIX tenden a ser distribuídas, interoperables e en rede, apoiadas sobre estándares abertos da internet.

#### **22.4.2. Clasificación dos SIX**

De acordo coa funcionalidade que integran e o tipo de problema que pretenden resolver, podemos distinguir os seguintes grupos de sistemas de información xeográfica.

1. *SIX profesional*: enfócanse cara a usuarios cun alto nivel de especialización e formación neste campo. Integra todas as funcións que se poden necesitar nun SIX no que atinxe á recompilación e edición de datos, administración de BD, análise e xeoproceso avanzado e todas as ferramentas específicas que poidan ser necesarias para mantemento da información.
2. *SIX de sobremesa*: enfócanse cara á explotación e utilización da información. Incorpora ferramentas de análise da información, ademais de mecanismos avanzados para a presentación de resultados como son informes, gráficos, mapas, etc. Presentan unha gran facilidade de manexo, co cal os usuarios non necesitan ser expertos no ámbito, ademais as ferramentas que integran son potentes e facilitan o acceso avanzado á información.



3. *Visualizadores SIX*: son ferramentas sinxelas que se centran exclusivamente na visualización da información, de distintos tipos e formatos..
4. *WebGIS*: consiste en proporcionar o acceso a datos cartográficos e ás funcionalidades (servizos) dos SIX a través da rede. Cada vez máis téndese cara á estandarización deste tipo de servizos liderado por OpenGIS.
5. *SIX de compoñentes*: coa expansión no campo da enxeñaría de software dos desenvolvementos baseados en compoñentes, alcanzouse a posibilidade de incorporar funcionalidades espaciais en todo tipo de aplicacións (captación espacial de aplicacións), o que supón un novo impulso para a xeneralización do uso da información espacial a novos campos, nos que se poden realizar interesantes sinerxías.
6. *SIX de dispositivos móbiles*: apóiase no uso de PDA e teléfonos intelixentes. Estes dispositivos teñen capacidade abondo como para soportar case todas as funcións dun sistema tradicional.

#### **22.4.3. Ámbitos de aplicación**

Os produtos SIX comerciais son cada vez máis comúns e populares, polo que recoller todos os ámbitos posibles de aplicación é unha ardua tarefa. Non obstante, no seguinte listado preséntanse os máis destacados ou onde o número de desenvolvementos é maior.



## Demografía

Nesta categoría recóllense todas as aplicacións que, se ben poden ser de natureza moi diversa, comparten o feito de que utilizan características demográficas e socioeconómicas, e a distribución espacial delas para a toma de decisións.

Os datos nos que se apoian este tipo de sistemas adoitan proceder de rexistros estatísticos confeccionados por algún organismo (oficial ou non).

As aplicacións dentro desta categoría adóitanse centrar na mercadotecnia, avaliación do impacto dun servizo, selección de lugares para o establecemento de negocios ou servizos, etc.

## Xestión e planificación urbana

Esta categoría oríéntase a actividades propias de xestión municipal como son a xestión de servizos de infraestrutura (iluminación, rede de sumidoiros, mobiliario urbano, etc.), a xestión do tráfico, a xestión de taxas e licenzas, a localización para instalacións e servizos comunitarios, etc.

Este tipo de sistemas adoita manexar escalas grandes e úsase como base o rueiro do concello en cuestión. Ademais este tipo de aplicacións adoita empregar un modelo de datos de tipo vectorial.

## Xestión de instalacións

Nesta categoría agrúpanse os desenvolvementos orientados a compañías de subministracións e servizos, como son electricidade, auga, ferrocarril, etc. As aplicacións tipo deste grupo pasan pola xestión do mantemento, a relación co cliente (notificacións de cortes de subministración), deseño de instalacións, etc.



Estes sistemas caracterízanse por:

- A precisión necesaria adoita ser elevada.
- Existe unha forte estrutura en rede, necesaria para a realización de análises.
- Establécense conexións con bases de datos externas.
- Existe unha xerarquía de compoñentes da rede.

### Aplicacións de xestión e inventario de recursos

Nesta categoría inclúense campos como a xestión forestal, a planificación agraria, a avaliación do impacto ambiental, a xestión do territorio, a do patrimonio natural e a do medio.

Normalmente manexan escalas pequenas con diversas calidades nos datos e mesmo sen contrastar. Estas aplicacións usan modelos de datos tanto vectoriais como ráster.

### Xestión catastral

Esta categoría oríéntase á xestión da propiedade inmobiliaria e, pola súa importancia, adquiriu un termo específico: sistemas de información territorial (SIT).

No noso país contamos co sistema de información catastral, que conta con datos e descrições das propiedades tanto do ámbito urbano como do rústico.



## 22.5. Bibliografía

- ☐ Introduction to Knowledge Management. Filemon A. Uriarte Jr.
- ☐ "SilverStripe: The Complete Guide to CMS Development". Ingo Schommer e 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

Colexiado do CPEIG



**23. MOTORES DE BUSCA.  
SERVIDORES DE MENSAXERÍA.  
SISTEMAS DE CORREO. LISTAS  
DE DISTRIBUCIÓN. GRUPOS DE  
NOTICIAS DE REDE (NNTP).  
SISTEMAS DE  
VIDEOCONFERENCIA.  
MENSAXERÍA INSTANTÁNEA.  
ACCESIBILIDADE E  
USABILIDADE. W3C. E-  
LEARNING. WEB 2.0. WIKIS.  
BLOGS. COMUNIDADES  
VIRTUAIS. REDES SOCIAIS.  
SINDICACIÓN DE CONTIDOS.  
PODCAST. SUITES DE  
OFIMÁTICA EM WEB.  
ALMACENAMIENTO EN WEB.  
ESCRITORIOS VIRTUAIS. P2P.  
WEB SEMÁNTICA.**



***Tema 23: Motores de busca. Servidores de mensaxaría. Sistemas de correo. Listas de distribución. Grupos de noticias de rede (NNTP). Sistemas de videoconferencia. Mensaxaría instantánea. Accesibilidade e usabilidade. W3C. E-learning. Web 2.0. Wikis. Blogs. Comunidades virtuais. Redes sociais. Sindicación de contidos. Podcast. Suites de ofimática en web. Almacenamento en web. Escritorios virtuais. P2P. Web semántica.***

## **ÍNDICE**

### **23.1 Motores de busca**

- 23.1.1 Spiders*
- 23.1.2 Directorios*
- 23.1.3 Sistemas mixtos (directorio e motor de busca)*
- 23.1.4 Metabuscadores*
- 23.1.5 Multibuscadores*

### **23.2 Servidores de mensaxaría**

- 23.2.1 Sistema de mensaxaría centralizada*
- 23.2.2 Sistema de mensaxaría distribuída*

### **23.3 Sistemas de correo**

- 23.3.1 Elementos do servizo de correo electrónico*
- 23.3.2 Enderezo de correo electrónico*
- 23.3.3 Proceso de envío de mensaxes*

### **23.4 Listas de distribución**

### **23.5 Grupos de noticias de rede (NNTP)**

### **23.6 Sistemas de videoconferencia**

### **23.7 Mensaxaría instantánea**

### **23.8 E-LEARNING**

- 23.8.1 Introducción*
- 23.8.2 Concepto*
- 23.8.3 Plataformas de e-Learning*
- 23.8.4 Vantaxes*
- 23.8.5 Inconvenientes*
- 23.8.6 Estandarización*
- 23.8.7 Plataforma Moodle*

### **23.9 Accesibilidade y Usabilidade**

- 23.9.1 Accesibilidade como calidade dos sistemas*
- 23.9.2 Limitacións na accesibilidade*
- 23.9.3 Promovendo a accesibilidade*
- 23.9.4 Usabilidade*



**23.10 W3C****23.11 Web 2.0****23.12 Wikis****23.13 Blogs****23.14 Comunidades virtuais****23.15 Redes sociais****23.16 Sindicación de contidos***23.16.1 Fonte web**23.16.2 Agregador de noticias**23.16.3 Formato RSS**23.16.4 Estándar Atom***23.17 Podcast***23.17.1 Podcasting fronte Streaming***23.18 Suites ofimáticas en web***23.18.1 Feng Office**23.18.2 Google Docs**23.18.3 Office Web Apps***23.19 Almacenamento en web***23.19.1 Dropbox***23.20 Escritorios virtuais***23.20.1 eyeOS***23.21 Mashups****23.22 P2P***23.22.1 Características**23.22.2 Tipos de redes P2P***23.23 Web semántica***23.23.1 Definición de web semántica**23.23.2 RDF, SPARQL e OWL***23.24 Bibliografía**



### 23.1 Motores de busca

Nos inicios, a internet comezaba a ofrecer unha gran cantidade de información que dificilmente podía ser catalogada e referenciada. Isto supoñía un impedimento á hora de realizar buscas de información relativas a temáticas concretas, provocando unha alta ineficiencia a nivel xeral no uso da internet e o acceso á rede para a busca e/ou divulgación da información. Como solución para este problema xurdiron os motores de busca.

Os motores de busca, tamén coñecidos como buscadores, son sistemas informáticos que se encargan de localizar sitios web relacionados cun determinado conxunto de termos clave que lle subministran. En esencia, son un sistema informático que consulta os ficheiros das web que se encontran almacenadas nos servidores web. Para realizar esta tarefa, habitualmente utilizan unha peza de software especificamente deseñada para analizar a rede en busca de webs e obter información que permita clasificalas mediante termos clave ou ben utilizando árbores xerárquicas por temas.

A solución ofrecida polos motores de busca non é total, senón parcial. Isto é debido a que en realidade non buscan na internet cada vez que realizamos unha consulta. A busca realízana nunha base de datos na cal almacenan referencias das páxinas accesibles xunto con datos concretos, metainformación, que serve para catalogalas. Esta información habitualmente recóllese a través dun programa (polo xeral robot) que é o que se encarga de realizar visitas periódicas por todo o contido dispoñible do web. Non existe unicidade nos criterios de selección para a agregar novas páxinas ás bases de datos dos motores de busca. O resultado é que cada base de datos contén información de moi diversa calidade, especificando os seus propios criterios de selección, e consecuentemente, establecendo categorizacións e resultados



diferentes para cada busca en función do motor de busca co que esteamos a traballar.

En liñas xerais, pódense distinguir cinco tipos básicos de motores de busca, diferenciados entre si fundamentalmente polo tipo de información que albergan, ou os mecanismos que utilizan para realizar a referenciación das páxinas que ofrecen. Estes cinco tipos son os spiders, directorios, sistemas mixtos, metabuscadores e multibuscadores.

### 23.1.1 *Spiders*

Os *spiders*, tamén coñecidos como arañas web, *crawlers* ou *rastrexadores web*, son programas que revisan as páxinas web de forma metódica e automática. Habitualmente realizan copias das páxinas web visitadas para un procesado posterior que consiste en indexar esas páxinas en función do seu contido, determinado por conxuntos de termos clave, para proporcionar un sistema de busca posterior máis optimizado.

O funcionamento é simple. A araña iníciase cunha lista de URL ou páxinas para visitar. A medida que vai consultando as páxinas, vai engadindo todos os hipervínculos que se encontran nelas a unha lista de URL que visitará de forma recorrente en función dunhas regras establecidas. As visitas realízanse de forma periódica, polo tanto, é posible que en determinadas ocasións o contido non apareza totalmente actualizado. A orde en que se mostran os resultados da consulta está determinada por diversos factores que dependen de cada buscador en particular.

A gran maioría dos buscadores que se utilizan habitualmente entran dentro da categoría de rastrexadores web. Son sistemas custosos, que fan uso intensivo dunha gran cantidade de recursos.



Algúns exemplos de *spiders* son Google, Bing ou Hotbot

### 23.1.2 Directorios

Os directorios son un tipo de motores de busca cun funcionamento totalmente distinto das arañas web. Os directorios son simplemente listas categorizadas de recursos que se estruturan xerarquicamente. Esta estrutura organízase en forma de árbore, permitindo visualizar os contidos con diferente grao de granularidade, dende os máis xerais aos máis específicos.

En realidade estes motores non dispoñen de ningún software específico que analice os contidos web, senón que realiza as clasificacións e categorizacións do material en función dun conxunto de criterios seleccionados de forma manual. Isto implica que a tecnoloxía na que se basean é barata e sinxela, non obstante o custo operacional é alto, xa que sempre se require de intervención humana.

Algúns exemplos de directorios son Yahoo! e Open Directory Project

### 23.1.3 Sistemas mixtos (directorio e motor de busca)

Os sistemas mixtos combinan características de directorios e motores de busca. Dispoñen habitualmente dalgunha peza software de tipo "araña" para realizar a análise da web, e ademais permiten engadir e presentar páxinas clasificadas en catálogos segundo o seu contido. A combinación destas características representa a tendencia actual nos buscadores máis importantes.

### 23.1.4 Metabuscadores



Os metabuscadores son un tipo de motores de busca que centran os seus resultados en buscas que realizan sobre outros buscadores. Isto significa que obteñen inicialmente un conxunto de resultados doutro buscador, e a continuación refinan eses resultados presentando unha selección propia.

Unha das principais vantaxes dos metabuscadores é que amplían o ámbito das buscas que realiza o usuario. Proporciona unha gran cantidade de resultados combinados en función dos criterios particulares de cada metabuscador. En moitas ocasións estes criterios de ordenación non resultan de todo claros.

Por outro lado, o problema principal é que os metabuscadores non distinguen entre as diferentes sintaxes dos buscadores, limitando a especificidade coa que os metabuscadores poden traballar para localizar información. Ademais, ao realizar as buscas en diferentes fontes (buscadores), a obtención de resultados adoita demorarse moito máis que ao utilizar outro tipo de motor de busca.

Alguns exemplos de metabuscadores son Metacrawler, Ixquick, Dogpile ou Metabuscador

### **23.1.5      Multibuscadores**

Os multibuscadores son un tipo de motores de busca similares aos metabuscadores, pero cunha diferenza notable; mentres que os metabuscadores non distinguen as diferentes sintaxes dos buscadores que utilizan para a obtención de resultados, os multibuscadores si o fan. Isto implica que poden lanzar varias buscas en motores seleccionados respectando o formato orixinal dos buscadores.

Os multibuscadores son útiles para realizar buscas en diferentes buscadores ao mesmo tempo. A súa operativa difire dos buscadores normais, dado que non dispoñen de compoñentes software que analicen e almacenen contido, senón que o único que conteñen é un rexistro de buscadores e os criterios de adecuación das expresións de busca asociadas a cada buscador. Os multibuscadores non almacenan información de



páxinas relativas a contido. Simplemente realizan a consulta axeitada a cada buscador dentro do seu rexistro, e realizan un filtrado dos enlaces repetidos, e aplican ademais criterios de selección como a relevancia de cada enlace nos diferentes buscadores, para xerar finalmente unha lista de resultados. Un exemplo de multibuscador é [iniciodirecto.com](http://iniciodirecto.com).

## 23.2 Servidores de mensaxaría

Un servidor de mensaxaría é unha aplicación que posúe a capacidade para manexar mensaxes entre dous ou máis entidades, xa sexa aplicacións de usuario ou outros sistemas de xestión de mensaxes. As mensaxes dun servidor de mensaxaría son enviadas a través dun *middleware* ou mediador, o que facilita a comunicación entre os distintos elementos do sistema, utilizando xeralmente un conxunto de regras e especificacións que posibilitan a comunicación entre as distintas partes. Outra das características dos servidores de mensaxaría é a capacidade de almacenaxe das mensaxes; este almacenamento prodúcese xeralmente nunha cola ata que é posible o envío deste cara ao seu destinatario, que polo xeral resulta ser outra aplicación.

É moi habitual encontrarse nunha empresa ou organización un sistema de mensaxaría funcionando nun servidor e esperando o envío de mensaxes á súa cola de entrada. Dende alí, o mediador analiza mensaxe a mensaxe determinando o destino de cada unha. Unha vez no servidor, unha mensaxe só ten dúas posibilidades de entrega, ou ser enviada de xeito local, ou que esta teña que ser redirixida a outro servidor de mensaxaría para que sexa el o que realice a entrega. Se a mensaxe ten que ser entregada a un destino local, daquela é enviada inmediatamente á caixa de correo local. Pola contra, se a mensaxe é determinada como remota, o servidor de mensaxaría debe enviar a mensaxe a outro servidor de mensaxaría dentro do seu contorno para que sexa este o que realice a entrega da mensaxe.

Polo xeral, se existen problemas de conexión entre os servidores ou non é posible determinar a localización do servidor de mensaxaría remoto, o usuario que realizou o envío da mensaxe é informado a través dunha mensaxe enviada polo servidor de



mensaxaría, informando da situación. Este tipo de mensaxe adoita ser só de notificación de que se están a ter problemas co envío da mensaxe, posto que o servidor de mensaxaría continuará intentando enviar a mensaxe ata que se esgoten o número máximo de intentos de envío ou ata que a mensaxe caduque, é dicir, exceda un límite de tempo de estancia no servidor.

Os modelos dos servidores de mensaxaría adoitan adaptarse a unha arquitectura centralizada ou seguir unha solución distribuída.

### **23.2.1      Sistema de mensaxaría centralizada**

Un sistema de mensaxaría centralizada fundaméntase nun núcleo de datos que aloxa todos os recursos e servizos dos servidores que conforman o sistema. Este núcleo de datos permite que calquera usuario do sistema de mensaxaría se conecte aos servizos de mensaxaría, xa sexa de forma local ou remota.

As características dun sistema de mensaxaría centralizado son:

- **Datos:** Todos os datos e a información encóntrase albergada e xestiónase dende o núcleo, mesmo cando os usuarios establecen unha conexión remota para a súa utilización. Esta centralización facilita en boa medida a administración dos servizos, posto que provoca que esta sexa máis sinxela.
- **Actualizacións:** As actualizacións débense realizar unicamente no núcleo central, onde se encontra todo o sistema.
- **Localización:** O centro de datos engade ao sistema dispositivos de illamento da alimentación ou sistemas de alimentación ininterrompida (SAI). Proporciona ademais a posibilidade de ofrecer servizos mesmo cando se produce algunha incidencia de carácter grave, xa que posibilita a réplica de todo o sistema dun xeito eficaz.

### **23.2.2      Sistema de mensaxaría distribuída**



Un sistema de mensaxaría distribuída está formado por unha serie de sucursais repartidas en distintas localizacións conectadas entre si. Cada sucursal posúe un servidor ou servidores de mensaxaría con todos os seus servizos de xeito independente do resto de sucursais. Cada un dos servidores de mensaxaría realiza o envío das súas mensaxes locais e redirixe aos outros servidores aquelas mensaxes que non son de dominio local e que si son capaces de resolver algún dos outros servidores.

- **Datos:** A información encóntrase tamén distribuída entre cada unha das sucursais e cada unha delas xestiona e administra esta información e os seus servizos, o que provoca un aumento na complexidade destas tarefas.
- **Actualizacións:** Cada vez que se leva a cabo unha tarefa de actualización esta débese realizar en cada unha das sucursais, para que teña efecto en todo o sistema.
- **Localización:** Cada sucursal posúe o seu propio centro de datos, que pode ofrecer os mesmos servizos que nunha arquitectura centralizada.

### **23.3 Sistemas de correo**

O correo electrónico, ou e-mail, está catalogado como un servizo de rede que lles proporciona aos usuarios a capacidade para enviar e recibir mensaxes e arquivos de forma rápida e eficiente a través de dispositivos dixitais. O sistema trata de representar unha analoxía co correo postal habitual, presentándose como unha alternativa para o envío de mensaxes de texto ou calquera tipo de ficheiro en formato dixital. Dado que o seu custo operacional é baixo e a súa eficiencia é elevada, o correo electrónico está actualmente a desprazar o correo postal orixinal.

A orixe do correo electrónico é mesmo anterior á rede internet. Os primeiros pasos para a creación do sistema de correo electrónico déronse no MIT contra 1961,



cando se desenvolveu un sistema que lles permitía a varios usuarios, dende terminais remotos, ingresar nun ordenador central no cal podían almacenar unha copia dos seus arquivos no disco. Este foi un dos pasos iniciais na implantación de mecanismos para a compartición de información. En 1965 comezou a utilizarse un sistema baseado no almacenamento de mensaxes compartidas entre os usuarios dunha supercomputadora dando lugar ao primeiro sistema de correo electrónico utilizado. Posteriormente, en 1971, incorporouse ao sistema de mensaxaría o uso da arroba (@) como elemento para dividir o nome dos usuarios da máquina na que se encontraban aloxados.

### 23.3.1 Elementos do servizo de correo electrónico

Tecnicamente falando, o correo electrónico é un servizo proporcionado na internet, soportado polo protocolo SMTP (Simple Mail Transfer Protocol) e o protocolo POP (Post Office Protocol). Como en todo servizo, a arquitectura do sistema consta dunha parte clienta, habitualmente utilizada polos usuarios para enviar e recibir correos, e unha parte servidora que proporciona os mecanismos axeitados para o almacenamento e transferencia dos correos entre os diferentes clientes. Os protocolos definen os esquemas de comunicación entre os clientes e os servidores para que as mensaxes se transmitan dun sitio a outro. O protocolo SMTP é o protocolo encargado do envío das mensaxes, mentres que o protocolo POP é o encargado da recepción das mensaxes.

O **cliente de correo electrónico**, tamén chamado **Mail User Agent (MUA)** é un programa que basicamente permite xestionar as mensaxes recibidas así como recibir correos novos. Habitualmente faise referencia a cliente de correo electrónico, ou a aplicacións autónomas que proporcionan un amplo abano de funcionalidades para a xestión do noso correo. Non obstante, actualmente a maioría dos provedores de servizo de correo permiten o acceso a través dos navegadores web, proporcionando interfaces web a modo de cliente para a consulta e xestión do noso correo electrónico. Existe unha gran diferenza respecto do funcionamento de ambas



as dúas opcións; cando se utiliza un cliente de correo electrónico, todas as mensaxes dispoñibles se descargan no ordenador no que se estea executando ese cliente de correo electrónico. Non obstante, cando se accede á conta de correo a través das interfaces web, as mensaxes seguen almacenadas no servidor, sendo accesibles a través do cliente web, dende calquera ordenador que dispoña dunha conexión á internet.

Algúns exemplos de clientes de correo electrónico autónomos son:

- **Microsoft Outlook:** Cliente privativo da compañía Microsoft. É o cliente de correo estándar de Microsoft, incluído no paquete Microsoft Office.
- **Mozilla Thunderbird:** Alternativa de software libre de Outlook, desenvolvido por Mozilla.

Algúns exemplos de provedores de servizo de correo web son Gmail, Hotmail ou Yahoo!.

En canto ao **servidor de correo electrónico** consiste nun conxunto de aplicacións informáticas situadas nun equipo servidor, xa sexa en rede local ou na internet, coa tarefa de realizar unha serie de procesos que teñen a finalidade de transportar información entre os distintos usuarios do servizo. O servidor de correo electrónico é o encargado de xestionar todos os usuarios rexistrados no sistema coas súas correspondentes identificacións (enderezos de correo electrónico) que servirán para poder interactuar entre si mediante o envío de correos.

O **servidor de correo electrónico**, dispón dunha peza software denominada Axente de Transferencia de Correo (MTA) ou Axente de Transporte de Mensaxes, co obxectivo de transmitir os datos dunha máquina a outra. En concreto, céntrase na parte de transferencia de datos entre distintos servidores, exercendo diferentes roles como servidor doutros servidores de correo, cliente doutros servidores de correo e como intermediario entre o cliente de correo que emite a mensaxe e outro servidor de correo externo.



Actualmente, con excepción das grandes corporacións que dispoñen da súa propia infraestrutura TIC, a maioría dos usuarios fai uso de servidores de correo electrónico que pertencen a algunha entidade provedora dese servizo. Existen diferentes empresas e entidades que ofrecen servizos de correo electrónico, tanto de forma gratuita como de pagamento. Os servizos de correo gratuito son os máis coñecidos polos usuarios, e entre eles podemos destacar os servizos de Gmail, Hotmail ou Yahoo!. As entidades rexistradoras de dominio son as que habitualmente ofrecen servizos de correo electrónico asociados á conta de dominio contratada. En canto ás solucións software para a implantación dun servidor de correo electrónico, poden destacarse Microsoft Exchange Server para plataformas Windows ou Sendmail, Qmail, Zimbra e Postfix para Unix/GNULinux.

### 23.3.2 Enderezo de correo electrónico

O enderezo de correo electrónico é unha secuencia de palabras que teñen por obxecto identificar un determinado usuario dun servizo de correo de forma inequívoca. Este enderezo de correo representa o identificador mediante o cal o usuario pode enviar e recibir correos.

A sintaxe dun enderezo de correo é a seguinte:

- **Nome de usuario:** É conxunto de palabras escollidas polo usuario que habitualmente adoita coincidir co nome ou algún identificativo da persoa ou usuario que utilizará a conta de correo. Pode conter letras, números e algúns signos.
- **@:** é o signo ou símbolo encargado de separar dúas partes importantes do enderezo de correo, concretamente o nome de usuario e o dominio.
- **Nome de dominio:** O nome de dominio na internet é unha identificación asociada a un dispositivo ou grupo de dispositivos. Habitualmente correspóndese co nome do provedor do servizo de correo.



### 23.3.3      **Proceso de envío de mensaxes**

Detállase a continuación o proceso de envío e recepción de correos electrónicos e os elementos e protocolos que interveñen entre un ordenador A e un ordenador B.

O ordenador co cliente A redacta un correo electrónico para o cliente B e envíallo. Ao realizar a operación de envío, o cliente de correo en A contacta co servidor de correo á través do protocolo SMTP, transfírelle o correo e dálle a orde de envialo. Ao recibir a petición, o servidor de correo A verifica que o correo pertence a outro dominio. Para resolver a dirección á que ten que enviar o correo, realiza unha consulta a un servidor de DNS para descubrir quen é o encargado de xestionar o dominio asociado ao cliente B. Unha vez obtida a resposta, e resolto o servidor de correo B, o servidor de correo A comunícase con el a través do protocolo SMTP, enviándolle o correo emitido dende o cliente A, e queda así este correo almacenado no servidor B. Posteriormente, cando o cliente B decida consultar o correo, accederá mediante o protocolo POP ao servidor de correo B e descargará as mensaxes almacenadas nese servidor que teñan por destinatario o usuario de cliente B.

### 23.4    **Listas de distribución**

As listas de distribución, ou listas de correo electrónico, son agrupacións de usuarios de correo electrónico. Mediante un software apropiado pódense configurar listados de enderezos de correo electrónico para o envío masivo de información a múltiples usuarios a un tempo. Cada lista de distribución de correo electrónico está á súa vez referenciada por un enderezo de correo electrónico. A grandes trazos, cada vez que un usuario autorizado emite un correo co enderezo da lista de distribución



como destinatario, en realidade a lista reenviará o correo a todos os usuarios adscritos a esa lista.

As listas de correo electrónico son unha das ferramentas cada vez máis utilizadas nas organizacións para manter os usuarios informados con noticias e información de interese. De forma habitual, é necesario que os propios usuarios se rexistren nesas listas das que están interesados en recibir noticias ou información.

As listas de correo electrónico están xestionadas polo propio servidor de correo, ou por software adicional específico para a súa xestión. Para a alta, baixa ou modificación dos datos dos usuarios, é habitual que os servidores de listas de correo electrónico poñan a disposición dos usuarios un ou varios enderezos de correo aos cales enviar comandos. Ademais algúns servidores de listas de correo permiten diferentes modos de subscrición:

- Modo individual: o usuario da lista recibe todas as mensaxes que formen parte dela. Da mesma forma, se o usuario dispón dos privilexios necesarios, pode enviar correos á lista de distribución.
- Modo non correo: o usuario non recibe as mensaxes que se envían á lista pero pode enviar correos á lista. Habitualmente esta opción permite a consulta dos correos a través de interface web.
- Modo resumo diario: tamén chamado modo colgante, consiste en que o usuario só recibe un correo diario que inclúe todas as mensaxes enviadas á lista de correo.

Tipos de listas de correo electrónico:

- Boletín electrónico: utilízase como medio unidireccional para a transmisión de información debido a que só poden enviar mensaxes á lista determinadas persoas encargadas da publicación e xestión dese boletín.



- **Lista de debate:** Neste tipo de lista, calquera subscritor pode enviar correos á lista de distribución, e o resto de usuarios pode contestalos do mesmo xeito. Desta forma pódense xerar debates e intercambios de información. As cadeas de correos van xerando fíos que poden ser contestados por calquera dos usuarios da lista.

Existen na internet diferentes servizos que permiten a creación de listas de correo electrónico de forma gratuíta, como por exemplo Google Groups, Yahoo! ou eListas.

No que atinxe á implantación, existen diferentes produtos baseados en software libre para a configuración e xestión do servizo de listas de correo, como por exemplo phpList, Sympa, Mailman e Gmane.

### **23.5 Grupos de noticias de rede (NNTP)**

Os grupos de noticias son un servizo proporcionado na internet ao cal os usuarios se poden subscribir para participar de forma similar ás listas de correo. En esencia, os grupos de noticias serían similares a un taboleiro electrónico de noticias categorizadas xerarquicamente por temáticas. O contido dentro do servizo de noticias organízase como un gran número de grupos, nos que se agrupan os diferentes temas. O nome de cada un dos grupos de noticias dispoñibles na rede consta dun conxunto de identificadores separados por puntos, habitualmente relacionados co dominio da entidade que xestiona o servidor de noticias, ou ben seguindo un estándar definido para as principais redes de grupos de noticias. Isto permite ao usuario subscribirse a un grupo ou grupos determinados que resulten do seu interese e recibir todas as mensaxes que o resto de usuarios envían a ese grupo.

Este funcionamento aseméllase bastante ao dunha lista correo, de feito



comparten un obxectivo intrínseco que consiste na xeración de espazos de debate e foros de discusión sobre algún tema concreto. A diferenza é que coas listas de correo se reciben directamente as mensaxes no cliente de correo, mentres que coas *news*, ou grupos de noticias, cómpre conectarse a un servidor de noticias e extraer os grupos nos que o usuario estea interesado.

Unha vez que o usuario se rexistra nun grupo ou grupos determinados, pode agregar mensaxes ao sistema. Pode tamén publicar noticias que contesten ou repliquen outras noticias previas, formando fíos de debate. Para a xestión das noticias necesítase software específico para este servizo. Actualmente a maioría dos clientes de correo web veñen preparados para desempeñar esta función. As funcionalidades básicas que deben proporcionar son as de permitir seleccionar os grupos de interese para o usuario, lectura de noticias publicadas por outros e envío de noticias ao servidor.

Dada a cantidade de mensaxes que se xeran a diario nos grupos de noticias, é habitual que os servidores de noticias públicos dispoñan de mecanismos para evitar a saturación dos seus sistemas de almacenamento. Un deles consiste en estipular un tempo de vida determinado para as noticias que se van almacenando no sistema. Ao cabo dese período de vida, o contido é eliminado.

Os grupos de noticias clasifícanse xerarquicamente en función das súas temáticas, proporcionando unha axuda importante á hora de localizar os temas de interese.

## **23.6 Sistemas de videoconferencia**

Os sistemas de videoconferencia teñen como principal característica permitir comunicación simultánea e bidireccional de sinais de audio e vídeo, o que proporciona



capacidade para manter reunións con persoas situadas en lugares afastados.

Este tipo de sistemas habitualmente integran capacidades que abranguen parte dos sistemas vistos con anterioridade xa que integran a capacidade de establecer comunicación escrita (chat) e xestión de mensaxaría instantánea. Ademais poden incluír capacidade para a transmisión de ficheiros e edición en ferramentas colaboradoras.

A base tecnolóxica dos sistemas de videoconferencia é a compresión dixital dos fluxos de audio e vídeo en tempo real.

En canto á categorización dos sistemas de videoconferencia, podemos clasificalos fundamentalmente en dous grandes grupos:

- **Sistemas de videoconferencia dedicados:** Dispoñen dos compoñentes hardware necesarios para realizar unha videoconferencia en remoto. Son sistemas de alta calidade, especiais para as circunstancias nas que se demanda unha alta fiabilidade e calidade dos datos transmitidos. Polo xeral, este tipo de dispositivos constan dunha cámara de vídeo de alta calidade e unha consola. Dentro deste tipo de sistemas podemos distinguir varios tipos de dispositivos hardware en función do obxectivo do sistema:
  - Grupos grandes: son dispositivos grandes, non portátiles, máis custosos utilizados para grandes salas e auditorios. Requiren de instalación e mantemento axeitados.
  - Grupos reducidos: non son portátiles, son máis pequenos e menos custosos, utilizados para salas de reunións pequenas. Requiren de instalación.
  - videoconferencia individual: son dispositivos portátiles, destinados a usuarios individuais, teñen cámaras fixas, micrófonos e altosfalantes integrados na consola.
- **Sistemas de videoconferencia de escritorio:** Os sistemas de escritorio, ou sistemas de usuario, baséanse na combinación de parte software e hardware. En canto á parte software, consiste nalgún cliente de mensaxaría con



capacidade para realizar videoconferencia mediante a transmisión dunha cámara web e un micrófono conectado ao ordenador. En canto aos dispositivos hardware, simplemente serían necesarios unha cámara web e un micrófono. Na actualidade, practicamente a gran maioría dos sistemas de mensaxaría instantánea soportan videoconferencia. É o caso por exemplo dos clientes de MSN Messenger ou Skype.

### 23.7 Mensaxaría instantánea

O termo chat é un anglicismo que fai referencia ao charla ou cibercharla. Utilízase para designar as comunicacións escritas realizadas en tempo real a través da internet entre dúas ou máis persoas. Pódese realizar a través de canles públicas, ou mediante canles privadas, dependendo do medio e protocolos que se utilicen. Este tipo de características determinan as diferentes tipoloxías de chat:

- **Webchat:** É un tipo de chat no que as mensaxes se transmiten a través de WWW. É un tipo de chat de doado acceso, dado que as interfaces están implantadas como aplicacións web, accesibles dende calquera navegador. Resulta sinxelo de utilizar dado que existen unha gran cantidade de compoñentes visuais que axudan a personalizar rapidamente os estilos de escritura e visualización neste tipo de aplicacións de acceso aos webchats, o que o fai resultar atractivo para usuarios noveis. Non obstante o uso dos webchat está a decaer dado que as tarefas de actualización do contido da páxina que carga o webchat, así como a inestabilidade dalgúns navegadores fai que manter as conversacións en tempo real sexa difícil nalgúns ocasións.
- **IRC (Internet Relay Chat):** Representa a forma máis coñecida e antiga de chat que existe. IRC é un protocolo de comunicación en tempo real baseado en texto que permite comunicación entre usuarios sen necesidade



de acordo previo de establecer a comunicación, é dicir, que dous usuarios que se encontren nunha canle poden comunicarse entre si sen necesidade de establecer unha comunicación previa. O IRC presenta un modelo cliente-servidor onde as aplicacións cliente dos usuarios, que habitualmente son aplicacións autónomas, se configuran e conectan contra un determinado servidor de IRC, permitindo establecer comunicación co resto de persoas que se encontran conectadas a ese servidor, ben mediante chat privado, ou a través de canles de libre acceso. Neste modelo de chat, existen axentes moderadores que interveñen na administración e control de todo o que sucede en cada servidor de IRC. O cliente máis habitual deste tipo de redes é o *mIRC*.

- **Mensaxaría instantánea:** Pode considerarse outra modalidade de chat. En esencia é similar ao IRC, dado que a arquitectura deste tipo de sistemas que proporcionan mensaxaría instantánea se basea nun modelo cliente-servidor no que aplicacións autónomas se conectan contra o servidor e permiten enviar e recibir mensaxes doutros usuarios conectados ao servidor. Non obstante a gran diferenza radica en que nos sistemas de mensaxaría instantánea, para que dous usuarios se comuniquen, deberá de existir un contacto previo mediante o cal ambos os dous usuarios accedan a establecer a comunicación. A maioría destes sistemas contan coa súa propia rede que unicamente é accesible mediante o cliente propio desa rede, desenvolvido por unha entidade ou compañía concreta. Nese sentido, preséntase como un modelo de comunicación limitado e controlado que se está a comezar a adoptar nalgúns empresas e corporacións para establecer un mecanismo de intercambio de información de forma económica, controlada e fiable. Algúns sistemas de mensaxaría instantánea máis comúns son o MSN Messenger, Yahoo Messenger ou ICQ.



## 23.8 E-LEARNING

### 23.8.1 Introducción

Nos últimos anos apareceron sistemas informáticos orientados ao ensino e aínda que o obxectivo de todos eles é moi similar, os medios mediante os que chegan a ese obxectivo varían en boa medida. O involucrar as novas tecnoloxías no ámbito do ensino, introducindo estas como ferramenta fundamental do proceso de aprendizaxe, desembocou na aparición dun novo termo coñecido como *e-Learning* ou *aprendizaxe en liña*. As tecnoloxías asociadas englóbanse nun conxunto de sistemas que tratan de proporcionar os medios e mecanismos adecuados para facilitar os procesos de aprendizaxe en practicamente todas as áreas de coñecemento. Non obstante, moitos destes sistemas, mal identificados como "sistemas de e-Learning", céntranse unicamente na xestión e clasificación de documentos para poñelos a disposición de alumnos e docentes, como é o caso dos sistemas de xestión de contidos, ou dos sistemas de xestión documental. Aínda que certamente facilitan a tarefa de busca e organización de información, este tipo de sistemas non realizan ningún tipo de seguimento do proceso de aprendizaxe dos alumnos.

A idea de aprendizaxe en liña, e, en consecuencia os sistemas de e-Learning, pretende precisamente abranguer esa fase do proceso de aprendizaxe, proporcionando os mecanismos necesarios para realizar o seguimento do proceso de forma íntegra.

### 23.8.2 Concepto

O concepto de aprendizaxe en liña defínese de moitas formas diferentes, fundamentalmente debido a que os actores que del fan uso son moi diversos, cada cal coa súa idiosincrasia e o seu ámbito de aplicación.

A nivel xeral, pódese definir o e-Learning como a educación a distancia completamente virtualizada a través das novas posibilidades tecnolóxicas que hai dispoñibles, como as novas redes de comunicación, e fundamentalmente a rede de



redes, a internet. Fundamentalmente utilízanse para iso ferramentas ou aplicacións de hipertexto, que proporcionan a vantaxe de ser totalmente portables e accesibles dende calquera plataforma. A idea é que este tipo de sistemas dean soporte aos procesos de ensino-aprendizaxe. Este tipo de sistemas poden englobarse como un subgrupo dos sistemas de xestión de contidos, entendendo estes últimos como unha xeneralización, e asumindo os sistemas de aprendizaxe en liña como unha especialización dos SXC para un propósito específico con funcionalidades propias.

Alguns teóricos dividen a aprendizaxe en liña en tres ramas diferentes:

- computer aided instruction (CAI)
- computer-managed instruction (CMI)
- computer supporter learning resources (CSLR)

O primeiro termo abrangue a porción de produtos de e-Learning que proporciona ensino como tutoriais, simulacións e exercicios. O segundo termo refírese aos produtos de aprendizaxe en liña que ten funcións de avaliación, seguimento e guía de estudo. Finalmente, o terceiro termo cobre os aspectos do e-Learning que dan soporte ao desempeño, á comunicación e ao almacenamento. Esta clasificación refírese só a partes do conxunto total representado pola aprendizaxe en liña.

### **23.8.3 Plataformas de e-Learning**

Na práctica, para levar a cabo un programa de formación baseado en aprendizaxe en liña, faise uso de plataformas ou sistemas de software que permitan a comunicación e interacción entre profesores, alumnos e contidos. Téñense principalmente dous tipos de plataformas:

- LMS (Learning Management Systems), utilizados para impartir e dar seguimento administrativo aos cursos en liña.
- LCMS (Learning Content Management Systems), empregados para a xestión



dos contidos dixitais. Seguen o concepto básico dos SXC, que é a administración de contidos, pero enfocados ao ámbito educativo.

Ás veces a diferenciación entre ambas as dúas é só funcional e en lugar de constituír dúas ferramentas software diferentes ofrécense nunha única aplicación, que en España se coñece polo nome de Plataforma Tecnolóxica ou de Teledocencia.

Entre as ferramentas máis utilizadas para os ambientes ou sistemas de aprendizaxe en liña están, como xa se dixo anteriormente, os sistemas de administración de aprendizaxe ou LMS, tamén amplamente coñecidos como plataformas de aprendizaxe. Un LMS é un software baseado nun servidor web que prové módulos para os procesos administrativos e de seguimento que se requiren para un sistema de ensino, simplificando o control destas tarefas. Os módulos administrativos permiten, por exemplo, configurar cursos, matricular alumnos, rexistrar profesores, asignar cursos a un alumno, levar informes de progreso e cualificacións. Tamén facilitan a aprendizaxe distribuída e colaboradora a partir de actividades e contidos preelaborados, de forma síncrona ou asíncrona, utilizando os servizos de comunicación da internet como o correo, os foros, as videoconferencias ou o chat.

O alumno interactúa coa plataforma a través dunha interface web que lle permite seguir as leccións do curso, realizar as actividades programadas, comunicarse co profesor e con outros alumnos, así como dar seguimento ao seu propio progreso con datos estatísticos e cualificacións. A complexidade e as capacidades das plataformas varían dun sistema a outro, pero en xeral todas contan con funcións básicas como as que se mencionaron. Entre as plataformas comerciais máis comúns encóntranse Blackboard e WebCT, mentres que as máis recoñecidas por parte do software libre son Moodle e Claroline.

#### **23.8.4 Vantaxes**



A aprendizaxe en liña permite superar algunhas das barreiras existentes nos sistemas de ensino asistido por ordenador. Algunhas delas son:

- Elimina as distancias e favorece a mobilidade dos alumnos.
- Aumenta o número de destinatarios que poden seguir un curso simultaneamente.
- Permite flexibilidade horaria.
- Permite alternar diversos métodos de ensino.
- Favorece a interacción entre alumnos. Está demostrado que a non presenza física minimiza a timidez e favorece o establecemento de comunicación entre os alumnos, especialmente na adolescencia.
- Anonimato.
- Seguimento e tutoría do progreso do alumno a través das canles de comunicación establecidas.
- Posibilidade de escoller entre gran variedade de materiais, cursos e especialidades.
- Minimiza os custos de formación continua na empresa.
- Favorece a convivencia familiar para alumnos con responsabilidades familiares ao seu cargo.

Ademais de polas vantaxes enumeradas, interveñen outros factores que favorecen a implantación de sistemas e-Learning:

- **Factores económicos:** Alcázase unha mellor relación custo-beneficio na produción e desenvolvemento aproveitando a reutilización de compoñentes tecnolóxicos e materiais de aprendizaxe. É un factor interesante á hora de aumentar os niveis de formación en países en desenvolvemento, cun alto ritmo de crecemento económico e con grandes necesidades de traballadores



cualificados.

- **Alta dispoñibilidade de recursos dixitais:** As grandes empresas multinacionais necesitan distribuír materiais de aprendizaxe a sitios xeograficamente dispersos, para que estean dispoñibles en calquera momento dende calquera lugar. A existencia dun gran número de recursos dixitais libres e gratuítos na internet (imaxes, clips de audio e vídeo, animacións, etc.) favorecen a súa reutilización e aproveitamento por parte das grandes empresas (ou terceiros, como pode ser unha empresa especializada na creación de cursos ou implantación de sistemas de aprendizaxe en liña) para a creación de cursos a través de sistemas e-Learning.
- **Penetración social:** A alta penetración na sociedade das novas tecnoloxías en xeral e da internet en particular, favorece a aceptación de novas vías de información e de comunicación.
- **Axudas estatais:** Os programas de subvencións por parte do Estado, as Comunidades Autónomas e o Fondo Social Europeo, incentivaron a creación e desenvolvemento dun sector empresarial dedicado á formación en liña. Estas subvencións fixeron posible a aparición de programas como os de formación continua de traballadores, que contribúen á adaptación dos traballadores ás máis novas tecnoloxías.

### 23.8.5 Inconvenientes

Alguns inconvenientes no emprego de sistemas de aprendizaxe en liña son:

- **Preparación do estudante:** É necesario un esforzo para asegurar que os estudantes teñen as habilidades e coñecementos técnicos, así como o acceso ao hardware e software necesarios para completar satisfactoriamente o curso baseado nas TIC. Tanto a xestión do tempo e as habilidades metacognitivas



están relacionadas coas actitudes e a motivación do estudante.

- **Persoal dedicado:** Ao igual que os estudantes, os profesores deben ter habilidades técnicas, coñecemento e acceso ao hardware e software, necesarios neste caso para facilitar o deseño e desenvolvemento do curso baseado nas TIC. E deben ter un excelente manexo do tempo e a motivación para proporcionar asistencia e levar o seguimento do estudante. Non obstante algúns autores diferencian rol do profesor, encargado da selección de contidos, seguimento e asistencia ao alumno, do rol do técnico encargado do deseño e creación do curso de aprendizaxe en liña a partir dos contidos, obxectivos e metodoloxías, establecendo desta forma a necesidade de diferentes perfís.
- **Xestión da información:** A pesar de que se posúan unhas habilidades técnicas e un manexo do tempo excepcionais, tanto os profesores como os alumnos requiren de interfaces que reduzan as cuestións loxísticas e técnicas. O uso de boletíns e listas de distribución poden axudar a manexar a sobrecarga de información.
- **Equidade:** Non todos os usuarios contan coas mesmas facilidades de acceso á internet. A tecnoloxía incrementa as diferenzas entre os que teñen e os que non teñen tales posibilidades.
- **Largo de banda:** Este é un dos maiores inconvenientes dende hai unha década e que está a desaparecer rapidamente coa chegada de liñas de banda larga. Actualmente, en Europa, o largo de banda é aceptable e permite transmitir con bos resultados audio e vídeo sincronizados sen os indesexables "saltos" de antano.

#### 23.8.6 Estandarización

Un dos principais problemas dos sistemas de aprendizaxe en liña sempre foi a



reutilización dos contidos, de forma que estes poidan ser utilizados en sistemas diferentes, debido a que a maioría dos sistemas definían os seus propios formatos de almacenamento e procesamento dos contidos educativos, así como a forma de acceder e manexalos. Esta falta de acordo débese en boa medida á descoordinación no desenvolvemento de estándares para e-Learning na década pasada.

Hoxe en día existen multitude de sistemas destinados ao ensino, xa sexa simples xestores de contidos, xestores do proceso de aprendizaxe ou sistemas máis completos capaces de dar soporte a procesos administrativos, ofrecer ferramentas de autoría e edición de cursos, etc. Non obstante, a pesar da variedade existente, a súa heteroxeneidade dificulta a compatibilidade entre eles. Non todos son de código aberto, algúns usan formatos propietarios e xeralmente non é posible reutilizar contidos e estruturas de aprendizaxe entre eles.

Estas incompatibilidades, xa sexan totais ou parciais, repercuten negativamente no custo asociado á implantación dun sistema de aprendizaxe en liña, posto que no mellor dos casos, unha vez superado o tempo de aprendizaxe das distintas aplicacións do sistema, sería necesaria a readaptación de material xa existente para outros sistemas, ou no peor dos casos, crear ese material dende cero. Unha especificación sobre aprendizaxe virtual asegura que o novo material siga funcionando exactamente igual independentemente da plataforma que se utilice, sempre que esas plataformas cumpran a mesma especificación.

### **23.8.7      Plataforma Moodle**

#### **Introdución**

Moodle é un acrónimo de Module Object-Oriented Dynamic Learning Environment. Consiste nunha plataforma que proporciona de forma integral mecanismos para a xestión de cursos. Moodle integra ademais as ferramentas necesarias para crear e xestionar comunidades virtuais orientadas á aprendizaxe en liña. Polo tanto, podemos categorizar a Moodle como unha plataforma tecnolóxica de



tipo LMS (Learning Management System).

Orixinalmente Moodle foi creado por Martin Dougiamas. Baseou o deseño da plataforma partindo de que o coñecemento se constrúe na mente do estudante en lugar de ser transmitido sen cambios a partir de libros. Existe tamén unha importante aposta polo modelo de aprendizaxe colaboradora. O propósito é construír un ambiente centrado no estudante que lle proporcione capacidade para xerar ese coñecemento, baseado nas habilidades e coñecementos propios dos titores ou profesores, en lugar de simplemente publicar e transmitir a información que se considera que os estudantes deben coñecer.

En conclusión, Moodle é un paquete de software para a creación de cursos e sitios web baseados na internet, orientado a dar soporte a un marco de educación construtivista. O sistema é multiplataforma e está rexistrado baixo licenza GNU/GPL.

En canto á arquitectura da plataforma, Moodle é unha aplicación web que se executa en servidores que soportan PHP e facendo uso dunha base de datos para a persistencia da información. Esa base de datos é única, e dende a versión 1.7 Moodle conta cunha capa de abstracción que lle permite seleccionar entre diversos motores de bases de datos, das que MySQL e PostgreSQL son as máis utilizadas.

## **Principais características**

Moodle, como sistema englobado dentro dos xestores de contido, e á súa vez como sistema específico de aprendizaxe en liña, ten as seguintes características:

- Promove unha pedagogía construtivista social fundamentada no traballo colaborador, a realización de actividades e debates.
- A súa arquitectura e ferramentas son apropiadas para clases en liña, ademais de servir como complemento da aprendizaxe presencial.
- Ten unha interface de navegador de tecnoloxía sinxela, lixeira, e compatible.



- Para a súa posta en produción, unicamente é necesaria unha plataforma que soporte PHP e a dispoñibilidade dunha base de datos. Grazas á súa capa de abstracción, Moodle soporta os principais sistemas xestores de bases de datos.
- É unha plataforma segura. Todos os formularios son revisados e as *cookies* cifradas.
- É adaptable e extensible. A maioría das áreas de introdución de texto poden ser editadas usando o editor HTML, tan sinxelo como calquera editor de texto.

## 23.9 Accesibilidade e usabilidade

### 23.9.1 Accesibilidade como calidade dos sistemas

A accesibilidade é unha calidade dos sistemas informáticos vinculada ao campo da interacción entre humanos e ordenadores. Fundamentalmente céntrase na capacidade de acceso ao uso da aplicación ou sistema informático que é obxectivo por parte do usuario. No campo concreto das tecnoloxías web, a accesibilidade fai referencia á capacidade de acceso á web e aos seus contidos por todas as persoas. A accesibilidade pretende facilitar o acceso a calquera tipo de usuario independentemente da minusvalía (física, intelectual ou técnica) que presenten. Tamén está relacionado con aquelas dificultades que se derivan do contexto de uso xa sexa tecnolóxicas ou ambientais. Esta calidade está intimamente relacionada coa usabilidade dos sistemas.

Á hora de deseñar contidos, hai que ter en conta os factores de accesibilidade que permitirán que calquera tipo de usuario poida acceder en condicións de igualdade á información almacenada. Existen mecanismos e estándares actualmente que traballan sobre iso, e as tecnoloxías proporcionadas pola maioría dos SXC permiten estruturar os nosos contidos tendo en conta este tipo de facetas. Un caso concreto dáse cos sitios que teñen un código XHTML semanticamente correcto, permitindo



proporcionar un texto equivalente alternativo ás imaxes e ás ligazóns. Isto supón que os usuarios cegos poden utilizar lectores de pantalla ou liñas Braille para acceder aos contidos. O mesmo acontece cando os vídeos dispoñen de subtítulos; usuarios con dificultades auditivas poderán entendelos perfectamente.

Os sistemas de xestión de contido actuais permiten ademais certa personalización das características do sitio. Factores como o tamaño de letra ou as proporcións da interface comezan a ser xa personalizables por cada tipo de usuario, proporcionando axuda para que os usuarios con problemas visuais poidan lelos sen dificultade.

### **23.9.2 Limitacións na accesibilidade**

Existen fundamentalmente catro tipos de limitacións na accesibilidade dos sitios web:

- **Visuais:** Abranguendo un amplo abano de patoloxías e de distintos graos de deficiencia visual, que poden ir dende a baixa visión á cegueira total, ademais de problemas para distinguir cores.
- **Motoras:** Dificultade ou a imposibilidade de usar as mans, incluídos tremores, lentitude muscular, debido a enfermidades como o Párkinson, distrofia muscular, parálise cerebral ou amputacións.
- **Auditivas:** Xordeira ou deficiencias auditivas.
- **Cognitivas:** Dificultades de aprendizaxe ou minusvalías cognitivas que afecten á memoria, a atención, as habilidades lóxicas, etc.

### **23.9.3 Promovendo a accesibilidade**



A tarefa de promover a accesibilidade no ámbito web corre a cargo do grupo de traballo Web Accessibility Initiative (WAI), que depende directamente do World Wide Web Consortium. En 1999 o WAI publicou a versión 1.0 das súas pautas de accesibilidade Web (WCAG). Co paso do tempo convertéronse nun referente internacionalmente aceptado ata que en decembro do 2008 as WCAG 2.0 foron aprobadas como recomendación oficial.

Estas pautas divídense en tres bloques orientadas especificamente para cada un dos principais perfís que forman parte dun proxecto de desenvolvemento web:

- **Pautas de accesibilidade ao contido na web (WCAG):** Están dirixidas aos profesionais do deseño e desenvolvemento web e proporcionan información e recomendacións acerca de como facer que os contidos do sitio web sexan accesibles.
- **Pautas de accesibilidade para ferramentas de autor (ATAG):** Están dirixidas aos desenvolvedores do software que usan os administradores de sitios web, co obxectivo de proporcionar un mellor soporte para a construción de sitios accesibles.
- **Pautas de accesibilidade para axentes de usuario (UAAG):** Están dirixidas aos desenvolvedores de axentes de usuario (navegadores e similares), para que estes programas faciliten a todos os usuarios o acceso aos sitios web.

#### 23.9.4 Usabilidade

A usabilidade é outro atributo vinculado aos sistemas informáticos, particularmente importante no campo da interacción home-computador. Actualmente a usabilidade está recoñecida como un importante atributo de calidade do software. Actualmente non chega con fabricar sistemas con alto rendemento e fiabilidade, senón que o obxectivo é crear sistemas que sexan cómodos e manexables, adaptados



para os usuarios finais. No marco da usabilidade xerouse un importante centro de servizos no que empresas especializadas desenvolven as súas actividades fundamentalmente orientadas á asesoría nestes campos.

Nos proxectos de desenvolvemento de software en xeral, e nos orientados á distribución e xestión de contidos en particular, o concepto de usabilidade é de importancia capital. Á hora de distribuír contido a través da rede, o portal está aberto a todo tipo de usuarios. Esta faceta comparte importancia co concepto anterior de accesibilidade. Pero ademais, a usabilidade permite incrementar o atractivo, desenvolvendo sistemas sinxelos e intuitivos que permiten un doado manexo e rápida aprendizaxe.

Dende un enfoque do deseño e avaliación de aplicacións software, falamos de usabilidade software como un conxunto de fundamentos teóricos e metodolóxicos que aseguran o cumprimento dos niveis de usabilidade requiridos.

## 23.10W3C

O World Wide Web Consortium, abreviado W3C, é o máximo organismo no ámbito mundial que se encarga de xestionar e publicar as recomendacións e estándares asociados ao World Wide Web. É dicir, o obxectivo deste consorcio é estandarizar os protocolos e as tecnoloxías utilizadas para construír a web, de maneira que o contido estea dispoñible para a maior parte posible da poboación do mundo. As principais actividades ás que se dedica son, á coordinación dos diferentes grupos de traballo no ámbito da xeración de:

- Especificacións e estándares: sobre tecnoloxías asociadas ao WWW.
- Directrices: e recomendacións de desenvolvemento para boas prácticas.
- Ferramentas: que permitan validar a aceptación e cumprimento dos estándares



e recomendacións propostas.

Está dirixida por Tim Berners-Lee, responsable do grupo de investigación que desenvolveu a URL, o protocolo HTTP (HyperText Transfer Protocol, Protocolo de Transferencia de Hipertexto), así como tamén da linguaxe de etiquetaxe HTML (Linguaxe de Marcado de Hipertexto) que son as principais tecnoloxías sobre as que se basea la Web.

Creouse en 1994 no MIT, actual sede central do consorcio. O consorcio está formado por unha gran diversidade de membros e entidades cada unhas das cales colabora nos ámbitos nos que o W3C exerce a súa función. Actualmente está integrado por tres tipos de figuras principais:

- **Membros adscritos do W3C:** garanten a fortaleza e o sentido do consorcio a través do investimento e a participación activa nas actividades do W3C. O W3C conta con máis de 400 organizacións membro provenientes de máis de 40 países, con intereses moi variados. Entre os membros do W3C inclúense provedores de produtos de tecnoloxía e servizos, provedores de contido, usuarios corporativos, laboratorios de investigación, organismos de estandarización e administracións, que traballan conxuntamente para alcanzar un acordo sobre a dirección que debe tomar a web.
- **Equipo do W3C (W3C Team):** O equipo do W3C inclúe máis de sesenta investigadores e enxeñeiros de todo o mundo que dirixen as actividades técnicas do W3C e xestionan as operacións do consorcio. A maioría dos compoñentes do equipo do W3C traballan nunha das tres institucións que albergan o W3C: O MIT/CSAIL, nos Estados Unidos; o ERCIM, coas oficinas centrais en Francia; e a Universidade de Keio, en Xapón. Están coordinados polo director Tim Berners-Lee, o director de operacións Steve Bratt, e un equipo de dirección, e os traballadores do W3C:
  - Mantéñense informados sobre as novas tecnoloxías, as flutuacións do mercado e as actividades de organizacións relacionadas, con intención de



orientar o W3C axeitadamente.

- Organizan as actividades do W3C para, así, cumprir o maior número de obxectivos dentro duns límites prácticos (tales como os recursos dispoñibles).
  - Promoven a cooperación entre os membros, á vez que buscan a súa diversidade, incentivan a innovación, e facilitan a súa activa participación.
  - Divulgan os resultados do W3C aos membros e á prensa, e promoven a súa aceptación na comunidade web; vexa a lista de presentacións públicas realizadas polo Equipo.
- Oficinas do W3C (W3C Offices): O obxectivo das oficinas do W3C é traballar coas comunidades rexionais para potenciar a adopción das recomendacións do W3C entre os desenvolvedores, os creadores de aplicacións, e os difusores de estándares, así como fomentar a inclusión das organizacións máis importantes na creación de futuras recomendacións a través da súa adscrición ao consorcio.

### 23.11 Web 2.0

O concepto de web 2.0 nace para describir aquelas aplicacións web que se centran no usuario, en contraposición á web tradicional (ou 1.0) que simplemente actúa como unha mera presentadora de datos. A web 2.0 fomenta a interacción co usuario, a interoperabilidade, e busca compartir información e a colaboración. Este concepto desenvolveuse principalmente mediante servizos web, redes sociais, wikis, blogs e sistemas de almacenamento de vídeos, entre outros.

Pódese entender á web 2.0 como a evolución dunha serie de aplicacións tradicionais noutras enfocadas ao usuario, non tanto dende un punto de vista tecnolóxico, senón



de concepto e intención. Así, a través do concepto 2.0, certas aplicacións irán abandonando o escritorio e irán migrando á web, sempre baseándose na colaboración co usuario e a interactividade. Para permitir o uso do software en liña e outras aplicacións multimedia, a extensión da banda larga foi un factor fundamental.

A web 2.0 é unha actitude de compartir información, colaboración, interacción, cambio continuo e a creación dunha plataforma global.

A web 2.0 é unha reinterpretación da web, que describe os pasos para chegar a un modelo de comunicación colectiva máis participativa e innovadora. Un dos principais cambios prodúcese na xestión dos datos. Na web 1.0, é a empresa a que xestiona a información, a través dunha rede de expertos, ou algún tipo de procesamento artificial. Pola súa banda, a web 2.0 é un concepto colaborador no que se espera que os datos sexan obtidos por medio dos usuarios.

En 2004, O'Reilly Media realiza unha conferencia sobre web 2.0, onde se comeza a sentar as bases deste concepto. Dende o primeiro momento déixase claro que non consiste nun cambio tecnolóxico na web, senón un cambio na forma na que desenvolvedores e usuarios utilizan a web. Non obstante, Tim Berners-Lee, o creador da World Wide Web, cualificou o termo 2.0 como "labia", xa que, segundo el mesmo, a web xa contiña eses valores dende un principio.

Con anterioridade ao concepto 2.0, a maioría de aplicacións web eran portais estáticos, programados en linguaxe HTML (Hyper Text Markup Language), cunha periodicidade de actualización baixa, e con escasa interacción co usuario, que simplemente "consumía" contido. Unha primeira aproximación ao concepto 2.0 foron



as aplicacións webs dinámicas, nas que as páxinas son servidas ao usuario dinamicamente a partir de contido obtido dunha base de datos.

Algúns expertos na web 2.0 manteñen que a web debe enfocarse á interacción e ás redes sociais, de modo que a web sexa un punto de encontro e dependa do usuario.

Dale Dougherty, de O'Reilly Media, usa por primeira vez o termo "web 2.0" xunto con Craig Cline, de MediaLive, para ilustrar a idea de que a web se estaba a refundar. Unha das principais críticas do concepto 2.0 é que non existe unha definición formal. Dougherty argumentou o seu punto de vista con exemplos de aplicacións web 1.0 e 2.0. Así, o que DoubleClick era na web 1.0, éo AdSense na 2.0. E o que era Ofoto na web 1.0, éo Flickr na 2.0. Outros exemplos son Google (que mide o impacto dos sitios web mediante o número de ligazóns a páxinas web, e non tanto por clics absolutos) ou a Wikipedia (un proxecto colaborador a partir dunha gran cantidade de pequenos usuarios, en lugar dun reducido equipo de expertos).

Deste modo, en outubro de 2004 realízase a primeira conferencia sobre web 2.0, con Dougherty, Cline e John Battelle. Un ano despois realízase unha segunda conferencia, onde se une Tim O'Reilly para resumir os principais trazos da web 2.0. Entre eles, encontramos a innovación, o deseño para múltiples plataformas, e a importancia da participación do usuario. Na web 2.0, a aplicación está orientada polo usuario, que é o que alimenta e modifica unha base de coñecemento da aplicación, a partir dun deseño interactivo e en rede (tal e como definiu Xavier Ribes en 2007).

Algunhas das características que definen a web 2.0 son:

- Permite o uso de aplicacións en liña, substituíndo potencialmente as aplicacións de escritorio, e dende un punto de vista multiplataforma. As aplicacións



executadas en web son independentes do navegador e do sistema operativo dende o que se executan, o que facilita o desenvolvemento.

- Permite a transferencia de información e contidos entre aplicacións web.
- Permite unha experiencia de usuario simple, cunha curva de aprendizaxe rápida.
- Permite engadir novas funcionalidades dun xeito simple e intuitivo.
- Permite gradualmente a virtualización das estruturas sociais no mundo en liña.
- Fai que o papel do usuario gaña importancia, levándoo ata o rol de codesenvolvedor, e fomentando un desenvolvemento colectivo. Pódese dicir que un elevado número de usuarios poden substituír nalgúns casos un reducido grupo de expertos.
- Fomenta a interoperabilidade das aplicacións web, a incrustación de código externo e o uso de API mesmo por usuarios non expertos. Neste punto é destacable a contribución da tecnoloxía RSS (Real Simple Syndication) que separa totalmente contido e presentación dos datos. RSS permite coñecer as actualizacións dun portal web sen necesidade de visitalo, e ademais fai posible crear sistemas de agrupación de información alimentado por distintas fontes (os chamados agregadores, como Google Reader). RSS baséase en XML, que foi outro actor importante na transmisión de información para a web 2.0.
- Fomenta a participación para a mellora continua da aplicación.

## 23.12 Wikis

Un wiki (do hawaiano "wiki wiki" = rápido) é un tipo de aplicación web que permite a edición dos seus contidos de forma concorrente, voluntaria e colaboradora por parte de usuarios, autenticados ou non, a través dun navegador web, e co obxecto de acumular coñecemento de xeito conxunto a xeito de repositorio centralizado. Os wikis



baséanse no esforzo democrático e compartido sobre unha base de igualdade e facilidade: todo o mundo debe poder achegar novo contido.

Cada páxina de contido do wiki correspóndese cun nome unívoco e simple que facilita a súa comprensibilidade, así como o seu enlazamento dende outras páxinas de wiki e portais externos. Ademais, existe unha linguaxe wiki que facilita a edición e creación de xerarquías, categorías, thesaurus e taxonomías por medio de enlaces internos. Isto crea unha estrutura descentralizada que fai que a navegación polos wikis sexa non lineal, xa que cada páxina contén numerosas ligazóns a outras páxinas.

A orixe dos wikis provén de Ward Cunningham, quen desenvolveu un servidor wiki como repositorio de patróns de deseño en Portland (Portland Pattern Repository) en 1995 (chamado WikiWikiWeb), e definiu o wiki como 'a base de datos en liña máis simple que podería funcionar'. Posteriormente, en 2001, Jimbo Wales e Larry Sanger usan un wiki como xerme do seu proxecto de enciclopedia Wikipedia, unha enciclopedia libre e en rede. Comezan utilizando o software UseMod, aínda que finalmente desenvolven un software propio, denominado Media Wiki, que se converteu nun estándar para outros wikis. Foi precisamente a Wikipedia, e outras enciclopedias colectivas, as que colaboraron no auxe dos wikis.

Cada artigo coescribese co resto da comunidade. Dentro da colaboración múltiple, o wiki posibilita un historial de actualizacións que actúa a xeito de control de versións temporal e por usuario. Ao traballar como un repositorio, os wikis permiten volver a versións anteriores con facilidade. Normalmente non existe unha supervisión, e a edición baséase na negociación entre usuarios, pero a tendencia actual é que exista un reducido grupo de usuarios cun rol especial que revisa os contidos e fai posible manter a calidade nos contidos e evitar incoherencias e sabotaxes. Noutros casos



requírese autenticación soamente para manter o historial de cambios e asinar os contidos.

Hoxe en día, a versión inglesa da Wikipedia é o wiki máis grande que existe. O resto de versións noutros idiomas, e outras aplicacións wiki máis específicas, contan con menor número de usuarios debido a que as súas comunidades son menos numerosas.

O que diferencia un wiki doutras aplicacións web de xestión de contidos é a súa rapidez para crear e modificar páxinas, así como a simplicidade e lexibilidade da súa interface. Isto fomentou o alto número de participantes nos proxectos wiki, destacablemente maior que noutros proxectos colaboradores web. Se ben o usuario posúe un alto grao de liberdade para editar contidos, existen equipos para que as páxinas wiki garden coherencia entre si, o que fai aumentar a calidade do repositorio. Estas estruturas predefinidas facilitan que a edición de contidos sexa o máis simple posible. Ademais, os wikis teñen normalmente un deseño sinxelo que non se adoita modificar.

Os wikis teñen un compoñente de altruísmo máis notable que outras aplicacións web. Se ben é colaborador, e o usuario fomenta a pertenza a unha comunidade con intereses comúns, o importante da colaboración é o ben común, e non tanto que un usuario destaque máis ou menos polas súas contribucións, xa que a autoría dos contidos rara vez é exclusiva. Non só iso, os contidos están en continua edición, de xeito indefinido, o que obriga á reflexión e aínda continua revisión das ideas.

É destacable a gran relación entre os wikis e o mundo educativo. Os wikis cambiaron radicalmente o modelo de consulta de coñecementos, pasando das mastodónticas



enciclopedias estáticas aos proxectos colaboradores actuais baseados en wikis onde o usuario xoga un papel fundamental.

Os software máis utilizados para o desenvolvemento de wikis son: MediaWiki, TikiWiki ou CitiWiki (en PHP), e JSPWiki ou XWiki (en Java), entre outros. A aplicación só define a presentación básica dos datos (estilos, ámbito...), pero a edición dos datos corre por parte dos usuarios.

### 23.13 Blogs

Un blog (ou weblog) é un sitio web de actualización frecuente que recompila artigos, presentando primeiro o máis recente, e que permite a interacción cos lectores por medio de comentarios sobre os artigos. O blog é cronolóxico (permite manter unha liña de tempo de publicación), é colaborador (poden publicar varios autores), e é interactivo (os lectores poden publicar os seus comentarios, de modo que o autor poida contestarlles para conformar un diálogo). O administrador do blog pode tomar distintas opcións de deseño, como non permitir comentarios, e administrar os artigos (borralos ou reordenalos).

A temática do blog é variada, sendo a súa motivación primixenia actuar a xeito de diario persoal ou bitácora. Dende aquela, hainos corporativos, xornalísticos, educativos, etc. O blog que se dedica esencialmente á publicación de fotografías denomínase fotolog ou fotoblog (e videoblog no caso de vídeos). Unha práctica habitual é proporcionar un gran número de ligazóns que amplíen a información para cada entrada. En ocasións as entradas permiten que se lles faga *trackback* (unha ligazón inversa) para coñecer quen enlazou a entrada dende outro sitio web.



As entradas do blog adoitan agruparse en categorías, e é frecuente a práctica de indicar palabras clave ou etiquetas para facilitar a busca por contido. Os blogs tamén proporcionan arquivos e índices mensuais e anuais que permiten unha navegación ordenada por data. Así mesmo, é habitual que as entradas proporcionen facilidades para ser compartidas noutros blogs, ou por correo electrónico, así como por sindicación de contidos, mediante o uso de tecnoloxía RSS ou Atom.

Gran parte do auxe dos blogs débese á súa facilidade de mantemento e ás numerosas alternativas gratuítas dispoñibles. Non son necesarios grandes coñecementos técnicos para administrar un blog, e nin sequera para crealo, o que os achegou ao gran público. Ademais, como calquera outro sitio web, un blog pode ter publicidade e xerar ingresos.

Existen principalmente dous tipos de solucións blog: as que proporcionan unha solución completa de software e aloxamento web gratuíto (como Blogger ou LiveJournal), e as que simplemente proporcionan software que precisa ser instalado nun sitio web (como WordPress). Este último é un tipo específico de xestor de contido (SXC).

### **23.14 Comunidades virtuais**

Unha comunidade virtual é aquela na que os vínculos, interaccións e relacións non teñen lugar nun espazo físico, senón nun espazo virtual como a internet.



As comunidades virtuais xorden coa internet e obteñen o seu modelo das comunidades non informáticas, existentes dende moito antes. A primeira comunidade virtual data dos anos 70, aínda que o seu maior desenvolvemento se produce nos 90, volvéndose, neste momento, accesibles ao público en xeral, grazas ao nacemento de da World Wide Web (WWW) e a expansión de ferramentas como os chats, correo electrónico ou mensaxaría instantánea. Ata este momento, as comunidades estaban restrinxidas ao ámbito científico e a expertos en informática.

Os usuarios sen acceso á internet, implantaron e popularizaron o uso do sistema de taboleiro de anuncios (BBS ou Bulletin Board System), que se consistía nun sistema que funcionaba mediante un acceso mediante módem telefónico a unha central (o BBS), o cal podía basearse nunha ou varias liñas de teléfono. Nos BBS podíanse manter conversacións, intercambiar arquivos, publicar comentarios, etc. Nesta época as comunidades eran independentes, o máis habitual era que os usuarios particulares empregasen os seus propios equipos domésticos para proporcionar servizo con ata un único módem de entrada.

Actualmente, as comunidades virtuais evolucionaron, converténdose nunha ferramenta moi útil dende o punto de vista empresarial. Isto débese á mellora que ofrecen ás organizacións na súa dinámica de traballo interno, nas relacións cos clientes ou no incremento da eficiencia dos seus procedementos.

Dende o punto de vista social, as comunidades virtuais permiten aos usuarios relacionarse cos demais, adquirindo así un carácter socializador.

Estímase que no ano 2000 máis de 40 millóns de persoas participaban en comunidades virtuais, as cales podemos caracterizar da seguinte forma:

- Asociación virtual de persoas



- Existe un propósito determinado que é a razón de ser da comunidade virtual
- Existe un gran desexo de interacción entre os usuarios para satisfacer unhas necesidades ou desempeñar uns roles concretos
- Existen sistemas que avalían e miden as interaccións e favorecen a cohesión entre os membros

O principal inconveniente ao que se enfronta o desenvolvemento das comunidades virtuais é a problemática da organización interna delas, que adoita ser moi difícil de que establecer e xestionar. En moitos casos, é demasiado custoso crear a estrutura da comunidade co que se pode chegar a perder o verdadeiro propósito da creación dela.

Unha comunidade virtual queda definida dende 3 puntos de vista:

- Comunidade virtual como lugar: localización onde os individuos poden manter relacións económicas ou sociais.
- Comunidade virtual como símbolo: os membros dunha comunidade desenvolven un sentimento de pertenza a unha estrutura maior.
- Comunidade virtual como virtual: a pesar das similitudes entre as comunidades físicas e as virtuais, unha comunidade virtual desenvólvese principalmente nun ámbito virtual que non pode ser asimilable cunha localización física.

### **23.15 Redes sociais**

O concepto de rede social está estreitamente ligado co de comunidade virtual, entendéndose unha comunidade virtual como un caso máis específico de rede social.



Unha rede social é unha estrutura de nodos onde distintos actores, que poden ser individuos ou organizacións, están conectados mediante un serie de relacións baseadas en propiedades comúns. Unha rede social aséntase sobre certo tipo de relacións, económicas, laborais, familiares, políticas, deportivas, etc.

Unha rede social é distribuída cando a súa localización non está limitada a un sitio en concreto senón que se distribúe a nivel xeográfico. No caso dunha rede social, é lóxico pensar que non todos os actores participantes nesa rede se encontren localizados nun único espazo.

Unha rede social apóiase no uso dalgunha tecnoloxía de comunicación que lles permite aos distintos usuarios interactuar entre si. Neste caso, o máis habitual é empregar a internet como tecnoloxía subxacente; non obstante, tamén existen redes sociais baseadas en tecnoloxías móbiles e mesmo en tecnoloxías non dixitais como o teléfono, o fax ou o correo postal.

Actualmente as redes sociais oríéntanse ao redor dun sitio web que lles ofrece aos usuarios unha serie de servizos, como son chat, mensaxaría instantánea, carga de imaxes, vídeos, grupos de debate, etc. Un dos servizos que teñen unha maior importancia son os do "software social", que abrangue todas aquelas aplicacións que simulan procesos sociais do mundo real. O exemplo máis habitual é a simulación do efecto "amigo dun amigo", neste caso a aplicación localiza os amigos dos nosos amigos para poder así facilitar o contacto con novos usuarios.

Dentro das redes sociais máis empregadas hoxe en día, podemos citar Facebook, Youtube, Twitter, Myspace, Orkut, Hi5 etc.



### 23.16 Sindicación de contidos

Antes de definir que é a sindicación de contidos, é necesario aclarar que se ben a expresión correcta é "redifusión web", o termo "sindicación web" está moi expandido no seu uso, especialmente no que se refire a contidos web, aínda que esta redifusión pode levarse mediante calquera medio de comunicación.

A sindicación de contidos (sindicación web ou redifusión web) consiste no reenvío ou redistribución de contidos dende un sitio web de orixe, ata outro sitio web receptor, o cal á súa vez se pode ver como un emisor dos contidos, posto que estes deixan de estar limitados aos usuarios do sitio web inicial. Esta redifusión de contidos faise habitualmente mediante unha licenza ou contrato entre os sitios web de orixe e destino.

Os contidos que se redistribúen adoitan codificarse en XML, aínda que isto non é obrigatorio e pode empregarse calquera outro formato soportado por http.

Existen dúas familias máis destacadas en canto a formatos de redifusión web, que son RSS e Atom. De feito, actualmente o termo RSS (Really Simple Syndication) empezouse a usar indistintamente para referirse a calquera dos 2 formatos de fontes web, o propio RSS ou Atom.

Para poder ler unha fonte web é necesario realizar unha subscrición mediante un agregador, o cal mostra os novos contidos que fosen publicados polo provedor da fonte web subscrita.



### 23.16.1 Fonte web

Unha canle web ou fonte web (Web feed) é un medio de redistribución de contidos web, que se emprega para subministrar información aos subscritores de xeito actualizado, os cales deben contar cun programa "agregador" para acceder a todas as fontes ás que están subscritos dende un mesmo lugar.

Como xa comentamos anteriormente, os dous principais formatos de fonte web son RSS e Atom, ambos os dous escritos en XML.

### 23.16.2 Agregador de noticias

Un lector RSS ou agregador de noticias (eventualmente só agregador) é unha aplicación que permite establecer unha subscrición a fontes de noticias en formatos Atom, RSS e outros derivados de XML/RDF. A función do agregador consiste en reunir todas as noticias e contidos publicados nos sitios con redifusión escollidos e mostralas de xeito unificado ao usuario, de tal forma que o usuario poida saber que webs incorporaron ou modificaron contidos dende a última lectura e, en cada caso, cal é o contido destas.

Os lectores RSS volvéronse máis populares coa implantación de XML e a web semántica, e hoxe en día existe un gran número de blogs e sitios web que ofrecen as súas actualizacións, que son administradas e agregadas nun único lugar grazas a ferramentas como as de Google Reader, Netvibes, etc.

### 23.16.3 Formato RSS



RSS é un formato XML para syndicar contidos web que se emprega para difundir información aos usuarios subscritos a unha fonte de contido. Este formato caracterízase por permitir a distribución de contidos sen necesidade de empregar un navegador, xa que se utiliza un agregador de contidos RSS, e aínda así é posible empregar o navegador para ver os contidos RSS.

De feito, as últimas versións dos navegadores permiten visualizar os RSS sen necesidade dun agregador.

Este formato desenvolveuse especificamente para aqueles sitios que se actualizan de forma habitual e mediante o cal se pode compartir a información e ser empregada noutros sitios web.

#### **23.16.4 Estándar Atom**

Atom fai referencia a 2 estándares relacionados entre si:

- Protocolo de publicación Atom (AtomPub ou APP): protocolo baseado en http para crear ou actualizar recursos en web.
- Formato de redifusión Atom: ficheiro en formato XML usado para redifusión web.

Para crear un contido que poida ser tratado con agregador ou por outro sitio web que redifunde os contidos da fonte, o propietario do sitio web pode empregar un software específico, como un sistema de xestión de contidos, que publica unha fonte web de artigos recentes nun formato estándar e lexible polos ordenadores.



O formato Atom foi desenvolvido como unha alternativa a RSS. Atom xorde pola incompatibilidade existente entre algunhas versións do protocolo RSS. O formato de redifusión Atom publicouse como un "estándar proposto" de la IETF con RFC 4287, mentres que o protocolo de comunicación se publicou como RFC 5023.

### 23.17 Podcast

*Podcasting* consiste en distribuír arquivos multimedia, xeralmente audio ou vídeo, mediante un sistema de redifusión que lles permita aos usuarios establecer subscricións, ao empregar un programa de descarga para poder visualizar o contido no momento que se desexe. Tamén existe a posibilidade de descargar os contidos sen unha subscrición previa. O *podcasting* é un tipo de sindicación onde os arquivos que se redistribúen son de contido multimedia.

Ao principio o *podcasting* referíase exclusivamente ás retransmisións de arquivos de audio, aínda que máis tarde se estendeu o concepto para facer referencia tanto a audio como a vídeo de xeito indistinto.

O contido dos *podcasts* é moi diverso: sobre tecnoloxía, política, noticias, contidos educativos, etc. En función do produtor do *podcast* a súa complexidade, número de participantes e estrutura varían significativamente. Algúns aseméllanse a programas de radio, con varios participantes e diversas opinións, e outros parécense máis a comunicados ou monólogos dunha única persoa, nos que a duración xeralmente é máis curta.



Os *podcasts* adoitan ser accesibles dende o sitio web en que foron colocados. Existen blogs que permiten realizar *podcasting* mediante o uso de complementos gratuítos. Ademais estes arquivos tamén se poden descargar.

### 23.17.1 Podcasting fronte Streaming

Antes da aparición do *podcasting*, a forma habitual de transmitir contidos multimedia era o *streaming* ou *webcasting*. Mediante este sistema, provedores de contidos como cadeas de televisión ou radios empregaban o *streaming* para emitir dende un servidor central.

*Podcasting* e *streaming* presentan certas diferenzas, as máis destacadas:

- Co *streaming* non se produce a descarga do ficheiro senón que este se reproduce en modo fluxo mentres se está a descargar. Cando remata a reprodución, o ficheiro non se almacena no equipo receptor. Isto presenta a vantaxe do aforro de espazo de almacenamento, non obstante, sempre que se queira ver ou oír novamente o arquivo volverá ser descargado e non pode ser reproducido se non existe unha conexión á internet.
- Co *streaming*, é necesario acceder ao sitio web onde está a canle desexada e indicar que a reprodución do contido debe iniciarse mediante algún tipo de ligazón, botón, etc. Non obstante, co *podcasting*, cando o contido está dispoñible, este descárgase de xeito automático e pode ser escoitado en calquera momento.



- O *streaming* presenta máis problemas de compatibilidade entre os distintos sistemas empregados que o *podcasting*.
- O *streaming* é máis sensible a problemas de conexión á internet ou de sobrecarga do servidor xa que a descarga se produce mentres se está a reproducir o arquivo.

### 23.18 Suites ofimáticas en web

Unha suite ofimática é un conxunto de ferramentas que se utilizan habitualmente en ámbitos de oficina para o traballo con documentos de calquera tipo. Normalmente inclúense nestes paquetes un editor de textos, un xestor de follas de cálculo, un xestor de bases de datos, un programa de creación de diapositivas de presentación etc. Tradicionalmente todas as suites ofimáticas eran ferramentas de escritorio. Actualmente existen distintas versións de suites ofimáticas pero en web, ás que se accede mediante o uso dun navegador.

#### 23.18.1 Feng Office

Feng Office é unha aplicación libre de tipo Web Office, antes coñecida como OpenGoo. É un sistema completo que proporciona funcionalidades para crear, publicar, colaborar e compartir documentos.

Feng Office permite crear e traballar entre outros, sobre:

- *Documentos*: permite aloxar documentos de todo tipo e editar directamente



algúns deles.

- *Listas de tarefas*: permite a creación de listas de tarefas asignadas a distintos usuarios, con opcións de notificación, categorización, etc
- *Correo electrónico*: permite centralizar a xestión das distintas contas de correo.
- *Calendario*: permite establecer reunións e unha xestión das actividades diarias.
- *Axenda*: permite realizar unha xestión de contactos.

Esta aplicación pode funcionar baixo un modelo SaaS (Software as a Service), onde os servidores do proveedor a través dun navegador nos permiten traballar coa aplicación, pero tamén é posible realizar unha instalación da aplicación nun servidor propio; neste caso os requisitos deste sistema pasan por un servidor web Apache, PHP e MySQL como base de datos.

### 23.18.2 Google Docs

Google Docs & Spreadsheets é un programa web gratuíto que permite crear e traballar sobre uns documentos de xeito individual ou en grupo.

Google Docs componse de:

- Procesador de textos
- Follas de cálculo
- Programa de presentación sinxelo
- Editor de formularios

Entre as vantaxes de Google Docs, encóntrase o feito de que pode ser usado tanto en liña como sen conexión. Nesta modalidade sen conexión, os cambios que se



introduzan nos documentos serán actualizados de forma automática en canto a conexión coa internet se restableza.

Ademais recentemente incorporouse a compatibilidade entre Google Docs e os dispositivos móbiles, de tal forma que se poida non só acceder aos documentos senón tamén editalos.

### **23.18.3 Office Web Apps**

Office Web Apps é a solución de Microsoft para as suites ofimáticas na web. É unha versión gratuíta baseada no conxunto de aplicacións de Microsoft Office.

Office Web Apps, componse de:

- Word Web App
- Excel Web App
- PowerPoint Web App
- OneNote Web App

Estas aplicacións permiten acceder aos documentos a través do navegador, así como compartir arquivos e traballar sobre eles de forma colaboradora.

### **23.19 Almacenamento en web**



Un servizo de almacenamento de arquivos en liña (servizo de aloxamento de arquivos ou centro de medios en liña) é un servizo de aloxamento en liña que ten por propósito facilitar o almacenamento de contido estático, como arquivos, documentos, etc. Por norma xeral este tipo de servizos prové de accesos a través de diversas interfaces, web, ftp, etc.

### **23.19.1      Dropbox**

Dropbox é un servizo de aloxamento de arquivos na nube, que permite almacenar e sincronizar arquivos entre distintos ordenadores e mesmo con distintos usuarios. Como característica principal cómpre destacar que consiste nun sistema multiplataforma e que presenta versións tanto gratuítas coma de pagamento.

O funcionamento é moi sinxelo, cada ordenador cliente instala un software que permite aos usuarios desprazar calquera contido a unha carpeta designada, que se integra no sistema de arquivos do sistema que se trate. Unha vez colocado un arquivo nesa carpeta, ou modificado, este é sincronizado na nube e con todos os demais ordenadores onde estea instalado o cliente Dropbox dese usuario. O acceso aos arquivos da carpeta de Dropbox tamén se pode realizar a través da web e mesmo ser compartido por varios usuarios. Aínda que Dropbox funciona como un servizo de almacenamento, o seu propósito céntrase máis na sincronización e compartición de arquivos.

#### **21.19.1.1    Funcionalidades de Dropbox**

- Historial de revisións: os arquivos borrados da carpeta Dropbox poden ser recuperados dende a web ou dende calquera dos ordenadores sincronizados.
- Historial do documento: pódese acceder ao historial dun documento, de tal



forma que se pode traballar sobre este, sen que isto afecte ás versións preexistentes.

- Optimización da conexión: ao modificar un arquivo nunha carpeta Dropbox, o sistema só cargará as partes do documento que foron modificadas cando se produza a sincronización.

## **23.20 Escritorios virtuais**

Un escritorio virtual consiste nun servizo de virtualización aplicado sobre un escritorio tradicional. Neste caso o escritorio do usuario execútase nun servidor, onde as ordes dese usuario se transmiten en liña ao servidor, que envía de volta os resultados desas accións.

A virtualización de escritorio é relativamente recente e describe a separación do contorno que percibe o usuario, que engloba os seus datos e programas, da máquina física na que estes se almacenan e executan. Neste caso, o usuario pode ter un sistema completo e empregar adicionalmente o escritorio virtual para certo tipo de tarefas, ou mesmo, o usuario pode contar cun sistema sinxelo tipo terminal, onde toda as tarefas do usuario se realizan directamente contra o servidor.

### **23.20.1 eyeOS**

eyeOS é un sistema libre e multiplataforma que se basea no estilo que teñen os escritorios nos sistemas operativos tradicionais, e inclúe a estrutura dun sistema operativo así como certas aplicacións ofimáticas como procesador de textos, calendario, navegador, xestor de arquivos, etc.



Este sistema diferénciase doutros en que non necesita de ningún software adicional para poder usalo, posto que todo o acceso se realiza mediante un navegador web. Recentemente, o sistema foi adaptado para poder utilizarse en dispositivos móbiles.

### 23.21 Mashups

Un mashup é unha aplicación ou páxina web que usa e combina funcionalidades e datos dunha ou máis fontes para crear novos servizos. Implica unha integración rápida e sinxela, xeralmente con API abertos e fontes de datos, para producir resultados enriquecidos. Tómanse unha serie de datos existentes e transfórmanse noutros cun valor engadido que son máis útiles tanto no ámbito persoal como profesional. Como principais características dos mashup pódese destacar a visualización, a combinación e a agregación.

Os mashup compóñense de 3 partes:

- Proveedor de contidos ou fonte de datos. Os datos están accesibles a través dun API e mediante distintos protocolos como RSS.
- Sitio mashup: aplicación web que ofrece un servizo a partir de distintas informacións que non son súas.
- Navegador web: é a interface coa que o usuario interactúa co mashup.

Un erro moi frecuente é confundir os contidos embebidos cos que existen nun mashup. Un sitio que permite embeber por exemplo un vídeo ou un arquivo de son,



non é un mashup, xa que non existiu ningún tipo de procesado nestes datos que permita incrementar o valor que estes teñen para o usuario.

Existe distintos tipos de mashups:

- *De consumidores:* é o máis coñecido. Intégranse datos de diversas fontes e accédese a través dunha interface sinxela. Exemplo: Google Maps.
- *De datos:* mestúranse datos de tipo similar de distintas fontes. Exemplo: combinación de múltiples agregadores RSS nun só.
- *Empresariais:* integra datos de fontes tanto externas coma internas. Exemplo: incorporar maior información a un informe estratéxico mediante datos existentes nalgún rexistro oficial.
- *De negocio:* é unha combinación dos 3 anteriores.

## 23.22P2P

Unha rede P2P (Peer-to-peer, rede de pares ou rede punto a punto) é unha rede de ordenadores na que todos ou algúns dos aspectos funcionan sen que existan clientes nin servidores fixos, senón unha serie de nodos que actúan como iguais entre si, onde cada un deles é á vez cliente e servidor.

Este tipo de redes permite o intercambio directo de información entre os ordenadores que están conectados, habitualmente para compartir ficheiros de calquera tipo, aínda que tamén se emprega para telefonía VoIP.

### 23.22.1 Características



A continuación detállanse algunhas características das redes P2P:

- *Anonimato*: é importante que o autor dun contido, o seu lector, editor e o servidor que o almacena sexan anónimos
- *Descentralización*: por definición os nodos P2P son iguais e a rede descentralizada. Ningún nodo é imprescindible para o funcionamento da rede.
- *Robustez*: ao tratarse de redes distribuídas, a robustez tamén se ve incrementada xa que en caso de se producir un fallo, ao existir unha réplica dos datos en múltiples destinos, a información desexada sempre se pode encontrar, ao non depender dun servidor central.
- *Seguridade*: consiste en identificar e evitar nodos maliciosos, así como o contido potencialmente perigoso, etc. Os mecanismos de seguridade máis destacados neste caso son: caixas de area, reputación, comunicacións seguras, comentarios sobre os ficheiros, cifrado multiclave, etc.
- *Escalabilidade*: canto maior número de nodos estean conectados a unha rede P2P mellor será o funcionamento. Cando se incorporan novos nodos, cos seus recursos, os recursos totais do sistema aumentan.

## 23.22.2 Tipos de redes P2P

### 21.22.2.1 Redes P2P centralizadas

Este tipo de rede caracterízase por:

- Arquitectura monolítica onde todas as transaccións se fan a través dun único servidor, o cal almacena e distribúe os nodos onde se almacenan os contidos.
- Todas as peticións dependen da existencia do servidor.
- Administración dinámica.
- Privacidade dos usuarios limitada.



- Falta de escalabilidade.

#### **21.22.2.2 Redes P2P híbridas, semicentralizadas ou mixtas**

Este tipo de redes caracterízase por:

- Existe un servidor que atende peticións pero non almacena información.
- O servidor administra os recursos, encamiñamentos e comunicación entre nodos.
- Os nodos son os encargados de almacenar a información.
- O servidor central recoñece a información que desexa compartir cada nodo.
- Pode existir máis dun servidor que xestione os recursos compartidos.
- Os nodos poden seguir en contacto directo entre eles en caso de que o servidor ou servidores caian.

#### **21.22.2.3 Redes P2P puras ou totalmente descentralizadas**

Este tipo de redes caracterízase por:

- Son as máis comúns e versátiles posto que non necesitan de ningún tipo de xestión central.
- Redúcese a necesidade de usar un servidor central.
- Cada nodo é á vez cliente e servidor.
- As conexións establécense entre usuarios, coa axuda dun terceiro nodo que permite ligar esa conexión.
- Non existe un encamiñador central.



### 23.23 Web semántica

A web influíu moito no modo de comunicación dos últimos tempos e se ben ten multitude de vantaxes, como o acceso a millóns de recursos independentemente da nosa localización, tamén existen dificultades como son a sobrecarga de información e a heteroxeneidade das fontes de información, o que nos leva a un problema de interoperabilidade.

Coa web semántica estes problemas soluciónanse, permitindo que os usuarios deleguen certas tarefas no software. Grazas á incorporación de maior "semántica" á web, o software é capaz de procesar o contido, combinalo, realizar deducións, etc.

#### 23.23.1 Definición de web semántica

A web semántica (semantic web) baséase na idea de incorporar metadatos ontolóxicos e semánticos á web. Esta información adicional describe o significado, contido e relación entre os datos. Ademais debe ser proporcionada de xeito formal para que poida ser avaliada automaticamente por equipos de procesamento. Ao enriquecer a web con máis significado, pódense obter solucións a problemas comúns na busca de información.

A web baséase fundamentalmente en documentos HTML, o cal non é demasiado versátil á hora de categorizar os elementos que configuran o texto. A función da web semántica é resolver estas deficiencias de tal forma que se poidan describir os contidos dunha web, mediante tecnoloxías como RDF, OWL, ademais de XML. Este tipo de tecnoloxías proporciona descrições explícitas dos distintos recursos incorporando unha serie de etiquetas interpretables polos xestores de contidos, de tal forma que sexa posible a interpretación dos documentos, o tratamento da súa información, etc.



### 23.23.2 RDF, SPARQL e OWL

A web semántica, para realizar unha axeitada definición dos datos, emprega fundamentalmente RDF, SPARQL e OWL.

- *RDF*: proporciona información sobre os recursos da web de forma simple e descritiva.
- *SPARQL*: é a linguaxe de consulta de RDF. Permite realizar buscas sobre os recursos da web semántica.
- *OWL*: é un mecanismo que permite desenvolver vocabularios específicos que se poidan asignar aos recursos. Proporciona unha linguaxe para definir ontoloxías que se poden usar a través de distintos sistemas.

As ontoloxías encárganse de definir os conceptos empregados para describir e representar unha área de coñecemento, inclúen as definicións de conceptos básicos e a relación entre estes.



## 23.24 Bibliografía

- Jerri L. SEO: Optimización de Posicionamento en Buscadores. Ledford Anaya Multimedia
- V. Canseco G. Gerónimo. Breve introducción a los sistemas colaboradores: Groupware& worflkow. 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 (xullo 1995). *The USENET Handbook (Nutshell Handbook)*. O'Reilly. [ISBN 1-56592-101-1](#).
- Kate Gregory, Jim Mann, Tim Parker, and Noel Estabrook (xuño 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 (xuño 1995). *Using Usenet Newsgroups*. Que. [ISBN 0-7897-0134-0](#).
- Mark Harrison (xullo 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

Colexiado do CPEIG



## **24. FERRAMENTAS OFIMÁTICAS. PROCESADORES DE TEXTO, FOLLAS DE CÁLCULO, BASES DE DATOS, PRESENTACIÓN.**



## **Tema 24: Ferramentas ofimáticas. Procesadores de texto, follas de cálculo, bases de datos, presentación.**

---

### **ÍNDICE**

#### **24.1 Ferramentas ofimáticas**

*24.1.1 Definición de suite ofimática*

*24.1.2 OpenOffice.*

*24.1.3 Microsoft Office.*

*24.1.4 Suites ofimáticas en Web*

#### **24.2 Procesadores de texto**

*24.2.1 Historia dos procesadores de textos*

*24.2.2 Procesadores ou editores de textos*

*24.2.3 Exemplos de procesadores de textos*

#### **24.3 Follas de cálculo**

*24.3.1 Microsoft Excel*

*24.3.2 Openoffice.orgCalc*

#### **24.4 Bases de datos**

*24.4.1 Características*

*24.4.2 Sistema de xestión de base de datos (SXBD)*

*24.4.3 Vantaxes das bases de datos*

*24.4.4 Desvantaxes das bases de datos*

*24.4.5 Ferramentas para a xestión de bases de datos presentes en suites ofimáticas*

#### **24.5 Presentación**

*24.5.1 Tipos de presentación*

*24.5.2 Ferramentas de presentación*

#### **24.6 BIBLIOGRAFÍA**



## ***24.1 FERRAMENTAS OFIMÁTICAS***

O termo ofimática pódese definir como a automatización, mediante sistemas electrónicos, das comunicacións e procesos administrativos nas oficinas. As actividades básicas dun sistema ofimático comprenden o almacenamento de datos en bruto, a transferencia electrónica deles e a xestión de toda a información (en formato dixital) relativa ao negocio. A ofimática axuda a optimizar ou automatizar os procedementos existentes na administración dunha empresa.

Dentro da clasificación tradicional do software, o software ofimático encádrase dentro da categoría software de aplicación.

Pódense clasificar os sistemas ofimáticos segundo distintos factores ou características:

- Por tipo de licenza.

Todo software ten unha licenza de uso que conforma un contrato entre o propietario do software e o usuario que define que é o que se pode facer con el. Unha primeira clasificación que ten en conta este feito permite clasificar o software de ofimática en dous grandes bloques: software libre e software privativo.

O software libre (en inglés FLOSS) é a denominación do software que prové o usuario de distintos niveis de liberdade sobre o produto software adquirido e polo tanto, unha vez obtido, pode ser usado, copiado, estudado, modificado e redistribuído libremente.

O software non libre (tamén chamado software privativo, software propietario, software privado, software con propietario ou software de propiedade) refírese a calquera programa no que os usuarios están limitados en



canto ás posibilidades de usalo, modifícalo ou redistribuílo (con ou sen modificacións). O habitual neste tipo de software é que o código fonte non está dispoñible ou o acceso a este código fonte está restrinxido dalgunha forma.

- Por función

Unha segunda clasificación ten en conta o obxectivo do software, isto é, a función que cobre unha aplicación dentro do esquema de funcionamento administrativo da organización. Os tipos ou clases de ferramentas ofimáticas segundo a función á que están dirixidos son:

- Sistemas xestores de bases de datos.
- Sistemas de traballo en grupo.
- Sistemas de tratamento, almacenamento e arquivo de textos e documentos.
- Follas de calculo.
- Edición e deseño de gráficos.
- Axendas e organizadores persoais.
- Correo electrónico e navegación pola internet.

#### **24.1.1**      *Definición de suite ofimática*

Os fabricantes de software ofimático habitualmente organizan as súas aplicacións en suites ou paquetes. Os diferentes sistemas comerciais ou non comerciais propoñen familias de aplicacións baixo unha mesma estrutura. O conxunto destes programas que proporcionan unha interface idéntica e teñen a posibilidade de intercambiar ficheiros denomínase suite ou paquete informático. Con estas aplicacións preténdese satisfacer as necesidades dos usuarios no terreo da ofimática.



Un suite ofimática é un conxunto de ferramentas que se utilizan habitualmente en ámbitos de oficina para o traballo con documentos de calquera tipo. Normalmente, tal como se vía no apartado de clasificación por funcións, inclúense nestes paquetes un editor de textos, un xestor de follas de cálculo, un xestor de bases de datos, un programa de creación de diapositivas de presentación etc. Tradicionalmente todas as suites ofimáticas eran ferramentas de escritorio que se executaban no ordenador do usuario. Actualmente existen distintas versións de suites ofimáticas en web, ás que se accede mediante o uso dun navegador, estando todos os arquivos e datos do usuario gardados en contornos de nube.

Na actualidade as suites ofimáticas dominantes no mercado son, por parte do software privativo, Microsoft Office, que posúe os seus propios formatos pechados de documentos (non existe unha publicación do estándar que utilizan para a súa gravación) para cada un dos seus programas. Respecto ao software libre, está OpenOffice.org, desenvolvido por Sun Microsystems, tamén cun formato para cada programa, pero de código aberto.

Características dunha suite:

- As diferentes aplicacións dun paquete informático teñen unha interface de usuario co mesmo aspecto; deste xeito o usuario entende a suite como un todo e só necesita coñecer e aprender unha interface.
- As suites créanse de maneira que as súas aplicacións poidan intercambiar ficheiros entre elas sen ningunha dificultade e sen perda de información.
- Pódense xerar arquivos nunha aplicación que serán inseridos dentro doutra, e poden ser modificados coa aplicación que os creou mediante un acceso directo.
- Algunhas ferramentas e utilidades son compartidas por todas as aplicacións. O dicionario de revisión ortográfica é unha destas ferramentas compartidas. Se agregamos unha palabra ao dicionario dende o procesador de textos, esta palabra estará dispoñible dende calquera outra aplicación.



### **24.1.2**      *OpenOffice.*

Foi o primeiro paquete informático en ofrecer ao mercado de usuarios software aberto con varias aplicacións e ferramentas, sen ningún custo económico. Foi traducido a máis de 35 idiomas diferentes e está dispoñible para distintas plataformas, tales como Windows, Mac OS, GNU/Linux e Solaris.

A suite de OpenOffice.org está conformada por cinco módulos:

- Writer, o procesador de textos
- Calc, a folla de calculo
- Impress, paquete de presentacións
- Draw, editor de debuxos vectoriais
- Math, editor de formulas matemáticas

Na páxina web de OpenOffice.org en español podemos ver a diferenza principal cun software privativo:

A misión de OpenOffice.org: "Crear, no contorno dunha comunidade, a suite de oficina internacional —líder— que traballará en todas as plataformas principais e permitirá un acceso a toda a funcionalidade e datos por medio de API baseados en compoñentes abertos e un formato de arquivos XML".

#### **24.1.2.1**      *Historia de OpenOffice.org*

Durante a pasada década dos 90, a empresa Sun Microsystems tentou comercializar a súa serie de estacións de traballo Ultra, que eran equipos de escritorio equipados co seu sistema operativo Solaris, unha versión de Unix. Entre as dificultades que encontraron para penetrar no mercado foi que o usuario normal non podía crear ou editar documentos de oficina como follas de calculo ou presentacións.



Daquelas unha das suites de oficina dispoñibles para Unix era StarOffice, software desenvolvido pola empresa alemá StarDivision. StarOffice era unha suite ofimática madura e nunha fase avanzada de desenvolvemento, en 1999 estaba na versión 5.0, contaba con folla de calculo (StarCalc), procesador de texto (StarWriter), paquete de presentacións (StarImpress), un editor de ecuacións (StarMath) e un programa de debuxo vectorial (StarDraw). Ademais, StarOffice integrábase nun escritorio propio e incluía un navegador co cal era posible ver páxinas HTML da internet. En 1998 Sun mostrou interese en StarOffice e decidiu a adquisición de StarDivision para facer fronte ao problema da falta de programas de oficina para Solaris.

Sun nunca tivo a intención de competir co Office de Microsoft e a súa motivación era darlle un valor agregado ás súas solucións en estacións de traballo. Debido a iso a empresa ofreceu o uso de StarOffice como *freeware*, é dicir, que calquera o podía usar sen pagar unha licenza. De xeito derivado, StarDivision tamén desenvolvera StarOffice para Windows e Linux.

StarOffice tivo unha discreta aceptación entre os usuarios de Windows, pero en Linux colocouse rapidamente como a suite de oficina preferida. Como única forma de crecer e desenvolvera, decidiuse que no ano 2000 Sun Microsystems liberase o código de StarOffice baixo a licenza GPL ao grupo de desenvolvemento de OpenOffice.org.

O código liberado de StarOffice 6.0 foi usado para lanzar en 2001 OpenOffice.org 1.0 e en 2003 —despois de miles de melloras feitas polo equipo de OpenOffice.org— lanzoouse a versión 1.1.

En outubro de 2010 como consecuencia da compra de Sun Microsystems (líder do desenvolvemento de OpenOffice.org) por Oracle e a orientación que esta empresa lle daba ao proxecto aparece LibreOffice. LibreOffice creouse como unha bifurcación (unha división dun proxecto de software libre en dúas partes con orientacións diferentes) de OpenOffice e está creada e mantida por unha comunidade liderada pola fundación The Document Foundation.



### **24.1.3      *Microsoft Office.***

É sen dúbida a suite máis popular e utilizada de todas as dispoñibles, tanto en aplicacións independentes coma en conxunto. Existen diferentes versións desta suite diferenciándose principalmente nas aplicacións incluídas e a licenza de uso de certas funcionalidades: estándar, para educación, para pequenas empresas e profesionais. Os seus programas principais son moi coñecidos e considerados como estándar nalgúns contornos.

Inclúe unha serie de aplicacións que poden adquirirse de xeito independente ou co paquete e son coñecidas como:

- WORD
- EXCEL
- ACCESS
- POWERPOINT
- OUTLOOK
- FRONTPAGE
- PUBLISHER
- PROJECT

#### **Historia de Microsoft Office**

Microsoft seguiu con Office o ronsel do paquete Lisa Office System de Apple que xa en 1983 ofrecía procesador de texto e folia de cálculo entre as súas sete aplicacións, baixo un sistema operativo con ventás, escritorio e papeleira, 12 anos antes do Windows 95. O paquete de Office para Windows data de 1990. O termo foi usado inicialmente en mercadotecnia para vender un conxunto de aplicacións que previamente se vendían por separado. O principal argumento de venda era que



comprar o paquete completo resultaba máis barato que comprar cada aplicación por separado. A primeira versión de Office contiña as aplicacións Microsoft Word, Microsoft Excel e Microsoft PowerPoint. Adicionalmente, unha chamada "versión profesional" de Office incluía tamén Microsoft Access e Schedule Plus.

Co transcurso dos anos as aplicacións de Office creceron substancialmente dende un punto de vista técnico, mesmo comparten funcionalidades, tales como: corrector ortográfico común, un integrador de datos OLE e a linguaxe de scripts de Visual Basic para aplicacións. Microsoft tamén sitúa Office como unha plataforma de desenvolvemento para a liña de software para negocios.

A versión actual é Office 2010 para Windows, presentada en 2010.

#### **24.1.4      *Suites ofimáticas en web***

##### **24.1.4.1      *Feng Office***

Feng Office é unha aplicación libre de tipo Web Office, antes coñecida como OpenGoo. É un sistema completo que proporciona funcionalidades para crear, publicar, colaborar e compartir documentos.

Feng Office permite crear e traballar entre outros, sobre:

- *Documentos*: permite aloxar documentos de todo tipo e editar directamente algúns deles.
- *Listas de tarefas*: permite a creación de listas de tarefas asignadas a distintos usuarios, con opcións de notificación, categorización, etc
- *Correo electrónico*: permite centralizar a xestión das distintas contas de correo.



- *Calendario*: permite establecer reunións e unha xestión das actividades diarias.
- *Axenda*: permite realizar unha xestión de contactos.

Esta aplicación pode funcionar baixo un modelo SaaS (Software as a Service), onde os servidores do proveedor a través dun navegador nos permiten traballar coa aplicación, pero tamén é posible realizar unha instalación da aplicación nun servidor propio; neste caso os requisitos deste sistema pasan por un servidor web Apache, PHP e MySQL como base de datos.

#### *24.1.4.2      Google Docs*

Google Docs&Spreadsheets, é un programa web gratuíto que permite crear e traballar sobre uns documentos de xeito individual ou en grupo.

Google Docs componse de:

- Procesador de textos
- Follas de cálculo
- Programa de presentación sinxelo
- Editor de formularios

Entre as vantaxes de Google Docs, encóntrase o feito de que pode ser usado tanto en liña como sen conexión. Nesta modalidade sen conexión, os cambios que se introduzan nos documentos serán actualizados de forma automática en canto a conexión coa internet se restableza.



Ademais recentemente incorporouse compatibilidade entre Google Docs e os dispositivos móbiles, de tal forma que se poida non só acceder aos documentos senón tamén editalos.

#### *24.1.4.3      Office Web Apps*

Office Web Apps é a solución de Microsoft para as suites ofimáticas na web. Consiste nunha versión gratuíta baseada no conxunto de aplicacións de Microsoft Office.

Office Web Apps, componse de:

- Word Web App
- Excel Web App
- PowerPoint Web App
- OneNote Web App

Estas aplicacións permiten acceder aos documentos a través do navegador, así como compartir arquivos e traballar sobre eles de forma colaboradora.



## **24.2 PROCESADORES DE TEXTO**

### **24.2.1 *Historia dos procesadores de textos***

As ferramentas de procesamento de texto foron o primeiro tipo de aplicación que se lle deu aos primeiros ordenadores.

Inicialmente, os mecanismos de comunicación entre os programadores, que se encargaban de editar o comportamento e operacións, e as máquinas de proceso realizábase fundamentalmente mediante o uso de tarxetas perforadas, ou a través de codificacións dificultosas e que implicaban un alto investimento de tempo. Co obxecto de cubrir esta necesidade de interacción entre os programadores e a operativa das máquinas, deseñáronse e construíron as primeiras aplicacións que permiten programar as instrucións de forma máis lexible, mediante a escritura de comandos en forma de texto. Isto deu orixe aos primeiros editores de textos.

A medida que os microcomputadores se van estendendo fóra de ámbitos específicos e o uso da informática se estende a nivel xeral, a evolución dos editores de texto como ferramentas de uso xeral vai cobrando forza. Hoxe en día os editores de texto son ferramentas case de propósito xeral que permiten a redacción de documentos, xeración de informes, etc, en todos os ámbitos profesionais e persoais. Este auxo deste tipo de ferramentas provocou que moitas empresas de deseño de software apliquen esforzos na construción e deseño de novos e mellores produtos, actualmente integrados en suites ofimáticas que abranguen non só a edición de textos senón o manexo doutras operativas habituais, como a xeración de gráficas ou equipos de cálculo.

### **24.2.2 *Procesadores ou editores de textos***

Os editores ou procesadores de textos son aplicacións informáticas que se converteron nas máis usuais e utilizadas dentro do ámbito da informática profesional e de usuario. Sobre o uso das antigas máquinas de escribir, os procesadores de texto



dixitais permiten a posibilidade de escribir documentos de centos de páxinas, realizar correccións e/ou modificalos, automatizar tarefas de edición, corrección, e finalmente imprimilos.

Son programas que permiten realizar todas as operacións necesarias para editar, gardar, recuperar, modificar e imprimir un texto. Ao escribir cun procesador de textos, non hai que se preocupar do final da liña, nin tampouco do fin de páxina, xa que o programa salta automaticamente á liña seguinte ou á folla seguinte cando unha estea completa. O propio procesador delimitará o tamaño da folla, liñas por páxina, etc.

Mediante sinxelos procedementos podemos mover, borrar, subliñar ou repetir parágrafos, frases e palabras dentro dun texto. Unha vez finalizada a manipulación do documento, podemos gardalo nun soporte magnético ou óptico, imprimilo ou ambas as dúas cousas. (Cando se edita ou crea un texto, este reside na memoria interna, polo que só permanece temporalmente, perdéndose ao desconectar o equipo. Para evitar a perda, antes debe gardarse nun soporte, se desexamos conservalo).

Os procesadores teñen acceso e comunicación con outros programas: bases de datos, follas de cálculo, dicionarios, outros arquivos, etc, aínda que isto xa non é imprescindible nos ámbitos Linux ou Windows, dadas as facilidades que ofrecen para a interrelación entre programas.

Todos os procesadores de textos permiten establecer cabeceiras e pés de páxina, definir a anchura do documento, o número de caracteres por liña, definir a lonxitude das páxinas, marcar marxes e tabuladores, mover, copiar ou borrar bloques de textos, definir atributos de texto (negra, subliñado...). Tamén ofrecen a posibilidade de crear de forma sinxela táboas, gráficos, debuxos e inserir mesmo imaxes. É normal a posibilidade de ver o documento en pantalla no seu formato definitivo, é dicir, tal e como se vai imprimir. Esta mostra previa é moi interesante para comprobar o seu aspecto final sen necesidade de imprimilo; coñécese como WYSIWYG (obtense o que se ve). Un programa que non é WYSIWYG, aínda que é o máis potente de todos é Tex, escrito por Donald Knuth da Universidade de Stanford (EUA), moi utilizado polos matemáticos e científicos en xeral, e conta con moitos defensores.



Respecto á seguridade, gardan automaticamente unha copia do documento anterior; outros teñen claves de acceso (PASSWORD) que protexen o texto, permitíndolles a súa manipulación soamente aos usuarios que coñezan o contrasinal. Non obstante Microsoft WORD ten niveis de seguridade moi deficientes.

Os procesadores de texto teñen a posibilidade de dispoñer dos tipos de letras do respectivo sistema operativo, aínda que é o tipo de impresora o factor limitativo da calidade dos resultados obtidos.

Os procesadores actuais contan con programas auxiliares como os dicionarios ortográficos, de sinónimos ou bilingües. Os ortográficos serven para revisar o documento completo detectando os erros de mecanografía e faltas de ortografía. Os de sinónimos (thesaurus) permiten consultar posibles alternativas a unha palabra. Os dicionarios bilingües permiten buscar a palabra noutro idioma. Outras posibilidades ata hai pouco tempo consideradas como avanzadas son: editores de fórmulas, posibilidade de definir macros, sombreados de marcos, escritura en columnas. Outros programas interesantes son os comprobadores de estilo que comparan os textos cunha serie de regras gramaticais. Detectan erros de puntuación, maiúsculas, palabras repetidas, palabras en desuso.

É tamén moi importante a existencia dunha axuda o máis sinxela e completa posible que evite a consulta constante do manual. Algúns programas inclúen discos titores e libros de aprendizaxe con exemplos.

O procesamento de textos non é escribir, aínda que se pense que ser un bo mecanógrafo é importante para utilizar un procesador de textos, hai algunhas aptitudes da mecanografía que son contraproducentes. Seguidamente lístanse os novos hábitos a ter en conta ao pasar ao uso dun editor de textos:

- Utiliza a tecla intro (ou retorno de carro ou fin de liña) só cando debas. Os axustes de liña prodúcense automaticamente.
- Utiliza guías de tabulación e marxes, non a barra espazadora para aliar columnas



- Non subliñes. Utiliza cursivas e grosas para resaltar o texto.
- Utiliza só un espazo despois dun punto. Débense evitar os espazos dobres
- Benefíciate dos caracteres especiais. Caracteres non comúns das máquinas de escribir fan que os escritos parezan máis profesionais.
- Sistemas de tratamento de textos e documentos.

### **24.2.3      *Exemplos de procesadores de textos***

#### **24.2.3.1      Microsoft Word:**

É un dos principais procesadores de texto existentes hoxe en día. A súa popularidade débese fundamentalmente á gran difusión das plataformas de Microsoft, e á facilidade de uso tanto do propio editor como da suite ofimática na que está integrado. O do paquete Microsoft Office converteuse case nun estándar de referencia precisamente dada a elevada porcentaxe de usuarios que o instalou no seu ordenador e o utiliza.

#### **24.2.3.2      Lotus Word Pro:**

Unha das alternativas máis populares a Microsoft Word é este procesador de textos, incluído no paquete de software de escritorio de Lotus. Lotus é outra firma comercial que distribúe os seus produtos ao igual que Microsoft baixo licenza comercial, e actualmente representa unha das alternativas máis robustas e fiables do mercado.

#### **24.2.3.3      Word Perfect:**

A mediados da década dos 90 Word Perfect representaba o editor de moda, aínda que actualmente se encontre en certa decadencia a favor de ferramentas comerciais amplamente máis estendidas como os paquetes de Microsoft ou Lotus, ou ben por alternativas libres como OpenOffice. Este procesador de textos presenta un despregamento de innovadoras características que demostran o interese por parte



dos seus promotores en volver facer deste programa un produto punteiro capaz de competir no mercado con Microsoft Word.

#### 24.2.3.4 Word Pad:

Instalado por defecto en todas as versións dos sistemas operativos de Windows, poderíase considerar o "irmán pequeno" de Microsoft Word. É a opción ideal para aqueles usuarios que necesitan de forma esporádica un procesador co que dar certo estilo estético aos seus escritos, sen buscar un rematado de aparencia profesional nin excesivos adobíos ornamentais.

#### 24.2.3.5 Block de Notas:

Tamén presente por defecto en todas as instalacións do sistema de Windows, independentemente da versión, este programa móstrase como a opción ideal para usuarios austeros. Ao ser unha aplicación de posibilidades reducidas, non terán que familiarizarse cun complexo contorno cheo de funcións que nunca van utilizar.

#### 24.2.3.6 OpenOffice Writer:

É o equivalente a Word en software libre. As súas posibilidades e forma de utilizalo son enormemente parecidas, coa vantaxe de que ao ser software libre podemos descargalo gratuitamente da internet e actualizalo tantas veces queiramos sen pagar dereitos de autor.

#### 24.2.3.7 LibreOffice Writer:

É o procesador de textos da suite ofimática LibreOffice e ofrece características moi similares ao de OpenOffice, posto que comparten unha base común. Pode protexer documentos con contrasinal, gardar versións do mesmo documento, inserir imaxes, obxectos OLE, admite sinaturas dixitais, símbolos, fórmulas, táboas de cálculo, gráficos, hiperligazóns, marcadores, formularios, etc.

Writer permite exportar arquivos de texto aos formatos PDF e HTML sen software adicional, o que permite que poida ser utilizado como un editor WYSIWYG para crear e editar páxinas web.







### **24.3 FOLLAS DE CÁLCULO**

Co avance da computación e o deseño de novos ordenadores máis potentes e compactos, as tendencias dos grandes sistemas de procesamento foron migrando cara á adquisición microordenadores con alta capacidade de proceso. O proceso de adaptación dos sistemas de planificación utilizados nos antigos sistemas de proceso deron paso a novas ferramentas de cálculo e xestión máis dinámicos, que integrasen ferramentas sinxelas de cálculo e que se asemellasen ás táboas de cálculos que habitualmente se utilizaban na xestión a base de lapis e papel.

Como solución a esta tendencia, nacen as follas de cálculo, tamén chamadas follas electrónicas de cálculo. As follas de cálculo son sistemas informáticos, habitualmente integrados dentro dalgunha suite ofimática, que representan táboas con celas nun formato que resulta cómodo para realizar operacións contables e xeración de informes ou cálculos estatísticos. Combinan ademais ferramentas de cálculo e diferentes funcionalidades engadidas. Na actualidade, as follas de cálculo conforman unha ferramenta de grande importancia no mundo empresarial.

En 1961 desvelouse o concepto dunha folla de cálculo electrónica no artigo "Budgeting Models and System Simulation" de Richard Mattessich. Pardo e Landau merecen parte do crédito deste tipo de programas, e de feito tentaron patentar (patente nos EUA número 4398249) algúns dos algoritmos en 1970. A patente non foi concedida pola oficina de patentes por ser unha invención puramente matemática. Pardo e Landau gañaron un caso nos tribunais establecendo que "algo non deixa de ser patentable só porque o punto da novidade é un algoritmo". Este caso axudou ao comezo das patentes de software. A primeira folla de cálculo comercializada masivamente (Visicalc, posteriormente comprada por Lotus) naceu en 1977 dunha idea dun estudante, Dan Bricklin, de Administración de Empresas da Universidade de Harvard (EUA).



Os cálculos empresariais manexan unhas cantidades de datos que adoitan variar dependendo dos supostos, polo que ata a invención da folla electrónica se rexeitara aplicar a informática a ese ámbito de traballo, onde cada caso orixinaría un programa distinto.

A folla de cálculo preséntase como unha táboa ou matriz de dúas dimensións (actualmente existen de tres) que consta dun número de filas e columnas variable segundo o modelo que sexa. (P. ex. en LOTUS 123, unha das primeiras no contorno IBM PC, tiña 8192 filas e 256 columnas, en total máis de 2000000 de celas). Cos datos pódense representar variados tipos de gráficos, de grande utilidade en ambientes empresariais. As filas son os datos horizontais e as columnas os verticais. O lugar onde se produce a intersección dunha fila e unha columna denomínase cela. As columnas habitualmente noméanse con letras (A, B... AA, AB...) e as filas dende o 1 en diante.

Unha folla de calculo é un programa que permite manipular datos numéricos e alfanuméricos dispostos en forma de táboas (que é a unión de filas e columnas). Habitualmente é posible realizar cálculos complexos con fórmulas e funcións e debuxar distintos tipos de gráficos.

#### **24.3.1      *Microsoft Excel***

Microsoft Excel é unha aplicación integrada dentro da suite ofimática Microsoft Office, distribuída baixo licenza comercial en todas as súas versións. Está fundamentalmente orientado ao manexo de follas de cálculo e utilízase normalmente en tarefas financeiras e contables.

Microsoft Excel ofrece unha interface de usuario axustada ás principais características das follas de cálculo xerais. O programa mostra as celas organizadas en filas e columnas. Cada cela contén datos ou fórmulas con referencias relativas, absolutas ou mixtas a outras celas.

Como un engadido inicial, Microsoft Excel destaca por ser a primeira folla de cálculo que permite ao usuario definir a aparencia do documento final, relacionado



coas fontes de letra, atributos dos caracteres, celas, etc. Tamén integra facetas como o recálculo intelixente de celas, consistente en que as celas nas que o valor depende doutras que se modifican, se actualizan ao instante.

Ademais destas facetas, Microsoft Excel integra un módulo gráfico que permite o deseño dunha gran cantidade de gráficas representativas dos valores cos que se traballa nas táboas, e integra ademais un motor de cálculo propio con capacidade para a resolución de problemas de optimización no ámbito da programación lineal.

Microsoft Excel inclúe tamén Visual Basic para Aplicacións (VBA), que é unha linguaxe de programación baseada en Visual Basic, e que engade a capacidade para a automatización de tarefas. Ademais permite ao usuario definir funcións para o seu uso nas follas de traballo. Nas versións máis recentes do paquete Office, Microsoft Excel conta xa cun contorno completo integrado de edición de VBA para o desenvolvemento de funcións e automatización de tarefas. Así mesmo, a xeración de macros pode producir código VBA para repetir as accións do usuario, o que permite a automatización de simples tarefas. VBA permite a creación de formularios e controis na folla de traballo para comunicarse co usuario. Admite o uso da linguaxe (pero non a creación) das DLL de ActiveX (COM); versións posteriores engadiron soporte para os módulos de clase permitindo o uso de técnicas de programación básicas orientadas a obxectos.

Como contrapartida á funcionalidade proporcionada por VBA e a automatización de tarefas, encóntrase a vulnerabilidade provocada polo ataque de virus en forma de macro.

### **24.3.2**      *Openoffice.orgCalc*

OpenOffice.org Calc é un software de creación e xestión de follas de cálculo integrado na suite ofimática OpenOffice.org. Trátase dunha folla de cálculo Open Source e software libre compatible con Microsoft Excel. OpenOffice.org Calc é perfectamente compatible con practicamente todas as plataformas existentes,



abranguendo Mac OS X, Windows, GNU/Linux, FreeBSD e Solaris, e encóntrase dispoñible baixo licenza LGPL.

OpenOffice.org Calc é unha folla de cálculo similar a Microsoft Excel, cun conxunto de características equivalentes en canto funcionalidade. Non obstante o seu tamaño é menor e proporciona un número de características non presentes en Excel, incluíndo un sistema que automaticamente define series para representar graficamente, baseado na disposición dos datos do usuario. Calc tamén é capaz de exportar follas de cálculo como arquivos PDF, conta con filtros e ademais pode realizar agrupacións en táboas dinámicas.

En canto á automatización de tarefas e programación de macros, Calc non é compatible co modelo de obxectos de Excel, e isto supón unha fortaleza fronte á gran cantidade de virus baseados en macros.



## **24.4 BASES DE DATOS**

Defínese unha base de datos como "unha serie de datos organizados e relacionados entre si, os cales son reunidos e explotados polos sistemas de información dunha empresa ou negocio en particular".

É dicir, unha base de datos é un conxunto de datos que pertencen ao mesmo contexto almacenados sistematicamente para o seu posterior uso, un "almacén" que nos permite gardar grandes cantidades de información de forma organizada para que despois poidamos encontrar e utilizar doadamente.

O termo bases de datos foi escoitado por primeira vez en 1963, nun simposio celebrado en California, USA. Unha base de datos pódese definir como un conxunto de información relacionada que se encontra agrupada ou estruturada.

Dende o punto de vista informático, a base de datos é un sistema formado por un conxunto de datos almacenados en discos que permiten o acceso directo a eles e un conxunto de programas que manipulen ese conxunto de datos.

Cada base de datos componse dunha ou máis táboas que garda un conxunto de datos. Cada táboa ten unha ou máis columnas e filas. As columnas gardan unha parte da información sobre cada elemento que queiramos gardar na táboa, cada fila da táboa conforma un rexistro.

### **24.4.1 Características**

Entre as principais características dos sistemas de base de datos podemos mencionar:

- Independencia lóxica e física dos datos.
- Redundancia mínima.



- Acceso concorrente por parte de múltiples usuarios.
- Integridade dos datos.
- Consultas complexas optimizadas.
- Seguridade de acceso e auditoría.
- Respaldo e recuperación.
- Acceso a través de linguaxes de programación estándar.

#### **24.4.2      *Sistema de xestión de base de datos (SXBD)***

Os sistemas de xestión de base de datos (en inglés DataBase Management System) son un tipo de software moi específico, dedicado a servir de interface entre a base de datos, o usuario e as aplicacións que a utilizan. Componse dunha linguaxe de definición de datos, dunha linguaxe de manipulación de datos e dunha linguaxe de consulta.

#### **24.4.3      *Vantaxes das bases de datos***

##### **24.4.3.1      *Control sobre a redundancia de datos:***

Os sistemas de ficheiros almacenan varias copias dos mesmos datos en ficheiros distintos. Isto fai que se desperdicie espazo de almacenamento, ademais de provocar a falta de consistencia de datos.

Nos sistemas de bases de datos todos estes ficheiros están integrados, polo que non se almacenan varias copias dos mesmos datos. Non obstante, nunha base de datos non se pode eliminar a redundancia completamente, xa que en ocasións é necesaria para modelar as relacións entre os datos.

##### **24.4.3.2      *Consistencia de datos:***

Eliminando ou controlando as redundancias de datos redúcese en boa medida o risco de que haxa inconsistencias. Se un dato está almacenado unha única vez, calquera actualización se debe realizar só unha vez, e está dispoñible para todos os



usuarios inmediatamente. Se un dato está duplicado e o sistema coñece esta redundancia, o propio sistema pode encargarse de garantir que todas as copias se manteñan consistentes.

#### *24.4.3.3      Compartición de datos:*

Nos sistemas de ficheiros, os ficheiros pertencen ás persoas ou aos departamentos que os utilizan. Pero nos sistemas de bases de datos, a base de datos pertence á empresa e pode ser compartida por todos os usuarios que estean autorizados.

#### *24.4.3.4      Mantemento de estándares:*

Grazas á integración é máis doado respectar os estándares necesarios, tanto os establecidos a nivel da empresa coma os nacionais e internacionais. Estes estándares poden establecerse sobre o formato dos datos para facilitar o seu intercambio, poden ser estándares de documentación, procedementos de actualización e tamén regras de acceso.

#### *24.4.3.5      Mellora na integridade de datos:*

A integridade da base de datos refírese á validez e á consistencia dos datos almacenados. Normalmente, a integridade exprésase mediante restricións ou regras que non se poden violar. Estas restricións pódense aplicar tanto aos datos, como ás súas relacións, e é o SXBD quen se debe encargar de mantelas.

#### *24.4.3.6      Mellora na seguridade:*

A seguridade da base de datos é a protección da base de datos fronte a usuarios non autorizados. Sen unhas boas medidas de seguridade, a integración de datos nos sistemas de bases de datos fai que estes sexan máis vulnerables que nos sistemas de ficheiros.

#### *24.4.3.7      Mellora na accesibilidade aos datos:*

Moitos SXBD proporcionan linguaxes de consultas ou xeradores de informes que lle permiten ao usuario facer calquera tipo de consulta sobre os datos, sen que sexa necesario que un programador escriba unha aplicación que realice tal tarefa.



#### 24.4.3.8 Mellora na produtividade:

O SXBD proporciona moitas das funcións estándar que o programador necesita escribir nun sistema de ficheiros. Basicamente, o SXBD proporciona todas as rutinas de manexo de ficheiros típicas dos programas de aplicación.

O feito de dispoñer destas funcións permítelle ao programador centrarse mellor na función específica requirida polos usuarios, sen ter que se preocupar dos detalles de implantación de baixo nivel.

#### 24.4.3.9 Mellora no mantemento:

Nos sistemas de ficheiros, as descrições dos datos está inmersas nos programas de aplicación que os manexan.

Isto fai que os programas sexan dependentes dos datos, de modo que un cambio na súa estrutura, ou un cambio no modo en que se almacena no disco, require cambios importantes nos programas nos que os datos se ven afectados.

Non obstante, os SXBD separan as descrições dos datos das aplicacións. Isto é o que se coñece como independencia de datos, grazas á cal se simplifica o mantemento das aplicacións que acceden á base de datos.

#### 24.4.3.10 Aumento da concorrencia:

Nalgúns sistemas de ficheiros, se hai varios usuarios que poden acceder simultaneamente a un mesmo ficheiro, é posible que o acceso interfira entre eles de modo que se perda información ou se perda a integridade. A maioría dos SXBD xestionan o acceso concorrente á base de datos e garanten que non acontezan problemas deste tipo.

#### 24.4.3.11 Mellora nos servizos de copias de seguridade:

Moitos sistemas de ficheiros deixan que sexa o usuario quen proporcione as medidas necesarias para protexer os datos ante fallos no sistema ou nas aplicacións. Os usuarios teñen que facer copias de seguridade cada día, e se se produce algún fallo, utilizar estas copias para restauralos.



Neste caso, todo o traballo realizado sobre os datos dende que se fixo a última copia de seguridade se perde e cómpre volver realizalo. Non obstante, os SXBD actuais funcionan de modo que se minimiza a cantidade de traballo perdido cando se produce un fallo.

#### ***24.4.4 Desvantaxes das bases de datos***

##### ***24.4.4.1 Complexidade:***

Os SXBD son conxuntos de programas que poden chegar a ser complexos cunha gran funcionalidade. É preciso comprender moi ben esta funcionalidade para poder realizar un bo uso deles.

##### ***24.4.4.2 Custo do equipamento adicional:***

Tanto o SXBD, como a propia base de datos, poden facer que sexa necesario adquirir máis espazo de almacenamento. Ademais, para alcanzar as prestacións desexadas, é posible que sexa necesario adquirir unha máquina máis grande ou unha máquina que se dedique só ao SXBD. Todo isto fará que a implantación dun sistema de bases de datos sexa máis cara.

##### ***24.4.4.3 Vulnerable aos fallos:***

O feito de que todo estea centralizado no SXBD fai que o sistema sexa máis vulnerable ante os fallos que se poidan producir. É por iso que se deben ter copias de seguridade (*Backup*).

#### ***24.4.5 Ferramentas para a xestión de bases de datos presentes en suites ofimáticas***

##### ***24.4.5.1 LibreOffice Base***

LibreOffice Base é un programa de base de datos. LibreOffice Base permite a creación e manexo de bases de datos, elaboración de formularios e informes que proporcionan aos usuarios finais un acceso doado aos datos. Ao igual que Microsoft Access, é capaz de traballar como un *frontend* para diversos sistemas de bases de



datos tales como o de Access (JET), fonte de datos ODBC e MySQL/PostgreSQL. Base forma parte da suite ofimática dende a versión 2.0 de OpenOffice (da cal se deriva LibreOffice).

#### *24.4.5.2 Microsoft Access*

Microsoft Access é un sistema de xestión de bases de datos relacionais para os sistemas operativos Microsoft Windows, desenvolvido por Microsoft e orientado a ser usado nun contorno persoal ou en pequenas organizacións. É un compoñente da suite ofimática Microsoft Office. Permite crear ficheiros de bases de datos relacionais que poden ser doadamente xestionadas por unha interface gráfica simple. Ademais, estas bases de datos poden ser consultadas por outros programas. Este programa permite manipular os datos en forma de táboas (formadas por filas e columnas), crear relacións entre táboas, consultas, formularios para introducir datos e informes para presentar a información.



## **24.5 PRESENTACIÓN**

Un programa de presentación é un paquete de software usado para mostrar información, normalmente mediante unha serie de diapositivas.

Tipicamente inclúe tres funcións principais: un editor que permite inserir un texto e darlle formato, un método para inserir e manipular imaxes e gráficos e un sistema para mostrar o contido en forma continua.

### **24.5.1      *Tipos de presentación***

Hai moitos tipos de presentacións, para educación, ou para comunicar noticias en xeral. Os programas de presentación poden servir de axuda ou substituír as formas tradicionais de dar unha presentación, como por exemplo panfletos, resumos en papel, encerados, diapositivas ou transparencias.

Un programa de presentación permite colocar texto, gráficos, películas e outros obxectos en páxinas individuais ou "diapositivas". O termo "diapositiva" é unha referencia ao proxector de diapositivas, un dispositivo que quedou obsoleto para estes fins dende a aparición dos programas de presentación.

As diapositivas pódense imprimir en transparencias e móstrase mediante un proxector de transparencias, ou visualízase directamente na pantalla do ordenador (ou nunha pantalla normal usando un proxector de vídeo) baixo o control da persoa que dá a presentación. A transición dunha diapositiva a outra pode ser animada de varias formas, e pódese animar tamén a aparición dos elementos individuais en cada diapositiva.

### **24.5.2      *Ferramentas de presentación***

#### **24.5.2.1      *Microsoft PowerPoint***

Microsoft PowerPoint é un software incluído dentro da suite ofimática de Microsoft Office orientado á realización de presentacións dixitais que permite integrar diferentes



medios multimedia na elaboración. Está dispoñible para plataformas Windows e MacOS e é utilizado amplamente en distintos campos como o ensino, negocios, etc.

O obxectivo principal é a realización de presentacións esquemáticas que combinan texto, imaxes, animacións, sons e mesmo vídeos. O obxecto é realizar contidos elegantes e doadamente comprensibles, e pódese facer uso das funcionalidades do programa para a completa edición de todos os elementos que se utilizan, aplicando diferentes deseños de fontes e tipos de letra, equipos predefinidos, animacións, e incorporación de imaxes predeseñadas ou ben importadas dende bibliotecas externas. Este tipo de presentacións adoita ser moi rechamante e moito máis práctico que os de Microsoft Word.

A facilidade de uso deste software, así como a popularidade das plataformas e familiaridade dos produtos de Microsoft, provocou que PowerPoint sexa un dos programas de presentación máis estendidos actualmente. Vén integrado no paquete Microsoft Office como un elemento máis, que pode aproveitar as vantaxes que lle ofrecen os demais compoñentes do equipo para obter un resultado óptimo.

#### 24.5.2.2      *OpenOffice Impress*

Os usuarios de OpenOffice.org Impress tamén poden instalar a Open Clip Art Library (Biblioteca Aberta de Clip Art), que agrega unha enorme galería de bandeiras, logos, iconas, estandartes e pancartas para presentacións xerais e proxectos de debuxo. Algunhas distribucións Linux, como Debian, Mandriva Linux e Ubuntu proporcionaron un paquete chamado openclipart listo para usar e doado de baixar dende os seus repositorios, instalando unha galería de imaxes e sons para o OpenOffice.org.

OpenOffice.org Impress é un programa de presentación similar a Microsoft PowerPoint que vén integrado na suite de OpenOffice.org, desenvolvida por Sun Microsystems. Entre as principais vantaxes engadidas que veñen integradas neste software, destaca a posibilidade de exportar as presentacións a formatos de arquivo SWF, utilizado por Adobe Flash. Isto permite certa portabilidade permitindo que estas



presentacións poidan ser executadas en calquera ordenador que dispoña dunha versión do reprodutor de Adobe Flash Player instalado. Como no resto de ferramentas do paquete de OpenOffice, Impress permite tamén a creación de arquivos PDF de forma directa. Como punto negativo, Impress non dispón do repertorio de deseños predefinidos cos que contan outras ferramentas como o caso do seu competidor directo de Microsoft, PowerPoint. De todos os xeitos, isto non supón un grande inconveniente dado que poden obterse sen dificultade modelos elaborados por terceiros a través da internet, proporcionados pola crecente comunidade de usuarios.

#### *24.5.2.3 LibreOffice Impress*

LibreOffice Impress, derivado de OpenOffice Impress é outro dos programas orientados ao desenvolvemento de presentacións multimedia baseadas en transparencias ou diapositivas. O formato nativo das presentacións é ODP, pero tamén ten a capacidade de ler e escribir no formato de arquivos de Microsoft PowerPoint (ppt), así como a capacidade de exportar as presentacións a arquivos PDF. Pode tamén exportar as presentacións ao formato SWF, permitindo que se reproduzan en calquera computadora cun reprodutor de Flash instalado.

### **24.6 BIBLIOGRAFÍA**

- 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. EnglewoodsCliffs:Prentice Hall. Laudon, K.C.; Laudon, J.P. (2002).



- Administración de los Sistemas de Información. Prentice-Hall. Laudon, K.C. e 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

Colexiado do CPEIG



# **25. ARQUITECTURA DAS REDES INTRANET E INTERNET: CONCEPTO, ESTRUTURA E CARACTERÍSTICAS. A SÚA IMPLANTACIÓN NAS ORGANIZACIÓNS.**



## **TEMA 25. ARQUITECTURA DAS REDES INTRANET E INTERNET: CONCEPTO, ESTRUCTURA E CARACTERÍSTICAS. A SÚA IMPLANTACIÓN NAS ORGANIZACIÓNS.**

### **25.1. INTRODUCCIÓN E CONCEPTOS**

### **25.2. INTERNET**

### **25.3. INTRANET/EXTRANET**

### **25.4. IMPLANTACIÓN DE REDES EN ORGANIZACIÓNS**

### **25.5. ESQUEMA**

### **25.6. REFERENCIAS**

### **25.1. INTRODUCCIÓN E CONCEPTOS**

Unha rede son dous ou máis nodos comunicados entre si. A partir de aí, a rede pode aumentarse en calquera número de nodos e conectarse a outras redes. **Internet** é unha rede de alcance mundial que conecta as diferentes redes físicas dun xeito descentralizado como unha rede lóxica única.

No mundo da informática un nodo pode ser calquera compoñente dunha rede, dende dispositivos de interconexión a equipos ou estacións de traballo, ou calquera outro tipo de cliente como equipos portátiles e dispositivos móbiles.

Por debaixo destas redes ademais teremos diferentes tipos de redes físicas, que tomarán diferentes medios e tecnoloxías. Internet proporcionará un mecanismo de comunicación común baseado na familia de protocolos TCP/IP, de maneira que calquera destas redes que implemente ou acepte esta familia de protocolos poderá comunicarse coas demais.

De entre todos os servizos que proporcionar Internet o buque insignia é o World Wide Web (WWW, ou a Web) o conxunto de protocolos que permite a



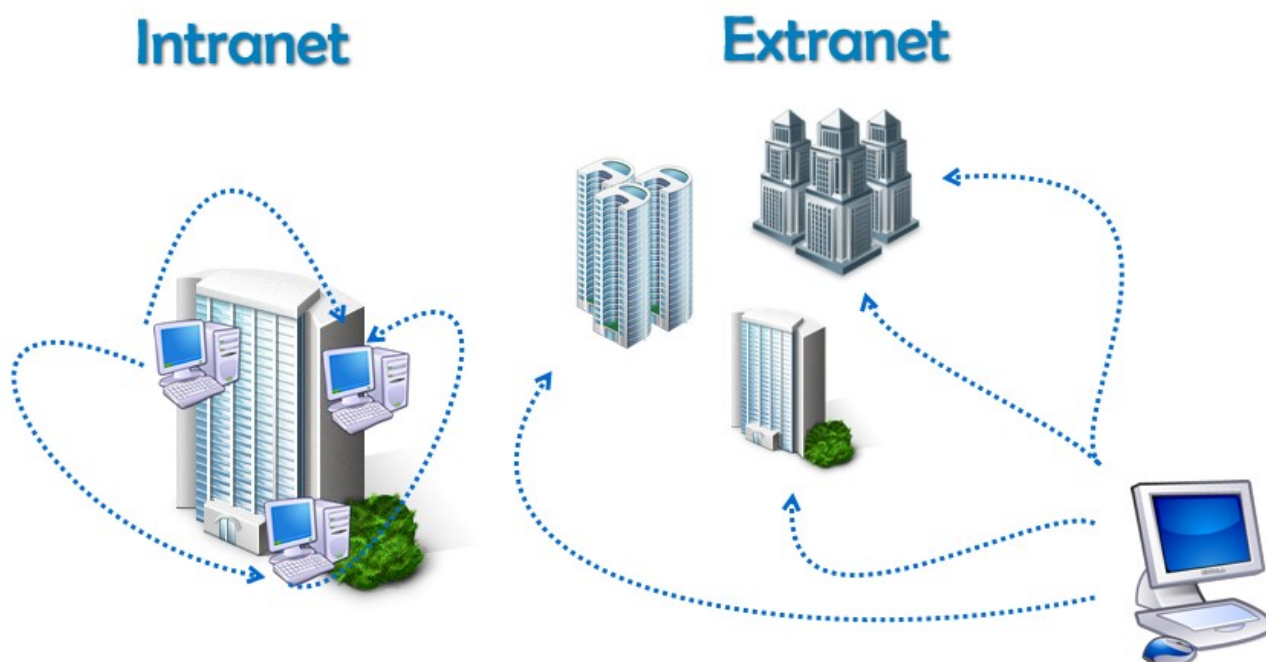
consulta de arquivos de hipertexto ou páxinas web emprazados en diferentes sitios de aloxamento ou sitios web.

Unha **Intranet** é unha rede interna a unha organización ou institución, que ten por obxecto proporcionar un conxunto de servizos accesibles exclusivamente dende a rede local ou un conxunto de redes illadas do exterior a través de Internet.

A idea principal dunha Intranet é que os seus servizos sexan só accesibles polos usuarios da organización ou institución, dun xeito privado. Estes servizos poden incluír servidores web, servidores de correo electrónico, sistemas de xestión de arquivos, contidos e utilidades de comunicación ou mensaxería.

Estendendo este concepto a Internet, cando os servizos están dispoñibles cara fóra, pero só para os usuarios da organización ou institución estarase falando dunha **Extranet**. No caso da Extranet establécese un mecanismo de seguridade ou autenticación dos usuarios para garantir que pertencen á organización ou institución. En consecuencia unha Extranet non será nin unha Intranet nin un sitio de Internet senón a publicación dos servizos dunha Intranet a través da Internet mediante un sistema de autenticación dos usuarios da organización ou institución.





***Figura 1: Intranet e Extranet***

## **25.2. INTERNET**

### **25.2.1. CARACTERÍSTICAS BÁSICAS**

Internet ten as súas orixes a finais da década dos 60, sendo unha evolución da rede experimental ARPANET (Rede da Axencia de proxectos de investigación avanzada), desenvolvida polo departamento de Defensa dos EUA.

A idea orixinal era dispoñer dunha rede na que en caso de acontecer danos ou a desaparición dalgún nodo ou punto da mesma a rede permanecera activa entre os nodos ou elementos restantes, garantindo así a supervivencia da información e o funcionamento do medio de comunicación. A partir deste concepto pode entenderse o funcionamento distribuído e completamente descentralizado que posúe o sistema actualmente, de xeito que cada nodo individual ten a mesma importancia e



peso no conxunto á hora de dar servizo ou comunicarse cos demais.

Posteriormente desenvolveuse sobre a rede un software básico de control da transmisión de información que terminaría por dar lugar á **familia de protocolos TCP/IP**. Esta familia de protocolos representa un conxunto de normas e estándares que definen o mecanismo de comunicación entre os diferentes nodos da rede. Calquera rede física que implemente ou dea soporte a este conxunto de protocolos poderá comunicarse con outras redes que tamén o fagan. A partir dun destes protocolos, podemos especificar outro dos factores fundamentais que explican o funcionamento desta rede o concepto de **Enderezo IP** (Protocolo de Internet). Este enderezo representa o enderezo ou nome de cada nodo da rede, sendo un identificador único para cada un deles. Os enderezos IP compóñense de catro cifras numéricas separadas por puntos que toman valores entre 0 e 255. Por exemplo: 192.168.1.1. Por mor de aumentar o rango de enderezos deseñouse o **IPv6** que pasa a valores de 128 bits, con oito grupos de catro díxitos hexadecimais, por exemplo: FEDC:BA98:7654:3210:FEDC:BA98:7654:3210.

Enderezo IP	Significado
::	Ausencia de enderezo
0:0:0:0:0:0:0:0	Ausencia de enderezo
::1	Loopback
::1.2.3.4	Compatible con IPv4
::ffff:0:0	Enderezo Ipv4 mapeado
Ff00::	Multicast
FF01:0:0:0:0:0:0:0:101	Multicast

**Táboa 1: Exemplos de enderezos Ipv6.**



Como este tipo de identificación pode resultar difícil de lembrar emprégase en conxunción o Sistema de Nomes de Dominio (**DNS**). Neste sistema diferentes nodos da rede fan as funcións de tradutores entre enderezos IP e nomes de Dominio, sendo estes varias palabras separadas por puntos, por exemplo [www.xunta.es](http://www.xunta.es), indicando en última instancia a zona ou tipo de organización á que pertence o sitio, neste caso España, co acrónimo 'es', a continuación a organización, institución ou mnemotécnico, neste caso 'xunta', e por último o usuario ou protocolo, neste caso 'www'.

Por último outro concepto fundamental é o de **clientes e servidores**. O obxectivo da rede será dobre comunicar e dar servizos. Neste caso podemos distinguir tres tipos de nodos:

1. **Servidores**. Proven de servizos á rede, tales como contidos web, correo electrónico, vídeo, xestión das comunicacións, seguridade, etc...
2. **Clientes**. Nodos que representan o equipo de traballo dun usuario final, o cal fai uso dun dos servizos da rede que lle proporciona un servidor.
3. **Elementos de interconexión**. Son nodos específicos de comunicación, encárganse de xestionar as comunicacións, retransmitir e dirixir as mensaxes.

O modelo de Internet pode aplicarse sobre redes máis pequenas, de menos equipos e unha extensión menor. A idea de Internet é unha rede global, con servizos e comunicación a escala mundial. Abstraendo funcionamento e protocolos, poden facerse rede máis pequenas cun servizo reducido ao seu ámbito. A partir disto temos a clasificación habitual das redes, que inclúe:

1. **Redes de área local**. (En inglés *Local Area Network* ou LAN). Interconexión de varios computadores e elementos de interconexión limitada fisicamente a un edificio ou contorno de arredor de 200

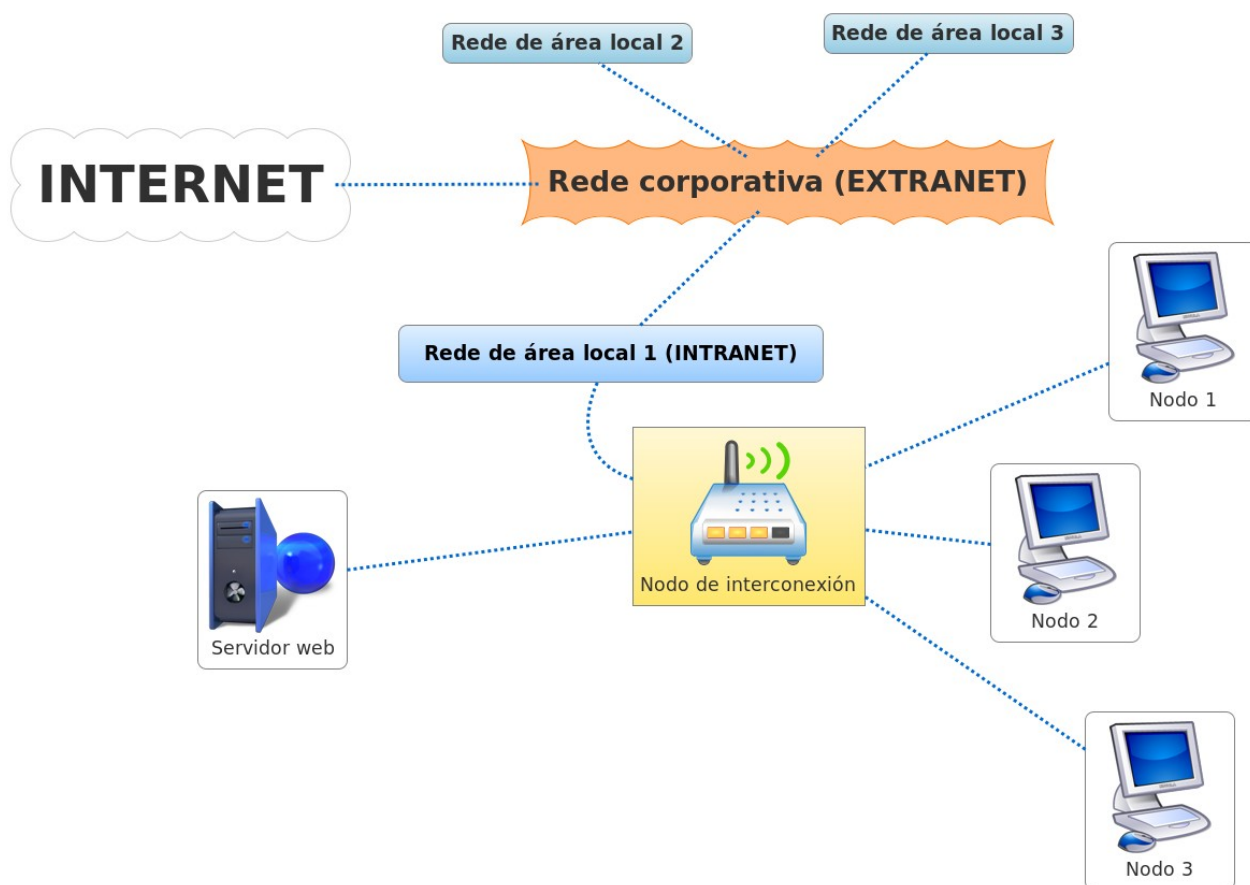


metros - 1 Quilómetro. Exemplos destas redes serían as redes corporativas ou institucionais dentro dun mesmo edificio, como pode ser a rede interna dunha Consellaría, e por norma xeral inclúen ademais servizos de Intranet.

2. **Redes de área metropolitana.** (En inglés *Metropolitan Area Network* ou MAN). Interconexión de varios computadores e elementos de interconexión nun área extensa, como pode ser unha cidade, provincia ou comunidade autónoma. Exemplo deste tipo de redes sería a rede corporativa da Xunta de Galicia. Por norma xeral este tipo de redes incorporan servizos de Intranet/Extranet.
3. **Redes de área ampla.** (En inglés *Wide Area Network* ou WAN). Interconexión de varios computadores e elementos de interconexión en distancias de 100-1000 Quilómetros. Exemplo deste tipo de redes sería a propia rede Internet.

Na seguinte figura, podemos ver un exemplo de rede de área local, con algúns elementos básicos. Diferentes equipos de traballo, conectados por nodos de interconexión e con algún servizo como o proporcionado polo servidor web. Esta pequena rede pode atoparse integrada nunha rede de maior alcance con servizos de Intranet e como medio de comunicación co resto do mundo a través de Internet.





**Figura 2: Exemplo de rede de área local conectada a unha rede corporativa e a Internet.**

### 25.2.2. TIPOS DE CONEXIÓN A INTERNET

Para conectar outra rede ou equipo cliente, xa sexa un ordenador de sobremesa, portátil, teléfono móbil, PDA, etc..., á rede Internet o primeiro paso será dispoñer dun **provedor de acceso ou ISP** (en inglés *Internet Service Provider*, provedor de servizos de Internet). Trátase de empresas que proporcionan e xestionan a conexión á rede aos seus clientes, empregando diferentes tecnoloxías. Por normal xeral os ISP proporcionan un hardware de conexión á rede específico e pode que un software para xestionalo.

Entre as tecnoloxías de conexión máis empregadas hoxe en día dispoñemos



de:

1. **RTC.** A rede telefónica conmutada, que emprega a mesma rede que os teléfonos fixos en Galicia. Neste caso se trata dun soporte analóxico polo que para enviar datos dixitais haberá que mudalos empregando un dispositivo denominado **Módem** (modulador - demodulador), ou variantes máis avanzadas con maiores características como a enrutación ao estilo dos **Módem-Routers**. Deste xeito un usuario que queira acceder a Internet precisará dispor dunha liña telefónica e un Módem ou Módem-Router. Estes dispositivos poden ser internos, como acontece normalmente nos dispositivos portátiles ou externos. Neste último caso a conexión co equipo de traballo realizarase conectando o dispositivo por un cable/porto (P.ex.: USB) ou con conectividade sen fíos. Actualmente esta tecnoloxía atópase nun estado practicamente obsoleto, debido a que non pode transmitir datos e voz á vez e que a súa velocidade máxima é moi baixa (arredor de 56 Kbps).
2. **ADSL.** A liña de aboado dixital asimétrica convirte a liña telefónica nunha liña de alta velocidade debido a que aproveita toda a potencia dos fíos establecendo tres canles independentes:
  - a) Canle de alta velocidade para transmitir datos.
  - b) Canle de alta velocidade para recibir datos.
  - c) Canle de alta velocidade para voz.

Deste xeito permítese que a través da mesma liña se envíen datos e voz á vez. O concepto de asimétrica ven de que as velocidades de subida e baixada de datos son diferentes sendo máis altas as velocidades de baixada, nunha interpretación de que as necesidades dos usuarios van neste senso. O hardware empregado neste caso serán Módem-Routers, proporcionados por un ISP. As velocidades de descarga acadadas son moi superiores ao RTC, indo de 512 Kbps a un máximo teórico para VSL (unha



evolución da ADSL de moi alta taxa de transferencia) de 55 Mbps, se ben os provedores en Galicia proporcionan bastante menos.

3. **Sen fíos.** Aínda que en orixe as redes sen fíos foron deseñadas para redes de área local actualmente tamén se empregan para posibilitar accesos a Internet. Baseados no conxunto de estándares Wi-Fi (en inglés *Wireless Fidelity*) chegan a acadar velocidades de arredor de 54 Mbps chegando ao máximo teórico de 600Mbps. O hardware necesario neste caso será un Router Wi-Fi sen fíos que faga as funcións de punto de acceso (en inglés *hotspot*) e no equipo de traballo unha antena receptora integrada nunha tarxeta de rede (interna ou externa).
4. **Cable.** Redes baseadas en tecnoloxías de fibra óptica o que implica que precisa unha liña de transmisión desta tecnoloxía. O hardware empregado é similar ao da ADSL, pero neste caso será un Cable-Módem o encargado de xestionar a comunicación, aínda que o termo Cable-Router sería máis axeitado neste caso, pois a xestión é máis avanzada ca no caso do Módem. As velocidades son moi elevadas, esta tecnoloxía tamén resulta moi cara en contrapartida, chegando a 10 Gbps de máximo teórico con 1 Gbps habituais. De cara ao usuario e en Galicia, o ancho de banda é moito menor, os provedores máis habituais acostuman a proporcionar velocidades similares ás da ADSL.
5. **Satélite.** A conexión vía satélite emprégase en emprazamentos con pouca infraestrutura onde non é posible aplicar as tecnoloxías anteriores, como ADSL ou Cable. En Galicia recórrase a este tipo de tecnoloxías en zonas do contorno rural ou zonas de alta montaña. Esta tecnoloxía ten un custe moi alto, pero presenta unha ampla cobertura. O hardware necesario require a instalación dunha antena



parabólica e na oferta habitual dos ISP proporcionan 2 Mbps de subida e baixada.

6. **Módem Móbil.** As últimas tecnoloxías desenvolvidas para teléfonos móbiles como GSM, GPRS, ou UTMS/3G permiten que os operadores ofrezan aos usuarios servizos de Internet ben directamente dende o **dispositivo móbil** ou ben conectando outro equipo de traballo á rede a través do mesmo. Empregan un protocolo específico denominado WAP (en inglés *Wireless Application Protocol*) e as velocidades de conexión varían dependendo da tecnoloxía de 56 Kbps a 2 Mbps coas tecnoloxías de última xeración. O hardware básico é un teléfono móbil que soporte estas tecnoloxías, podendo precisar algún elemento de conexión extra para conectalo con outros equipos de traballo.
7. **PLC.** (Do inglés *Power Line Communication*) Esta tecnoloxía ofrece conexión a Internet a través da rede eléctrica. Como a ADSL esta tecnoloxía aproveita unha infraestrutura de cableado xa existente para ampliar os canais empregando medias e altas frecuencias. Require hardware específico, os denominados **Módem PLC**. Acada velocidades de ata 134 Mbps, e a pesar de que o ancho de banda é mesmo superior ao da ADSL e a infraestrutura de cableado eléctrico pode ser mesmo superior que o telefónico, en Galicia o uso desta tecnoloxía está moi pouco estendido.

### **25.2.3. SERVIZOS DE INTERNET**

O fin último de acceder a Internet ou a outra rede é facer uso dos **servizos** que se atopan nela, e que veremos a continuación:

#### **1. WWW**



No caso de Internet o servizo máis empregado é a Rede global mundial ou **WWW** (siglas en inglés de *World Wide Web*), trátase dun sistema de publicación e intercambio de información distribuído que relaciona uns contidos con outros a través de ligazóns. Este sistema estendeuse rapidamente grazas á súa facilidade de uso.

Neste contexto xorde o **Hiperligazóns**, que ven sendo un texto ou outro obxecto que contén unha ligazón premendo nas cal se accede a outra información emprazada noutra zona do documento ou noutro documento distinto. Esta funcionalidade permite relacionar uns documentos con outros, ou o que é o mesmo uns nodos con outros formando un rede denominada arañeira (en inglés *web*), de aí que cada documento pasara a denominarse páxina web. Cando se trata de texto as ligazóns soen aparecer resaltados en cor azul e subliñados, e mesmo pode cambiar o estilo do punteiro do rato para que non pasen desapercibidos.

Os documentos denominados páxinas web, son documentos en linguaxes estándar como HTML ou XML que poden incluír diferentes tipos de información: texto, hiperligazóns, gráficos e outros elementos multimedia. Estas páxinas web alóxanse en servidores web distribuídos por todo o mundo no que se coñece como **sitios web**. Cando o servidor se atope conectado á rede a conxunción de enderezos IP e nomes de dominio permitirá acceder aos documentos do sitio e visualizalos mediante uns programas denominados **navegadores** de Internet. Este tipo de programas implementan o protocolo HTTP que funciona sobre a familia de protocolos TCP/IP encargándose de xestionar a comunicación entre o cliente e o servidor web. Para acceder ao enderezo dunha páxina web podemos facelo tanto mediante ligazóns como directamente dende a barra de enderezos do navegador sen máis que inserir directamente nela o nome ou enderezo web do sitio.



Os enderezos web serven para identificar os recursos da rede, e denomínanse **URL** (en inglés *Uniform Resource Locator*) ou localizador uniforme de recurso. As URL poden ser da forma: <https://www.xunta.es:80/ruta/index.htm> tendo os seguintes compoñentes:

- a) O protocolo da rede que se emprega para recuperar a información do recurso especificado, neste caso 'https', sendo un dos máis habituais xunto a 'http', 'ftp', 'mailto', 'file', ou 'ldap'. Normalmente o protocolo HTTP é opcional na maioría dos navegadores xa que se trata do protocolo máis utilizado.
- b) O nome de dominio, ou servidor co que se comunica, neste caso '[www.xunta.es](http://www.xunta.es)'.
- c) O porto de comunicación que emprega ese protocolo no servidor, neste caso ':80', sendo este opcional pois os protocolos acostuman levar un porto asociado por defecto.
- d) A ruta do recurso no servidor, (en inglés *path*), neste caso '/ruta'.
- e) O nome do arquivo aloxado nesa ruta ou directorio, neste caso '*index.htm*'.
- f) Outros campos como parámetros ou propiedade propias de determinados protocolos.

## **2. Correo electrónico**

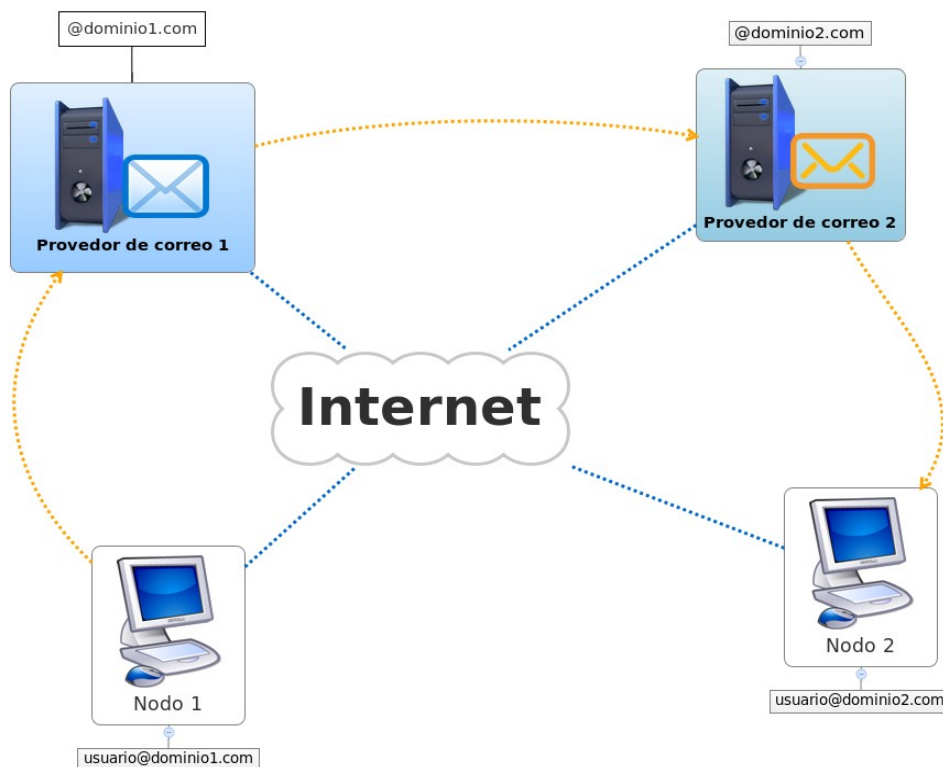
O servizo de **correo electrónico** (e-Correo) proporciona os mecanismos para facilitar o envío e recepción de mensaxes que poden incluír texto e outras achegas a modo de arquivos multimedia. Neste servizo identifícase cada usuario cunha conta que levará o seu nome de usuario para o dominio dese servidor de correo seguido do símbolo '@' (arroba) e o nome de dominio (DNS) dese servidor. Por exemplo: [usuario@xunta.es](mailto:usuario@xunta.es). O acceso ao correo electrónico pode facerse con dous sistemas diferentes, ou ben accedendo cun correo web ou ben cun cliente de correo electrónico. No



**correo web** (en inglés *webmail*) accédese dende un navegador a unha páxina de administración do correo, que require a autenticación do usuario e permite facer as operacións como en calquera outro sitio web. Neste caso é idéntico a calquera outro servizo *www*, é dicir emprega os protocolos HTTP ou HTTPS segundo o nivel de seguridade do servidor. Non require software adicional e calquera equipo que teña instalado un navegador permitirá acceder a un servidor de correo remoto.

A alternativa é empregar software a modo de clientes de correo electrónico específicos que permiten conectar un software de xestión de correo co servidor a través dos protocolos de correo **POP3** ou **IMAP**. Estes dous protocolos permiten obter e enviar mensaxes de correo dende e cara un servidor remoto. A diferenza entre ambos protocolos é que POP3 atópase máis orientado cara a recepción de correo que para o envío, co cal ao conectarse descarga todas as mensaxes ao equipo cliente e as elimina do servidor, mentres que o protocolo IMAP as mantén. En liñas xerais IMAP proporciona máis funcionalidades que POP3, sendo un pouco máis complexo polo que non se atopa implantado en todos os servidores de correo. Para especificar como deben encamiñarse os correos empréganse os **Rexistros MX** (en inglés *Mail eXchange Record*), recursos DNS que indican os servidores de correo por prioridade. O MTA (en inglés *Mail Transfer Agent*) solicita o Rexistro MX perante unha petición DNS encamiñando posteriormente o envío. Existen moitos riscos de seguridade asociados aos correos, ademais da posibilidade de envío de virus, Hoax ou troianos, algúns servidores permiten o envío aberto ou Open Relay.





**Figura 3: Funcionamento do correo electrónico.**

### **3. Transferencia de arquivos (FTP)**

O servizo de transferencia de arquivos ou FTP (en inglés *File Transfer Protocol*) é un protocolo que define os estándares para o servizo de transferencia de arquivos a través de Internet. Trátase dun sistema cliente-servidor ao estilo dos anteriormente comentados onde un equipo cliente pódese conectar cun servidor de arquivos remoto para descargar ou enviar un ou máis ficheiros independentemente do sistema operativo do equipo cliente. Coma acontecía co correo electrónico pode xestionarse dende un navegador empregando o servizo *www*, ou ben cun cliente FTP que faga transparentes e usables as diferentes funcionalidades do servizo. Unha conta de usuario especial é a que ten como usuario e contrasinal 'anonymous' que se emprega para acceder a servidores FTP anónimos ou públicos, trátase dun estándar de facto para permitir acceder a calquera persoa aos contidos dun directorio público dun servidor FTP. Ampliacións



deste protocolo no eido da seguridade dan lugar á evolución a **SCP** (en inglés *Secure Copy*) e **SFTP** (en inglés *SSH File Transfer Protocol*) ambos engaden a seguridade **SSH** (en inglés *Secure Shell*) no primeiro limitado a transferencia de arquivos e no segundo con máis opcións.

#### **4. Conexión ou acceso remoto (Telnet)**

Este servizo permite o acceso remoto a outro equipo a través da rede e traballar con ela dende o noso equipo a través dunha consola coma se estiveramos conectados directamente a ela, coma un usuario desa máquina. **Telnet** é o protocolo de rede que permite realizar este tipo de comunicacións, que precisan que no servidor remoto estea activado o servizo de Telnet para aceptar as comunicacións. Require unha conta de usuario e contrasinal para o servidor de Telnet, que en moitos casos pode coincidir cun usuario do equipo remoto. Os problemas de seguridade das versións iniciais do Telnet arranxáronse coa súa evolución a **SSH** unha nova versión do sistema con técnicas de cifrado e con novas funcionalidades. Como ocorría co Telnet, SSH é tanto o nome do protocolo coma o do programa que o implementa, e como acontecía co FTP e co correo electrónico existe software de xestión que facilita ao usuario a conexión vía Telnet ou SSH. A posibilidade de estar nunha computadora mentres se traballa en outra resulta moi útil para tarefas administrativas, sobre todo para os administradores de rede ou para situacións de teletraballo.

Un paso máis alá, os **terminais en modo gráfico** permiten ademais de texto amosar imaxes, co cal accederíamos dende o noso equipo a un escritorio idéntico a como o veríamos se nos atopáramos fisicamente no equipo remoto. Os clientes deste servizo empregan os protocolos RDP ou X11 segundo o sistema operativo, para sistemas Windows e Unix/Linux respectivamente.

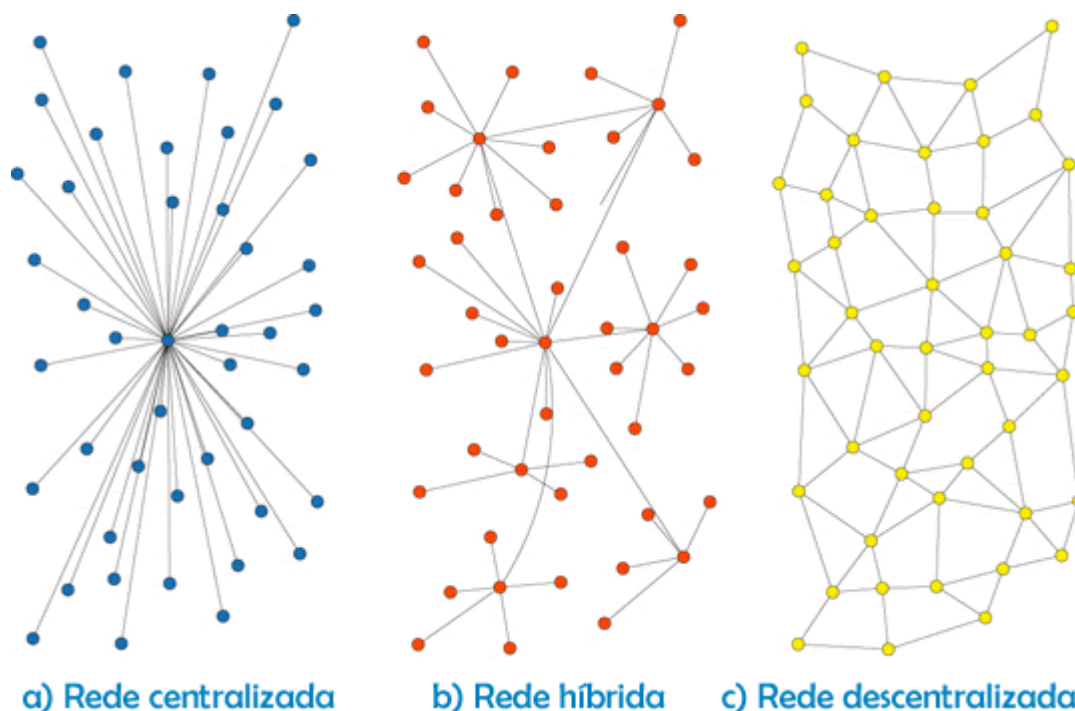


## **5. P2P**

O servizos P2P teñen a súa orixe no concepto das redes entre iguais (en inglés *peer-to-peer*). A característica principal deste tipo de redes é que todos os nodos que participan na rede teñen o mesmo peso na mesma, todos actúan como clientes e como servidores. Trátase de subredes dentro da Internet establecidas a partir dun determinado software de xestión para P2P. Nun principio estas redes podían ter nodos centrais para xestionar as comunicacións se ben a base do intercambio de arquivos seguía sendo distribuída. Cada nodo, equivalente a un usuario conectado á rede P2P comparte os seus recursos con todos os demais nodos, se ben o xeito máis habitual é o de compartir arquivos en ocasións permiten realizar cálculos de custe elevado ou procesamentos de datos masivos con orientación científica. Segundo dispoñan de nodos centrais podemos falar dos seguintes tipos de redes:

- a) Redes P2P centralizadas, en forma de estrela cun servidor central que monopoliza a xestión e administración da rede.
- b) Redes P2P híbridas, onde ademais do nodo central existen nodos de segundo nivel que centralizan a xestión de subredes.
- c) Redes P2P descentralizadas, onde todos os nodos son clientes e servidores co mesmo peso.



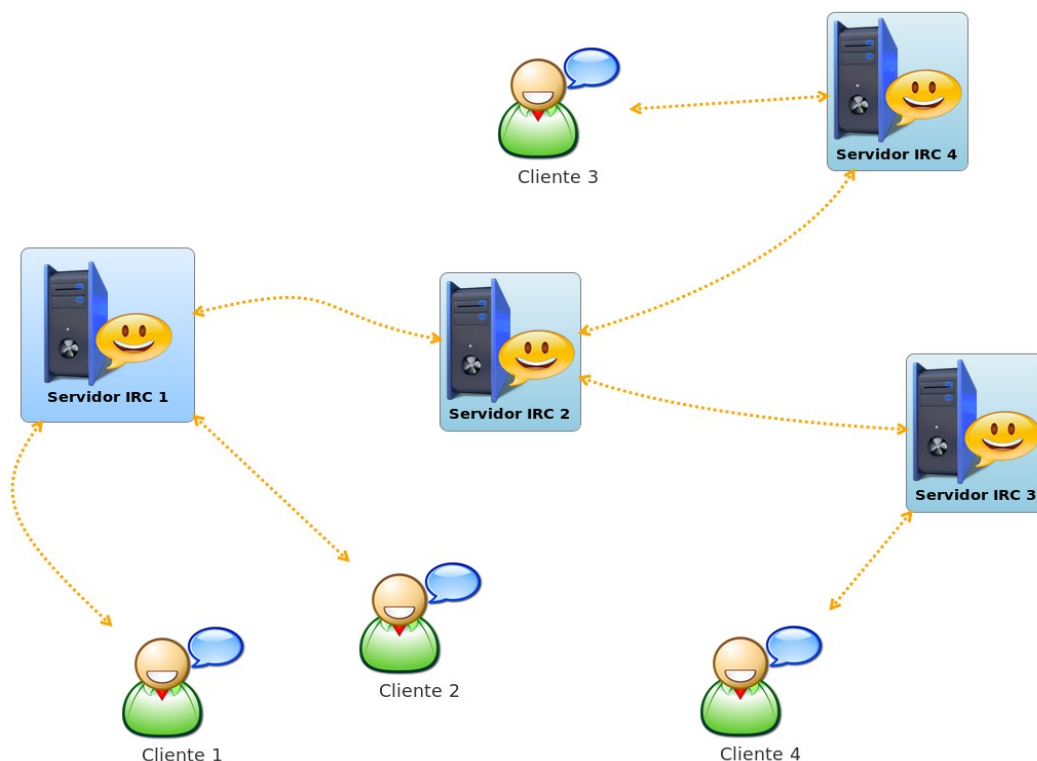


**Figura 4: Topoloxías habituais das redes P2P.**

## 6. Conversa (Chat)

Este servizo permite que dous ou máis usuarios conectados simultaneamente a Internet sosteñan conversas interactivas en tempo real. O IRC (en inglés *Internet Relay Chat*) é o protocolo de comunicación baseado en texto que sustenta o servizo. As conversas teñen lugar nos denominados canles de IRC de maneira que cada canle pode soste unha conversa paralela entre dous ou máis nodos calquera da rede. Existen múltiples clientes que como ocorría cos servizos anteriores facilitan o uso do servizo aos usuarios. Nas súas orixes permitía unicamente o envío de mensaxes de texto, pero evolucionou ata permitir o envío de arquivos, transmisión de voz e vídeo e mesmo conexión de escritorio remota.





**Figura 5: Exemplo dunha rede IRC.**

#### **25.2.4. MOTORES DE BUSCA**

Outra característica de Internet froito da gran cantidade de información que almacena sería a existencia dunhas ferramentas denominadas **Motores de busca** (en inglés *browser*). Estas ferramentas buscan os arquivos almacenados nos servidores web e os indexan para poder proporcionar resultados de buscas de palabras chave nos mesmos nun tempo óptimo. Os motores de busca empregan un **robot** (ou simplemente *bot*) que fai as funcións de rastrexador da web. Periodicamente este robot recolle información sobre os sitios e páxinas web que recorre dende un punto de partida ás ligazóns de cada documento que percorre. Deste xeito pode descubrir novos documentos nun sitio sempre e cando estean vinculados dende outros documentos xa atopados do sitio. En determinadas ocasións podemos non desexar que o documento sexa incorporado aos buscadores, polo que poderemos perante código HTML indicarlle ao robot que salte ese



documento. A información que recolle un robot inclúe o texto e parte do código da páxina web, non podendo interpretar imaxes, animacións ou vídeos non sendo a través da súa descrición. Para que un sitio web pase a existir cómpre dalo de alta en polo menos un dos buscadores, abondando con incluír a páxina principal do sitio sempre e cando o resto de páxinas estean ligadas dende ela.

A partir da información recollida polo robot elabórase un **índice** ou catálogo de documentos orientado a facilitar a busca de información. Con cada nova busca a información de rastreo deberá actualizarse e consecuentemente tamén o índice ou catálogo, incorporando as novas páxinas descubertas, eliminando as que foron borrados así como os cambios de cada documento.

De cara ao usuario o motor de busca proporcionará unha **interface de busca** vía web ou cliente software onde a partir dun termo inserido daralle como resultado as ligazóns atopadas que mellor se correspondan por orde de relevancia. Os factores chave do resultado dun buscador serán por tanto o tempo de resposta, optimizado grazas ao índice e a relevancia ou adaptación dos resultados aos termos empregados na busca.

Por norma xeral os buscadores implementan os seus propios algoritmos de relevancia ou **posicionamento**, que establece un peso para cada páxina en función do número de visitas, número de páxinas que a enlazan, aspectos comerciais, valoración dos usuarios e un longo etcétera. No resultado dunha busca aparecerán primeiro as páxinas que teñan un maior posicionamento ou relevancia.

### **25.3. INTRANET/EXTRANET**

En liñas xerais unha Intranet compórtase igual que Internet, sendo unha Internet limitada ao ámbito da organización para a que da servizo, é dicir



unha Internet privada. Unha Intranet sería unha Internet que restrinxe o acceso aos sistemas de información. A efectos de alcance e servizos poderemos dispor das mesmas posibilidades en cada tipo de rede. No tocante ao seu funcionamento tamén é idéntica ao de Internet, cada equipo ou nodo tamén disporá dun enderezo IP, pero neste caso non se corresponderá cos enderezos IP de Internet senón que será un enderezo IP privado, para uso interno. Se parte dos equipos atópanse abertos a Internet pasaremos a falar de Extranet, podendo convivir ambas na mesma organización.

Noutra variante unha Extranet pode comunicar dúas Intranets con distinta localización xeográfica establecendo por exemplo unha Rede privada virtual ou **VPN** (en inglés *Virtual Private Network*) que define unha rede privada lóxica sobre unha rede pública. Existen varias arquitecturas de VPN:

- 1) **VPN de acceso remoto.** Conecta directamente os usuarios a rede a través de Internet tendo en conta tanto só que o usuario se autentica de maneira correcta.
- 2) **VPN punto a punto (Tunneling).** Require un servidor VPN que responde ás conexións a través de Internet e crea un túnel VPN, que consiste en enmascarar un protocolo de rede sobre outro. Deste xeito pódense transmitir os paquetes con protocolos cifrados como SSH.
- 3) **VPN LAN.** Nesta solución non se emprega Internet para o acceso remoto senón que se fai sobre a propia rede da organización. En redes sen fíos, permite establecer un nivel de seguridade engadido onde ademais dos protocolos de seguridade da Wi-Fi se inclúen as credenciais de seguridade do túnel VPN.

Particularizando e concretando os servizos que ofrece Internet podemos definir unha serie de **servizos** básicos que pode proporcionar unha Intranet/Extranet:



**a) Acceso a sistemas de información**

- ✓ Acceso a documentación: manuais, publicacións, guías e formularios internos.
- ✓ Acceso a sistemas de información e bases de datos corporativas.
- ✓ Consulta e edición de informes, formularios e listaxes.
- ✓ Axenda, calendarios e planificación de traballo en grupo.
- ✓ Acceso a información de contacto da organización.
- ✓ Páxinas de novas e ligazóns de interese.

**b) Recursos compartidos**

- ✓ Acceso a recursos compartidos: conexión a Internet, impresoras, escáners, etc...
- ✓ Acceso a sistemas de intercambio de arquivos.
- ✓ Buscadores de recursos e información.

**c) Fluxos de traballo**

- ✓ Xestión de usuarios e perfís.
- ✓ Acceso a aplicacións/equipos remotos.
- ✓ Acceso a repositorios de versións.
- ✓ Acceso a aplicacións de xestión e control de incidencias.
- ✓ Soporte a traballadores móbiles ou tele-traballadores.

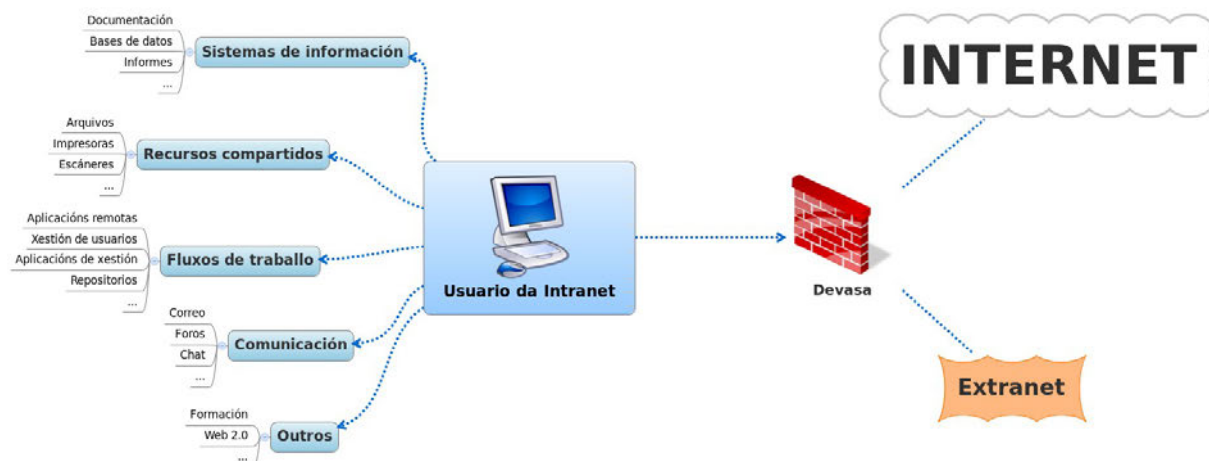
**d) Comunicación**

- ✓ Servizos de mensaxería interna, correo electrónico, foros e videoconferencia.

**e) Outros**

- ✓ Realización de actividades de formación.
- ✓ Acceso a ferramentas da Web 2.0: portais, blogs, wikis, redes sociais, etc...





**Figura 6 : Servizos básicos dunha Intranet/Extranet.**

Para poder soportar esta longa lista de servizos unha Intranet debería estar dotada dos seguintes **compoñentes** básicos:

### **1. Soporte e infraestrutura de rede.**

O modelo máis sinxelo de Intranet sería dous equipos dunha organización conectados en rede. A partir de aí a rede pode medrar tanto como requira a organización, podendo incluír calquera número de redes, subredes, equipos e elementos de interconexión. A implantación disto equivalería á dunha rede LAN, sendo preciso definir unha topoloxía, un conxunto de tecnoloxías (Ethernet, Fibra óptica, Wi-Fi, etc...), dispositivos de interconexión e de seguridade e unha política de asignación de enderezos IP e nomes de dominio DNS. Nun paso máis alá habería que estender a rede cara o exterior no caso de que se queira definir parte da mesma como Extranet, tendo en conta tamén este conxunto de características. A este respecto hai que ter en consideración o Plan de Direccionamento e Interconexión de Redes de Área Local na Administración (2010), o cal establece os rangos de asignación de enderezos IP. Para a Xunta de Galicia estable o seguinte **rango**:



**10.179.0.0 -  
10.180.0.0***Xunta de Galicia*

Así como outras **recomendacións**, tales como:

- ✓ Cada entidade ou organismo pode xestionar independentemente os seus plans de numeración IP pero seguindo o plan para evitar enderezos duplicados cos outros organismos.
- ✓ Empregar máscaras de rede de 24 bits, para ter redes de 254 nodos por segmento, co cal teríamos a máscara 255.255.255.0 independentemente da situación física do nodo.
- ✓ A asignación do grupo de bits para o *Host* realizárase de xeito ascendente para permitir subredes nas zonas aínda non asignadas.
- ✓ Empregar valores de enderezos IP baixos para servidores e equipos de comunicacións.
- ✓ Valores por riba dos anteriores para equipos de usuarios, ordenadores persoais e estacións de traballo.
- ✓ Seguir o plan de numeración en cada subrede.
- ✓ Manter ao día a documentación dos cambios que se producen no mesmo.

## **2. Servidores.**

Os servidores serán os provedores de servizos na Intranet/Extranet atopándonos diferentes requirimentos hardware e software segundo a función que van desempeñar. Segundo o seu perfil dentro da rede atopamos dous tipos:

- a) **Servidores adicados.** Invisten toda a súa potencia en dar servizo á rede, adicando todos os seus recursos a tal función.



- b) **Servidores non adicados.** Funcionan tanto como servidor como estación de traballo, repartindo os seus recursos nas dúas funcións.

Por outra banda atendendo ao tipo de servizo que proporcionan, poderían clasificarse do seguinte xeito:

- 1) **Servidores de arquivos.** Almacenan arquivos e directorios e xestionan o acceso aos mesmos por parte dos usuarios da Intranet. En sistemas avanzados proporcionan información de versións, permisos e servizos de transferencia, sincronización, replicación e soporte de protocolos SMB/NetBIOS, CIFS, NFS e FTP así como funcionalidades de integración do estilo de Samba.
- 2) **Servidores de impresión.** Controlan as impresoras, fax ou escáneres en rede, realizando tarefas de xestión de colas, asignación de prioridades e detección de erros, con soporte para protocolos IPX/SPX, LDP, IIP, CUPS ou vía Socket.
- 3) **Servidores de comunicacións.** Realizan a xestión das comunicacións de telefonía, voz sobre IP (VoIP) ou videoconferencia. Sistemas avanzados inclúen contestadores automáticos e sistemas de resposta robótica paralela automática. Deben soportar diferentes protocolos como TCP/IP, IPX, PPP, SLIP/CSLIP, SNMP, LAT ou NetBEUI.
- 4) **Servidores de correo.** Almacenamento e xestión de mensaxes exclusivo para usuarios da Intranet/Extranet, con soporte para SMTP, IMAP, POP3 e seguridade SSL/TLS.
- 5) **Servidores de mensaxería instantánea.** Xestionan as comunicacións de *Chat* ou conversa instantánea entre os usuarios da Intranet/Extranet con soporte IRC, MUC, SIMPLE, MNP ou XMPP.
- 6) **Servidores de rede.** Realizan funcións de interconexión das redes e subredes que forman a Intranet/Extranet. Xestións de cachés (en inglés *proxy*), encamiñamento, servizos de devasa (en inglés *firewall*), NAT, DHCP, etc...



- 7) **Servidores de acceso remoto.** Xestionan a conexión remota de equipos dende outras localizacións con protocolos XDMCP, NX, RFB ou RDP. Optimizan a elevada carga do uso de aplicacións e escritorios de maneira remota e incorporan mecanismos de autenticación avanzados.
- 8) **Servidores de aplicacións.** Permite que os clientes traballen con aplicacións de custe de implantación/configuración elevado ou cunha alta demanda de recursos de maneira remota. As solucións máis habituais baséanse nas plataformas JEE, .NET, PHP e Coldfusion.
- 9) **Servidores de copias de seguridade.** Permiten manter un sistema de control de almacenamento de copias de seguridade de datos ou servidores en discos duros redundantes ou cintas, en ocasións noutras localizacións pero adicados ou SAN. O obxectivo destes sistemas é restaurar o sistema a un estado funcional e seguro logo dun erro, caída ou desastre que provoque a perda da funcionalidade da rede, convertendo as redes locais en NAS. Actualmente coa mellora das conexións van gañando forza os *backups* na nube.
- 10) **Servidores de Bases de datos.** Provéen os servizos de acceso ás Bases de datos así como a xestión das mesmas dende ordenadores con máis recursos que as estacións de traballo. Resulta habitual a súa comunicación con outros servidores para proporcionar servizos conxuntos. Algúns dos exemplos máis representativos son Oracle, DB2, SQL Server, MySQL e PostgreSQL.
- 11) **Servidores web.** Soportan o servizo de contidos web a nivel interno controlando o acceso ás páxinas e documentos HTML e XML. Os dous exemplos máis representativos son Apache e IIS.
- 12) **Outros.** Calquera outro servizo de importancia para a Intranet/Extranet debería ter un servidor adicado especializado que destinara todos os seus recursos á xestión e soporte dese servizo. Algúns destes servizos poderían ser o control e xestión de usuarios



(Servidores LDAP), servidores de informes, control de versións, etc...

### **3. Control de Seguridade**

En todas as redes e sistemas de comunicación é importante adicar recursos ao control da seguridade, principalmente nas partes visibles dende fóra, é dicir as partes da Extranet, pero tampouco hai que esquecer as partes propias da Intranet. A maior parte da seguridade recae sobre as **devasas**, que filtran as comunicacións co exterior, restrinxen aplicacións e controlan os enderezos IP e físicos das máquinas segundo unha serie de regras e filtros de control. Así mesmo dispoñen de ferramentas de monitorización e rexistro que permiten facer seguimentos e auditorías da rede.

Por outra banda, pódese restrinxir a comunicación co exterior empregando equipos de interconexión ponte que dean servizo ao resto da rede. Estes dispositivos de **xestión de caché**, fan a función de repetidores na rede, pero illan aos equipos internos e permiten centralizar e reforzar a seguridade e o control neste equipo, en lugar de en toda a rede. Os equipos pasarela deberían incluír todos os servizos básicos como Web, FTP ou mensaxería instantánea.

A seguridade debe contemplarse tamén nos **clientes**, aínda que unha correcta xestión nos servidores protexe por extensión aos equipos de traballo. Cómpre prestar especial atención ao control dos usuarios de cada equipo, controlar accesos físicos, xestión de contrasinais, e dotalos dun software antivirus axeitado. En ocasións pode ser necesario controlar o acceso dos usuarios ao mesmo equipo distinguindo en diferentes perfís usuario/administrador ou máis segundo as necesidades da organización.

### **4. Administración da rede.**



O papel de administrador da rede resulta fundamental para asegurar o correcto funcionamento e seguridade do sistema, ademais de para dar soporte e participar da resolución de incidencias.

Mención especial merece o control das comunicacións que se realizan entre os usuarios, tendo especial coidado con temas como correos masivos ou SPAM, envío masivo ou non autorizado fóra da organización. Entre as labores ou **funcións** do administrador ou administradores atoparíamos:

- ✓ Establecemento e mantemento de políticas de xestión de usuarios e roles, permisos e accesos.
- ✓ Mantemento e soporte físico do hardware da rede.
- ✓ Configuración e mantemento de devasas, antivirus e cachés, así como calquera outro equipamento ou software de conexión da rede.
- ✓ Avaliación da calidade do servizo.
- ✓ Realización de auditorías periódicas de control e avaliación da seguridade e rendemento.
- ✓ Atención aos usuarios, soporte e resolución de incidencias.
- ✓ Documentación do deseño e descrición da rede, configuracións de servidores e protocolos de restauración/recuperación da rede en caso de erro ou desastre.

## **25.4. IMPLANTACIÓN DE REDES EN ORGANIZACIÓNS**

Cando as organizacións conectan a súa rede e servizos con Internet teñen varias alternativas:

**1) Integrar por completo a rede corporativa en Internet.** Deste xeito cada equipo da organización pasa a ser un nodo de Internet, con enderezos IP de Internet. Dende calquera localización poderase ter acceso aos servizos e equipos da rede directamente, sen restricións. Esta solución plantexa riscos de seguridade, xa que o conxunto da rede queda exposto a



ataques dende o exterior, e cada nodo convértese nun potencial punto feble.

**2) Integrar parcialmente a rede e equipos.** Neste tipo de solucións a maioría da rede aparece oculta ao exterior, fóra de Internet, para evitar os riscos de seguridade. Dende o exterior pódense ver algúns servidores da organización e do mesmo xeito dende a rede se pode acceder a servidores externos, pero restrinxido os servizos e comunicacións. Cando a integración é parcial pódese falar de redes dos tipos Intranet e Extranet. Partindo disto xorden outras moitas cuestións, número de usuarios, distribucións físicas que abarcará a rede, servizos que se implantarán, e un longo etcétera. Resulta obvio que o primeiro paso da implantación dunha Intranet/Extranet será a **planificación**. Na planificación abordaranse os seguintes puntos:

### **1. Plantexamento de obxectivos.**

Os **obxectivos** da rede quedará definidos polo seu alcance. Habería que realizar tarefas tales como:

- ✓ Estimación do número de usuarios e a súa posible evolución.
- ✓ Determinar o emprazamento da rede e posibles subredes, situación de posibles redes externas e as necesidades de comunicación coas mesmas.
- ✓ Considerar os sistemas de información e necesidades de acceso aos mesmos, tendo en conta tamén os fluxos e procesos internos.
- ✓ Definir os servizos que se proporcionarán na Intranet/Extranet polo miúdo, con estimacións de carga predicións da súa evolución futura, etc...

A partires dos obxectivos pode irse elaborando a lista de **requisitos** da Intranet/Extranet, como paso previo ao deseño da rede. A documentación en ambos puntos debería ser o máis completa posible.



## **2. Selección de tecnoloxías**

Nun segundo paso tomando os obxectivos e requisitos habería que seleccionar as tecnoloxías máis axeitadas tanto a nivel de hardware, físico, como de software. A nivel físico acostuma a optarse por redes Ethernet, pero poden ser precisas redes sen fíos, así mesmo cada rede precisaría diferentes elementos de interconexión dependendo da tecnoloxía e o mesmo para clientes e servidores. Dende sistemas operativos a software de xestión e control de cada servizo, xestores de contidos e outros propósitos, debería seleccionarse atendendo ás necesidades especificadas polos obxectivos e requisitos sen esquecer outros aspectos como custes, a existencia e dispoñibilidade de soporte para cada tecnoloxía e complexidade de instalación, configuración e mantemento.

## **3. Definición dos recursos necesarios.**

Unha vez seleccionadas as tecnoloxías que tomarán parte no deseño da selección habería que definir o número de **recursos** necesarios para a implantación. Isto inclúe, número, tipo e software dos equipos clientes e o mesmo para os equipos de interconexión da rede e o servidores. Este paso sería o **deseño** da rede en si, dende o cableado á lista completa de software necesario. Segundo as particularidades da Intranet/Extranet podería ser preciso o desenvolvemento de software a medida, o cal habería que incluír tamén nesta fase do deseño.

## **4. Definición de políticas de seguridade.**

Paralelamente haberá que establecer e documentar os protocolos e políticas de seguridade como:

- ✓ A asignación de contas de usuario e contrasinais e usuarios dos equipos e servidores, así como caducidade e revisión do cambio de contrasinais nas mesmas.



- ✓ Equipos que comunican co exterior e que precisan máis seguridade e equipos que pertencerán á DMZ (zona desmilitarizada).
- ✓ Filtros de aplicacións, de enderezos IP e enderezos físicos.

A continuación viría a **implantación** en si, sendo o recomendable establecer un período de proba e realimentación previo para ofrecer un produto de maior calidade.

### **1. Período de proba.**

Durante este período realizaranse probas completas do funcionamento da Intranet/Extranet. Deberían incluírse casos de proba para os diferentes servizos e as comunicacións entre as diferentes subredes e redes externas. Coa calidade como obxectivo habería que realizar probas máis complexas como probas de carga, buscando os picos de demanda de recursos, e casos de ataques de seguridade controlados. Todo elo mentres se realiza a monitorización e posteriores probas de auditoría do sistema permiten elaborar un informe completo dos límites e deficiencias da rede, útil para detectar puntos febles que arranxar antes da posta en servizo.

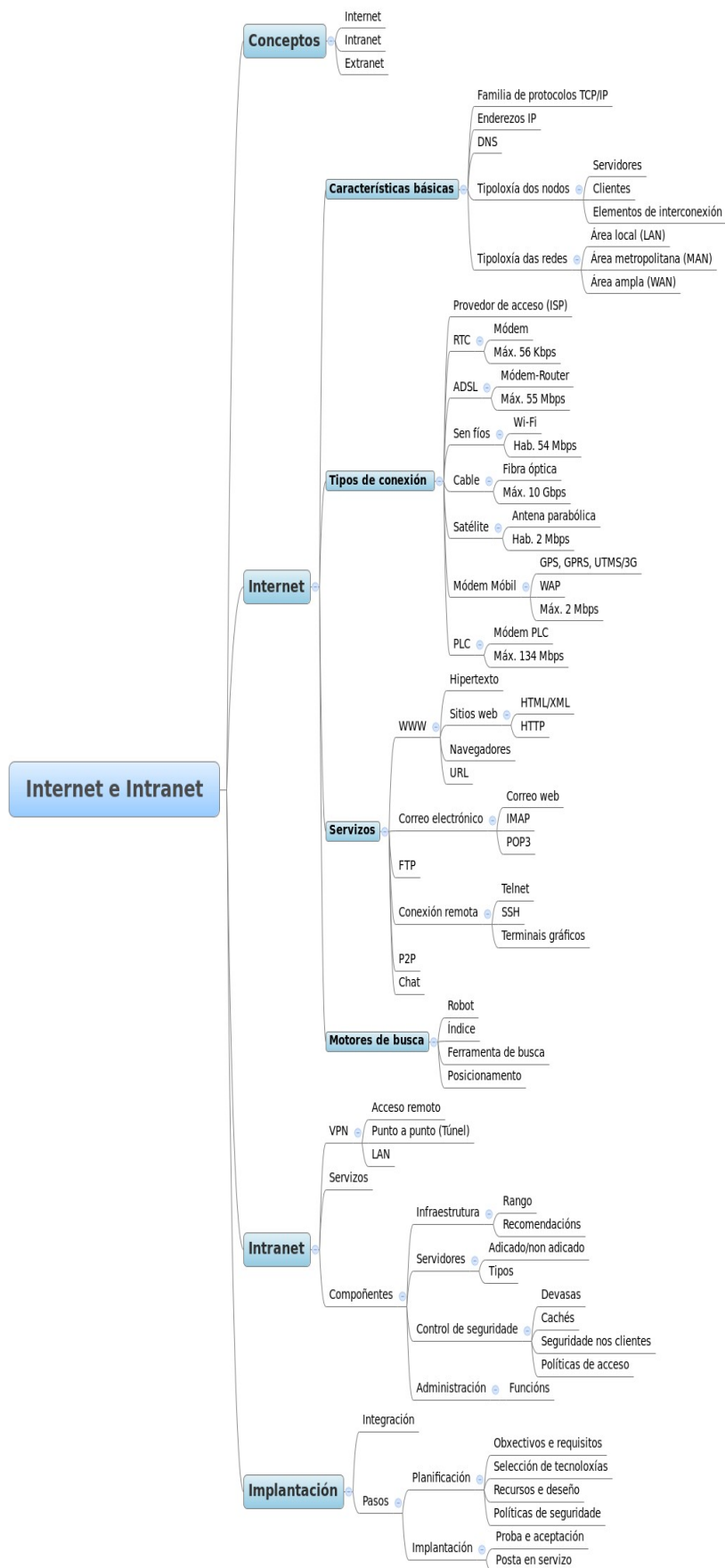
### **2. Posta en servizo.**

Logo das sucesivas probas e unha vez obtidos resultados de aceptación pode levarse a cabo a posta en servizo definitiva da Intranet/Extranet, abríndoa a todos os usuarios. Nos primeiros momentos da posta en servizo cómpre realizar a monitorización e auditoría os sistemas do mesmo xeito que se fixo durante o período de proba pois neste primeiro momento poderán detectarse problemas e debilidades reais que puideron pasar desapercibidas durante as probas controladas realizadas anteriormente.



## 25.5. ESQUEMA







## **25.6. REFERENCIAS**

Abel Rodríguez Ávila.

Iniciación a la red Internet. Concepto, funcionamiento, servicios y aplicaciones de Internet. (2007).

Irene Rodil e 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 e 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 APLICACIÓN, SERVIDORES DE DATOS, GRANXAS DE SERVIDORES. SCRIPTS DO CLIENTE.**



## TEMA 26. MODELO DE CAPAS: SERVIDORES DE APLICACIÓNS, SERVIDORES DE DATOS, GRANXAS DE SERVIDORES. SCRIPTS DO CLIENTE.

### 26.1. INTRODUCCIÓN E CONCEPTOS

### 26.2 MODELO DE CAPAS: SERVIDORES DE APLICACIÓNS, SERVIDORES DE DATOS, GRANXAS DE SERVIDORES.

### 26.3 SCRIPTS DO CLIENTE.

### 26.4. ESQUEMA

### 26.5. REFERENCIAS

### 26.1. INTRODUCCIÓN E CONCEPTOS

Nunha rede formada por equipos informáticos os **nodos** acostuman realizar tres **funcións** diferenciadas:

1. Facilitar a comunicación e interconexión dos nodos da rede.
2. Proporcionar servizos ou información a outros nodos.
3. Realizar funcións de equipo de traballo, facendo uso das comunicacións, servizos e información dispoñible.

Neste contexto os nodos ou equipos que realizan as funcións de proporcionar servizo ou información ao resto denomínanse **Servidores** e os que fan uso dos servizos **Cientes**. Formando no seu conxunto o que se da en denominar **Arquitectura Cliente-Servidor**. Trátase dunha das arquitecturas máis estendidas nos contornos distribuídos, permitindo a heteroxeneidade nos clientes e un acceso transparente á información. Os servidores permanecen á escoita da rede en todo momento para atender ás solicitudes ou demandas dos clientes.



O esquema de **funcionamento básico** seguiría o seguinte modelo:

1. O cliente solicita un servizo ao servidor a través da rede.
2. O servidor á escoita recibe a petición do servizo e a pon na cola de demanda.
3. O servidor obtén o resultado da petición.
4. O servidor envía a resposta da petición ao cliente a través da rede.
5. O cliente obtén o resultado e o procesa.

A partir destes conceptos básicos podemos extraer as **características básicas** da arquitectura Cliente-Servidor:

- a) **Servizos.** Son a base das peticións entre os clientes e o servidores, trátase de calquera entidade susceptible de ser demandada por un ou máis clientes.
- b) **Recursos compartidos.** Elementos e servizos da rede, tanto lóxicos (software, datos e información), como físicos (hardware, impresoras, unidades en rede, etc...).
- c) **Comunicación asíncrona baseada no envío de mensaxes.** Este tipo de arquitecturas empregan protocolos de comunicación asimétricos onde os clientes inician conversas e os servidores esperan que se estableza a comunicación escoitando a rede. Toda a comunicación se realiza mediante o envío de mensaxes e respostas.
- d) **Transparencia.** A localización, a organización lóxica e física, así como a implementación dos servizos resulta transparente aos clientes. O uso dos mesmos limítase a facer unha petición á rede e obter a resposta.
- e) **Escalabilidade.** Horizontal nos clientes á hora de permitir engadir novos nodos sen máis que engadilos á rede e vertical nos servidores, de xeito que administrando un único punto pode mellorarse a potencia, o rendemento, o mantemento e a recuperación de erros.



Froito da escalabilidade desta arquitectura xorden as **granxas de servidores**, consistentes en empregar varios servidores á vez subministrando o mesmo servizo e repartíndose as peticións ou carga do sistema. A xestión dunha granxa de servidores será complexa debido á necesidade de balancear a carga para obter o maior rendemento posible.

Un dos servizos máis estendidos actualmente é o web ou **WWW**, (siglas en inglés de *World Wide Web*), trátase dun sistema de publicación e intercambio de información distribuído que relaciona uns contidos con outros a través de ligazóns. Neste servizo os clientes solicitan información a modo de páxinas web, tratándose de documentos en linguaxes estándar como HTML ou XML que inclúen diferentes tipos de información: texto, hiperligazóns, e elementos multimedia. Entre estes elementos multimedia atopamos:

- a) **Texto.** Distinguindo entre sen formato, con formato ou enriquecido (tipo de letra, tamaño, cor, cor de fondo, etc...) e hipertexto texto cun vínculo ou ligazón a outro texto ou documento.
- b) **Son.** Dixitalización da fala, a música ou outros sons.
- c) **Gráficos.** Representan esquemas, planos, debuxos vectoriais, etc... son documentos que se constrúen a partires dunha serie de primitivas: puntos, segmentos, elipses, etc... aplicándolles a continuación todo tipo de transformacións ou funcións: rotación, cambio de atributos, escalado, efectos, etc...
- d) **Imaxes.** Representacións fieis da realidade, como fotografías. Son documentos formados exclusivamente por píxeles, punto a punto e por tanto non se estruturan ou dividen en primitivas.
- e) **Animación.** Representación dunha secuencia de gráficos por unidade de tempo, para ofrecer a sensación de movemento. Así mesmo ofrece posibilidades de interacción ante eventos.



- f) **Vídeo.** Representación dunha secuencia de imaxes por unidade de tempo, para ofrecer a sensación de movemento.

Documentos complexos agrupan diversos compoñentes multimedia nunha mesma páxina ou documento. Os sistemas de publicación actuais permiten que a **multimedia dixital en liña** poda transmitirse **en fluxo** (en inglés *streaming*), que se atopa dispoñible tanto en liña en tempo real coma baixo demanda. Neste modelo non é necesario descargar ou acceder á totalidade do documento para acceder aos contidos senón que se proporciona acceso directo a calquera parte do fluxo e reprodución dende ese punto.

Outra característica das páxinas web ou documentos HTML/XML é que poden incluír **código de script para os clientes**. Este código representa un guión ou secuencia de instrucións a xeito dun programa sinxelo. Este programa pode ser interpretado polo navegador do equipo cliente cuns permisos limitados no equipo e focalizados principalmente na páxina ou documento web no que se atopan incrustados ou dende o que son chamados. Existen diferentes tecnoloxías para estas linguaxes de *script* sendo as máis coñecidas: Javascript, Visual Basic Script, Flash, e a evolución de Javascript: AJAX, aceptados con maior ou menor fortuna polos navegadores actuais, moitas delas denominadas tecnoloxías RIA nun achegamento das aplicacións web ás aplicacións de escritorio.

## **26.2 MODELO DE CAPAS: SERVIDORES DE APLICACIÓNS, SERVIDORES DE DATOS, GRANXAS DE SERVIDORES**

A distribución dos sistemas de información foi evolucionando ao longo do tempo en función das demandas e crecemento das redes e o aumento da complexidade das arquitecturas de rede.



### **26.2.1. Arquitectura nunha capa: Superordenador central.**

A arquitectura máis simple estaría formada por un **superordenador central** (en inglés *mainframe*) que centraliza toda a capacidade de procesamento e almacenamento da rede, tamén denominada monolítica. Neste modelo o acceso á información faise directamente a través da computadora principal ou ben a través de clientes lixeiros que se limitan a facer as funcións de terminais. Nesta arquitectura centraliza todo o custe de administración e mantemento adícase ao servidor central. Os terminais carecen de programas propios, e teñen recursos de memoria ou disco mínimos, podendo mesmo carecer de disco. Calquera instalación ou mellora no servidor repercute ao momento na rede de xeito que calquera programa instalado estará dispoñible para todos os clientes. Por contra, se temos en conta a sostibilidade do sistema en caso de caída ou erros no servidor central toda a rede vese afectada, do mesmo xeito que se un cliente sobrecarga o sistema todos os demais veranse afectados en canto a rendemento. Á súa vez os mainframes organizáanse segundo arquitecturas paralelas tipo **SNA** (en inglés *Systems Network Architecture*) cun deseño de rede con comunicación P2P a través de **APPN** (en inglés *Advanced Peer-to-Peer Networking*).

### **26.2.2. Arquitectura en dúas capas: Modelo Cliente-Servidor.**

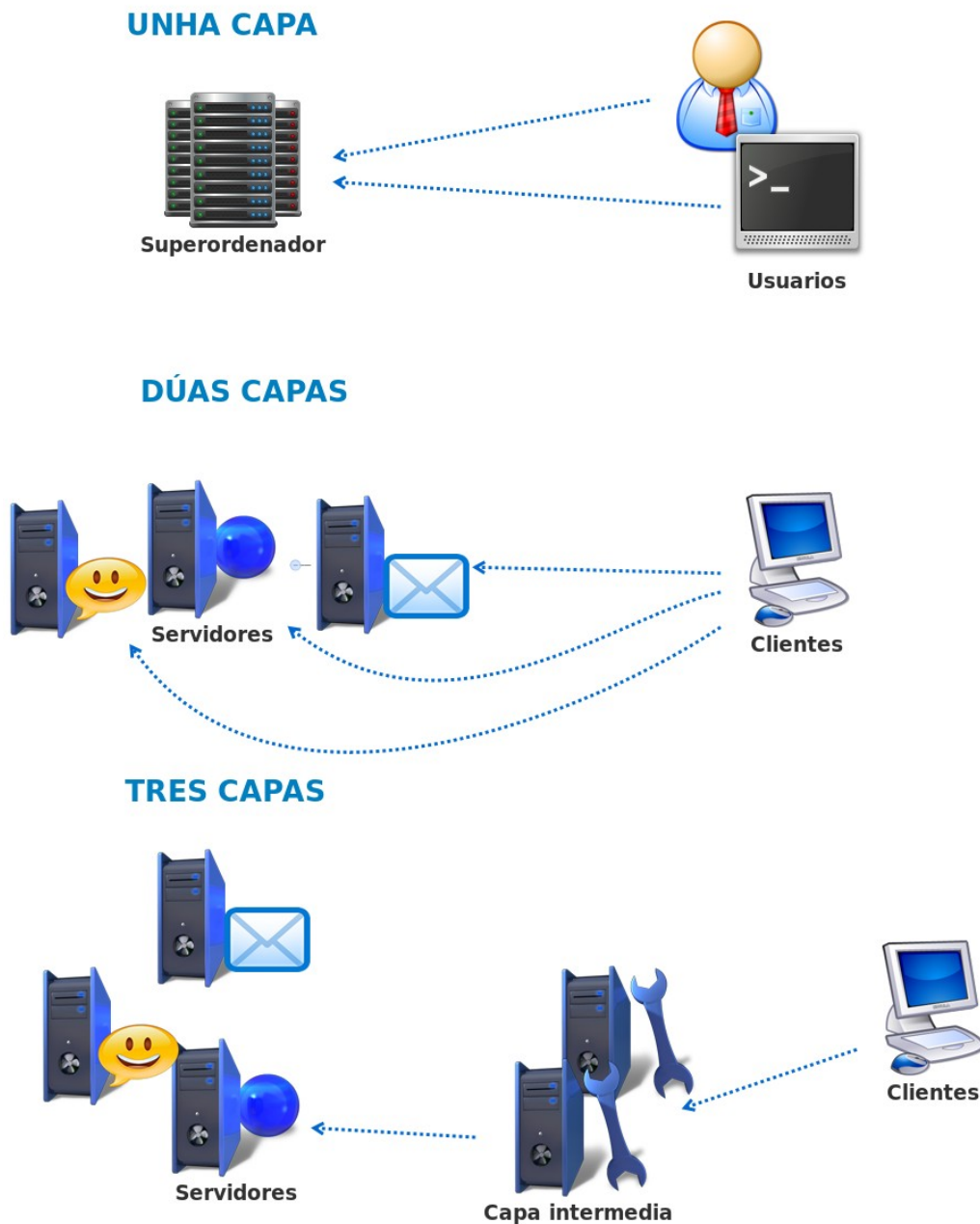
Neste modelo o sistema se estrutura en dúas capas, unha capa a nivel de usuario que almacena e procesa parte da información e outra capa remota a nivel de servizos que almacena e da funcionalidade á totalidade de clientes da rede. Deste xeito conséguese descargar de parte da carga da rede aos servidores centrais e mantén a capa de servizos transparente aos usuarios coa posibilidade de escalar o sistema mellorando ou aumentando o número de servidores sen que estes cheguen a notar o cambio en algo



máis que o rendemento. Por contra, un modelo máis distribuído no tocante aos clientes obriga a un maior mantemento dos mesmos por parte dos administradores. Outro dos puntos a ter en conta é a consistencia dos datos entre cliente e servidor, de xeito que cómpre coordinar cada servizo por separado.

Nesta liña o uso de protocolos de comunicación soporta o uso efectivo por parte dos clientes dos servizos da rede permitindo a heteroxeneidade dos clientes sempre e cando os implementen.





**Figura 1: Arquitecturas en capas**

### **26.2.3. Arquitectura en tres capas: Granxas de servidores.**

Por mor dos inconvenientes dos sistemas dunha única capa, que obrigan a manter un servidor central de tamaño demasiado grande para un mantemento e rendemento eficientes, e de dúas capas, que obrigan a



manter cada servidor independente do resto para un único servizo, optouse polo establecemento dunha capa máis entre as de cliente e servidor.

Nesta capa agrúpanse varios servidores nunha DMZ soportando o mesmo servizo dando lugar a unha redundancia que ten como vantaxes unha maior tolerancia a fallos e unha mellora de rendemento. A efectos da rede a granxa de servidores proporcionan un único servizo lóxico ou virtual integrado por calquera número de servidores físicos. Cada servidor da granxa debe ser unha réplica exacta do servidor lóxico en canto a datos e software instalado. Para escalar o sistema engádese á granxa un novo servidor réplica do virtual e aumenta a dispoñibilidade de recursos. O exemplo máis habitual de granxa de servidores é un servizo web, onde se por exemplo un servidor atende a mil usuarios e temos previstos picos de dez mil usuarios simultáneos poremos unha granxa de dez servidores, para atender o servizo e outros dous máis en previsión de caída dalgún ou picos puntuais aínda máis altos. A escalabilidade das granxas en función dos servizos ofertados define tipoloxías básicas como *Datacenters*, servidores de aplicacións, de importación, de *front-end* existindo elementos específicos para control de carga, Teredo ou de dominio, entre outros.

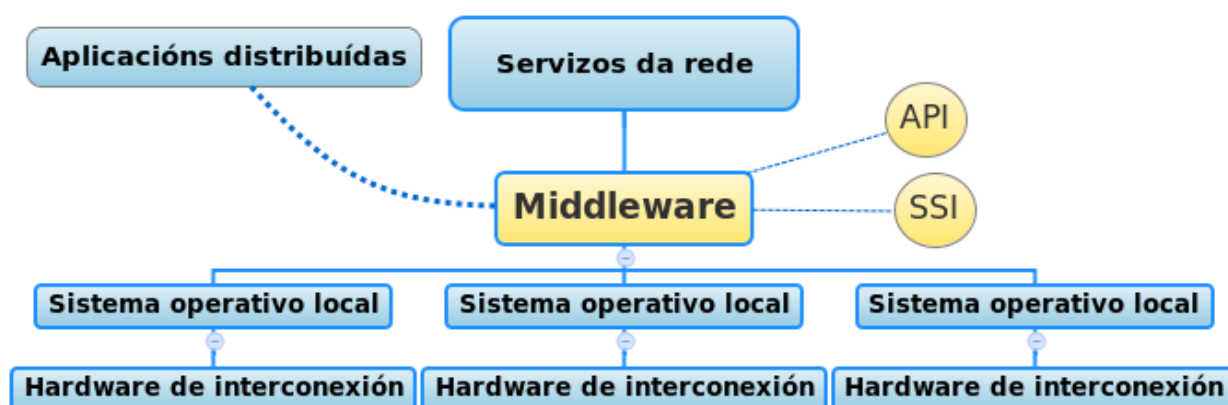
#### **26.2.3.1 Compoñentes intermedios: *Middleware***

Para dar o efecto de transparencia aos clientes, ese sistema require dunhas serie de compoñentes intermedios, é dicir que se atopan “polo medio” (en inglés *middleware*) das capas principais. Estes compoñentes se encargan de recibir e repartir as peticións dos clientes entre os servidores da granxa, coidar o balanceo de carga, o mantemento da sesión, etc... O ***middleware*** descríbese coma un condutor ou intermediario entre sistemas, dirixindo as peticións de datos e servizos a outros nodos da rede. Entre as súas principais características destacarían:

- a) Simplificar o desenvolvemento de aplicacións ao capsular comunicacións entre sistemas.



- b) Facilitar a interconexión dos sistemas de información con independencia da rede física.
- c) Mellorar a escalabilidade do sistema, aumentando a capacidade sen perda de funcionalidade.
- d) Mellorar a tolerancia a fallos do sistema, fiabilidade.
- e) Aumentar a complexidade de administración e soporte.



**Figura 2: O middleware nun sistema distribuído.**

Nun sistema distribuído o *middleware* é o software de conectividade que permite dispoñer dun conxunto de servizos sobre plataformas distribuídas heteroxéneas. Actúa coma unha capa de abstracción das funcións do sistema distribuído facendo transparente na rede os sistemas operativos e o hardware de interconexión das redes de comunicacións. Proporciona unha Interface de Programación de Aplicación (**API**) para a comunicación e acceso a aplicacións e servizos distribuídos. Por outra banda proporciona unha interface única de acceso ao sistema denominada **SSI** (do inglés *Single System Image*), a cal da ao cliente a sensación de acceder a un único servidor, o virtual.

Para garantir a heteroxeneidade na comunicación dos sistemas o *middleware* estruturase en tres **capas ou niveis de comunicación** separados:



- 1) **Protocolo de transporte.** Protocolos de comunicacións comúns á capa de transporte da rede, como TCP ou UDP. Establecen niveis de seguridade, control de sesións, etc...
- 2) **Sistema Operativo en Rede ou NOS** (en inglés *Network Operating System*). Extensión do sistema operativo dos clientes que captura as peticións e as dirixe cara o servidor axeitado para devolver a continuación a resposta do mesmo ao cliente.
- 3) **Protocolo de servizo.** Protocolo específico do servizo ou aplicación no sistema Cliente-Servidor.

Os middleware acostuman a clasificarse segundo o tipo de comunicación que realizan no Sistema Operativo en Rede e aos parámetros que comunican (infraestrutura, acceso a datos, aplicacións, etc...), os **tipos de middleware** máis habituais serían:

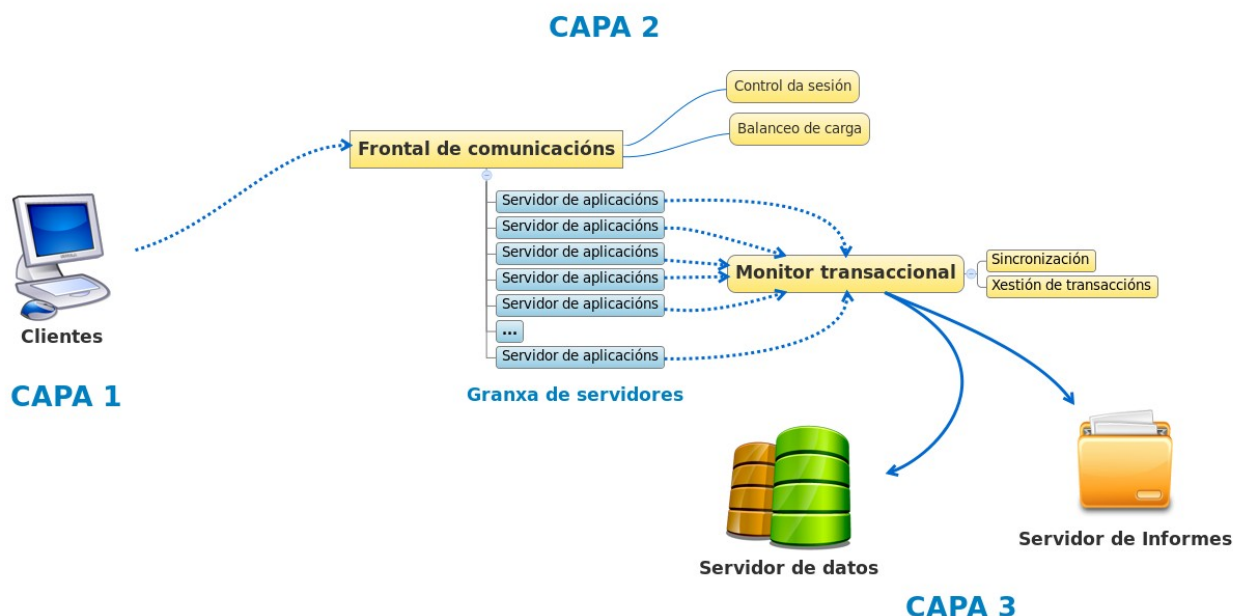
1. **Chamadas a procedementos remotos** (en inglés *Remote Procedure Call* ou RPC). Os Clientes invocan directamente procedementos ou funcións de procesos que se executan en servidores remotos, permitindo distribuír a lóxica da aplicación remota a través da rede. As chamadas poden realizarse de maneira asíncrona ou síncrona. Mantén ao mínimo a información da sesión e en caso de ruptura da mesma o cliente reinicia a comunicación de cero.
2. **Publicación/subscrición.** Este middleware realiza unha monitorización do sistema detectando os servizos e procesos activos. Os compoñentes rexistran o seu interese en determinados eventos en cando estes eventos son detectados polo monitor envía esa información aos subscritores. A interacción é asíncrona recaendo por completo no servizo de notificación/monitorización.
3. **Middleware orientado a mensaxes** (en inglés *Message Oriented*



*Middleware* ou MOM). A comunicación basease no envío de mensaxes asíncronos por parte dos nodos (cliente, servidor, servizo ou aplicación). As mensaxes recóllense en colas priorizadas no nodo destino e almacénanse ata que poden responderse. O funcionamento do sistema é análogo a como funciona un servizo de correo electrónico.

4. **Middleware baseado en obxectos** (en inglés *Object Request Broker* ou ORB). Incorpora a RPC os paradigmas de orientación a obxectos. Define unha arquitectura cliente servidor onde os servizos devolven obxectos, sendo estes a unidade de comunicación. Os nodos piden os obxectos polo nome sendo estes entregados por un servizo de resolución de nomes. Exemplos de implementacións deste *middleware* serían: CORBA, RMI, COM, .NET Remoting, etc...
5. **Middleware de acceso a datos** (en inglés *Oriented Data Access Middleware*). Proporcionan a API transparente de acceso a datos agrupando a operación de manexo da conexión coas bases de datos. Exemplos deste tipo de API serían JDBC e ODBC. Por norma xeral realizan conexións síncronas e operacións transaccionais.
6. **Arquitecturas orientadas a servizos** (en inglés *Service Oriented Architecture* ou SOA). As funcionalidades ou procedementos se publican dende calquera servidor a modo de servizos. Os servidores publican o servizo e permanecen á escoita ata que chega unha petición, a procesan e devolven unha resposta ao cliente do servizo. Exemplos deste *middleware* son os Servizos Web e os Servizos CORBA.





**Figura 3: Granxa de servidores.**

As granxas de servidores complétanse con dous compoñentes fundamentais, funcións que de xeito xeral realiza o *middleware*:

1. **O frontal de comunicacións.** O frontal de comunicacións (en inglés *front-end*) é o punto de acceso único á granxa de servidores, simulando un único servidor lóxico. Aínda que cada servidor da granxa poida ter o seu propio enderezo IP o normal é que o frontal teña un propio e sexa esta a forma de que os clientes accedan a el, ou ben directamente ou ben a través dun nome de dominio (DNS).
- ✓ **Balanceo de carga.** Consistir en dividir a carga de traballo dos clientes entre os servidores da granxa. Pode implementarse vía hardware, software ou unha combinación de ambas. Para facelo vía hardware o frontal de comunicacións debe dispoñer dun equipo específico, aínda que hai enrutadores que permiten esta funcionalidade.
  - ✓ **Control da sesión.** Se por mor do balanceo da carga diferentes



peticións dun mesmo cliente van cara servidores diferentes cómpre coordinar aos servidores no seguimento dunha sesión ou que poidan compartila.

- ✓ **Priorización.** En caso de ter peticións simultáneas o frontal debe ser capaz de atender primeiro aos clientes críticos ou de maior prioridade así como de asignar o procesamento das súas tarefas a aqueles servidores dotados de máis recursos.

2. **Os monitores transaccionais.** Os monitores transaccionais son os encargados de manter a consistencia dos datos e os procesos que se procesan simultaneamente nos servidores da granxa. Ten que garantir que unha modificación de datos froito dunha petición se realiza como unha transacción, é dicir, ou se realiza completamente ou non se realiza en absoluto. Cada transacción ten que ter lugar independente de que teña lugar outra simultánea, deben procesarse de xeito illado. As principais funcións serán:

- ✓ **A xestión das transaccións.** Encárgase de controlar a atomicidade e secuencialidade das transaccións, para garantir a consistencia de datos e operacións. A xestión de transaccións debe garantir o correcto funcionamento do sistema cando a carga de traballo ou o número de usuarios son moi elevados. Debe contemplar a posibilidade de erros nas aplicacións e caídas de elementos do sistema durante as transaccións, permitindo operación de volta atrás. Vista polo miúdo a xestión de transaccións constará de:

1. **Xestor de transaccións** (en inglés *Transaction Manager*). Controla o inicio de transacción, rexistra os recursos que precisa e xestiona as operacións de confirmación da transacción (en inglés *commit*) e de volta atrás e recuperación do estado inicial da transacción (en inglés



*rollback*).

2. **Xestor de rexistro** (en inglés *Log Manager*). Gardar os estado dos recursos que están uso por parte das transaccións, elaborando un historial de versións dos mesmos. Esta información é compartida polos distintos xestores de transaccións sendo o que permite garantir a consistencia dos recursos empregados.
  3. **Xestor de bloqueos** (en inglés *Lock Manager*). Xestiona o acceso simultáneo por parte de varios procesos aos recursos, permitindo bloquealos para evitar que dous ou máis accesos á vez dean lugar a inconsistencias. Así mesmo leva a cabo a detección de cando se libera un recurso e envía unha notificación ao xestor de transaccións.
- ✓ **Sincronización.** A sincronización das comunicacións resulta complexa neste modelo, xa que un cliente pode acceder a un servizo empregando diferentes servidores da capa intermedia, mesmo simultaneamente. Segundo o comportamento do servizo podemos atopar solucións síncronas, onde se espera sempre á resposta do servidor, simple pero con risco de bloqueo. Fronte as asíncronas onde se envía a petición e xa chegará a resposta, co cal non hai bloqueos, pero a resposta podería non chegar nunca sen máis opción que detectalo a través de tempos de espera esgotados.

#### **26.2.6. Arquitecturas en n-Capas.**

As arquitecturas en tres capas poden estenderse a n-capas cando na capa intermedia incorpóranse outros elementos de interconexión como distribuidores ou sistemas de devasa. Tamén pode dividirse a capa de servidores por servizos ou diferentes capas de acceso a datos ou presentación de información. A separación en capas é unha organización



lóxica do sistema co cal pode establecerse calquera número de capas segundo as necesidades do mesmo. Calquera especialización de servidores que se queira facer na rede e provoque un novo agrupamento podemos identificala cunha nova capa.

### **26.2.7. Arquitecturas para Rede entre iguais (P2P).**

Nos modelos distribuídos de igual a igual ou **P2P** (en inglés *Peer-to-Peer*), todos os equipos (agás os elementos de interconexión) teñen o mesmo rol dobre de cliente/servidor na rede. Fan uso de servizos e os proporcionan. Nesta arquitectura por tanto non se poden agrupar os nodos e perde sentido falar de capas. Estas redes non resultan óptimas para todo tipo de servizos, en moitos casos por exemplo á hora de funcionar como servidor web, requirirían un custe moi alto para control da consistencia do sitio web, mantemento e configuración do servidor, etc... por contra, en situacións onde os nodos caen a miúdo, por exemplo un servidor atacado continuamente, a redundancia de nodos garante que o sistema siga funcionando. Outros servizos como o intercambio de arquivos, ou o procesamento compartido presentan máis vantaxes á hora de empregar unha arquitectura deste tipo como solución de implantación. Os modelos máis habituais son centralizados, puros ou descentralizados ou híbridos, segundo o peso de cada nodo individual na rede ou da existencia de servidores con responsabilidade de control e xestión no modelo. A adaptación deste modelo por parte dos ISP da lugar a P2P híbridas de servizo denominadas P4P (en inglés *Proactive network Provider Participation for P2P*). Outros modelos similares serían os P2M, que actúan en arquitecturas híbridas empregando o correo electrónico como soporte do envío de datos.

A xestión deste tipo de redes realízase fundamentalmente vía software.



Neste caso o software deberá realizar as mesmas funcións xa vistas para a arquitectura de tres ou máis capas: frontal de comunicacións e xestión de transaccións. Por debaixo acostuman implementar servidores propios como Kademia, eDonkey, Gnutella, FastTrack, BitTorrent ou OpenNap entre outros.

## **26.2.8. SERVIDORES.**

### **26.2.8.1. Servidor web.**

Trátase de servidores que provén o servizo WWW a través do protocolo HTTP. En esencia trátase dunha aplicación executándose nun servidor á espera de peticións HTTP por parte dun cliente respondendo cos documentos solicitados xeralmente páxinas web e os obxectos que enlazan: imaxes, arquivos de *script*, animacións, etc... En funcións máis avanzadas estes servidores engaden seguridade a través de conexións encriptadas con protocolos tipo HTTP Seguro ou HTTPS. Por regra xeral, os servidores de aplicacións intégranse en arquitecturas de mínimo tres **capas**:

- 1) **Primeira capa.** Capa de interacción cos usuarios, principalmente a través de navegadores web.
- 2) **Capa intermedia.** Capa dos servidores web, que poden estar distribuídos nun modelo de granxa de servidores. Cada servidor incorporaría os módulos necesarios para seguridade, linguaxes de servidor interpretados, correo, mensaxería, acceso a datos e outras funcionalidades.
- 3) **Terceira capa.** Capa de servidores de acceso a datos, como servidores de arquivos, base de datos ou informes.

Entre os **servidores web de uso máis estendido** actualmente atoparíanse:



- ✓ **Apache.** Un dos máis utilizados, por se un servidor libre que ofrece prestacións a nivel doutras solucións propietarias ademais dunha grande facilidade de uso e configuración.
- ✓ **Internet Information Server (IIS).** Servidor propietario con soporte para aplicacións .NET ou ASP entre outras.
- ✓ **Outros:** Java Web Server, AOLServer, Cherokee, Tomcat, lightHttpd, etc...

Os servidores web poden dispoñer de módulos para a execución de programas de servidor interpretados, como son os das tecnoloxías Python, PHP, ASP, JSP, Tcl, ...

#### **26.2.8.2. Servidor de aplicacións.**

Os servidores de aplicacións son servidores web con capacidade de procesamento ampliada, podendo executar aplicacións e compoñentes de lóxica de negocio e recursos relacionados como o acceso a datos. Debido a isto permiten realizar o procesamento de aplicacións de cliente no propio servidor. Proporcionan soporte como *middleware* ou software de conectividade e para diferentes tecnoloxías de servidor. Por regra xeral, os servidores de aplicacións intégranse en arquitecturas de mínimo tres **capas**:

- 1) **Primeira capa.** Capa de interacción cos usuarios, principalmente a través de navegadores web.
- 2) **Capa intermedia.** Capa dos servidores de aplicacións, que poden estar distribuídos nun modelo de granxa de servidores. Un subconxunto dos servidores de aplicacións darán servizo aos usuarios/clientes mentres outro grupo encargárase de soportar a operativa común do dominio, como librarías ou aplicacións e servizos web dos que fagan uso as aplicacións para usuarios/clientes.
- 3) **Terceira capa.** Capa de servidores de acceso a datos, como



servidores de arquivos, base de datos ou informes.

O servidor de aplicacións acostuma ter integrado un servidor web, para xestionar de maneira independente o servizo WWW a través do protocolo HTTP.

Ademais deste servizo presenta un amplo conxunto de **ferramentas**:

- ✓ Servidor web integrado.
- ✓ Contedor de programas de servidor (en inglés *servlets*).
- ✓ Contedores de obxectos de lóxica de negocio (por exemplo EJBs).
- ✓ Sistemas de mensaxería.
- ✓ Software de conectividade con bases de datos.
- ✓ Balanceo de carga.
- ✓ Xestión de límites e colas de conexións (en inglés *Pool*) para bases de datos e obxectos.
- ✓ Etc...

En esencia un servidor de aplicacións realiza as mesmas funcións que un servidor web, pero cando a demanda de uso é grande e estamos ante un sistema complexo a solución pasa por empregar un servidor de aplicacións que ofrezca as seguintes **vantaxes**:

- ✓ **Centralización.** Centraliza nos servidores a administración e configuración da lóxica de negocio das aplicacións, de maneira que aspectos como o mantemento dos accesos a base de datos poden realizarse de xeito centralizado. Así mesmo cambios derivados de actualizacións, migracións ou recuperacións ante erros teñen lugar dende un único punto.
- ✓ **Seguridade.** Ao existir un único punto de acceso a datos pode reforzarse a defensa e os sistemas de control de erros nese punto, mellorando a súa xestión e protección.
- ✓ **Rendemento.** Como punto intermedio permite xestionar as peticións



dos clientes á Base de datos.

- ✓ **Escalabilidade.** Un mesmo servidor de aplicacións pode dar servizo a varios clientes, e por tanto aumentando o número de servidores mellórase o rendemento do sistema.

Entre os **servidores de uso máis estendido** actualmente atoparíanse:

- ✓ **Jboss, Glassfish.** Servidores de aplicacións libres baixo licenza GPL.
- ✓ **BEA Weblogic, IBM Websphere, Oracle Application Server.** Alternativas propietarias integradas en paquetes de aplicacións con funcionalidades de xestión e monitorización estendidas.
- ✓ **Tomcat, Internet Information Server (IIS), Jetty.** Proporcionan funcións parciais de servidores de aplicacións, co cal en ocasións se definen máis ben como contedores de programas de servidor.

### **26.2.8.3. Servidor de acceso a datos.**

Os servidores de acceso a datos ocuparían a última capa dos sistemas de información encargándose do acceso directo aos datos, existindo diferentes tipos segundo o sistema de información empregado para o seu almacenamento ou publicación:

- ✓ **Servidores de arquivos.** Neste tipo de servidores a información se almacena directamente en arquivos, por tanto a función destes equipos será a de permitir o acceso remoto aos mesmos dende os clientes ou outros servidores. Os protocolos máis habituais ofrecen servizo só dende redes locais pero en sistemas avanzados poden proporcionar servizos como FTP ou WebDAV para conexión remota a través de Internet. Actualmente o termo empregado para referirse a estes servidores é NAS (en inglés *Network-Attached Storage*), pero esta tan só sería a tecnoloxía máis habitual fronte a outras como DAS (en inglés *Direct Attached Storage*), baseada en SCSI ou SAN (en inglés *Storage Area Network*) baseada en fibra óptica. Non requiren



un software moi específico coma outros tipos de servidores senón máis ben soporte para diferentes protocolos e tecnoloxías. Por norma xeral acostuman a estar dipostos en [RAID](#) (en inglés *Redundant Arrays of Independent Disks*), equipos de almacenamento redundante.

- ✓ **Servidores de bases de datos.** Albergan un ou máis sistemas de xestión de bases de datos (en [inglés](#) *database management system*, ou DBMS), software de xestión que se encarga da comunicación entre as aplicacións e as bases de datos. Permiten realizar operacións de definición, manipulación e seguridade dos datos a través dunha API de comunicación coas aplicacións e un linguaxe estruturado de consulta como o SQL. Permiten accesos simultáneos aos datos, seguridade e xestión de transaccións.

Estes sistemas soen presentar ademais programas ou consolas de administración avanzadas para realizar as tarefas xerais de xestión da base de datos.

- ✓ **Servidores de informes.** Poden considerarse unha capa intermedia entre os servidores de datos e os de aplicación, onde se establecen servidores ou granxas de servidores que serven os datos en documentos predefinidos multiformato: follas de cálculo, PDF, XML, HTML, etc... O software deste tipo de servidores acostuma incorporar software de xestión para o servidor, e software de auto-edición de informes, para definir modelos de informes compostos de cabeceiras, imaxes, fórmulas, subinformes, etc... que se xerarán dinamicamente os informes a partir de consultas sobre os datos.



### CAPA DE USUARIO

### CAPA DE SERVIDOR DE APLICACIÓNS



### CAPA DE ACCESO A DATOS

**Figura 4: Arquitectura en 3 capas con servidor web, de aplicacións e base de datos.**

Entre os **servidores de uso máis estendido** actualmente atopáanse:

- ✓ **Servidores de arquivos.** Non requiren xestores especializados pero si soporte software aos protocolos: CIFS, NFS, SMB, FTP, WebDAV, etc... Así como utilidades tipo Samba ou FreeNAS.
- ✓ **Servidores de bases de datos.**
  - ✓ De licenza libre: PostgreSQL, MariaDB, Firebird, SQLite, Apache derby, ...
  - ✓ Dual, dependendo do seu 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 DO CLIENTE.

Os *scripts* do cliente son programas interpretados deseñados para executarse nos navegadores co obxectivo de dotar ás páxinas de maior interactividade co usuario e dinamismo nunha aproximación ás aplicacións de escritorio. O **funcionamento básico** dun *script* consiste en interpretar unha serie de comandos a través dos cales pode modificar e manipular obxectos e reaccionar ante eventos da interface como respostas a periféricos (rato, teclado, etc...), ou cambios nos elementos do documento (botóns, elementos de formularios, etc...). Os seus usos básicos son validacións, manipulación de formularios, procesamento de funcións e carga asíncrona de datos.

Os *scripts* de cliente proporcionan as seguintes **vantaxes**:

- ✓ Modificar o contido da páxina sen recargala do servidor en función das interaccións co usuario.
- ✓ Modificar parámetros de configuración do navegador e outros elementos da páxina web.
- ✓ Mellorar a interacción entre o usuario e o documento, en xeral a usabilidade.

Por contra, presentan un serie de inconvenientes ou **desvantaxes**:

- ✓ Problemas de accesibilidade, pois complícase a posibilidade de presentar alternativas a usuarios que non soporten a tecnoloxía de *script*.
- ✓ Problemas de seguridade, pois toda a lóxica de interacción aparece sen protección descargada no equipo do usuario, co cal dispón do código fonte do programa de *script*.

As **tecnoloxías de *script*** máis empregadas son Visual Basic Script,



Javascript, coa súa evolución AJAX e PerlScript. O principal problema á hora de seleccionar unha tecnoloxía cando se diseña unha páxina web é o soporte que recibirá por parte dos navegadores, pois hai que lembrar que as linguaxes de *script* serán en última instancia interpretados no navegador.

- a) **Javascript.** Baseado na linguaxe Java, é unha das linguaxes de script de uso máis estendido. É de sinalar, que ademais ten aplicación con outras tecnoloxías ademais da web coma en documentos PDF ou aplicacións de escritorio. Para permitir a interacción cos elementos dun documento web esta linguaxe dispón dunha API que implementa o DOM (en inglés *Document Object Model*) ou Modelo de Obxectos para a Representación do Documento, estandarizado polo W3C (en inglés *World Wide Web Consortium*). Nun intento por estandarizar esta linguaxe xorde o ECMAScript unha especificación da linguaxe aceptada como estándar ISO,
- b) **Visual Basic Script.** Similar ao Javascript no tocante a funcionamento e estrutura pero baseado en Visual Basic. Ten menor soporte dentro dos diferentes navegadores, agás no Internet Explorer.
- c) **PerlScript.** Baseado en linguaxe C o seu uso non está tan estendido coma as tecnoloxías anteriores aínda que ten menos limitacións. Debido a isto foi derivando cara linguaxe de servidor.
- d) **AJAX.** Acrónimo de Javascript Asíncrono e XML (en inglés *Asynchronous Javascript And XML*). Esta é unha tecnoloxía de *script* de cliente asíncrona, de xeito que pode realizar cargas de datos sen que afecten á recarga da páxina. AJAX é un conxunto de tecnoloxías que fai uso de:
  - 1) XHTML e follas de estilo en fervenza (CSS) para a estrutura e deseño dos contidos.
  - 2) A linguaxe Javascript como linguaxe de programación para funcións e definición do programa cliente.

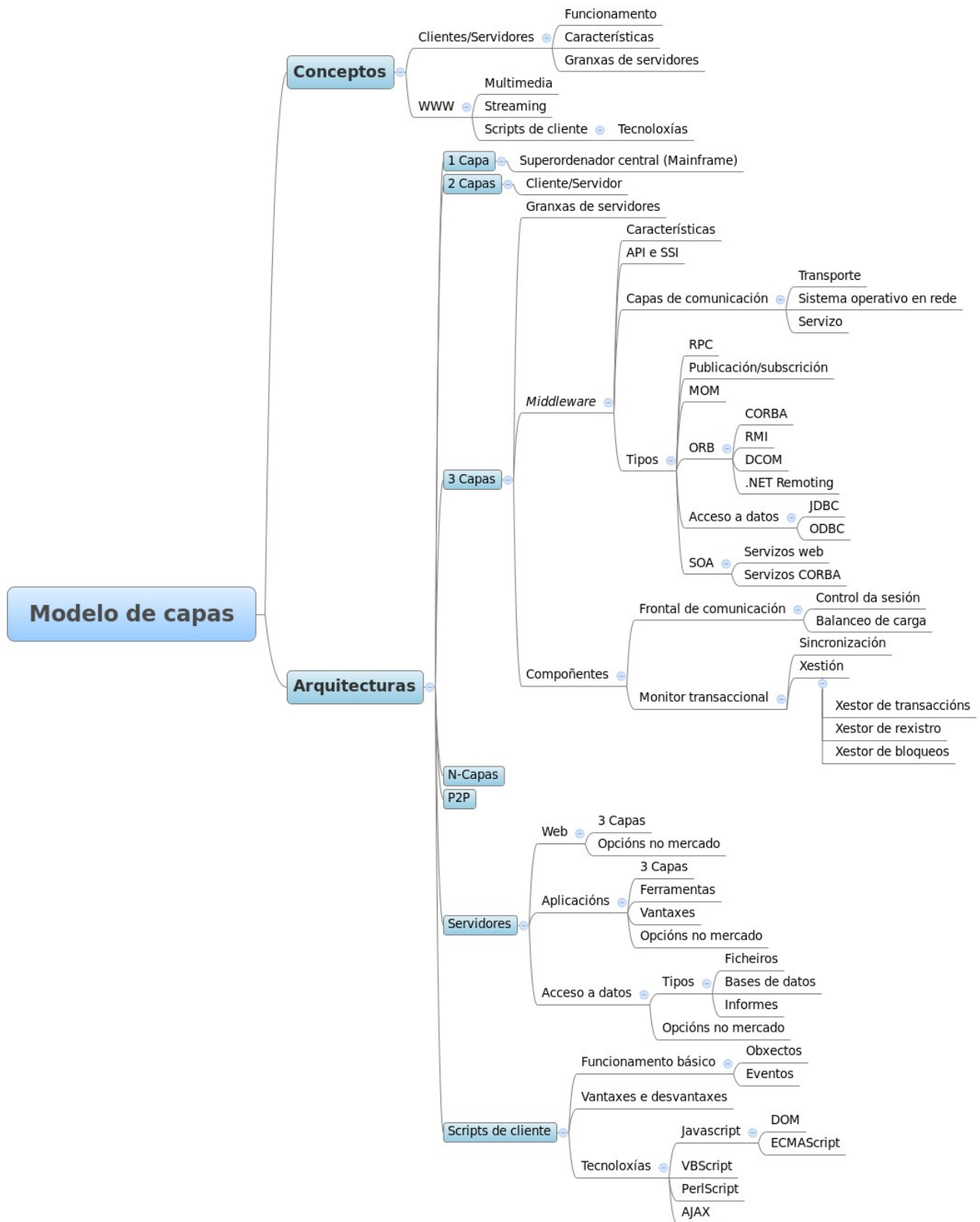


- 3) O obxecto *XMLHttpRequest* para intercambio de información asíncrona co servidor. Por tanto, cómpre que o navegador soporte este obxecto, sendo empregado en ocasións o obxecto *Iframe*.
- 4) XML e DOM como estándares asociados para intercambio de datos e manipulación do documento.



#### **26.4. ESQUEMA**







## **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 E CONCEPTOS**

### **27.2 ARQUITECTURA WEB EN .NET**

### **27.3 ARQUITECTURA WEB EN J2EE**

### **27.4. ESQUEMA**

### **27.5. REFERENCIAS**

### **27.1. INTRODUCCIÓN E CONCEPTOS**

O desenvolvemento de aplicacións, servizos web e outros compoñentes software ten hoxe en día dúas vertentes principais, a plataforma .NET e a plataforma J2EE, ou JEE nome co que é coñecida actualmente ao cambiar de versión. A rivalidade entre estas dúas tecnoloxías é moi forte, pois proporcionan solucións similares fortemente soportadas polas compañías dun e do outro bando.

**.NET** é a plataforma de desenvolvemento proposta pola empresa Microsoft para o mundo dos servidores de aplicacións que funciona como ferramenta de deseño e programación ademais de proporcionar un amplo conxunto de utilidades estendidas de apoio ao desenvolvemento neste tipo de contornos. Pola súa banda **JEE**, (en inglés *Java Platform, Enterprise Edition*) ou Java EE, é unha evolución da plataforma Java para desenvolver e soportar compoñentes software segundo un conxunto de especificacións de xeito que poidan operar nun servidor de aplicacións, incluíndo tamén ferramentas de deseño e programación.

A pesar de que ambas plataformas perseguen o mesmo obxectivo, teñen unha serie de particularidades ou diferenzas que se ven acentuadas polas guerras comerciais entre as empresas que as soportan. Mentres JEE ten



soporte multiplataforma, .NET funciona unicamente baixo a familia de Sistemas Operativos Windows. Mentres JEE basease exclusivamente na linguaxe Java, en .NET permítense moitos linguaxes de alto nivel, aínda que na práctica os principais sexan C# e VB .NET. JEE leva máis anos de experiencia no mercado, mentres que .NET é máis recente. Así mesmo JEE presenta maior soporte en canto a solucións e posibilidades de software libre, que son moi escasas e de pouca calidade en .NET. Con JEE pódese instalar unha infraestrutura de alto rendemento de xeito completamente gratuíto.

## **27.2 ARQUITECTURA WEB EN .NET**

**.NET** é, segundo a empresa Microsoft, unha plataforma para o desenvolvemento de servidores, clientes e servizos. Representa un conxunto de tecnoloxías que teñen como núcleo principal o .NET Framework, un marco de desenvolvemento e compoñente software que pode instalarse en Sistemas Operativos da familia Windows (Windows 2003, Vista, Windows 7, ...). Existe unha versión adaptada para móbiles dispoñible en Windows Mobile. A norma ISO/IEC 23271 recolle un conxunto funcional mínimo que deben cumprir os produtos software desenvolvidos para que poidan funcionar dentro do marco de traballo. Esta e máis normas se recollen nos estándares:

- a) **Estándar ECMA-334.** Especificación da linguaxe C#. (2006).
- b) **Estándar ECMA-335.** Especificación da linguaxe de infraestrutura común (CLI). (2010).

Por contra, outros compoñentes coma ASP .NET, *Windows Forms* ou ADO .NET non se atopan estandarizados. Paralelamente, unha vez publicados os documentos de especificación da arquitectura .NET apareceu o **Proxecto Mono** co obxectivo de implementar o marco de traballo .NET Framework empregando código aberto, para a partir de aí desenvolver aplicacións para sistemas UNIX/Linux.



### 27.2.1 .NET Framework

Marco de traballo que proporciona o conxunto de ferramentas e servizos para o desenvolvemento de compoñentes software, aplicacións, servidores e servizos web. Pode dividirse en tres bloques principais:

- 1) O **Contorno de Execución Común** (en inglés *Common Language Runtime* ou CLR). Encárgase da xestión de código en execución, control de memoria, seguridade e o outras funcións relacionadas co Sistema Operativo.
- 2) A **Biblioteca de Clases Base** (en inglés *.NET Framework Base Classes*). Realizan a función de API de servizos a disposición dos desenvolvedores para tarefas como xestión de ficheiros, mensaxería, procesos en varios fíos, acceso a datos, encriptación, etc...
- 3) **Control de acceso a datos**, que permite realizar as operacións de acceso a datos a través de clases e obxectos do *framework* incluídos no compoñente ADO.NET.
- 4) O **Motor de Xeración da Interface de Usuario**, que permite crear interfaces para aplicacións de escritorio ou web empregando compoñentes específicos coma ASP.NET para web, *Web forms* para aplicacións de escritorio ou *Web services* para servizos web.

### 27.2.2 Contorno de Execución Común (CLR)

O Contorno de execución común ou CLR é o encargado de xestionar o código en tempo de execución. De xeito análogo á Máquina virtual de Java este contorno permite executar aplicacións e servizos web ou de escritorio en calquera cliente ou servidor que dispoña deste software. A diferenza da Máquina virtual de Java o soporte de .NET é multilinguaxe permitindo C++, C#, ASP .NET, Visual Basic, Delphi, e moitos outros. Ademais permite integrar e herdar compoñentes entre diferentes linguaxes, con maior ou menor fortuna á hora de sacar proveito de linguaxes antigas.



O contorno de desenvolvemento compila o código fonte en calquera das linguaxes soportadas a un código intermedio denominado **CIL** (en inglés *Common Intermediate Language*) de maneira análoga ao BYTECODE de Java. A esta linguaxe intermedia chégase empregando a especificación CLS (en inglés *Common Language Specification*) onde se especifican unhas regras necesarias para crear o código intermedio CIL compatible co CLR. Así mesmo, o CLR dispón de compiladores coma JIT (en inglés *Just In Time*) ou AOT (en inglés *Ahead Of Time*) adaptados a cada linguaxe.

**JIT** xera o código máquina real en cada máquina a partir dese código intermedio conseguindo independencia do hardware. Esta compilación faise en tempo de execución a medida que a aplicación ou servizo invoca métodos ou funcións. Para axilizar o procesamento este código máquina obtido en tempo de execución gárdase na memoria caché actualizándose tan só cando se produce algún cambio no código fonte, momento no se que se repite o proceso. Por contra **AOT**, compila o código antes de executarse co cal logra un maior rendemento en execución pero menos independencia da plataforma. No tocante a JIT acostuma a distinguirse entre:

- 1) **Jitter estándar.** Compila o código CIL a nativo baixo demanda.
- 2) **Jitter económico.** Non optimiza, traduce cada instrución así precisa menos tempo e memoria de compilación.
- 3) **Prejitter.** Realiza unha compilación estática dun compoñente software completo.

As principais **vantaxes** deste modelo de compilación son:

- ✓ A **reutilización** de compoñentes escritos en diferentes linguaxes nunha mesma aplicación ou servizo web.
  
- ✓ **Modularidade** grazas á implementación do patrón Interface para cada compoñente ou librería xa que será accesible dende calquera



linguaxe a través da súa API (ASP, C#, Java, Python, etc ...)

- ✓ **Integración multilinguaxe**, xa que cada linguaxe cun compilador a CIL pode integrarse na plataforma co cal cada compoñente nesa linguaxe pode integrarse unha aplicación ou servizo web .NET.
- ✓ **Seguridade**. Polo illamento do código de usuario respecto dos accesos a datos e outras partes críticas do Sistema Operativo.



**Figura 1: Estrutura multilinguaxe do CLR**

O CLR cumpre ademais a función de proporcionar unha ampla gama de servizos ás aplicacións. A través da API de cada servizo os compoñentes web poden ter acceso a funcionalidades comúns, como:

- a) **Seguridade acceso ao código ou CAS** (en inglés *Code Access Security*). Controla que tipo de operacións pode realizar un código segundo se identifique na sinatura do ensamblado ou na orixe do código. Así mesmo recoñece directivas de administración do sistema para compoñentes ou a nivel de *host*. Cando un compoñente trate de acceder a recursos protexidos do sistemas lanzarase o CAS para comprobar os permisos, pero a este nivel non se poden establecer comprobacións dinámicas, por exemplo contra Bases de datos.
- b) **Atributos de protección do *host* ou HPA** (en inglés *Host Protection Attributes*). Mantén unha lista de atributos protexidos, denegando o acceso ou modificación dos mesmos. Algúns destes

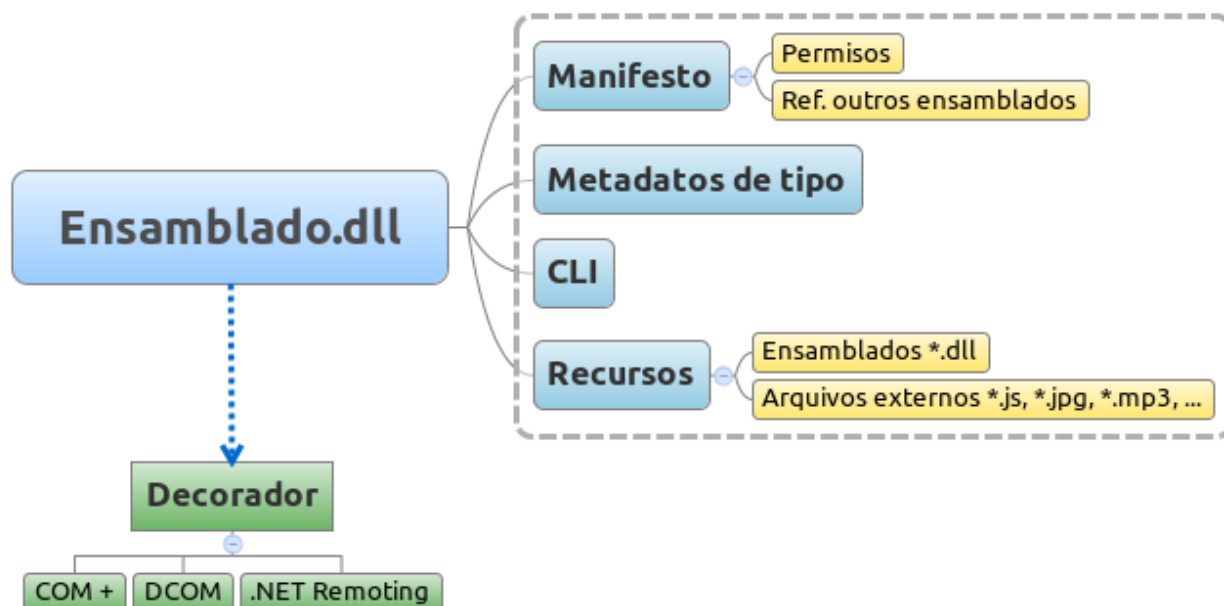


atributos serían *SharedState*, para estados compartidos, *Synchronization*, para permitir a capacidade de sincronizar procesos no *host* ou *ExternalProcessmgmt* que indica se os procesos no *host* se poden controlar externamente a través da API.

- c) **Dominios de aplicación.** Definen dominios illados de código para restrinxir o acceso dos compoñentes software, creando unha zona reservada para un subproceso. Por norma xeral o CLR crea para cada aplicación un dominio en tempo de execución pero pode precisar dominios específicos para compoñentes DLL ou externos.
- d) **Comprobación da seguridade de tipos.** Reserva espazos de memoria para cada obxecto segundo o especificado para o seu tipo. Cando o espazo é aleatorio ou queda algún oco fóra do espazo do obxecto, quere dicir que ese obxecto non ten seguridade de tipos. Coa compilación JIT realízase unha comprobación en tempo de execución para verificar se cada obxecto ten seguridade de tipos. Do mesmo xeito impide variables sen valores iniciais ou *cast* non seguros.
- e) **Cargador de clases.** Permite cargar en memoria clases e tipos de datos a partir da interpretación dos metadatos. Existe ademais a posibilidade de crear cargadores personalizados, aínda que só para Java, xa que por motivos de rendemento resulta máis óptimo empregar un ensamblado con outras linguaxes. Na compilación o cargador realiza a función de evitar código innecesario a través de funcións *stubs* que substitúe co código correcto baixo demanda.
- f) **Recolección de lixo** (en inglés *Garbage Collector*). Este servizo execútase de xeito continuo para buscar e eliminar de memoria os obxectos que non sexan referenciados ou rematen o tempo de espera de utilización.



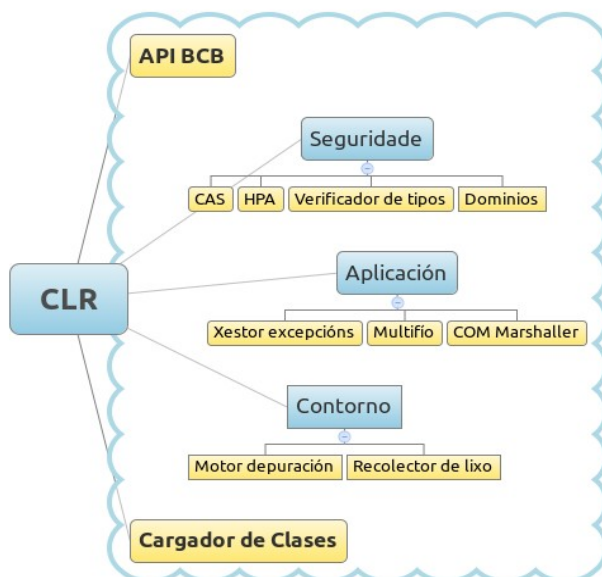
- g) **Motor de interacción COM.** Realiza funcións de conversión de datos e mensaxes ou *marshaling* dende e cara obxectos COM, o que permite a integración con aplicación *Legacy*.
- h) **Motor de depuración.** Permite realizar un seguimento da execución do código aínda que mesture diferentes linguaxes.
- i) **API multifío** (en inglés *multithread*). Proporciona unha API e as clases necesarias para xestionar a execución de fíos paralelos.
- j) **Xestor de excepcións.** Realiza a xestión estruturada e integración con *Windows Structured Exception Handling* de excepcións aínda que o erro proveña de diferentes linguaxes nun mesmo compoñente e mesmo no código aínda non executado. Este código pode incluír excepcións SHE do tipo C++ ou resultados HRESULTS típicos de COM.
- k) **API da Biblioteca de Clases Base (BCB).** Interface coa BCB do marco de traballo que realiza a integración do código co motor de execución.



**Figura 2: Ensamblados.**



No .NET Framework cando se compila un programa ou aplicación web xérase un arquivo denominado **ensamblado** que contén o código compilado á linguaxe intermedia CLI e un manifesto con permisos e referencias a outros ensamblados, compoñentes software ou servizos web. Son paquetes ou librarías EXE ou DLL destinadas ao control de versións, seguridade e comprobacións de implementación polo miúdo. Os ensamblados levan unha indicación que define o contorno de execución no que se debe lanzar: COM+, DCOM, .NET Remoting, ...



**Figura 3: Servizos do CLR**

### 27.2.3 Biblioteca de Clases Base (BCB)

A Biblioteca de Clases Base é unha API de alto nivel para permitir acceder aos servizos que ofrece o CLR a través de obxectos nunha xerarquía denominada **espazo de nomes**. Agrupa as funcionalidades de uso frecuente permitindo a súa redefinición. Atópase implementada en CIL polo que pode integrarse en calquera outra linguaxe. É un conxunto de clases, interfaces e tipos valor que son a base sobre as que se crearán as aplicacións, compoñentes e controis do .NET Framework. Permite realizar operacións como: soporte para diferentes idiomas, xeración de números



aleatorios, manipulación de gráficos e imaxes, operacións sobre datas e outros tipos de datos, integración con APIS antigas, operacións de compilación de código adaptada ás diferentes linguaxes de .NET, elementos para interfaces de usuario, tratamento de excepcións, acceso a datos, encriptación, administración de memoria, control de procesos, etc...

Espazo de nomes	Utilidade e obxectos
<b>System</b>	Tipos básicos, táboas, excepcións, datas, recolector de lixo, etc...
<b>System.Collections</b>	Manipulación de coleccións como pilas, colas, <i>hash</i> , etc...
<b>System.Data</b>	Arquitectura ADO.NET (Obxectos <i>DataSet</i> , <i>DataTable</i> , <i>DataRow</i> , <i>DataView</i> , ...)
<b>System.IO</b>	Manipulación de E/S arquivos e outras orixes de datos
<b>System.Net</b>	Xestión de comunicacións de rede (TCP/IP, <i>Sockets</i> , ...)
<b>System.Security</b>	Xestión das políticas de seguridade do CLR
<b>System.XML</b>	Acceso e manipulación de datos en documentos XML con compatibilidade co W3C (Transformacións en <i>System.Xml.Xls</i> e serialización para servizos web en <i>System.XML.Serialization</i> )
<b>System.Web</b>	Servizos para xestión de caché, seguridade e configuración para Servizos Web, estado das sesións e interfaces de usuario
<b>System.Web.Services</b>	Xestión dos requirimentos de Servizos Web
<b>System.Web.UI</b>	Controles para interfaces de usuario <i>HTMLControl</i> para mapeo de etiquetas HTML e <i>WebControl</i> para estruturar controis de usuario avanzados coma <i>DataGrids</i>
<b>System.Windows.Forms</b>	Creación da IU do cliente
<b>System.Drawing</b>	Acceso a funcionalidades gráficas básicas da GDI+ (Funcionalidades avanzadas en <i>System.Drawing.Imaging</i> , <i>System.Drawing.Text</i> e <i>System.Drawing.Drawing2D</i> )
<b>System.Reflection</b>	Acceso a metadatos sobre os ensamblados, módulos, membros, parámetros e outras entidades do código



	administrado
<b>System.JSON</b>	Proporciona compatibilidade baseada en estándares JSON, notación de objetos JavaScript (en inglés <i>JavaScript Object Notation</i> )
<b>System.Threading</b>	Manipulación de procesos e fíos de execución
<b>System.Text</b>	Proporciona clases para manipular a codificación de caracteres UNICODE e UTF-8 conversión de bloques de caracteres en bloques de <i>bytes</i> e viceversa
<b>System.Transactions</b>	Contén clases que permiten crear e administrar transaccións, admitindo participantes distribuídos, notificacións de fase e inscricións duradeiras
<b>System.Resources</b>	Proporciona clases e interfaces que permiten crear, almacenar e administrar recursos de localización
<b>System.Runtime.Remoting</b>	Proporciona a interface para acceso remoto e marco para a implantación de sistemas de compoñentes distribuídos
<b>Microsoft.CSharp</b>	Clases para realizar a compilación e execución de código en C# (O mesmo para outras linguaxes)

**Táboa 1: Principais espazos de nomes.**

#### **27.2.4 Control de acceso a datos.**

O control de acceso a datos, documentos XML e servizos de datos no marco .NET Framework recóllese na arquitectura ADO .NET, coma evolución do *ActiveX Data Objects*. A súa orientación principal é o acceso a datos do xestor de base de datos relacional *SQL Server*, orixes XML e orixes de datos vía obxectos OLE DB e ODBC. As **conexións** realízanse identificando os provedores de datos a través dos obxectos (Connection, Command, DataReader e DataAdapter). Unha vez establecida a conexión entra en escena o principal elemento do marco, o *Dataset* que recolle os **resultados** cargados a partir dunha orixe. Á súa vez pode particularizarse con outros elementos da base de datos con obxectos coma: *DataTable*, *DataRow*, *DataColumn* ou *Constraint*. Os **obxectivos** de deseño principais deste marco son:



- ✓ Soporte á tecnoloxía ADO previa.
- ✓ Integración con tecnoloxías baseadas en XML.
- ✓ Soporte a modelos de arquitectura multicapa.

Nas últimas versións incorpórase o **Marco de Entidades** (en inglés *Entity Framework*) que permite realizar Consultas Integradas nas Linguaxes (en inglés *Language Integrated Query*) ou **LINQ**.

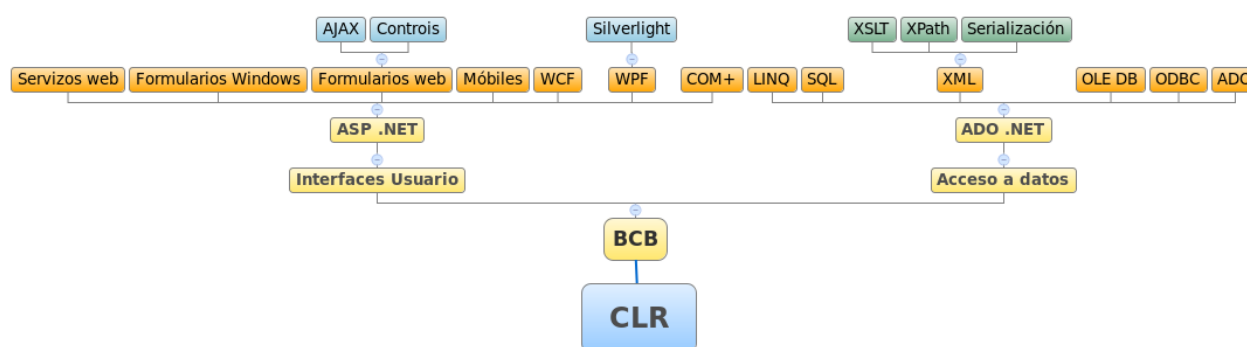
Este marco permitirá empregar LINQ sobre moitos compoñentes de acceso a datos novos: *LINQ to SQL*, *LINQ to DataSet* e *LINQ to Entities*. Asocia unha chave lóxica ás entidades, dotando ao modelo relacional ou conceptual de orientación a obxectos. Utilidades do marco de traballo coma **SQLMetal** permiten a xeración automática de clases a partir da base de datos ou documentos XML.

#### **27.2.5 Motor de Xeración de Interfaces de Usuario**

A parte do marco de traballo encargada da xeración de interfaces de usuario e servizos web sería a arquitectura ASP .NET. O conxunto de clases correspondentes agrúpanse nos espazos de nomes: System.Web, System.Web.Services, System.WebUI. As páxinas web desenvolvidas con ASP .NET teñen a extensión **ASPX** e son coñecidas con **Formularios Web** (en inglés *Web Forms*). Paralelamente a esta tecnoloxía de presentación atoparíamos **Formularios Windows** (en inglés *Windows Forms*) para aplicacións de escritorio e tecnoloxías para Móviles, sendo a primeira a máis empregada para aplicacións de servidor. No espazo System.Web.UI recóllense as dúas clases principais de controis os HTML para acceso directo ás etiquetas estáticas destas linguaxes e os controis web que incorporan código de servidor dinámico. A arquitectura recomendada emprega o modelo **Code-Behind** no que se crea un arquivo separado co código de servidor, a diferenza da arquitectura DNA para ASP anterior. Os



**Controis de usuario** (en inglés *User Controls*) seguen a mesma estrutura que os Formularios Web, pero derivan do espazo `System.Web.UI.UserControl`, e gárdanse en arquivos **ASCX**, que deberían seguir tamén o modelo Code-Behind. O marco incorpora tamén elementos para control do estado e a sesión así coma outros para seguridade, autenticación de usuarios, uso de roles, uso do servizo Indigo ou **WCF** (en inglés *Windows Communication Foundation*). Así mesmo, permite a integración con AJAX, a través da incorporación dun **Toolkit** na aplicación, ou **WPF** (en inglés *Windows Presentation Foundation*) baseado en XALM e marco para Silverlight.



**Figura 4: Arquitectura xeral**

### 27.2.6 Niveis lóxicos.

Nos modelos e solucións que propón esta estrutura establécense unha serie de servizos xenéricos presentes na maioría de aplicacións corporativas actuais. Esta división permite definir un deseño e unha arquitectura específicos para cada nivel, facilitando o desenvolvemento e soporte da aplicación.

1. **Servizos de usuarios.** Atópanse na primeira liña de interacción cos usuarios e proporcionan a interface de acceso ao sistema que deriva en chamadas aos compoñentes do nivel de Servizos corporativos. En ocasións considéranse dentro deste nivel procesos fóra das interfaces



de usuario, como procedementos de control ou automatizados que non requiren a presenza dun usuario.

2. **Servizos corporativos.** Encapsulan a lóxica corporativa proporcionando unha API das funcionalidades básicas do sistema. Isto permite abstraer os servizos de usuario da lóxica corporativa e manter diferentes servizos de usuario a partir das mesmas funcionalidades. Cada funcionalidade pode precisar dispoñer de varios servizos corporativos.
3. **Servizos de datos.** Sería a parte máis illada do usuario, proporcionando o acceso a datos e a outros sistemas ou servidores. Establecen diferentes API xenéricas das que poden facer uso os Servizos corporativos. Conteñen unha ampla gama de orixes de datos e sistemas de servidor, encapsulando regras de acceso e formatos de datos.

### **27.2.6 Solucións de integración.**

Coa tecnoloxía .NET existen tres principais plantexamentos de solucións arquitectónicas:

- 1) **SOA** (en inglés *Service Oriented Architecture*). Entende a comunicación entre aplicacións e compoñentes coma servizos, non necesariamente servizos web, demandados por clientes ou subscritores e proporcionados e publicados por provedores.
- 2) **MOA** (en inglés *Message Oriented Architecture*). A comunicación realízase por paso de mensaxes forzando un modelo SOA distribuído.
- 3) **EAI** (en inglés *Enterprise Integration Application*). Especifica unha serie de requirimentos de integración e comunicación en sistemas regulados polos patróns de integración Mediación e Federación, onde un sistema EAI fai funcións de *Hub* ou *bus* de comunicacións.

## **27.3 ARQUITECTURA WEB EN J2EE**



JEE representa un conxunto de especificacións para plataformas de desenvolvemento baseadas en linguaxe Java para servidores de aplicacións en arquitecturas de múltiples capas. O **JCP** (en inglés *Java Community Process*) é o organismo encargado de validar os requisitos de conformidade para cada plataforma e aceptala. As plataformas JEE constan dos seguintes compoñentes:

- 1) Un conxunto de especificacións.
- 2) Un test de compatibilidade ou CTS (en inglés *Compatibility Test Suite*)
- 3) Unha implementación de referencia para cada especificación.
- 4) Un conxunto de guías de desenvolvemento e boas prácticas denominadas JEE Blueprints.

As principais **especificacións** que inclúe JEE dan soporte a Servizos web, RPC baseado en XML, Mensaxería XML, despregues, servizos de autorización, conexión remota RMI, JSP, JSF, JSTL, Servlets, Portlets, Applets, JavaBeans, esquemas XML, acceso a datos JDBC, documentación Javadoc, transformacións XSL, etc...

O soporte multiplataforma do Java ten a súa base na **Maquina Virtual** (VM/JVM ou KVM/CVM para móbiles), unha plataforma lóxica capaz de instalarse en equipos con diferente hardware e Sistema operativo e interpretar e executar instrucións de código **Java bytecode**. A especificación da VM tamén se recolle como especificación pola JCP e do mesmo xeito están dispoñibles test de compatibilidade. A forma máis habitual para a VM é mediante un compilador JIT pero tamén permite interpretación. Do mesmo xeito permítese execución segura mediante o modelo das Java Applets. Programas de cliente que se executan nunha VM dentro do navegador logo de descargar vía HTTP código do servidor, que se executa nunha *Sandbox* moi restrinxida.



Os compoñentes software web e de negocio dentro desta tecnoloxía despréganse a través de **Contedores** (en inglés *containers*). Os contedores son implementacións de arquitecturas JEE que proporcionan os servizos do servidor de aplicacións aos compoñentes, incluíndo seguridade, acceso a datos, manexo de transaccións, acceso a recursos, control de estados, xestión do ciclo de vida e comunicacións entre outros. Antes de executarse un compoñente software debe configurarse coma un servizo JEE e despregarse dentro dun contedor. Os principais serían os contedores web para Servlets e JSP, os contedores EJB para compoñentes da lóxica de negocio, e contedores de aplicacións cliente e contedores de *Applets* para os programas de cliente e código de cliente para o navegador respectivamente.

Os principais **servizos** que proporciona JEE xunto coas súas respectivas API serían:

- 1) **HTTP e HTTPS**. Para control das comunicacións web e SSL a través destes protocolos. As API de servidor veñen dadas polos paquetes de clases Servlets e JSP e a de clientes no paquete Java.Net.
- 2) **JDBC** (en inglés *Java Data Base Connection*). API de acceso a datos en sistemas xestores de bases de datos relacionais vía SQL. Por un lado aporta a interface para ser empregada polos compoñentes software e por outra a interface para que os provedores poidan desenvolver os controladores específicos. As versións máis recentes son as JDBC 3.0 e 4.0, que inclúen os paquetes *java.sql* e *javax.sql*.
- 3) **JSTL** (en inglés *Java Server Pages Standard Tag Library*). Proporciona as funcionalidades de para etiquetas nas páxinas JSP.
- 4) **RMI-IIOP** (en inglés *Remote Method Invocation-Internet Inter-ORB Protocol*). Proporciona a API para permitir comunicacións en aplicación distribuídas a través de JAVA RMI, por exemplo para acceder a compoñentes EJB. Os protocolos máis habituais son JRMP, de RMI e IIOP, de CORBA.



- 5) **IDL** (en inglés *Java Interface Definition Language*). Permite a comunicación de clientes con servizos CORBA a través do protocolo IIOP, servizos SOAP ou RPC.
- 6) **JNDI** (en inglés *Java Naming and Directory Interface*). Proporciona o servizo de nomes e directorios, indicando o contexto de cada obxecto e as relacións entre eles. Divídese en dúas interfaces, a API de programación e unha SPI que permite conectar con provedores de servizos de nomes e directorios sendo os principais LDAP, CORBA e RMI.
- 7) **JAXP** (en inglés *Java API for XML Processing*). Soporta o procesamento de documentos XML que cumpra cos esquemas do W3C a través de DOM, SAX e XSLT.
- 8) **JMS** (en inglés *Java Message Service*). Proporciona a API de envío de mensaxes para comunicarse cun MOM (en inglés *Message-Oriented Middleware*), unha abstracción independente do provedor para comunicacións entre sistemas.
- 9) **JavaMail**. Proporciona a interface para controlar o envío e recepción de correos electrónicos. Pode soportar o formato MIME grazas á súa integración con marco de traballo JAF.
- 10) **JAF** (en inglés *Java Beans Activation Framework*). API que proporciona o marco de traballo para activación que soporta as peticións doutros paquetes.
- 11) **JTA** (en inglés *Java Transaction API*). Orientada cara o manexo de transaccións e a permitir a comunicación entre contedor e compoñentes do servidor de aplicacións coma os monitores transaccionais e os administradores de recursos.
- 12) **JAX-RPC** (en inglés *Java API for XML-based RPC*). Proporciona soporte para comunicacións remotas de tipo RPC entre clientes e servizos web cos estándares HTTP e SOAP. Soporta outros estándares coma



WSDL, así coma SSL e TTL para autenticación. O SAAJ (en inglés *SOAP with attachments API for Java*) engade a posibilidade de arquivos ou notas achegados coas mensaxes.

Cada compoñente denomínase **Módulo** JEE de xeito que unha aplicación estará formada por un conxunto de módulos sendo cada un un compoñente para un contedor. Existen tres tipos de módulos:

- 1) **Arquivos JAR** (en inglés *Java Archive*). Agrupación de arquivos Java e recursos segundo o formato ZIP. Empaquetan compoñentes EJB segundo a estrutura de directorios do código, engadindo unha carpeta especial, META-INF, con metadatos.
- 2) **Arquivos WAR** (en inglés *Web Application Archive*). Agrupan nun único arquivo unha aplicación web, incluíndo Servlets, arquivos JSP, contido estático e outros recursos web.
- 3) **Arquivos EAR** (en inglés *Enterprise Application Archive*). Agrupa nun único arquivo varios módulos dunha aplicación coma arquivos WAR ou compoñentes EJB e outras librarías en arquivos JAR empaquetados cos seus respectivos recursos. Así mesmo inclúese o descriptor de despregue da aplicación na carpeta META-INF.
- 4) **Arquivos RAR** (en inglés *Resource Adapter Archive*). Contén un adaptador de recursos de xeito análogo a un controlador JDBC e similar aos EAR, podendo ir contido nun arquivo deste tipo. O formato vén definido na especificación JCA (en inglés *Java EE Connector Architecture*).

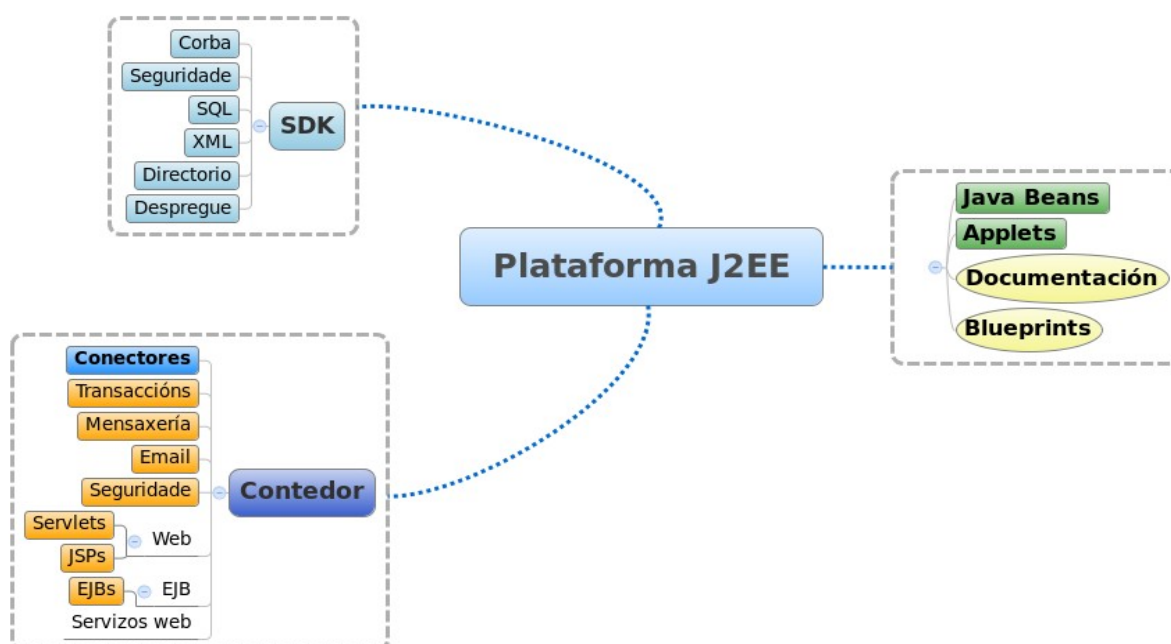
De xeito xeral convén considerar á plataforma como JEE, se ben, existen diferentes **edicións**, sendo as principais:

- 1) **J2ME**. (en inglés *Java 2 Platform Micro Edition*). Para desenvolvemento de aplicacións para dispositivos móbiles, electrodomésticos e equipos PDA. Desenvolveuse mediante o JPC baixo a especificación JSR 68.



- 2) **J2SE**. (en inglés *Java 2 Platform Standard Edition*). Para desenvolvemento de aplicacións de uso xeral en estacións de traballo. Desenvolveuse mediante o JPC baixo diferentes especificacións segundo as versións existentes: 1.4, 5.0 e 6.
- 3) **J2EE**. (en inglés *Java 2 Platform Enterprise Edition*). Para desenvolvemento de aplicacións destinadas a servidores de aplicacións para dar soporte a sistemas distribuídos en N capas. Estandarizada polo JPC a partir da versión 1.4 acostuma a denominarse JEE.

Para cada edición pode distinguirse entre a SDK (en inglés *Software Development Kit*), co software e recursos destinados ao desenvolvemento de aplicacións e o JRE (en inglés *Java Runtime Environment*) co contorno e librarías principais para permitir a execución das aplicacións.



**Figura 5: Plataforma J2EE.**

### 27.3.1 Modelo de desenvolvemento

O modelo de desenvolvemento máis habitual na arquitectura JEE é un



modelo separado en múltiples capas sendo o habitual un mínimo de tres, pero podendo chegar a 5 ou 7 segundo a complexidade do sistema. O obxectivo é minimizar o solapamento entre elas para que os cambios e modificacións se limiten ao mínimo necesario co ideal de que se poida cambiar unha das múltiples capas sen ter que modificar o resto. Mellórase a sostibilidade, crecemento do sistema e a reutilización de compoñentes no mesmo e entre sistemas. Así mesmo permítese unha maior heteroxeneidade de clientes ou elementos de cliente e presentación ao modificar só a capa máis próxima ao usuario. O deseño do modelo é análogo a solucións en .NET pero a implantación diferente. As 5 capas máis habituais se describirán a continuación.

#### **27.3.1.1 Capa de cliente.**

Agrupa os elementos da interface de usuario máis próximos ao cliente. Exemplos destes elementos serían o código (X)HTML/XML e Javascript, os Applets, arquivos de recursos e tecnoloxías RIA. Os tipos de aplicacións cliente máis habituais serían os navegadores web, as aplicacións de escritorio e actualmente cobran forza as aplicacións para dispositivos móbiles. Un aspecto importante neste modelo é garantir que os EJB da lóxica de negocio sexan accesibles tan só dende interfaces remotas a través do patrón *SessionFacade*. A variedade de interfaces actual fai que aparezan *frameworks* de xeración dinámica dos mesmos baseados na linguaxe **XUL** (en inglés *XML User-Interface Language*), baseada en XML. Permite incrustar XHTML e outras linguaxes coma MathML ou SVG ademais de CSS. Existen varias alternativas de librarías XUL coma Luxor, XWT, Thinlets ou SwingML.

#### **27.3.1.2 Capa de presentación.**

Contén toda a lóxica de interacción directa entre o usuario e a aplicación. Encárgase de xerar as vistas máis axeitadas para amosar a información a través de formatos e estilos adecuados. Compóñense dunha serie de



Servlets e páxinas JSP que se encargan de devolver o código que irá á capa de cliente logo de comunicarse coa capa de lóxica de negocio para obter os resultados. Pode localizarse nunha aplicación de escritorio ou nun contedor web. Ademais de ensamblar as diferentes vistas, controla o fluxo de navegación e fai funcións de autenticación, permisos de acceso e autorización de usuarios, etc... O patrón máis habitual nesta capa será o MVC (en inglés *Model View Controller*). Entre as tendencias actuais atópanse implementacións deste modelo coma Swing o JFace para aplicacións de escritorio, mentres que dentro dos *frameworks* web atoparíanse Struts, JSF, Tapestry, Expresso e moitos outros. Sendo Struts unha especie de estándar de feito. Estes *frameworks* ademais incorporan outros servizos como etiquetas personalizadas JSTL para interfaces de usuario, manexo de XML, acceso a datos, modelos, filtros, etc...

#### **27.3.1.3 Capa de lóxica de negocio.**

Contén os compoñentes de negocio reutilizables EJB ou POJO, que representan o conxunto de entidades, obxectos, relacións, regras e algoritmos do dominio ou negocio no que opere o sistema. Nesta capa a solución POJO é unha opción sinxela que pode mesturar elementos das capas de integración e datos, mentres que os EJB distinguen os obxectos de sesión e as entidades, recomendado nos Blueprints de Sun, emprazando cada un na súa correspondente capa. O patrón básico nesta capa será o *SessionFacade* onde un único Bean de sesión encárgase de recibir as chamadas de cliente-presentación e dirixilas dentro do contedor de EJB illando esta capa.

De xeito análogo establécese o modelo dos Bean de mensaxería, que realizan comunicación asíncrona mediante JMS nun servidor MOM (en inglés *Messaging Oriented Middleware*), que pode ser un servidor externo ao servidor de aplicacións. Tamén funciona cun patrón Fachada centralizando as chamadas remotas.



#### **27.3.1.4 Capa de Integración.**

Agrupa os compoñentes encargados do acceso a datos, sistemas *legacy*, motores de regras de *workflow*, acceso a LDAP, etc... Poden realizar cambios de formato na información, pero transformacións máis complexas deberían realizarse na capa de lóxica de negocio, restrinxido esta á lóxica de acceso a datos ou DAO (en inglés *Data Access Objects*) e os encapsuladores de datos e entidades VO (en inglés *Value Object*). Os VO poden implementarse como POJO ou EJB de entidade. Como ocorría anteriormente no caso dos POJO gáñase en facilidade pero pérdense servizos e funcionalidades coma os de persistencia. Os EJB suman complexidade, pero melloran o rendemento en memoria. No tocante ao acceso a datos aparecen as seguintes alternativas:

- 1) **JDBC.** Para POJO ou Beans de entidade con control de persistencia. É unha solución sinxela, con poucas funcionalidade pero que fai uso dunha API de uso estendido.
- 2) **DAO.** Vai un paso máis alá que o JDBC incorporando interfaces para abstraer o acceso a datos e facelo independente da linguaxe do xestor. Cada interface terá unha implementación diferente para cada xestor de bases de datos.
- 3) **Frameworks de persistencia.** Fan as funcións de motores de correspondencia de obxectos a bases de datos relacionais definindo entidades e relacións vía XML. Realizan gran parte das funcións de acceso a datos automaticamente. As solucións de uso máis estendido son os *frameworks* Hibernate, iBatis, TopLink, JPA ou a través de EJB.
- 4) **JDO** (en inglés *Java Data Objects*). Sistema de persistencia estándar a partir dunha especificación JEE, engadindo ademais da correspondencia entre o modelo relacional e os obxectos a posibilidade de permitir definir os obxectos sobre a base de datos. As implementacións de uso máis estendido son OJB, XORM, Kodo JDO ou LiDO.



### **27.3.1.5 Capa de sistemas de información.**

Atópase integrada polos sistemas de bases de datos, ficheiros, sistemas 4GL, ERP, Data Warehouse, Servizos web e calquera outros sistema de información da organización. Nesta capa irían os conectadores para diferentes sistemas de información heteroxéneos e os propios recursos que integran os sistemas de información. As solucións máis habituais enuméranse a continuación.

- 1) **JCA** (en inglés *J2EE Connector Architecture*). Define unha interface de acceso común independente do sistema, coa mesma API para todos. Basease no concepto de adaptador de recursos, sendo cada adaptador un controlador específico para un sistema de información. As operacións básicas que define a especificación son: xestión das conexións de acceso a JCA, seguridade, transaccións, multiproceso, paso de mensaxes e portabilidade dentro dos servidores de aplicacións.
- 2) **JMS** (en inglés *Java Message Service*). O servizo de mensaxes Java emprega colas de mensaxes para o traspaso de información entre compoñentes software establecendo unha infraestrutura MOM con dous modelos de API, Punto a punto, entre dous únicos clientes ou Publicador/subscritor onde varios clientes segundo o seu rol envían ou len mensaxes.
- 3) **Servizos web**. Permiten a comunicación entre sistemas heteroxéneos a través do acceso á URL de aplicacións empregando protocolos baseados en XML como SOAP ou SAAJ. Os clientes acceden ao servizo a partir da súa interface definida perante WSDL (en inglés *Web Service Definition Language*) ou inserida nalgún rexistro de servizos web.





**Figura 6: Modelo de desenvolvemento en capas.**

### 27.3.2 Servidores de aplicacións.

O servidor de aplicacións será o encargado de soportar a maioría das funcionalidades e servizos da tecnoloxía JEE, sendo o núcleo desta arquitectura. Cando un servidor de aplicación implementa a tecnoloxía JEE ten que proporcionar todos os compoñentes definidos na especificación e por tanto calquera aplicación JEE poderá despregarse e executarse no devandito servidor.

O servidor de aplicacións disporá de diferentes contedores para Applets e aplicacións clientes, web e EJB, sendo estes últimos os que se encargarán de operar coa lóxica do dominio, xestión de transaccións, persistencia, control do fluxo, etc...

- a) **Tomcat.** Sen presentar tódalas funcionalidades dun servidor de aplicacións, este servidor libre de Apache incorpora o servidor web Apache e soporte para JSP e Servlets co contedor Catalina. Presenta diferentes módulos de soporte de aplicación como seguridade SSL, SSO, JMX, AJP, JSF, conector Coyote para peticións HTTP, soporte



para Comet, Recolector de lixo reducido así como ferramentas web para despregue e administración.

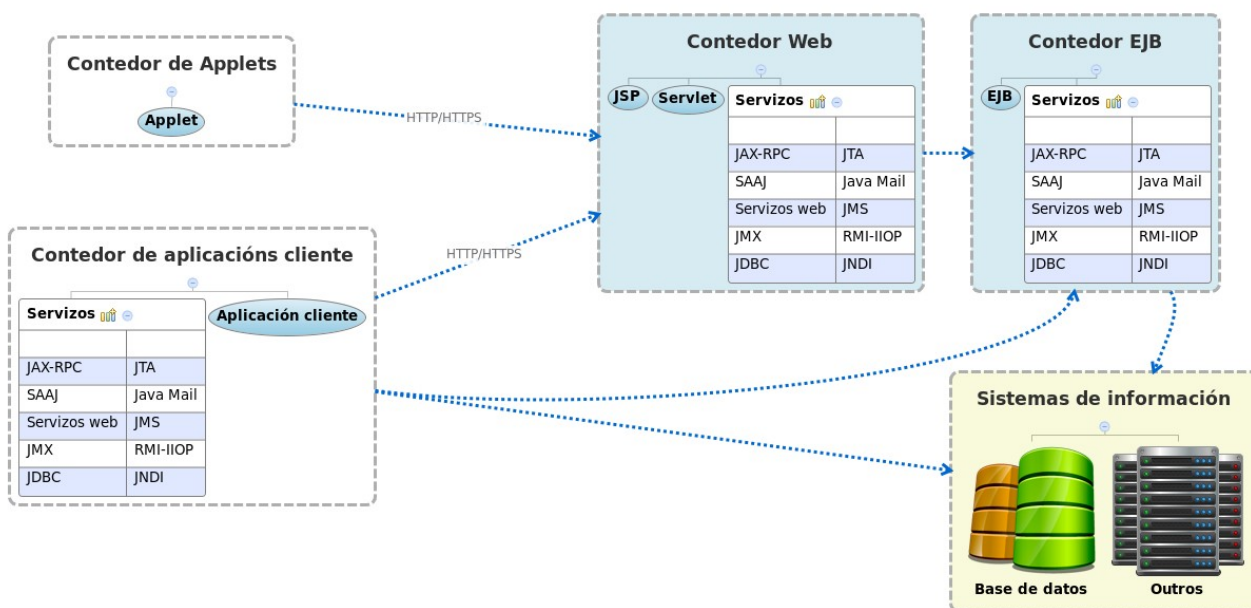
- b) **JBoss**. Un dos servidores de aplicacións libres de uso máis estendido composto por un Contedor de Servlets para JSP e Servlets e un Contedor de Beans. A diferenza de Tomcat implementa todo o conxunto de servizos especificados por JEE. Como contedor de Servlets emprega unha adaptación de Tomcat ou o contedor Jetty. Entre os módulos e funcionalidades que soporta destaca que permite a creación de *cluster*, soporte EJB, JMX, Hibernate, JBoss AOP para dotar a clases Java de persistencia e funcionalidade transaccional, sistema caché, JSF, Portlets, JMS, Servidor de correo, xestión de contidos foros e portais, entre outras moitas.
- c) **Geronimo**. Outro produto libre de Apache, compatible con JEE que inclúe JDBC, RMI, JM, Servizos web, EJB, JSP, Servlets e outras tecnoloxías. A principal característica deste servidor é que integra un gran número doutras solucións xa existentes: Tomcat e Embarcadero como contedores web, OpenEJB como contedor de Servlets, OpenJPA, Apache Axis, Apache CXF e Scout Apache para servizos web, Derby para o acceso a datos, e WADI para establecer clusters e balanceo de carga, entre outros.
- d) **JonAS**. Outra alternativa libre a JBoss, aínda que non soporta por completo JEE. Permite integración con Tomcat ou Jetty como contedores web e ten contedor de EJB. Entre módulos e servizos incorpora: Xplus, Hibernate, TopLink, OpenJPA, JORAM como implementación de JMS, varios protocolos RMI (IIOP, JRMP, IRMI), soporte LDAP, servizos web Axis e outros moitos.
- e) **Glassfish**. Alternativa libre de Sun, agora Oracle, que ten como base o *framework* para persistencia Toplink. Incorpora ademais módulos para soporte EJB, JAX-RS, JSF, RMI, JMS, servizos web, na liña dos anteriores, e novidades como Apache Félix, unha implementación de OSGi (en inglés *Open Services Gateway*) e Grizzly que fai uso da nova



API de Java de E/S (NIO) para mellorar a escalabilidade.

- f) **WebSphere.** Alternativa comercial de IBM cunha versión de libre distribución. A versión libre, máis lixeira, basease no servidor Geronimo diferenciándose deste en que inclúe soporte para DB2, Informix, soporte RAC de Oracle e outras bases de datos así como mellores librarías para XML. Outras tecnoloxías serían: os Servlets SIP (en inglés *Session Initiation Protocol*) que utilizan elementos multimedia en tempo real, mensaxería instantánea e xogos en liña; o *framework* Spring; protocolos de seguridade Kerberos e SAML. A diferenza con outros servidores e que posúe ferramentas de administración máis avanzadas, sobre todo para sistemas en cluster e soporte para *mainframes*.
- g) **Weblogic.** Alternativa comercial de Oracle baseada en Glassfish, incorporando servizos do Weblogic server sobre a JVM JRockit. En concreto Weblogic Server proporciona os Servizos web Oracle WebLogic Server Services Web, a Application Grid coma solución de *grid* de datos, soporte de conectividade con Tuxedo (WTC), soporte de RAC para Oracle, SAML, unha API de integración con .NET a JMS.NET, Spring e o *framework* de diagnose WLDF.
- h) **Coldfusion.** Alternativa comercial de Adobe das máis valoradas actualmente, diferenciándose polo soporte a tecnoloxías RIA principalmente Flash. Implementa parte dos servizos JEE pero pode integrarse con outros servidores de aplicacións como WebSphere ou Jboss, podendo despregarse como aplicación Java. Ademais leva incorporado o servidor de aplicacións Adobe JRun. Destaca polo soporte en tecnoloxías AJAX, Flex, PDF, RSS, Flash Remoting, integración .NET e ferramentas de administración avanzadas.

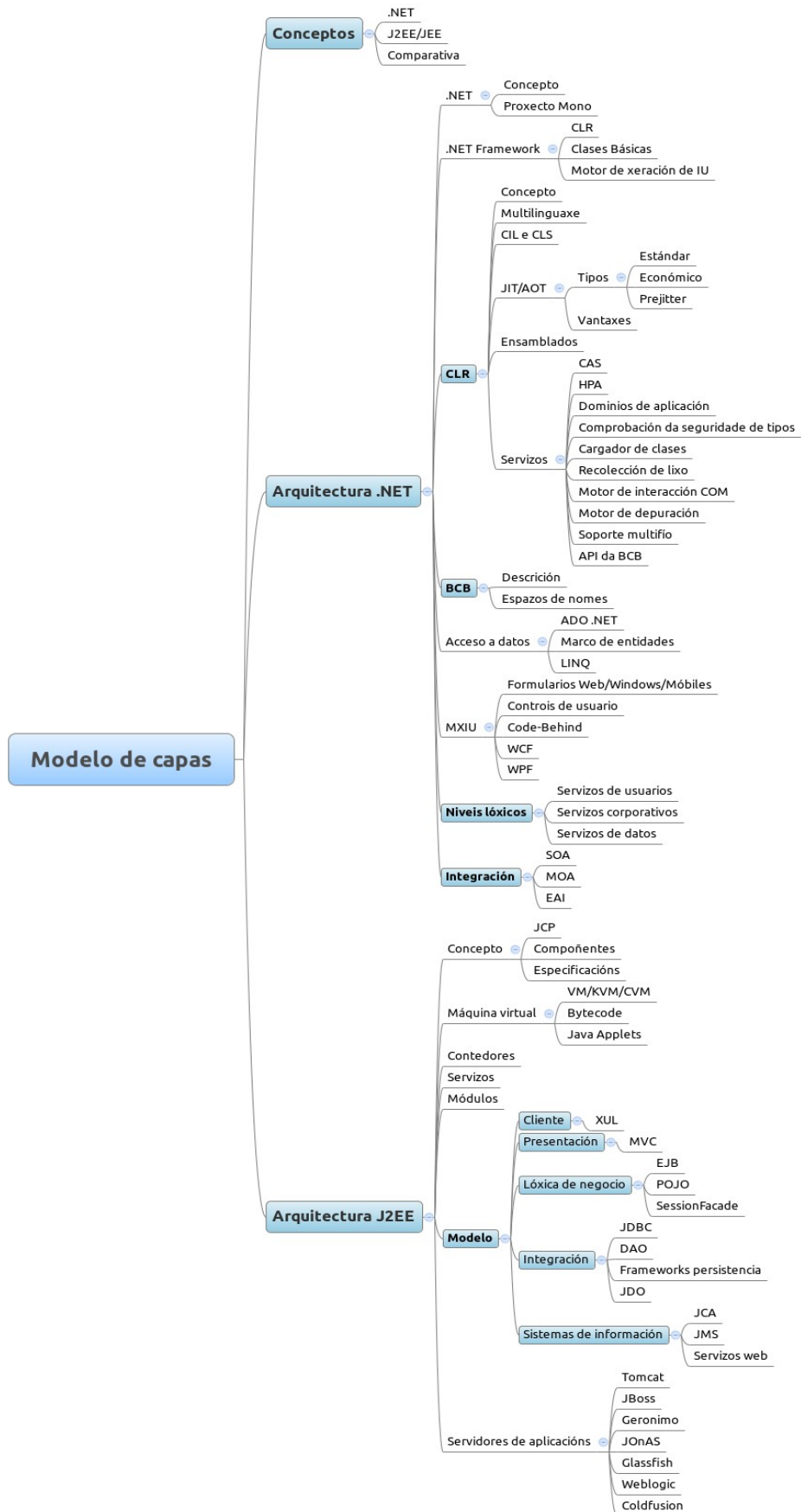




**Figura7: Arquitectura J2EE**

## 27.4. ESQUEMA







## **27.5. REFERENCIAS**

Varios autores.

Biblioteca MSDN de Microsoft. (2003).

Jef Ferguson e outros.

La biblia de C#. (2003).

Benjamín Aumaille.

J2EE. Desarrollo de aplicaciones Web. (2002).

I. Singh, B. Stearns e outros.

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. TECNOLOXÍAS XML.**



## TEMA 28. ARQUITECTURA SOA. SERVICIOS WEB. TECNOLOXÍAS XML.

### 28.1 INTRODUCCIÓN E CONCEPTOS

### 28.2 ARQUITECTURA SOA

### 28.3 SERVICIOS WEB

### 28.4 TECNOLOXÍAS XML

### 28.5 ESQUEMA

### 28.6 REFERENCIAS

#### 28.1. INTRODUCCIÓN E CONCEPTOS

Os sistemas actuais teñen unha grande complexidade debido á integración de múltiples compoñentes heteroxéneos. A comunicación e relación entre estes compoñentes é un dos grandes problemas actuais sendo **SOA** (en inglés *Service Oriented Architecture*) un dos actuais modelos de solución. Esta arquitectura entende a comunicación entre aplicacións e compoñentes coma servizos, non necesariamente servizos web, demandados por clientes ou subscritores e proporcionados e publicados por provedores. As arquitecturas para servidores de aplicacións de uso máis estendido coma .NET e JEE acostuman a definir unha **capa de integración** que agrupa os compoñentes encargados do acceso a datos, sistemas *legacy*, motores de regras de *workflow*, acceso a LDAP, etc... Para a comunicación dos compoñentes desta capa existen varias solucións coma JCA, JMS e servizos web, está última unha das máis aceptadas actualmente.

Os **Servizos web** permiten a comunicación entre sistemas heteroxéneos a través do acceso á URL de aplicacións empregando protocolos baseados en XML como SOAP ou SAAJ. Os clientes acceden ao servizo a partir da súa interface definida perante WSDL (en inglés *Web Service Definition Language*) ou inserida nalgún rexistro de servizos web.



O uso de XML convértese nun estándar de integración, sendo a base das comunicacións nesta capa, tanto para estruturar coma para almacenar e intercambiar información, estendendo o seu uso a outros ámbitos. As **tecnoloxías XML** son un conxunto de módulos que ofrecen servizos como: XSL/XSLT para deseño de documentos, Xpath como linguaxe de rutas para acceso a documentos, a linguaxe de consulta XQL, e outros como XLink ou XPointer.

## **28.2 ARQUITECTURA SOA**

SOA define unha arquitectura orientada a servizos que busca simplificar o modelo de integración de sistemas distribuídos heteroxéneos. Nesta arquitectura os compoñentes publican e invocan servizos na rede a través de mecanismos de comunicación coma JCA, JMS, SOAP, RPC ou Servizos web. Os servizos son funcionalidades da lóxica de negocio que poden invocarse de xeito remoto para obter un resultado. Defínense perante unha interface explícita, por exemplo a través de WSDL, independente da súa implementación empregando estándares de comunicación baseados en XML. SOA define tres **bases** fundamentais:

- 1) **Orientación ao intercambio de mensaxes.** A base do sistema é a comunicación entre os nodos do sistema.
- 2) **Abstracción de compoñentes.** Cada sistema redúcese á súa interface e o conxunto de servizos que define, co cal permite a integración entre calquera tipo de sistema.
- 3) **Metadatos.** Descricións e información asociada a servizos e mensaxes, mellorando as capacidade semántica do sistema.

A nivel lóxico os principais compoñentes nunha arquitectura SOA son:

- a) **Servizos.** Entidades ou funcionalidades lóxicas definidos en interfaces públicas, que poden ou non requirir autenticación.
- b) **Provedor de servizos.** Compoñente software que implementa un



servizo e publica a súa interface.

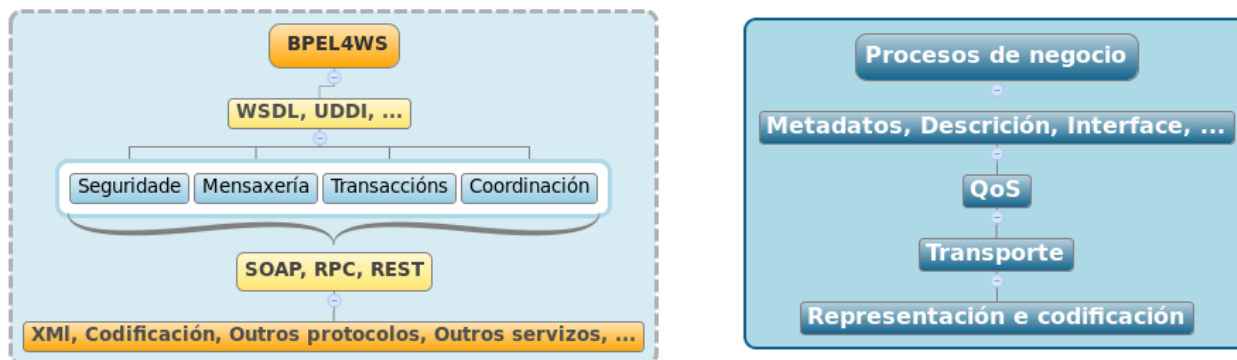
- c) **Ciente de servizos.** Compoñente software que invoca un servizo dun provedor.
- d) **Localizador de servizos.** Provedor de servizos que rexistra as interfaces e permite aos clientes buscar no rexistro e acceder á súa localización.
- e) **Servizo de interconexión.** Provedor que comunica solicitudes de servizo a outros provedores.

O concepto de **BPM** ou Xestión de procesos de negocio (en inglés *Business Process Management*), está moi relacionado con SOA. BPM é un modelo de xestión centrado en procesos de negocio e de como integrar as súas funcionalidades en sistemas heteroxéneos. A partir da identificación e xestión dos procesos da organización pode implantarse unha solución BPM a través dunha arquitectura SOA. Froito desta idea aparecen solucións coma:

- ✓ **BPMN.** Notación para o modelado de procesos de negocio.
- ✓ **BPEL.** Linguaxe de execución de procesos de negocio con servizos web para a orquestración de servizos. Xeralmente se realiza unha conversión de BPMN a BPEL.
- ✓ **BPEL4WS.** Linguaxe de definición e execución de procesos de negocios empregando servizos web (en inglés *Business Process Execution Language for Web Services*). BPEL4WS é resultado da converxencia de WSFL (en inglés *Web Services Flow Language*) e XLANG, permitindo compoñer Servizos web coma servizos compostos denominados Servizos de negocio.



## Arquitectura SOA



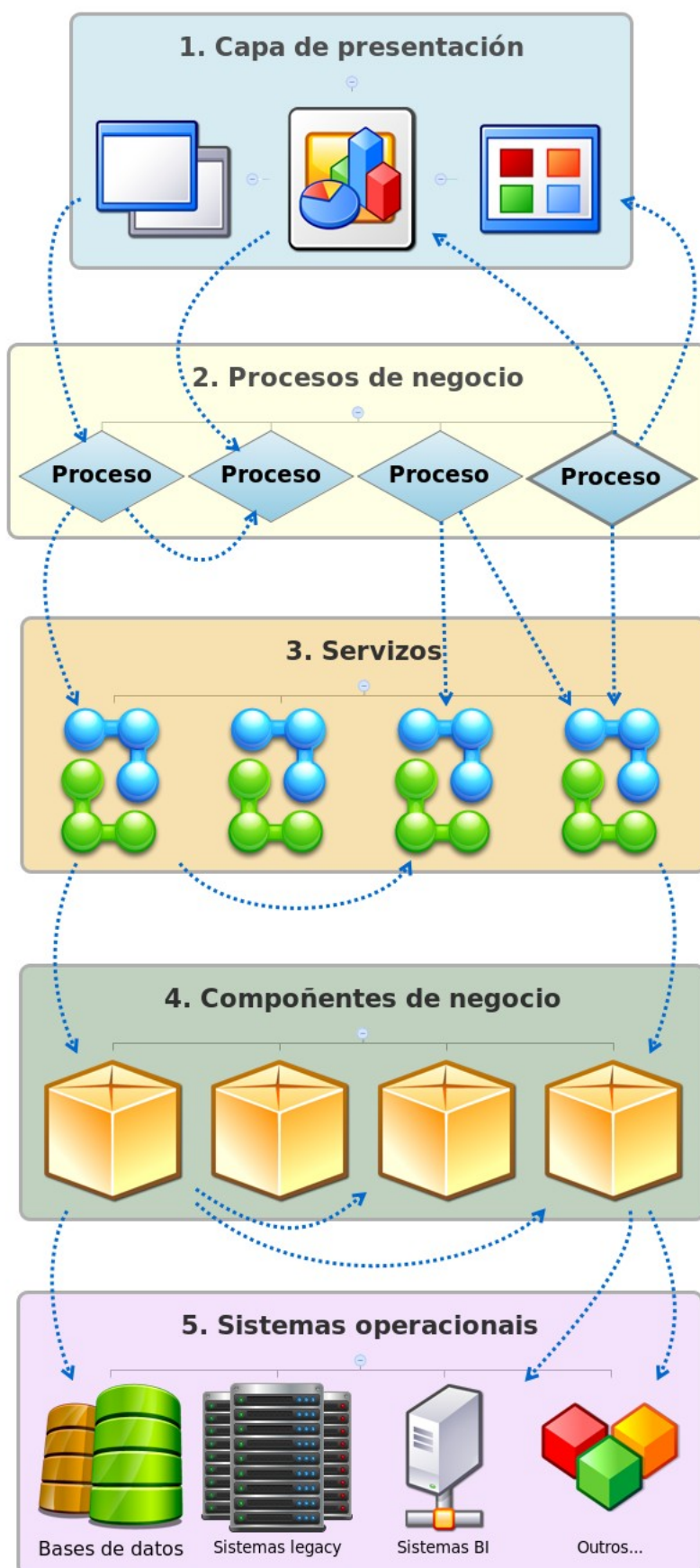
**Figura 1: Arquitectura SOA.**

A integración na arquitectura SOA dos BPM permite establecer o que se da en chamar **SOA Governance**, unha estrutura para a toma de decisións e establecemento de responsabilidades na organización a través da implantación de políticas, monitorización de servizos, incorporación de boas prácticas, principios arquitectónicos, mellora continua dos procesos de negocio, etc... en definitiva análise e deseño de solucións que permitan cumprir con éxito a implantación de SOA nunha organización.

A **nivel conceptual SOA** describe unha serie de guías ou patróns para servizos aliñados cun modelo de negocio. Un modelo conceptual de SOA permite definir un deseño en múltiples capas onde os servizos se relacionan con procesos de negocio e sistemas de información. A división máis habitual considera 7 capas diferenciadas, que se poden ver graficamente na Figura 2:

- 1) **Capa de Presentación.** Interfaces de usuario en sitios web, aplicacións e portais que invocan funcionalidades dos procesos de negocio.





6 Capa de Integración (ESB)

7 Capa de SOA Governance



**Figura 2: Modelo conceptual SOA.**

- 2) **Capa de Procesos de negocio.** Representa os procesos e fluxos operativos ou workflows que invocan os clientes dende a capa de presentación ou orquestración de servizos. Os procesos por norma xeral representaranse con BPEL e implementarase con algunha ferramenta de transformación.
- 3) **Capa de Servizos.** Funcionalidades dos compoñentes da lóxica de negocio que se publican para uso dos clientes. Referéncianse a partir da interface sendo transparentes á súa implementación.
- 4) **Capa de Compoñentes de negocio.** Son os encargados de proporcionar as funcionalidades que se publicarán nos servizos así como calquera outra intermedia ou común ao sistema. A este nivel irían os servidores de aplicacións, outros servizos web, aplicacións, paquetes e librarías.
- 5) **Capa de Sistemas operacionais.** Neste nivel irían os sistemas de información da organización, sistemas *legacy*, sistemas CRM ou ERP, aplicacións de BI, etc...
- 6) **Capa de Integración.** Axiliza a integración de servizos a través de sistemas tipo Buses de Servizos Empresariais ou ESB, que realizan funcións de enrutamento, monitorización e administración, transformacións das mensaxes, etc... dentro da área de comunicacións.
- 7) **Capa de SOA Governance.** Realiza funcións de administración, monitorización e control da calidade do servizo en áreas como seguridade, dispoñibilidade e outros factores xerais non recollidos na capa de integración.

Existen **patróns de deseño** tomando como punto de partida esta arquitectura trátase de patróns para Servizos web e patróns **POSA** (en inglés *Service-Oriented Architecture Patterns*). Algúns dos principais son:



- **Service Oriented Architecture.** Patrón que define a arquitectura SOA establecendo regras, relacións e dependencias entre os compoñentes do sistema. Permite buscar servizos dinamicamente con independencia da plataforma e sen requirir implementación, con transparencia. Este patrón é unha variante ampliada do **Broker Pattern** de POSA. Neste patrón un Servizo ou nodo intermedia axuda a localizar o servizo e pode obrigar a realizar todas as comunicacións a través del ou ben unha vez establecida deixar que esta sexa directa entre o cliente e o servizo.
- **Architecture Adapter.** Patrón xenérico que facilita a comunicación entre diferentes arquitecturas grazas á independencia de usar XML/SOAP e a xeración de clases proxy. Este patrón é implementado por *frameworks* para Servizos web como Apache Axis (Java).
- **Service Directory.** Facilita a localización de Servizos web a partir dunha especificación robusta das interfaces a través do catálogo UDDI de interfaces WSDL.
- **Service Factory.** Permite a selección de servizos do provedor illando o código de comunicación UDDI. Do mesmo xeito o patrón de estendido Service Factory Cache fai funcións de caché no servizo. Simplifica en parte a API do patrón Service Directory.
- **Service Facade.** Proporciona un servizo web controlador que actúe como punto de entrada da lóxica de negocio ou obxecto de fachada. Pode empregar simultaneamente outros mecanismos de comunicación coma CORBA.
- **Event Monitor.** Emprégase para notificar que un Servizo web de longa duración invocado remotamente completa a solicitude. Cando o Servizo non dispón de mecanismos de notificación cómpre establecer un intermediario.
- **Business Object.** Un BO engloba un concepto do dominio, equiparable a un VO para contornos distribuídos.
- **Business Process.** Un BP engloba un proceso da lóxica de negocio,



representando a xerarquía formada polas diferentes implementacións das súas funcionalidades e a interface do servizo.

- **Asynchronous Business Process.** Este patrón encárgase de xestionar a chamada e notificación de resposta ao cliente cando estas poden ser de longa duración.
- **Business Object Collection.** Agrupa diferentes procesos de negocio nun mesmo BOC.
- **Observer Services.** Basease nun rexistro de servizos onde o observador notifica ao cliente sobre eventos que derivados dos servizos nos que esta rexistrado.
- **Publish/Subscribe Services.** Evolución do patrón Observer Services incorporando un sistema de notificacións para substituír ao rexistro. Emprégase cando os servizos web non incorporan un sistema de notificación e precisan un intermediario.
- **Data Transfer Object.** Permite enviar múltiples obxectos nunha mesma chamada reducindo o número de conexións.
- **Partial Population.** Permite que os clientes seleccionen parte da información do mensaxe de resposta á solicitude de servizo buscando un mellor aproveitamento do ancho de banda.
- **Microkernel.** Separa un núcleo de funcionalidade mínimo de partes especificadas polo cliente.
- **Web Service Interface.** Proporciona unha interface que pode empregarse dende os clientes para invocar os métodos dun proxy de Servizo web xenérico en lugar de depender da clase proxy xerada a partir da WDSL.

REST (en inglés *Representation State Transfer*) representa un modelo de comunicación onde cada petición HTTP contén a información necesaria para responder á petición sen ter que almacenar o estado da sesión. En REST todo os servizos son recursos, identificados por URIs e se deseñan as súas representacións mediante XML, JSON ou microformatos. REST representa unha arquitectura SOA que non fai uso de Servizos web, SOAP



nin RPC.

Actualmente dentro do marco da Web 2.0 xorde unha nova variante nas arquitecturas SOA, o concepto de **Mashup**, un sitio ou aplicación web que fai uso de contido doutras aplicacións ou servizos vía HTTP. Este contido é recuperado nun modelo de Servizos web a través da súa API pública evitando caer no Web Scraping. Para empregar os Mashups coma XML empréganse linguaxes específicos coma EMMML (en inglés *Enterprise Mashups Markup Language*). As arquitecturas Mashup constan de tres compoñentes:

- ✓ **Os provedores de servizos.** Orixes de datos que publican a través dunha interface os métodos de acceso aso mesmos e permiten a súa consulta vía Atom, RSS, REST, JSON, Bases de datos ou interfaces WSDL de Servizos web.
- ✓ **Aplicación ou Servizo web Mashup.** Proporciona un novo servizo a partir da información obtida dos provedores.
- ✓ **Cientes.** Usuarios finais, ou outras aplicacións ou servizos que fan peticións ao Mashup. Nos clientes acostuman a empregarse tecnoloxías RIA do tipo de AJAX ou Comet.

Outro concepto que se pode relacionar con SOA é o da **Nube** (en inglés *Cloud Computing*). A nube fundamentase en empregar a rede Internet para publicar servizos, que poden ou non requirir identificación. Na nube todo son servizos, aplicacións, bases de datos, redes, e xestiónanse e accédense como tal. A arquitectura da nube, estruturase habitualmente en tres capas:

- 1) **Software como servizo** ou SaaS (en inglés *Software as a Service*). Sería o nivel máis alto, orientado aos usuarios e clientes finais. Incluiríanse as aplicacións propias e aplicacións de terceiros do estilo de Google Aps. A mesma infraestrutura do provedor serve a múltiples organizacións finais.
- 2) **Plataforma como servizo** ou PaaS (en inglés *Platform as a Service*).



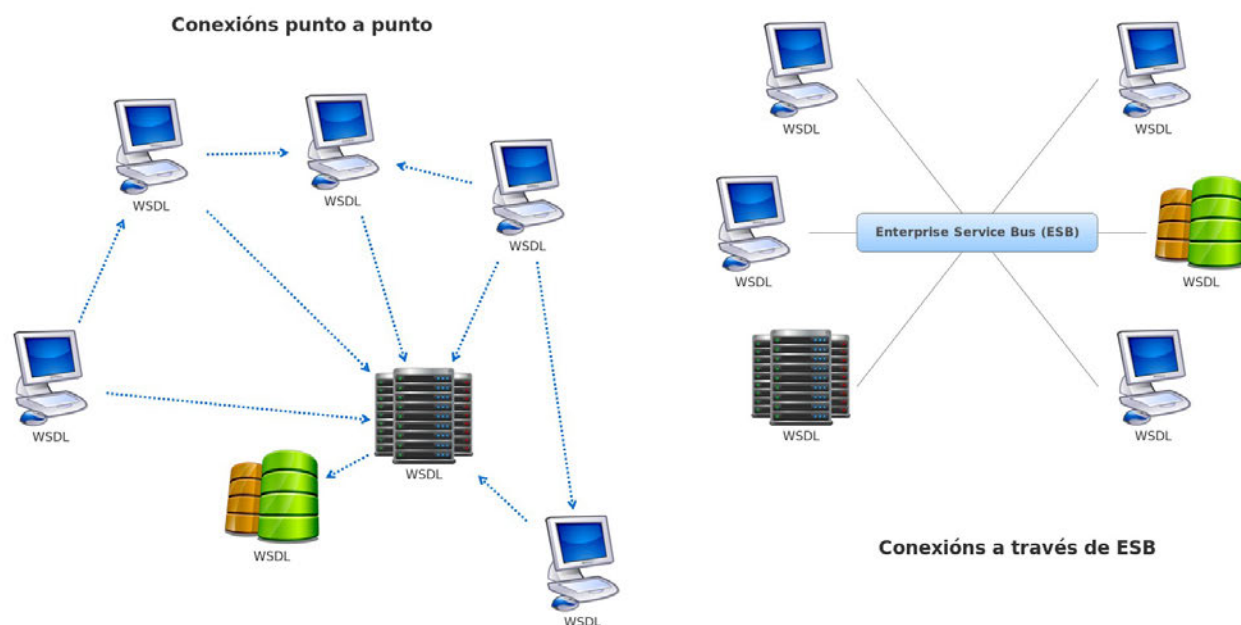
Serían a capa intermedia encargada de encapsular sistemas ou *middleware* permitindo correr aplicacións sobre elas. Exemplos deste servizo serían Google App Engine ou Windows Azure. Deste xeito unha organización externa proporciona un servizo de infraestrutura e soporte para outras organizacións.

- 3) **Infraestrutura como servizo** ou IaaS (en inglés *Infrastructure as a Service*). Neste caso ofrécense servidores para computación, rede, almacenamento ou bases de datos a través de diferentes técnicas como por exemplo a través de máquinas virtuais.

O **Bus de Servizos Empresariais** ou ESB (en inglés *Enterprise Service Bus*) representa outras das características de SOA, aínda que non é imprescindible nunha arquitectura deste tipo. Trátase dun compoñente que fai abstracción do sistema de mensaxes da organización a través dun sistema único para todos os elementos dun sistema SOA. O ESB proporciona funcións de transformación, adaptación, conexión e enrutamento que poden ser implementadas en SOA. Nunha arquitectura SOA cun ESB todas as aplicacións e servizos conéctanse a un punto único e central que administra as comunicacións, realizando funcións de *middleware*. O ESB constrúese sobre tecnoloxías XML, XSLT, XPath, JMS ou propias de Servizos web. Fai uso de elementos denominados Contedores de Servizos ou Brokers que fan a función de servidores de comunicacións. Entre os servizos que proporciona se atopan:

- ✓ Funcionalidades de enrutamento, fraccionamento e combinación de mensaxes baixo a base dos patróns EIP (en inglés *Enterprise Integration Pattern*).
- ✓ Funcións de supervisión e control da calidade do servizo a través de Acordo de Nivel de Servizos ou SLA dos servizos.
- ✓ Funcións de monitorización, seguridade e mediación de protocolos.





**Figura 3: Bus de Servizos Empresariais (ESB).**

O Bus substitúe a comunicación directa entre dous aplicacións ou servizos, de xeito que a comunicación faise de xeito transparente a través do ESB. Emprega un sistema de mensaxería, como por exemplo Tibco, soportando varios MEP ou patróns de intercambio de mensaxes, así como colas para reenviar as peticións aos provedores de servizos e as respostas ás solicitudes aos clientes. Existen *frameworks* que recollen os compoñentes precisos para implantar un ESB como Mule ESB, este é un *framework* lixeiro destinado a mensaxería e control de eventos. Permite integracións con outros *frameworks* coma Struts ou Spring e soporta moitos compoñentes de servizo coma JMS, SOAP, BPEL, JBI (en inglés Java Business Integration), e outros.

Por último sinalar que as arquitecturas SOA poden completarse con módulos específicos segundo as necesidades da organización, como:

- ✓ **Seguridade.** Coa adopción de diferentes tecnoloxías, SSL, Kerberos, X.509, Sinaturas XML, Encriptación XML, XML Canonicalization, SAML



(en inglés *Security Assertion Markup Language*) ou XKMS (en inglés *XML Key Direction Specification*), que administra a chave pública ou PKI das infraestruturas.

- ✓ **Orquestración e coreografía de servizos.** Neste modelo a interacción entre servizos non se produce directamente senón que se define unha entidade que define a lóxica de interacción, facilitando a colaboración que será un servizo de control primario. En BPEL o servizo primario será un proceso BPEL, pero tamén se pode definir con BPEL4WS, WSFL ou XLANG. Mentres a orquestración precisa dun director de orquestra ou servizo central o modelo de coreografía establece interaccións punto a punto a partir de regras de colaboración xerais. Para a coreografía existen linguaxes específicas como WS-CDL (en inglés *Web Services Choreography Description Language*) que teñen definida a forma de representar as interaccións.
- ✓ **Xestión transaccional.** Existen varias tecnoloxías que coordinan as transaccións entres servizos autónomos. O BTP (en inglés *Business Transaction Protocol*) onde ningún dos servizos xestiona unha transacción, senón que esta se comunica a todos e deciden se se unen ou non, con comunicacións baseadas en XML nun formato propio. Outros mecanismos como WS-Transaction e WS-Coordination encárganse de xestionar transaccións levadas a cabo por varios servizos á vez, con protocolos SOAP e WSDL. Pola súa banda JEE dispón da especificación JAXTX para transaccións complexas co obxectivo de illar estas dos contedores.

### **28.3 SERVIZOS WEB**

Os Servizos web son un dos modelos de implementación de SOA. Un Servizo web que proporciona un servizo vía web nunha rede a través dunha interface que lle permite recibir peticións e transmitir respostas. Para soportar este sistema se desenvolveron unha grande variedade de



protocolos e tecnoloxías. Os principais son o HTTP/HTTPS para peticións e respostas e o XML como formato de intercambio. Os principais **compoñentes** comúns aos servizos web serían:

- a) **SOAP** (en inglés *Simple Object Access Protocol*). O protocolo de comunicación, sobre a capa de transporte baseado en XML, que serve para invocar os servizos a través dun protocolo sendo os máis habituais HTTP ou SMTP, pero realmente é independente e permite outros como POP3 ou JMS. Permite tanto describir o contido da mensaxe e regras de codificación dos tipos de datos, coma aspectos de seguridade e transaccionalidade. Atópase estandarizado polo W3C, o que garante a comunicación entre sistemas heteroxéneos que o implementen.
- b) **UDDI** (en inglés *Universal Description, Discovery and Integration*). Directorio onde se publican os servizos proporcionando a información necesaria para permitir a súa invocación. Presenta dúas API que permiten aos servizos publicar as súas funcionalidades e aos clientes enviar as peticións e obter os resultados. Cada servizo publícase no UDDI proporcionando a URL da súa WSDL e meta-información. De xeito xeral enténdese que UDDI proporciona tres tipos de servizos: información xeral sobre os provedores dos servizos (páxinas brancas), categorías e clasificacións de servizos (páxinas amarelas) e as regras de negocio ou información técnica sobre os servizos (páxinas verdes).
- c) **WSDL** (en inglés *Web Services Description Language*). Linguaxe baseado en XML e XML Schema que permiten a descrición da interface dos Servizos web e que está estandarizado polo W3C. Nun documento WSDL defínense os tipos de datos, as mensaxes, os *endpoints*, os *bindings* e os servizos.
- d) **Serialización de datos**. Empréganse definicións de XML Schema para especificar como codificar os datos en conxunción coas regras



de codificación de SOAP. Aínda que o mecanismo máis habitual sexa o SOAP Document/Literal existen outros mecanismos coma: RPC/Encoding, Document/Encoding ou RPC/Literal.

Segundo o visto anteriormente para as arquitecturas SOA en xeral, pódense definir varios tipos de servizos segundo a súa complexidade, comezando polos de nivel básico aos de niveis máis complexos.

	<b>Servizos de nivel básico</b>	<b>Servizos de alta complexidade</b>
<b>Función</b>	Integración da funcionalidade dunha aplicación	Elemento chave dunha arquitectura SOA
<b>Protocolos e tecnoloxías</b>	SOPA, UDDI, WSDL	ebXML, BPEL, BTP, RossetaNet, Apache Axis, ...
<b>Tipo de contido</b>	Plano	MIME, PDF, ...
<b>Comunicacións</b>	Punto a punto	Multiparty, ESB, ...
<b>Mensaxería</b>	JMS, RPC, ...	Colaboración e <i>workflows</i>
<b>Transaccionalidade</b>	Non transaccional	Transaccional
<b>Seguridade</b>	SSL, autenticación, ...	Sinatura dixital, XML-encryption, Kerberos, ...

***Táboa 1: Complexidade dos servizos web.***

Segundo o tipo de comunicacións as APIs máis habituais son:

- API de mensaxería.** Clientes e servizos dispoñen de sistemas de mensaxería que lles permiten comunicarse en formato XML. Ao estar orientadas cara os sistemas de mensaxería presentan unha alta QoS.
- API de RPC.** A solución máis habitual, que emprega un compilador intermedio de WSDL para xerar o *stub* e o *skeleton* para cliente e



servidor respectivamente, tal e como acontece con CORBA. Este sistema é o que acostuman empregar os *frameworks* actuais como Apache Axis.

- c) **API para servidores de aplicacións (JEE/.NET).** Estas API veñen dispoñibles nas bibliotecas de clases de cada arquitectura, por exemplo en JEE dispónse de: JAXM (en inglés *Java API for XML Messaging*) para intercambio de mensaxes; JAX-RPC (en inglés *Java API for XML-based RPC*) , que permite enviar peticións remotas a terceiros e recibir resultados; e JAXR (en inglés *Java API for XML Registries*), que proporciona acceso a rexistros de negocio e mecanismos para compartir información .

O **proceso de implementación dun servizo web** consiste en implementar as funcionalidades do servizo a reutilizando as clases xeradas a partir dun WSDL ou dunha API (JAX-RPC, Axis, ...). Pódense aproveitar as ferramentas existentes nos IDE, ou outras máis específicas con Ant ou *WsdI2java*. Unha vez implementadas as clases coa lóxica do servizo xéranse as clases nun war e despréganse nun contedor de Servlets ou nun IIS. Sobre o modelo de programación empréganse diferentes variantes:

- a) **Estilo CORBA.** Xéranse todas as clases ao compilar empregando clases das API (Axis, JAX-RPC, ...)
- b) **Dynamic Proxy.** A interface WSDL créase ao compilar, pero o proxy no cliente só se compila en tempo de execución.
- c) **Dynamic Invocation Interface.** Tanto WSDL como cliente xéranse en tempo de execución. O cliente busca e invoca o servizo vía *broker*.

Outro dos factores a considerar é o tema da **seguridade** nos Servizos web, que pola propia natureza das arquitecturas SOA resulta un tema complexo. O principais elementos de seguridade no que respecta a JEE, aínda que



moitas serían extensibles a .NET serían:

- ✓ Para JAX-RPC a **API XWS-Security** que facilita a integración de aspectos de seguridade.
- ✓ O estándar **XML-DigitalSignature** para sinatura dixital.
- ✓ O estándar **XML-Encrytion** para encriptación de mensaxería.
- ✓ **Certificados X.509** para autenticación.
- ✓ Bases de datos de certificados baseadas en **JKS** (en inglés *Java Key Store*).

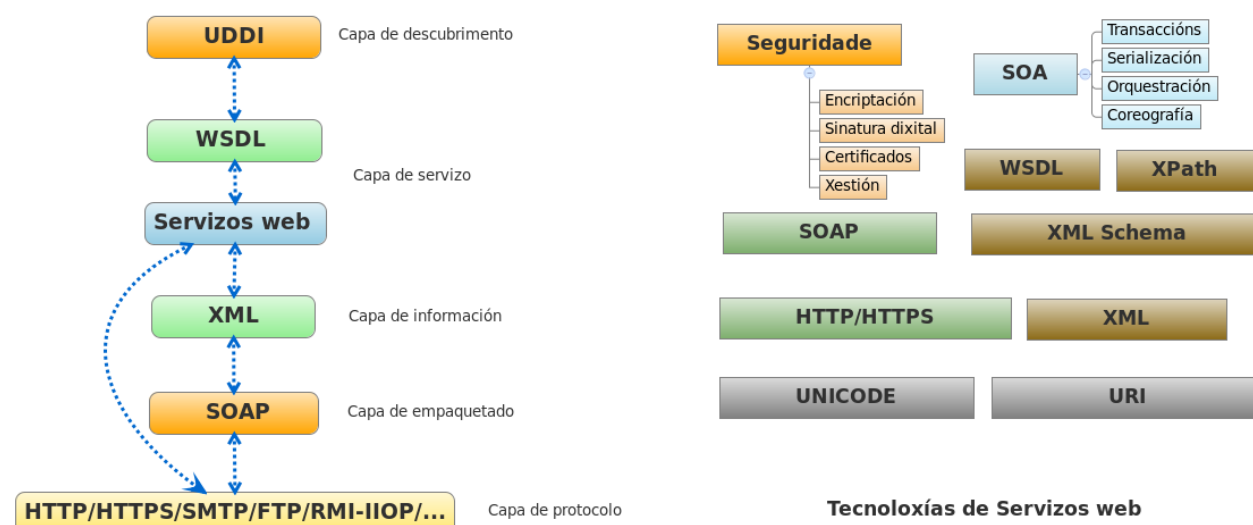
Dentro da arquitectura os diferentes mecanismos integraríanse nivel a nivel do seguinte xeito:

- 1) **Nivel de transporte.** Autenticación básica, autenticación por certificado vía SSL/TLS. Codificación de usuario/contrasinal nos *stubs* e regras de seguridade para *endpoints*.
- 2) **Nivel de mensaxe.** Sinatura de contidos con certificados XML-DigitalSignature, certificados X.509 e encriptación.

Entre as **posibilidades** existentes para implementar Servizos web atópanse:

- ✓ APIs Java: JAX-RPC, JAXM, SAAJ (mensaxes SOAP como obxectos), JWSL (Acceso a descrições WSDL), JAXR (Acceso ao UDDI), *framework* Apache Axis, ...
- ✓ .NET: ASP .NET, MS SOAP Toolkit, ...
- ✓ Outras tecnoloxías: NuSOAP para PHP, Axis para C++, ...





**Figura 4: Tecnoloxías de Servizos web.**

## 28.4 TECNOLOXÍAS XML

A linguaxe XML (en inglés *extensible Markup Language*) é unha metalinguaxe para etiquetado desenvolvido polo W3C. En SOA o XML representa o estándar para intercambio de información estruturada entre sistemas heteroxéneos. En esencia XML é unha linguaxe de marcas que permite a creación doutras linguaxes de marcas, con diferentes usos en SOA, cada unha destas linguaxes denomínase Aplicación XML e representa un modelo de datos de acordo a un esquema semántico.

O XML resulta máis estrito que outras linguaxes como HTML, admitindo varios mecanismos de validación ou corrección:

- ✓ **Formación.** Un documento XML denomínase “ben formado” cando segue as regras léxicas (tipo de caracteres, codificación, ...) e sintácticas (anidación correcta, marcas de apertura e peche de estrutura, ...) debidas.
- ✓ **Validación.** Un documento XML considérase “validado” cando



cumpre un conxunto de regras e restricións denominadas en conxunto gramática ou Esquemas XML (en inglés *XML Schema*). Existen varios formatos para gramáticas: os DTD herdados do SGML, os XML-Schema, que son recomendación do estándar polo W3C e outros máis específicos coma o XML Data.

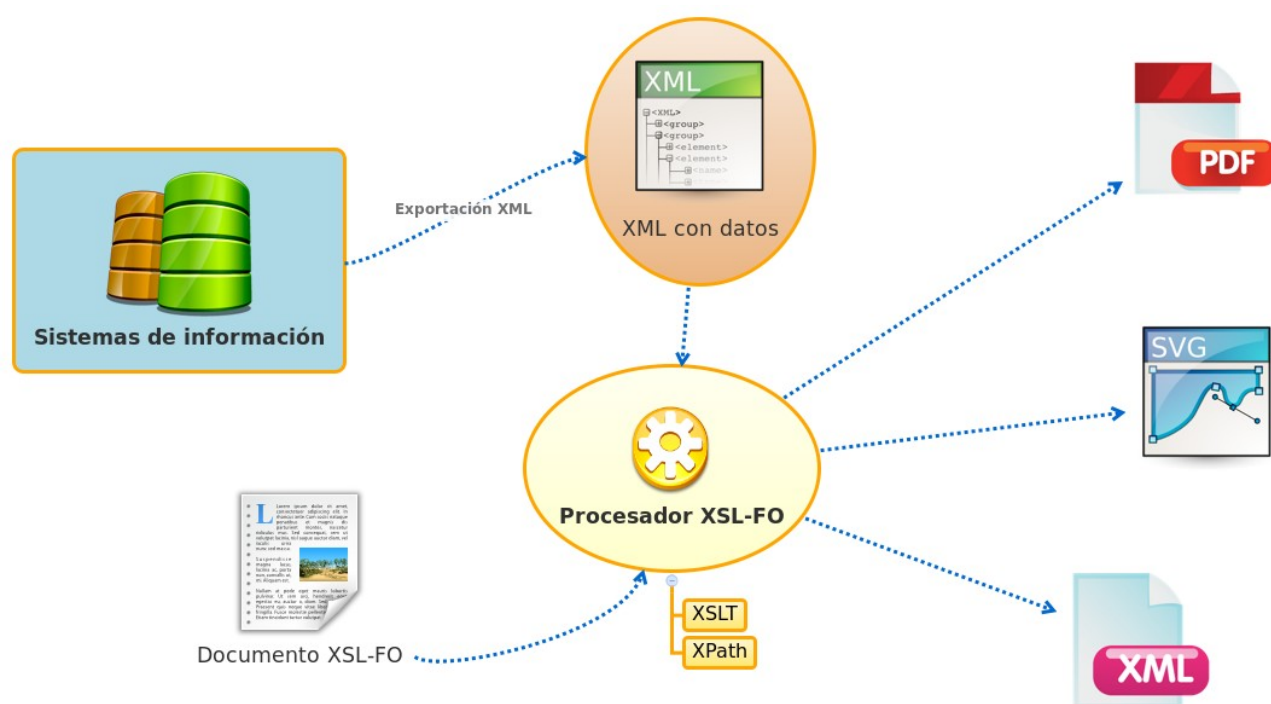
As **tecnoloxías principais máis empregadas** do modelo XML en arquitecturas SOA veñen dadas por:

- ✓ **XML Schema.** Linguaxe de esquema para describir a estrutura e regras de validación dun documento XML. Diferenciase do DTD en que permite un grande número de tipos de datos. Os documentos de esquema son de extensión XSD (en inglés *XML Schema Definition*). A programación de esquemas basease nos espazos de nomes e os elementos e atributos que conteñen. Logo de validar un documento contra un XSD pódese expresar a súa estrutura e contido en termos do modelo de datos do esquema. Esta funcionalidade denomínase PSVI (en inglés *Post Schema Validation Infoset*), e permite transformar o documento nunha xerarquía orientada a obxectos.
- ✓ **XSL.** XSL funciona como unha linguaxe avanzada para crear follas CSS transformando e realizando outras operacións sobre documentos XML, dándolles formato. Á súa vez pode descompoñerse en tres linguaxes ou dialectos XML, todos recomendacións do W3C, que integran a familia XSL:
  - ✓ **XSLT** (en inglés *Extensible Stylesheet Language Transformations*). Estándar para documentos XML que permiten transformar documentos XML en base a modelos dunha sintaxe a outra permitindo estruturas de programación, funcionando a xeito de intérprete. As regras dos modelos defínense programaticamente e a principal capacidade desta linguaxe é que permite separar o contido da presentación, ou diferentes presentacións en documentos XML o que se adapta perfectamente aos modelos de



separación en capas vistos.

- ✓ **XSL-FO** (en inglés *Extensible Stylesheet Language Formatting Objects*). Documentos XML que especifican formatos de datos ou obxectos para a súa presentación. A utilidade básica destes documentos é a presentación, co cal se complementan con XSLT na saída de datos das aplicacións. Permite a xeración de documentos multiformato: XML, (X)HTML e mesmo PDF. Existen procesadores específicos para este tipo de operacións coma Apache FOP.
- ✓ **XPath ou XML Path Language**. Permite identificar partes dun documento XML, accedendo aos seus atributos e elementos coma se foran nodos, a través da construción de expresións que recorren e procesan un documento XML. En XSL permite seleccionar e percorrer o documento XML de entrada da transformación, pero por extensión ten outros moitos usos actualmente, servindo de base para outras linguaxes XML.



**Figura 5: Familia XSL.**



- ✓ **XPointer.** Recomendación do W3C que permite localizar puntos concretos ou fragmentos nun documento que expande as funcionalidades de XPath a través de rangos .
- ✓ **XLink** . Recomendación do W3C que define un mecanismo para engadir hiperligazóns en arquivos XML ou outros recursos, coa opción de navegar nos dous sentidos, con ligazóns bidireccionais ou varios arquivos enlazados, multiligazóns. En XLink todo na rede é un recurso, e se pode enlazar dende un localizador, definindo as relacións entre recursos con arcos. Así mesmo permite agregar a un vínculo información sobre si mesmo coma metadatos.
- ✓ **XQuery.** Linguaxe de consulta desenvolvido polo W3C para recuperar coleccións de datos XML, con moitas similitudes con SQL. Permite extraer e manipular información de documentos XML ou calquera outro sistema de información que permita representación vía XML como Bases de datos ou documentos ofimáticos. Emprega XPath para acceder aos documentos engadindo unhas expresións propias denominadas FLWOR. Así mesmo realiza transformacións de documentos XML e buscas de elementos textuais na web ou en recursos XML/(X)HTML. Nas arquitecturas SOA resultan especialmente útiles para recuperar información de Bases de datos e presentala a través de Servizos web.
- ✓ **XForms.** Linguaxe de definición de Interfaces de usuario desenvolvida polo W3C, centrada especialmente na parte de formularios web e a súa integración en documentos (X)HTML, ODF ou SVG. Aplícase o paradigma de separar o contido, propósito e estrutura. En aplicación do MVC incorpora un modelo declarativo de composto de regras e validación para datos e tipos de datos dos formulario, así como envío de parámetros; unha capa de vista



composta dos controis da interface de usuario; un controlador para orquestrar as manipulacións de datos, interaccións entre o modelo e a vista e envíos de datos. Outras evolucións desta tecnoloxía serían AJAXForms e XSLTForms, incorporando AJAX e XSLT a esta tecnoloxía. Por outra banda, existen outras linguaxes relacionadas coas interfaces de usuario que seguen dialectos de XML, moitas delas con capacidade para interactuar con XForms como: XAML, XUL, UIML, UsiXML, AUIML, ...

O grupo de tecnoloxías anteriores poderían definirse como linguaxes XML de propósito xeral. En moitas ocasións o XML emprégase de xeito concreto para representación de datos complexos ou con necesidades específicas para o noso dominio. Na táboa 2 recóllense algúns exemplos de linguaxes XML empregadas para representación ou adaptación de información a necesidades concretas, ben para contornos de traballo como XHTML e WML ou ben en dominios específicos como aplicacións de información xeográfica ou deseño gráfico.

	<b>Función</b>
<b>XHTML</b>	HTML con especificacións máis estritas para presentar unha maior compatibilidade coa web semántica e os outros estándares XML
<b>MathML</b>	Expresar formulacións matemáticas
<b>SVG</b>	Especificación para describir gráficos vectoriais e animacións
<b>SMIL</b>	Permitir a integración multimedia en XHTML e SVG
<b>WML</b>	Adaptación do HTML para móbiles e PDA
<b>VoiceXML</b>	Converter fala en XML a partir de gramáticas de recoñecemento de voz
<b>SSML</b>	Para fala sintética
<b>GML/KML</b>	Para sistemas de modelado e información xeográfica



	Función
<b>X3D</b>	Representación de gráficos en 3D
<b>EBML</b>	Para almacenar xerarquías de datos en formato binario de lonxitude variable

### ***Táboas 2: Linguaxes XML complementarias.***

O uso tan estendido do XML obriga a dispoñer de ferramentas que permitan o tratamento doado dos documentos, percorrer, manipular, procesar, etc... Moitas tecnoloxías dispoñen de *frameworks* específicos para **tratamento de XML**:

- ✓ **DOM** (en inglés *Document Object Model*). Especificación do W3C dunha API ([org.w3c.dom](http://org.w3c.dom)) para manipular documentos XML/HTML, acceder ao seu contido, estrutura e estilos, a través dun analizador sintáctico. DOM xera unha árbore xerárquica en memoria onde almacena todo o documento. A través dun procesador permítese acceder a calquera nodo da árbore, ou inserir/eliminar novos nodos. O principal inconveniente deste modelo é que precisa gran cantidade de memoria pola necesidade de cargar todo o documento, pero ten as vantaxes de ser moi sinxelo de implementar e de permitir a xeración de XML. Apoiase en tecnoloxías XSLT e Xpath. Frameworks como Xerces baséanse en DOM para tratamento de XML así como outros baseados en AJAX do tipo de JQuery, Prototype, Dojo, etc... Así mesmo existen alternativas recentes similares a DOM, deseñadas explicitamente para JEE que resultan máis doadas de empregar, JDOM ([org.jdom](http://org.jdom)) e DOM4J ([org.dom4j](http://org.dom4j)).
- ✓ **SAX** (en inglés *Simple API for XML*). API inicialmente para Java ([org.xml.sax](http://org.xml.sax)), pero que despois evolucionou a outras linguaxes coma C++, Perl, Python, ... , que dispón dun analizador que xera eventos ao acadar puntos chave do documento analizado. Percorre o documento de xeito secuencial a través dun administrador de eventos, o



DocumentHandler, evento a evento, co cal non precisa cargar o documento en memoria pero non permite volta atrás sen ir de novo ao inicio. Isto o fai moi axeitado para documentos de gran tamaño. Outros *frameworks* máis completos baséanse á súa vez en SAX, como: Xerces, Crimson, Piccolo ou Oracle XML Parser.

- ✓ **StAX** (en inglés *Streaming API for XML*). Define un analizador sintáctico de fluxo de datos integrado en JEE, con soporte para xeración de XML. Empréganse dous estilos de análise Cursor API e Iterador Event Iterator API, ambos baseados en iteracións para solventar as limitacións de SAX e DOM. Neste modelo o documento XML transmítese nun fluxo de datos onde o se vai solicitando o seguinte evento (*Pull*) co fin de optimizar recursos de memoria. Distínguese entre Streaming Pull Parsing onde o cliente só obtén os datos solicitados previamente (SAX) e o Streaming Push cando o analizador envía ao cliente datos do XML ao localizar un elemento.
  
- ✓ **JAXP** (en inglés *Java API for XML Processing*). API de Java (javax.xml.parsers e javax.xml.transform) que proporciona acceso a través de dúas factorías abstractas para traballar con instancias de analizadores DOM e SAX a través de diferentes implementacións, así como soporte para StAX, espazos de nomes e XSLT (Xalan). Tamén leva incorporado o analizador Crimson. Acostuma integrarse en contornos con Servizos web para agrupar nun mesmo *framework* todas as posibilidades de tratamento de XML.
  
- ✓ **JAXB** (en inglés *Java Architecture for XML Binding*) . API JEE (javax.xml.bind) que proporciona un conxunto de interfaces para analizar e xerar XML de xeito automático. A partires do modelo definido en XML realiza a xeración de clases Java equivalentes. O esquema acostuma definirse vía DTD, a partir do cal un desenvolvedor pode construír unha árbore de obxectos Java que se



corresponden co XML. Deste xeito evítanse as limitacións de memoria de DOM.

Paralelamente dispórase de compoñentes específicos para contornos baseados en **Servizos web** coma WSDL, os máis habituais serían:

- ✓ **SAAJ**. API (javax.xml.soap) de SOAP e SOAP con achegas (en inglés *SOAP with Attachments*) que permite enviar documentos XML e achegas en formato MIME que poden ser ou non XML. Acostuma empregarse a baixo nivel por outras API para operacións de mensaxería.
- ✓ **JAX-RPC**. API de JEE para facilitar o desenvolvemento de compoñentes software que fagan uso de XML para comunicacións a través de chamadas a procedementos remotos (RPC), na liña de IDL-CORBA e RMI. A diferenza destas alternativas JAX-RPC emprega XML como soporte a Servizos web. Permite correspondencia entre obxectos e estruturas XML. En arquitecturas SOA o JAX-RPC sería a tecnoloxía a través da que o cliente envía a petición de servizo. Por debaixo emprega SOAP, pero este nivel permanece transparente á API. As súas funcións abarcan: Mensaxería asíncrona, Enrutamento de mensaxes, Mensaxería con entrega garantida.
- ✓ **JAXR** (en inglés *Java API for XML Registries*). API de JEE para acceso a rexistros de servizos en estándares abertos como ebXML ou UDDI. Permite aos servizos a posibilidade de auto-rexistrarse. Así mesmo soporta o uso de consultas SQL para a busca de rexistro a través do obxecto SQLQueryManager. Fai uso de JAXM para mensaxería.
- ✓ **JAX-WS** (en inglés *Java API for XML Web Services*). Compoñente do servizo web base Metro, que sería evolución e ampliación de JAX-RPC e se atoparía integrado con JEE (javax.xml.ws). Fai uso de anotacións Java para describir elementos das clases, como metadatos e



permiten automatización de tarefas.

Por último dentro do ámbito da **seguridade**, existen unha serie de solucións derivadas da capacidade insuficiente ante arquitecturas SOA de TLS ou SSL. Deste xeito cómpre destacar as seguintes tecnoloxías:

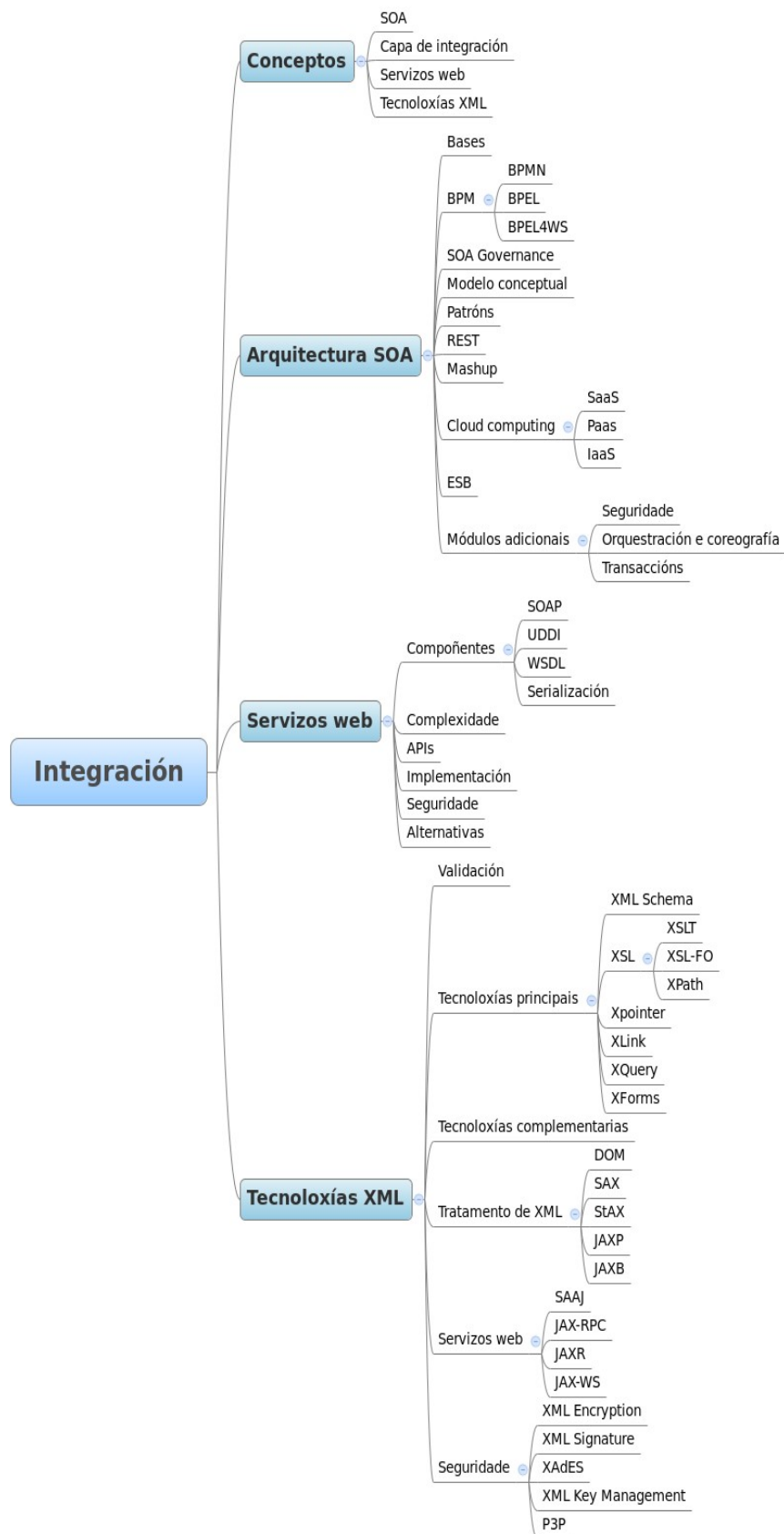
- ✓ **XML Encryption.** A encriptación XML é unha recomendación do W3C que especifica o proceso para cifrar datos ou documentos completos e representar esa información encriptada nun documento XML. Permite supercifrado e soporta os algoritmos TripleDES, AES e RSA.
- ✓ **XML Signature.** Sinatura dixital que garante a integridade das partes nunha comunicación. Así mesmo proporciona autenticación de mensaxes, integridade de datos, soporte de transaccións sen repudio e sinaturas para calquera contido dixital ou XML. No documento engádese un elemento Signature que encapsula o contido da sinatura dixital incluíndo unha referencia ao obxecto asinado, a indicación do algoritmo de canonización, e o valor resultante da sinatura.
- ✓ **XAdES** (en inglés *XML Advanced Electronic Signatures*). Sinatura dixital avanzada XML, que engade un conxunto de extensións a XML Signature, permitindo por exemplo que as sinaturas sexan válidas durante longos períodos de tempo
- ✓ **XML Key Management.** Protocolo para distribuír e rexistrar chaves públicas e certificados evitando a complexidade de PKI. Está composto de dúas partes: X-KRSS ou rexistro de chave pública, un conxunto de protocolos que soportan o rexistro de pares de chaves; e X-KISS información de chave pública, que define un conxunto de protocolos para procesamento e envío de información asociada en identificada con XML Signature e cifrada con XML Encryption.
- ✓ **P3P** (en inglés *Platform for Privacy Preferences*). Especificación do



W3C que define un estándar de xestión de datos e de privacidade, así como un formato XML para expresar políticas de privacidade, co obxectivo de permitir aos usuarios se e como queren revelar información persoal.

#### **28.4. ESQUEMA**







## **28.5. REFERENCIAS**

Varios autores.

Web Services Architecture. W3C Working Group. (2004).

César de la Torre e Roberto González.

Arquitectura SOA con tecnología Microsoft. Buenas prácticas y diseño de aplicaciones empresariales. (2008).

Joan Ribas Lequerica.

Web Services. (2003).

Patrick Cauldwell e outros.

Servicios Web XML. (2002).

**Autor: Juan Marcos Filgueira Gomis**

**Asesor Técnico Consellería de Educación e O. U.**

**Colegiado del CPEIG**





## **29. PATRÓNS DE DESEÑO E FRAMEWORKS. MVC, JSF. ANTIPATRÓNS.**



## TEMA 29. PATRÓNS DE DESEÑO E FRAMEWORKS. MVC. JSF. ANTIPATRÓNS.

### 29.1 INTRODUCCIÓN E CONCEPTOS

### 29.2 PATRÓNS DE DESEÑO E FRAMEWORKS

### 29.3 MVC

### 29.4 JSF

### 29.5 ANTIPATRÓNS

### 29.6 ESQUEMA

### 29.7 REFERENCIAS

### 29.1 INTRODUCCIÓN E CONCEPTOS

Para moitos problemas de deseño que se repiten tanto en desenvolvementos software como en implantacións hardware existen solucións comúns de aplicación dentro do mesmo contexto. Estas solucións recorrentes denomínanse **patróns de deseño** e se basean no concepto de reutilización e aproveitamento de solucións xa existentes en problemas novos. Os patróns segundo o autor acostuman a dividirse en diferentes familias, sendo a clasificación máis habitual en patróns de deseño, de arquitectura e interacción. Á súa vez dentro dos patróns de deseño clasifícanse segundo creacionais, estruturais e de comportamento. Así mesmo defínense os patróns de programación como patróns específicos para linguaxes de programación ou sistemas concretos. Cómpre sinalar que nunha mesma solución ou deseño pode convivir calquera número de patróns que sexa necesario, xa que en moitos casos trátase de solucións parciais a problemas concretos non de solucións xerais. Fronte ao concepto de patrón xorde o de **antipatrón** que definen erros de deseño comúns ou problemas que se repiten a miúdo para axudar a identificalos. Os patróns e antipatróns poden definirse perante linguaxes de definición do estilo da



Linguaxe Unificada de Modelado ou **UML** (en inglés *Unified Modeling Language*), que soportado polo OMG é un dos máis empregados actualmente.

Nas aplicacións JEE e .NET un dos patróns de uso máis estendido é o Modelo Vista Controlador ou **MVC** (en inglés *Model View Controller*) que se basea na separación dunha aplicación en tres capas ou compoñentes diferenciados, interface de usuario, lóxica de negocio e sistemas de información.

Este patrón intégrase en **frameworks**, ou compoñentes software que implementan funcionalidades comúns a conxuntos de aplicacións, e que poden seguir os modelos de patróns de deseño. O patrón sería a solución de deseño abstracta e o *framework* unha implementación do mesmo concreta. Algúns *frameworks* como Struts representan o esqueleto dunha aplicación, con implementación do patrón MVC entre outros, aportando así directamente todas as funcionalidades precisas para o seu funcionamento interno. Se nunha mesma aplicación engadimos novos *frameworks* dispoñemos de funcionalidades engadidas. Deste xeito o *framework JSF* (en inglés *Java Server Faces*) proporciona, complementariamente a Struts ferramentas que facilitan o desenvolvemento de interfaces de usuario.

## **29.2 PATRÓNS DE DESEÑO E FRAMEWORKS**

As principais vantaxes do emprego de patróns en solucións software pasan por facilitar a comunicación interna entre compoñentes, aforrar tempo e outros recursos, mellorar a calidade das operacións de todo o ciclo de vida de desenvolvemento e facilitar a aprendizaxe. A día de hoxe é un feito consumado que a súa aplicación correcta reporta un beneficio directo en calquera desenvolvemento ou implantación.

### **29.2.1 Clasificación xeral**



Existen moitas clasificacións dos patróns, segundo o autor(es) pero a máis habitual fai referencia ao ámbito de aplicación do patrón tomando como referencia a enxeñaría do software:

- 1) **Patróns de deseño.** Proporcionan un esquema de aplicación en partes dun sistema software. Definen estruturas que resoven un problema de deseño de utilidade en diferentes aplicacións.
- 2) **Patróns de arquitectura.** Proporcionan un esquema ou organización estrutural para definir sistemas completos ou subsistemas incluíndo responsabilidades e relacións entre sistemas.
- 3) **Patróns de interacción.** Proporcionan un deseño de interface para aplicacións ou aplicacións web.
- 4) **Patróns de programación** (en inglés *Idioms patterns*). Patróns a baixo nivel para linguaxes de programación ou tecnoloxías específicas. Definen representacións de implementacións de compoñentes e relacións considerando funcionalidades propias de cada linguaxe.

### **29.2.2 Clasificación de patróns para tecnoloxías de servidores de aplicacións**

A maiores foron xurdindo patróns para outros ámbitos de aplicación, como programación multifío, fluxos de traballo para procesos de sistemas empresariais, arquitecturas SOA ou integración de sistemas. En definitiva, pode concluírse que o concepto de patrón pode estenderse a calquera problema que nos atopemos e o nivel de abstracción que precisemos na solución. Existen diferentes catálogos de patróns, sendo os máis coñecidos:

- ✓ **GoF**, (en inglés *Gang of Four*) para problemas de deseño. (1995).
- ✓ **POSA**, (en inglés *Pattern Oriented Software Architecture*) para solucións en arquitecturas SOA. (1996).



- ✓ **J2EE**, para solucións específicas desta tecnoloxía. (2003).
- ✓ **PoEAA** (en inglés *Patterns of Enterprise Application Architecture*). Para sistemas complexos en arquitecturas empresariais distribuídas en capas. (2003).
- ✓ **GRASP** (en inglés *General Responsibility Assignment Software Patterns*). Patróns xerais para asignación de responsabilidades e transicións. (2005).

En concreto para o ámbito dos servidores de aplicacións como .NET e JEE en canto a arquitectura e análise poden destacarse os seguintes atendendo ao seu nivel de utilización:

- a) **Patrón de análise Party (Grupo)**. Agrupa as responsabilidades similares dos tipos de colectivos dunha organización nun supertipo. Emprégase para facilitar o modelado de estruturas en organización, sendo cada tipo unha organización, empresa, rol ou papel e almacenar os datos persoais de cada membro. Situacións especiais obrigan a adaptacións deste patrón como ocorre no Party Type Generalizations que permite a xeneralización de tipos de grupo que herdan dun subtipo, por exemplo para unha persoa ten varios roles a un tempo.
- b) **Patrón de análise Accountability**. Establece unha relación de responsabilidade entre dúas partes ou perfís. Cos tipos Accountability e Accountability Type permite expresar a clase de relación entre ambos. Pode facer uso do patrón Party para obter unha maior flexibilidade. Segundo sexan as relacións pode dar lugar a patróns máis complexos como Hierarchic Accountability ou Xerarquía de responsabilidade que engade restricións aos elementos de responsabilidade; ou que ten aplicación á hora de delegar tipos de responsabilidade a un subpatrón Party.
- c) **Patrón arquitectónico MVC** (en inglés *Model View Controller*).



Estrutura un compoñente software en 3 capas, o modelo coa lóxica de negocio, funcionalidades e sistemas de información, a vista coa interface de usuario e o Controlador que recibe os eventos da entrada e coordina as actividades da vista.

- d) **Patrón arquitectónico PAC** (en inglés *Presentation Abstraction Control*). Similar ao MVC este patrón define un sistema interactivo baseado nunha xerarquía de axentes cooperantes que realizan funcionalidades concretas. Divídese en tres capas: Presentación con interacción persoa-máquina, Abstracción coa lóxica e sistemas de información e o Control que centraliza as comunicacións entre axentes, procesa eventos externos e actualiza o modelo. A principal diferenza co MVC radica en que se poden facer diferentes axentes ou subsistemas de aplicación, operando de forma independente ou xerarquizada.
- e) **Patrón arquitectónico Capas** (en inglés *Layers*). Representaría a abstracción xenérica dos patróns anteriores a un sistema multicapa, orientado cara a distribución xerárquica de roles e responsabilidades. Permite aumentar ou diminuír o nivel de abstracción, máis ou menos capas, illando o mantemento e actualización de cada capa. Cada nivel ou capa ofrece servizos á capa superior e usa os da inferior.
- f) **Patrón arquitectónico Pipes and Filters**. Orientado tamén a arquitecturas SOA, neste modelo cada compoñente posúe un conxunto de entradas e saídas. Representa a lectura de fluxos de datos, transformándoos nun fluxo de saída sen ter que procesar toda a entrada, como ocorre nos modelos Streaming e de aí que se denominen Filtros aos compoñentes que reciben as entradas e tuberías ou condutos aos que encamiñan o fluxo cara a saída. Permite representar procesamentos en paralelo así coma execución concurrente.
- g) **Patrón arquitectónico Blackboard**. Proporciona un modelo de solucións aproximadas, cando non se pode aplicar unha solución



concreta. Permite reutilizar as fontes de coñecemento e un mellor soporte de cambios e mantemento da solución aproximada.

- h) **Patrón Microkernel.** Dentro dos patróns para sistemas adaptables, este modelo separa un kernel funcional mínimo do estendido para soportar sistemas software con requirimentos que cambian ao longo do tempo. Ideado para sistemas operativos, cada un deles sería unha vista do Microkernel central, permitindo que se poida estender o sistema de xeito doado.
- i) **Patrón Reflection.** Outro patrón sistemas adaptables que modela un mecanismo para mudar a estrutura e comportamento dun sistema dinamicamente. Establece dous niveis: Metadatos para que os software leve unha descrición de si mesmo e Lóxica de aplicación. Os cambios de comportamento poden reflectirse nos metadatos, pero isto pode pasar inadvertido.
- j) **Patrón arquitectónico Broker.** Orientado a arquitecturas SOA e sistemas distribuídos onde varios clientes fan peticións a un servidor ou servizo remoto. O axente Broker encárgase de coordinar a comunicación entre o cliente e o provedor do servizo. As principais vantaxes deste patrón son permitir a transparencia de localización do servizo, permitir cambios e ampliación de novos compoñentes sen que o sistema se vexa afectado, mellora da portabilidade e interoperabilidade con outros axentes Broker.
- k) **Patrón Publisher Subscriber.** Orientado a arquitecturas SOA e sistemas distribuídos insire unha capa entre clientes e servidores que se encarga de levar conta da comunicación de xeito transparente. Representa unha arquitectura de mensaxería sen acoplamento.

No tocante ao deseño, os principais patróns acostuman a agruparse en tres grandes categorías: Patróns creacionais, estruturais e de comportamento. Os **creacionais** incluírían:



- a) **Abstract Factory.** Prove unha interface que permite a creación de familias de obxectos dependentes ou relacionadas sen ter que especificar as clases completas. Exemplos deste patrón serían os Widgets e compoñentes de interfaces gráficas.
- b) **Builder.** Construtor virtual que separa a construción dun obxecto complexo da súa representación, de tal xeito que se obteñen diferentes representación nun mesmo proceso.
- c) **Factory Method.** Patrón que define unha interface para a creación de obxectos deixando que as subclases decidan que clase instanciar, facendo que o proceso de xeración do subtipo sexa transparente ao usuario.
- d) **Prototype.** Permite a creación de novos obxectos clonándoos dunha instancia dun obxecto xa existente.
- e) **Singleton.** Patrón de instancia única que asegura que dunha clase só existirá unha única instancia definindo un punto de acceso común á mesma.
- f) **Object Pool.** Patrón para a obtención de obxectos por clonación. Crease unha instancia dun tipo de obxecto da clase a clonar. Está pensado para casos onde a creación teña un custo moi alto e se permita a utilización de obxectos xenéricos do Pool.

Por outra banda, dentro dos **estruturais**:

- a) **Adapter.** Patrón que convirte a interface dunha clase noutra interface adaptada a necesidades específicas como determinados clientes ou interfaces requiridas por compatibilidade.
- b) **Bridge.** Ou patrón Handle/Body, separa unha abstracción da súa implementación de xeito que ámbalas dúas podan mudar de forma de maneira independente, sen que cambios nunha afecten a outra.
- c) **Composite.** Patrón que permite manipular obxectos compostos coma



se de un simple se tratase. Fai uso da composición recursiva e a estruturas en forma de árbore para poder presentar unha interface común.

- d) **Decorator**. Responde á necesidade de engadir funcionalidades a obxectos dinamicamente. Crea unha xerarquía de clases onde as fillas herdan da nais as funcionalidades e incorporan as súas propias.
- e) **Facade**. Proporciona unha interface común de acceso a un conxunto de interfaces dun sistema. Facilita o emprego do sistema interno con outras interfaces de alto nivel. Os clientes só poden comunicarse a través da interface única que fai de fachada.
- f) **Flyweight**. Permite eliminar a redundancia entre obxectos que presentan a mesma información. Factoriza os atributos comúns a estes obxectos nunha clase lixeira.
- g) **Proxy**. Proporciona un punto de control de acceso ou intermediario para o control doutro(s) obxecto(s). Presenta diferentes niveis de aplicabilidade:
  - ✓ *Proxy remoto*. Representa a un obxecto remoto de xeito local, codificando a petición e argumentos antes de enviala ao obxecto remoto.
  - ✓ *Proxy virtual*. Crea obxectos de alto custe baixo demanda, con posibilidade de caché da información dos mesmos limitando os custes de acceso.
  - ✓ *Proxy de protección*. Controla o acceso a obxectos remotos comprobando que os clientes dispoñen dos permisos necesarios.
  - ✓ *Proxy de referencia intelixente*. Análogo a un punteiro con operacións adicionais sobre un obxecto para temas de concorrencia, acceso a memoria, etc...

O último bloque serían os patróns de comportamento:

- a) **Chain of responsibility**. O patrón cadea de responsabilidade



permite establecer a liña que deben levar as mensaxes, denominada cadea de obxectos receptores, permitindo que varios obxectos podan capturar unha mensaxe, como pode ser unha excepción Java. Calquera dos receptores podería responder á petición segundo o criterio establecido.

- b) **Comando ou Orde.** Patrón que encapsula unha operación nun obxecto, de xeito que se poidan facer operacións estendidas como almacenamento e colas de peticións e soporte de accións de facer e desfacer.
- c) **Intérprete.** Define unha representación para a gramática dunha linguaxe xunto co seu intérprete.
- d) **Iterator.** Patrón co obxectivo de permitir percorrer obxectos compostos como poden ser as coleccións sen necesidade de contemplar aspectos de implementación ou representación interna dos mesmos. Define unha interface onde se ofrecen diferentes métodos para percorrer o obxecto complexo.
- e) **Mediador.** Define un obxecto que facilita a interacción entre outros de distinto tipo, coordinando a comunicación entre eles. O obxectivo é encapsular a interacción deses obxectos para evitar o acoplamento entre eles.
- f) **Memento.** Representa o estado dun obxecto ou sistema complexo para permitir o seu almacenamento e modificación, de xeito que se poida restaurar volvendo a estados anteriores no tempo.
- g) **Observador.** Permite definir unha dependencia dun a moitos, de xeito que eventos ou modificacións de estado disparen a notificación dos cambios a todos os obxectos ou sistemas dependentes.
- h) **Estado.** Emprégase para permitir que un obxecto cambie o seu comportamento no caso de modificarse o seu estado. Deste xeito diferentes clases poden representar a un mesmo obxecto ao longo do



tempo.

- i) **Estratexia.** Permite definir unha familia de algoritmos ou métodos de resolución, permitindo seleccionar dinamicamente cales aplicar e que deste xeito sexan intercambiabiles.
- j) **Template Method.** Define o esqueleto dun algoritmo para unha operación, delegando partes do mesmo ás clases concretas. Deste xeito as subclasses poden redefinir pasos concretos do método de resolución.
- k) **Visitor.** Representa un algoritmo ou operación realizada sobre a estrutura dun obxecto, permitindo a definición de novas operacións sen altera o tipo dos elementos sobre os que se realiza a operación.

Nunha última categoría poderían incluírse os patróns propias de linguaxes de programación ou tecnoloxías concretas, sendo os **patróns JEE**, a maioría Core J2EE Patterns, aqueles cun uso máis estendido dentro do mundo dos servidores de aplicacións e os servizos web:

- a) **Intercepting Filter.** Intercepta as peticións da capa de presentación antes ou despois do seu procesamento permitindo realizar operacións sobre os datos como auditorías, comprobacións de seguridade, conversións ou validacións. Permiten conectarse en fervenza e activar ou desactivar sen que afecte ao funcionamento xeral dunha aplicación. Permite diferentes estratexias como: Custom Filter, Estándar, Base Filter e Template Filter.
- b) **Front Controller.** Centraliza o control das peticións da capa de presentación, dirixíndoas cara o compoñente axeitado para validación de parámetros, invocación de elementos da lóxica de negocio, etc... Un controlador encárgase de recoller as peticións e factorizar o código repetitivo.
- c) **View Helper.** Prove unha clase que engloba código común, con aplicación tanto para a capa de negocio como para a de



presentación. Cada vista contén código para formato, delegando as responsabilidades de procesamento nas clases de axuda implementadas como Java Beans ou Custom Tags. Así mesmo poden almacenar modelos de datos intermedios facendo adaptacións previas do negocio, como conversións ou validacións, o lóxico pola separación en capas é que estas operacións non sexan moi complexas.

- d) **Composite View.** Define unha xerarquía de vistas compostas de diferentes vistas particulares permitindo modificar as partes en tempo de execución e a partir de modelos. Deste xeito inclúense dinamicamente as vistas concretas en vistas compostas da aplicación a través dos mecanismos que dispoñen para tal efecto JSP e Servlets.
- e) **Service to worker.** Agrupa varios patróns a modo de *framework* para permitir combinar un controlador (Front Controller), e un Dispatcher ou controlador de vistas (View Helper), para manexar as peticións dos clientes e xerar a presentación dinamicamente como resposta. Os controladores solicitan o contido aos Helpers que enchen o modelo de negocio intermedio.
- f) **Dispatcher View.** Cunha estrutura similar á do Service to worker, neste modelo tanto Controlador como Dispatcher teñen responsabilidades máis limitadas xa que lóxica de procesamento e control da vista son básicas.
- g) **Business Delegate.** Permite a abstracción a implementación de compoñentes complexos como EJB ou JMS da capa de presentación. Deste xeito poden crearse clases Proxy que almacenen e encolen as peticións podendo proporcionar control de prioridades, xestión de excepcións ou caché. O patrón emprega un compoñentes denominado Lookup Service, responsable de ocultar os detalles de implementación do código de busca dentro da lóxica de negocio.



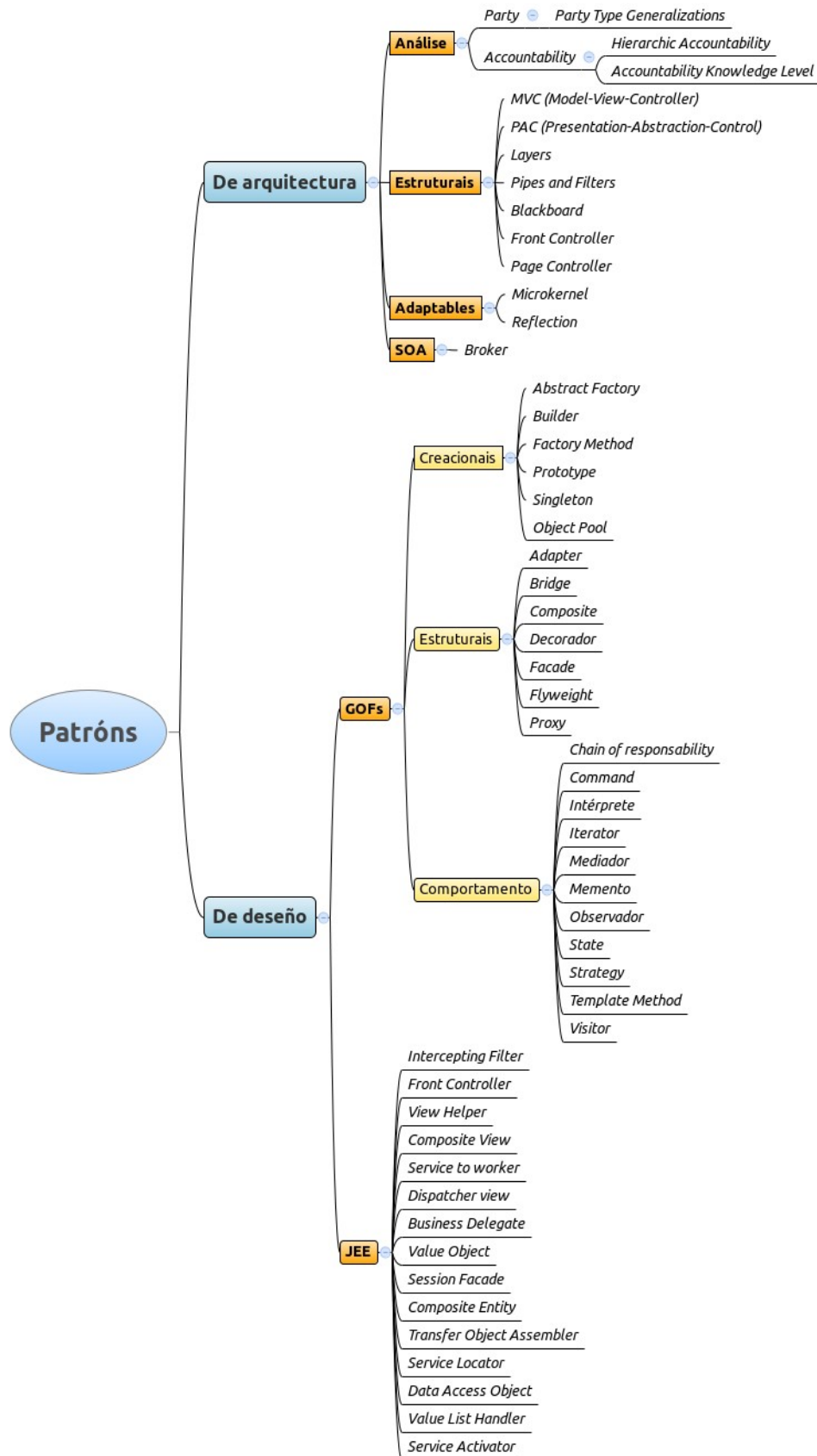
- h) **Value Object (VO).** Encapsula un conxunto de datos que representan un obxecto ou entidade do negocio. Cando se solicita a un Bean un conxunto de información este pode crear o obxecto Value Object e encher os seus atributos para devolvelo ao cliente.
- i) **Session Facade.** Emprega un Bean de sesión como fachada para encapsular as interaccións dos compoñentes de negocio e ofrecer un servizo de acceso uniforme, a través dos interfaces requiridos unicamente a través dos casos de uso. Proporciona unha abstracción de alto nivel implementada a modo de Bean.
- j) **Composite Entity.** Permite ampliar os Beans de entidade cando estes son de pequeno tamaño, deste xeito poden aumentarse mantendo a compatibilidade. O abuso deste patrón considérase un antipatrón xa que pode dar lugar a estruturas moi complexas. Un Bean Composite Entity representa un grafo de obxectos, por tanto debe empregarse con coidado.
- k) **Transfer Object Assembler.** Simplifica o acceso aos sistemas de información a través dun conector común. Cada obxecto de negocio terá un Transfer Object (TO) cos detalles de acceso a datos (Beans, JDO, JDBC, ...) e un Bean de sesión funcionará como interface común.
- l) **Service Locator.** Emprégase para abstraer a utilización de JNDI a través dun obxecto Service Locator e para ocultar as complexidades da creación do contexto inicial, así como da busca e instanciación de EJBs a través dun punto de acceso común.
- m) **Data Access Object (DAO).** Emprégase un obxecto como medio de acceso a sistemas de información, en especial Bases de datos. Abstrae e encapsula as operacións relacionadas coa tecnoloxía de persistencia empregada (JDBC, JDO, LDAP, Beans, TopLink, Hibernate, iBATIS, etc...). Controla os parámetros de conexión, obtención de datos e almacenamento proporcionando unha interface de acceso



común.

- n) **Value List Handler.** Implementado coma Beans de sesión, encárgase de manexar a execución de consultas SQL, cachealas e procesar os resultados. Accede directamente a un DAO que se encarga á súa vez de facer a conexión co sistema de información e recuperación dos datos. Unha vez obtidos almacénaos como TO ou VO permitindo ao cliente percorrelos grazas á implementación do patrón Iterador.
- o) **Service Activator.** Proporciona un modelo para mensaxería asíncrona como JMS. O Service Activator recibe as mensaxes e localiza e chama aos métodos dos compoñentes de negocio que se van encargar de resolver a petición.











**Figura 1: Resumo dos principais patróns en arquitecturas de servidores de aplicacións.**

A implementación destes patróns non acostuma a facerse a medida senón que se recorre aos **frameworks**. Moi relacionados entre si, os *frameworks* representan unha arquitectura de pequeno tamaño que proporciona unha estrutura xenérica integrando diferentes patróns de xeito que poidan ser reutilizados ou integrados de xeito doado nas aplicacións. Nun *framework* os patróns teñen unha implementación concreta sobre a definición abstracta do patrón. En última instancia son un conxunto de clases e interfaces que cooperan para ofrecer un software reutilizable.

### 29.3 MVC

O patrón Modelo-Vista-Controlador é o máis empregado para estruturar unha aplicación atendendo a unha correcta separación en capas: entrada, procesamento e saída. As súas principais **vantaxes** son unha redución do acoplamento, facilidade de desenvolvemento, claridade no deseño, mellora no mantemento, maior escalabilidade, unha maior cohesión con cada capa fortemente especializada, e unha maior flexibilidade e axilidade nas vistas, permitindo a súa modificación dinámica, sincronización, aniñamento e a existencia de múltiples vistas.

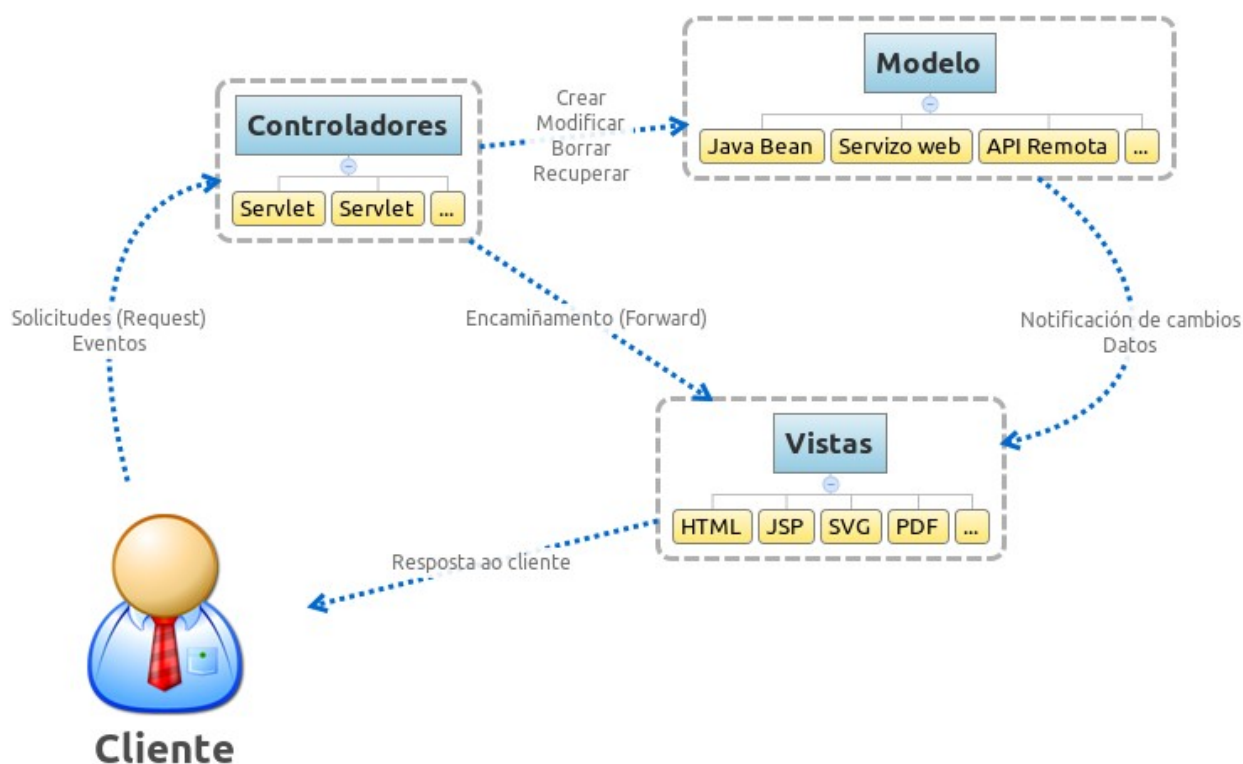
As **capas** do modelo concrétanse en:

- 1) **Modelo** (en inglés *Model*). Encapsula tanto datos como as funcionalidades ou casos de uso. Ten que funcionar independentemente de calquera representación que tomarán os datos na saída e calquera comportamento que se especifique na entrada do sistema. A todos os efectos será unha caixa negra que



recibe peticións de devolve resultados, encargándose de manexar os datos e controlar as súas transformacións. Normalmente implementa os patróns DAO, VO e Fachada.

- 2) **Vista** (en inglés *View*). Capa na que se integran todos os compoñentes que afecten á interface de usuario. Recibe as peticións do usuario e as envía cara o controlador, obtendo deste as respostas. Permítense múltiples vistas do mesmo modelo, pero toda a lóxica de presentación debe ir nesta capa.
- 3) **Controlador** (en inglés *Controller*). Recibe peticións da vista, tales como eventos, refrescos, etc... que recolle cun xestor de eventos ou Handler e son traducidos a solicitudes de servizos ou casos de uso, enviando as peticións ao modelo. A miúdo implementan patróns como Comando ou Front-Controller para encapsular as accións.



**Figura 2: Modelo-Vista-Controlador con tecnoloxías JEE.**



Como patrón de arquitectura o MVC pode conter á súa vez os seguintes **patróns** de deseño:

- ✓ **Observador.** Para prover o mecanismo de publicación e subscripción que permita notificar cambios do modelo nas vistas.
- ✓ **Composite View.** Para permitir a creación de vistas compostas nunha xerarquía.
- ✓ **Estratexia.** Para levar conta da relación entre as vistas e os controladores, xa que permite modificar dinamicamente aspectos do control.
- ✓ **Factory Method.** Para especificar ao controlador unha vista coma predeterminada.
- ✓ **Decorador.** Para engadir funcionalidades adicionais ás vistas.
- ✓ **Proxy.** Para distribuír a arquitectura en diferentes emprazamentos e mellorar características de rendemento.

O modelo MVC impleméntase tanto en *frameworks* .NET (Windows Forms, ASP .NET, Spring .NET, Maverick .NET, MonoRail, ...) como en JEE (Struts, Spring, Tapestry, Aurora, JSF, etc..). Así mesmo é un modelo que se atopa estendido a moitas outras tecnoloxías coma PHP, Ruby, Perl, Python, etc ...

Os *frameworks* que implementan o MVC acostuman presentar unha serie de **características xerais**, comúns a todos eles e que inclúen:

- ✓ Implementación de diferentes patróns de deseño orientados á reutilización de deseño e código.
- ✓ Controis de validación de campos de formularios.
- ✓ Control de erros e excepcións.
- ✓ Mensaxería e localización de cadeas de textos.
- ✓ Librarías de etiquetas ou compoñentes (TagLibs, Widgets, etc...)
- ✓ Compoñentes da Interface de Usuario como etiquetado de compoñentes de formularios, pestanas, controis AJAX, etc...
- ✓ Presentación de información a través de listados e táboas con paxinación.

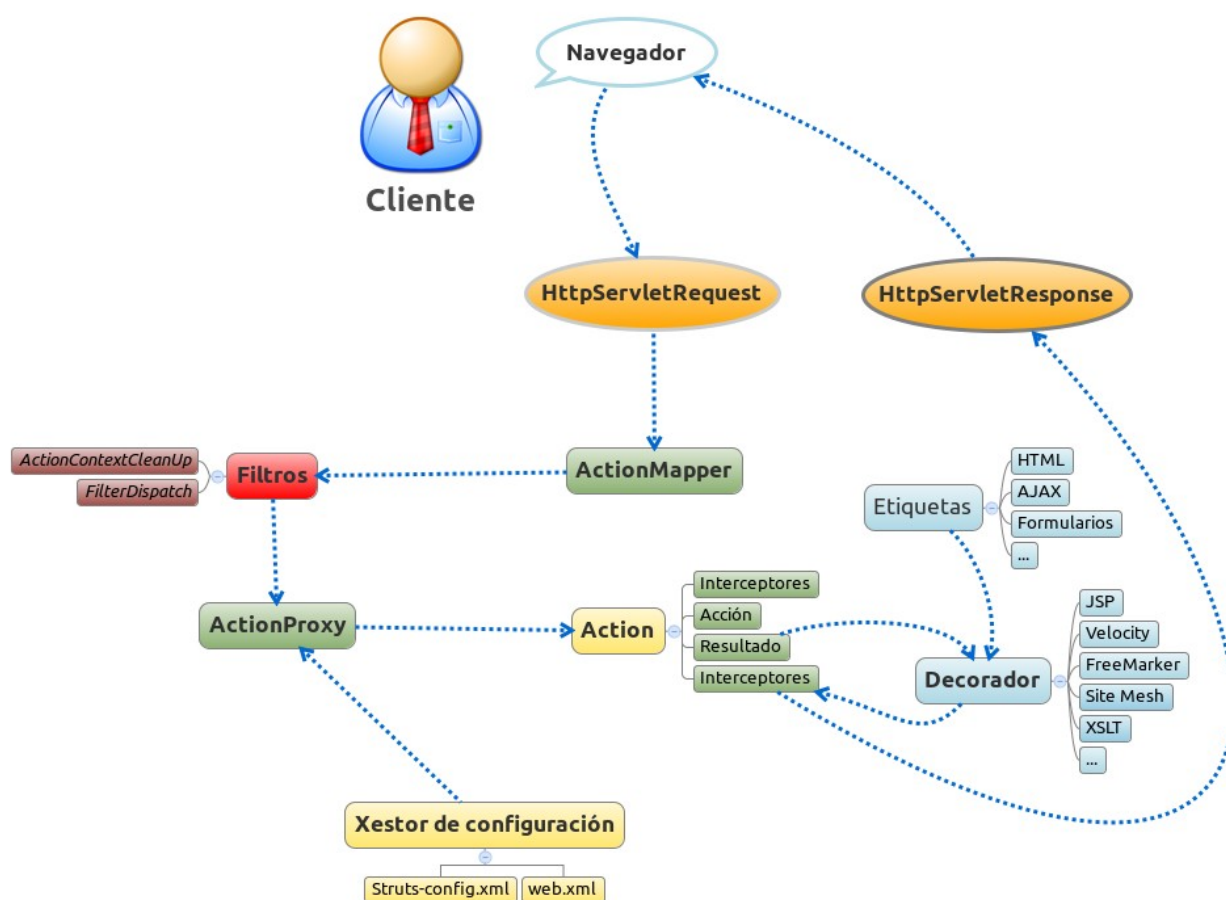


- ✓ Integración con *frameworks* co patrón Decorador ou baseados en modelos como Tiles, FreeMarker, Velocity, etc...
- ✓ Acceso datos en diferentes Sistemas de Información: Bases de datos, XML, etc...
- ✓ Abstracción de enderezos URL, Request e sesións.
- ✓ Autenticación e control de usuarios, roles e filtros.

Entre os frameworks que implementan o MVC destaca Apache Jakarta **Struts** (que ten unha evolución en Struts 2.0 ao fusionalo con WebWork), un dos máis empregados en tecnoloxías JEE e que resulta case un estándar de facto debido á súa integración noutros *frameworks* con máis funcionalidades. Emprégase para a implementación de aplicacións web baseadas en Servlets e JSP. Proporciona un conxunto de etiquetas JSP personalizadas (en inglés *Custom Tags*) que permiten encapsular funcionalidades na vista. Co modelo de Struts ten implantación directa o modelo MVC e outros patróns de deseño pre-construídos, permitindo a configuración directa de obxectos reutilizables perante a configuración de XML. Ademais proporciona as características anteriormente especificadas: validación, localización, modelos, etc...

Transporta automaticamente os datos inseridos polo cliente ata o controlador a través de Accións (en inglés *Actions*) mediante formularios ActionForms integrados no *framework* e vice versa para a súa presentación. Distingue entre unha parte común a calquera aplicación que faga uso do *framework* que fai de Controlador (ActionServlet) e outra parte configurable a través de arquivos de configuración en XML (struts-config.xml, web.xml, ...). A súa principal desvantaxe é non abarcar ata o nivel de acceso a datos, facendo que sexa necesario o emprego doutros *frameworks* especializados nesta capa para a elaboración de DAO, VO e outras operación complementarias.





**Figura 3: Funcionamento interno Struts/Struts 2.0.**

A principal alternativa a Struts sería **Spring** Framework, aínda que tamén permiten integración conxunta e con outros *frameworks* como JSF, Tapestry ou WebWork. Aínda que a súa orientación principal sexa a plataforma JEE, está dispoñible en .NET a través do *framework* Spring .NET. Ten soporte para JTA, JDO, JDBC e ODBC, e permite integración con terceiros como Acegi, Hibernate, iBatis e OJB. Como novidade permite programación orientada a aspectos ou AOP (en inglés *Aspect-Oriented Programming*) que busca empregar os servizos secundarios como seguridade, rexistro de log, manexo de transaccións, etc... das funcionalidades do modelo. Con AOP poden empregarse os servizos da aplicación de forma declarativa, ou perante arquivos XML de configuración ou mediante estándares JSR. Así mesmo realiza Inversión de Control ou IoC, que promove o baixo

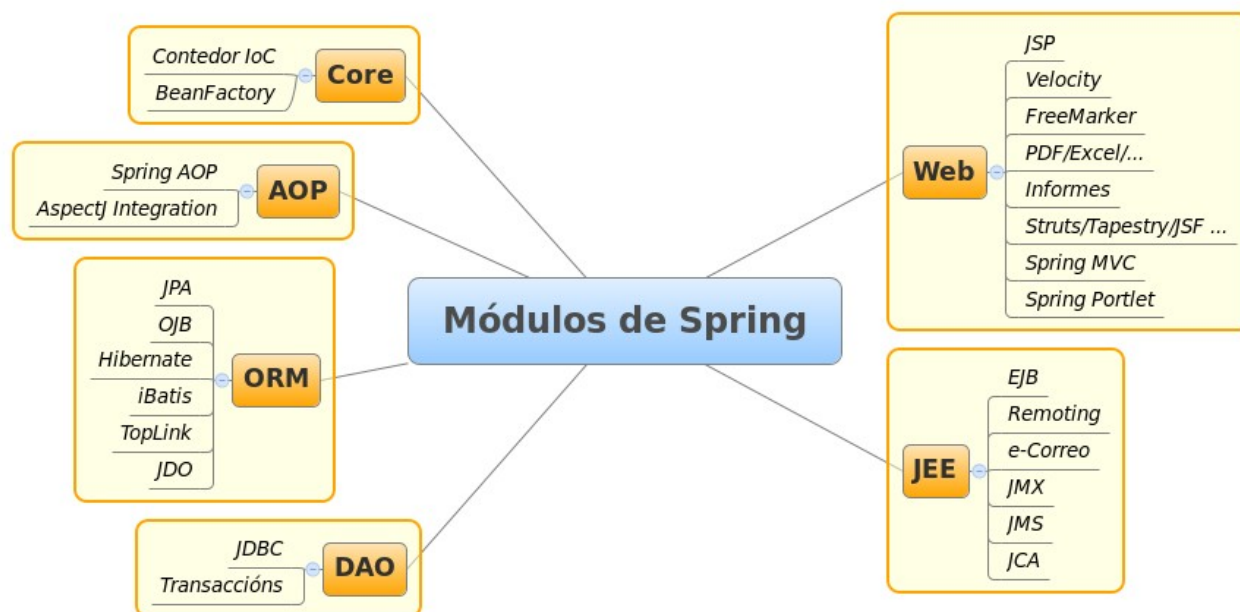


acoplamento a partir da inxección de dependencias entre obxectos. As principais desvantaxes de Spring son que implica unha configuración complexa, xa que cada servizo leva o seu XML propio, aínda que existe a alternativa do JSR. O seu contedor non resulta lixeiro o que impide que teña aplicación práctica nalgúns contornos como poden ser os dispositivos móbiles.

A **arquitectura de Spring** está composta polos seguintes compoñentes:

- ✓ **Core.** O núcleo que aloxa o contedor principal ou BeanFactory.
- ✓ **Módulo AOP.** Prove a implementación de AOP, permitindo desenvolver interceptores de método e puntos de ruptura para desligar o código do modelo das funcionalidades transversais.
- ✓ **Módulo DAO.** Prove a capa de abstracción de acceso a datos e sistemas de información sobre os diferentes conectadores dispoñibles. Ademais prove de manexo de transaccións vía AOP e outros servizos.
- ✓ **Módulo ORM.** Prove integración para as distintas API de correspondencia entre obxectos e entidades de bases de datos con soporte de diferentes tecnoloxías e integración con *frameworks* de terceiros.
- ✓ **Módulo JEE.** Integración con aplicacións e servizos JEE.
- ✓ **Módulo Web.** Aporta compoñentes especiais orientados a desenvolvemento web e integración con *frameworks* alternativos como Struts ou JSF, ademais dunha implementación do paquete Spring MVC.





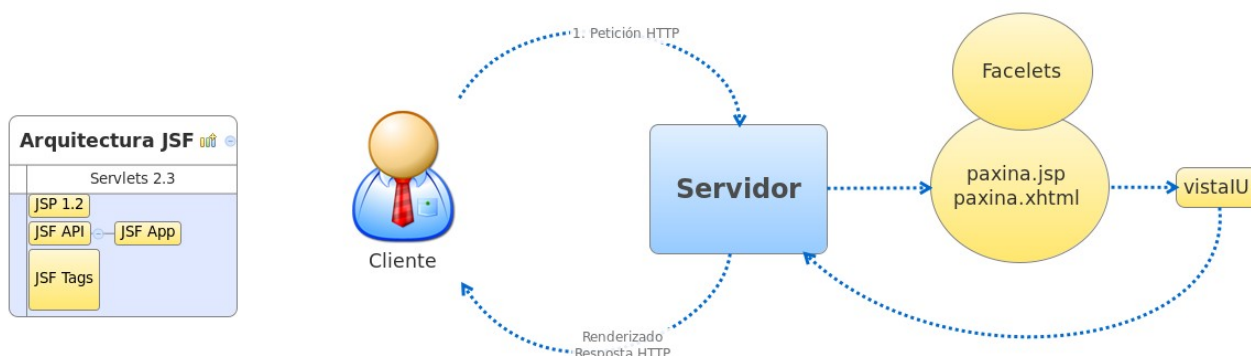
**Figura 4: Arquitectura de Spring.**

## 29.4 JSF

A tecnoloxía Java Server Faces proporciona un *framework* de interface de compoñentes de usuarios para o lado do servidor de aplicacións. Na súa base emprega JSP pero permite outras tecnoloxías para interfaces de usuario como XUL. Entres os **compoñentes** de JSF atópanse:

- 1) Un conxunto de APIs para representar e manexar compoñentes da interface de usuario. Entre as opcións que xestionaría atoparíanse control de estado e eventos, validacións de formularios, conversión de datos, control de navegacións e soporte de localización e accesibilidade.
- 2) Un conxunto de compoñentes da interface de usuario reutilizables.
- 3) Dúas librarías de etiquetas personalizadas (en inglés *Custom Tags*) para JSP.
- 4) Modelo de eventos para o lado do servidor.
- 5) Soporte para Managed Beans de control de eventos.





**Figura 5: Arquitectura JSF e funcionamento básico.**

Un dos compoñentes de JSF é o *framework* JavaServer **Facelets**, destinado á xestión de modelos (en inglés *templates*). As principais características deste *framework* son:

- ✓ Custe de tempo cero para o desenvolvemento de etiquetas de compoñentes da Interface de Usuario.
- ✓ Facilitade de creación de modelos de páxinas e compoñentes reutilizables.
- ✓ Soporte para UEL (en inglés *Unified Expression Language*) e validacións EL.
- ✓ Compatibilidade con calquera RenderKit.
- ✓ Intégrase plenamente con JSTL cousa que en JSF pode ocasionar problemas.
- ✓ Compilación máis rápida que con JSP.

Actualmente existen numerosas **implementacións** de JSF que poden complementar á especificación oficial JEE. Existe a posibilidade de combinar diferentes implementación nunha mesma aplicación, sendo as máis habituais:



- a) **MyFaces Tomahawk/Sandbox.** Desenvolvido por Apache proporciona un conxunto de compoñentes reutilizables compatibles coas especificacións JSF 1.1, JSF 1.2 e JSF 2.0.
- b) **Trinidad.** Subproxecto de MyFaces, a partir da inclusión dos compoñentes ADF Faces e outras melloras. Prove dos seguintes elementos: Unha implementación de JSF, varias librarías de compoñentes Widgets, a extensión MyFaces Orchestra e módulos de integración para outras tecnoloxías e estándares como MyFaces Portlet Bridge.
- c) **Tobago.** Outro proxecto baseado en MyFaces nunha aproximación do deseño de páxinas web ao de aplicacións de escritorio. Proporciona unha serie de compoñentes da Interface de Usuario como abstraccións do HTML. Presenta un conxunto de temas para clientes HTML con vistas independentes de HTML/CSS/Javascript.
- d) **ICEfaces.** Contén diversos compoñentes de interfaces de usuario enriquecidas baseadas en AJAX e compatibles con SSL, como editores de texto, reprodutores multimedia, etc... Soporta Facelets e Seam, ademais de ser compatible con Spring, WebWork e Tomahawk.
- e) **RichFaces.** Outro *framework* AJAX que inclúe ciclo de vida, validacións, conversións e xestión de recursos nas aplicacións. Soporta Facelets e Seam, ademais de ser compatible con Spring e Tomahawk.
- f) **Ajax4JSF.** Outra alternativa máis que proporciona un *framework* AJAX que inclúe ciclo de vida, validacións, conversións e xestión de recursos nas aplicacións. Soporta Facelets e Seam, ademais de ser compatible con Spring e Tomahawk. Inclúe os seguintes compoñentes:
  - ✓ *Ajax Filter.* Filtro de peticións para AJAX.
  - ✓ *Ajax Action Components.* Envían as peticións dende o cliente.
  - ✓ *Ajax Containers.* Interface que describe zonas dentro das JSP.



- ✓ *Javascript Engine*. Motor no lado do cliente que actualiza diferentes zonas das JSP en función da resposta AJAX.

## 29.5 ANTIPATRÓNS

Contrarios ao **concepto** de patróns, os antipatróns representan malos usos habituais, ou solucións que, sobre todo ao longo do tempo, presentan máis problemas dos que resolven, trátase en definitiva de malas prácticas. Existen dúas variantes principais, os que describen unha mala solución para un problema habitual e que produce consecuencias difíciles de arranxar ao longo do tempo; e aqueles que describen como poñer remedio a un problema e convertelo nunha boa solución. Por norma xeral os antipatróns vense como unha boa idea ao comezo, que falla de mala maneira á hora da súa implementación.

As **motivacións** ou razóns para ter en conta os antipatróns como caso de estudo atenden aos seguintes puntos:

- ✓ Permiten identificar solucións de risco para problemas habituais.
- ✓ Proven experiencia do mundo real para detectar problemas que se repiten ao longo do tempo, ofrecendo posibles solucións ou alternativas para as súas implicacións máis habituais.
- ✓ Proven dun marco común para a identificación e documentación dos problemas e deseño das solucións.

Como acontecía cos patróns, os antipatróns acostuman a agruparse en diferentes **categorías**, sendo as principais:

- 1) **Antipatróns de desenvolvemento software**. Definen problemas asociados ao desenvolvemento software a nivel de aplicación, ao



nivel dos patróns de deseño.

- 2) **Antipatróns de arquitectura de software.** Céntranse na distribución e relacións das aplicacións, servizos e outros compoñentes software a nivel de organización.
- 3) **Antipatróns de xestión de proxectos software.** Identifican escenarios críticos sobre a comunicación entre persoas e a resolución de problemas en equipos, vendo como afectan a un proxecto ou proceso software.

Así mesmo os antipatróns teñen aplicación en moitas outras áreas como metodoloxía, xestión da configuración, TDD, deseño web, accesibilidade, usabilidade, etc...

Dentro dos **antipatróns de desenvolvemento** software atopámonos entre os máis comúns:

- a) **Blob ou obxecto todopoderoso** (en inglés *God Object*). Emprégase un único obxecto, clase ou módulo para aglutinar un amplo conxunto de funcionalidades que deberían atoparse divididas. Con este patrón cáese nun código amplamente desorganizado e moi acoplado.
- b) **Fluxo de lava ou lava seca** (en inglés *Lava Flow*). Representa aqueles tipos de programación por impulsos ou erupcións de código, de xeito desestruturado, desorganizado e con pouca documentación. O sistema medra de xeito desproporcionado e pasado un tempo os bloques de código máis antigos considéranse metaforicamente solidificados no tocante á dificultade de solucionar calquera tipo de problema no que se atopen involucrados.
- c) **Descomposición funcional.** Deseño non orientado a obxectos, froito da migración dende linguaxes estruturadas a POO.



- d) **Poltergeists.** Ou clases pantasma debido ao descoñecemento dentro da aplicación de cal é o obxectivo dalgunhas clases, sendo en moitos casos súa única función transmitir información entre clases.
- e) **Martelo dourado.** Empregar a mesma solución para calquera problema que xorda, sen contemplar outras posibles alternativas.
- f) **Código spaghetti.** Fai referencia a código de aplicación cunha estrutura complexa e incomprendible con multitude de tecnoloxías mesturadas. A analoxía faise a partir das relacións entre o código que semellan un grande número de fíos mesturados e enrolados.
- g) **Programación copiar e pegar.** Solución na que en lugar de crear solucións xenéricas cópanse e adaptacións solucións xa existentes.

No tocante aos **antipatróns de arquitecturas** software destacan por ser os máis habituais:

- a) **Reinventar a roda.** Implementar compoñentes xa dispoñibles ou que poden aproveitarse con lixeiras modificacións. Dáse pola tendencia a facer todo un mesmo ou o descoñecemento da arquitectura e solucións dispoñibles no mercado ou alternativas de código aberto.
- b) **Vendor Lock-In.** Construír unha arquitectura dependente dun produto de terceiros, en especial cando se trata de software privativo. Ponse en perigo a escalabilidade do sistema e aumentan os custos de mantemento.
- c) **Illamento na organización.** Nunha mesma organización ou conxunto de sistemas créanse diferentes unidades illadas entre si que medran en paralelo solucionando problemas comúns de xeito



independente. Neste modelo pode medrar sobre maneira o custe de integración chegada a necesidade do mesmo.

- d) **Deseño por comité.** Demasiadas persoas participan dos requirimentos do proxecto dando lugar a un deseño demasiado abstracto e excesivamente complexo por mor de contemplar demasiados puntos de vista particulares. Complícase a toma de requisitos dando a lugar a demasiadas reunións de longa duración, que dificultan e provocan erros ao longo de todo o ciclo de vida de desenvolvemento.
- e) **Arquitectura por implicación.** Non existe documentación da arquitectura do sistema, nin dos procesos, nin das tarefas automatizadas máis habituais.

No tocante aos **antipatróns de xestión de proxectos** software destacan por ser os máis habituais:

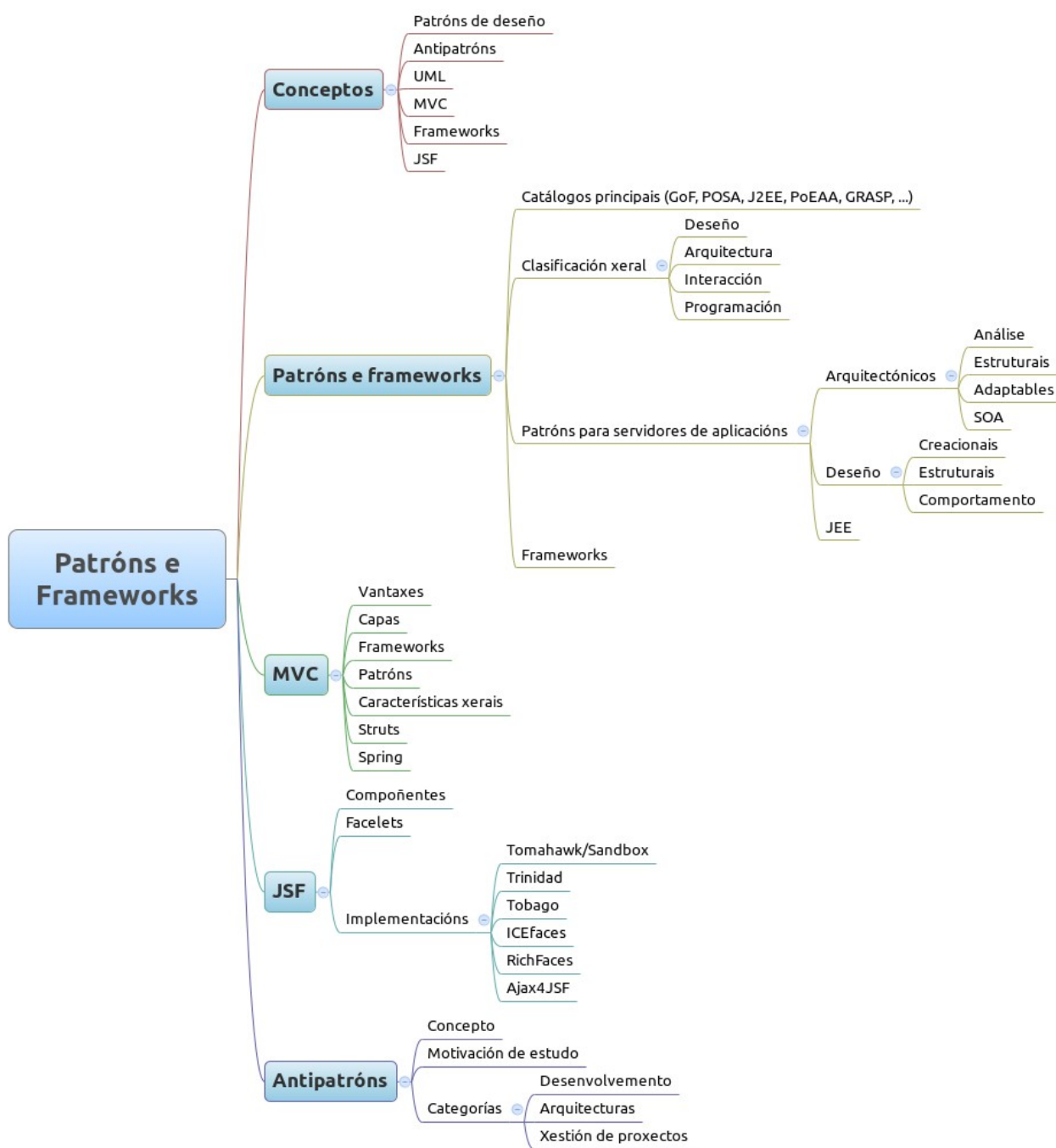
- a) **Parálise de análise.** Os procesos de análise e deseño prolóngase tanto que o proxecto remata morrendo nel sen chegar a implementarse. Son desenvolvementos opostos aos modelos baseados en prototipos e iterativos.
- b) **Morte por planificación.** Demasiada planificación e reunións sen chegar a concretar puntos de partida para o desenvolvemento. De novo son desenvolvementos opostos aos modelos baseados en prototipos e iterativos.
- c) **Persoas problemáticas** (en inglés *corncob*). Persoas difíciles de participar en equipos, ou con pouca capacitación ou aptitude, obstrúen, desvían e mesmo sabotan o desenvolvemento.
- d) **Xestión irracional.** A falta de decisión e capacitación sumado á nula planificación poden dar lugar á toma de decisións con posterioridade e desenvolvementos de urxencia.
- e) **Proxectos sen xestión.** Non se atende ao análise e o deseño, só a



implementación. Vanse arranxando incidencias segundo acontecen en modo pila, as últimas primeiro.

## **29.6 ESQUEMA**







## **29.7 REFERENCIAS**

Deepak Alur e outros.

Core J2EE Patterns. Best Practices and Design Strategies. (2003).

William Crawfor e Jonathan Kaplan.

J2EE Design Patterns. (2003).

Steven Metsker e 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). ENXEÑARÍA DO SOFTWARE.**



## TEMA 30. HTML. APLICACIÓNS DA INTERNET ENRIQUECIDAS (RIA).

### 30.1 INTRODUCCIÓN E CONCEPTOS

### 30.2 HTML

### 30.3 AJAX

### 30.4 RIA PARA MULTIMEDIA E ANIMACIÓNS

### 30.5 OUTRAS TECNOLOXÍAS RIA

### 30.6 ESQUEMA

### 30.7 REFERENCIAS

### 30.1 INTRODUCCIÓN E CONCEPTOS

Os documentos web e MIME poden definirse a través da linguaxe de marcado HTML, (en inglés *HyperText Markup Language*) que permite describir a estrutura e contido dun documento a través de etiquetas, ademais de permitir outros elementos complementarios como estilos ou *scripts*. Variantes desta linguaxe permiten un dominio máis estrito adaptado a ámbitos concretos como o XHTML ou MathML e máis abertos coma SGML (en inglés *Standard Generalized Markup Language*). A última versión da linguaxe HTML5 incorpora etiquetas específicas para contidos multimedia, grandes conxuntos de datos, melloras en formularios, web semántica e outras opcións, que responden á evolución das necesidades da web hoxe en día.

En extensión, as aplicacións de Internet enriquecidas ou **RIA** (en inglés *Rich Internet Applications*), son un conxunto de tecnoloxías que buscan achegar as interfaces das aplicacións web ás das aplicacións de escritorio dotándoas de novas funcionalidades, de aí a riqueza, e axilizando aspectos como as recargas de datos. Por norma xeral precisan dun *framework*, compoñente adicional ou *plug-in* no navegador que permitan a súa interpretación. Nas aplicacións RIA a maior parte da comunicación faise de



maneira asíncrona en comunicacións transparentes ao usuario que evitan gran parte das recargas de páxinas para realizar actualizacións de datos. Fronte a estas vantaxes no tocante á usabilidade en canto a mellora das funcionalidades e actualizacións de datos a principal desvantaxe será a accesibilidade da páxina para usuarios que presenten dificultades de acceso á información na web.

Moitas destas **tecnoloxías** pertencen ao mundo do software propietario atopando gran dependencia respecto das compañías que as desenvolven. As principais tecnoloxías atópanse nas plataformas Flash, Flex e AIR de Adobe, Silverlight de Microsoft, OpenLaszlo, incontables *frameworks* AJAX e Javascript, e outras tecnoloxías como as xa maduras, Applets e Java WebStart e as emerxentes como XUL, JavaFX, GWT ou Bindows. Calquera destas tecnoloxías localízase na capa de vista, usuario ou cliente como complemento á (X)HTML/CSS e intégranse coas tecnoloxías de servidores de aplicacións como .NET/JEE.

Dentro dos **casos de éxito** desta tecnoloxía, actualmente a maior parte das aplicacións e servizos web globais de maior uso dentro da Web 2.0 fan uso de tecnoloxías RIA nas súas interfaces, sendo Google Maps, Gmail, Flickr, Meebo, Orkut, e un longo etcétera.

## **30.2 HTML**

O conxunto de estándares HTML presentan un conxunto de normas definido polo W3C, sendo a versión estándar máis habituais a HTML 4.01, a recente HTML 5 e a variante XHTML 1.0. Un documento HTML describe a estrutura e contido dunha páxina ou sitio web, permitindo acceso ao DOM e facendo



uso de arquivos multimedia e *scripts* externos que á súa vez poden adoptar a forma de tecnoloxías RIA ou DHTML. Neste caso no documento HTML especificase un obxecto incrustado na tecnoloxía correspondente e calquera navegador identifica e reproducéoo segundo o correspondente Plugin. Grazas ao HTML pódese definir un documento cos seguintes elementos:

- Publicación de contidos en liña, como textos, encabezados, listas, táboas, parágrafos, formatos e estilos, etc...
- Asociación de documentos vía ligazóns de hipertexto.
- Deseño de formularios para permitir a entrada de información e comunicación con servidores remotos.
- Incrustación nos documentos de obxectos externos como elementos multimedia ou obxectos de aplicacións externas como follas de cálculo, documentos Pdf e outros.

Os documentos HTML especifican tamén a **codificación de caracteres** internacional do documento sendo UTF-8 ou Latin-1. Esta identificación da codificación permite que outras aplicacións, como navegadores ou procesadores de texto, poidan recoñecer e presentar adecuadamente a información do documento. Así mesmo dentro desta codificación, os caracteres especiais represéntanse a través de **entidades HTML**, que definen de maneira unívoca o carácter especial a través dun código HTML específico.

Carácter especial	Entidade HTML
Á	&Aacute;
á	&aacute;
Espazo en	&nbsp;



branco	
Ñ	&Ntilde;
ñ	&ntilde;
Ü	&Uuml;
>	&gt;
<	&lt;

***Táboa 1: Exemplos de entidades HTML.***

### **30.2.1 ACCESIBILIDADE WEB**

Outro aspecto asociado á definición de documentos web é o de **acesibilidade web**, onde o W3C elaborou as Pautas de Accesibilidade ao Contido Web 1.0 ou WCAG adaptadas no caso español coa Norma UNE 139803. Estas guías definen unha serie de regras e normas que deben cumprir os documentos web para permitir o acceso á súa información a persoas con discapacidade ou que presentan dificultades específicas.

As WCAG 1.0 son catorce pautas ou patróns de deseño web con solucións comúns para resolver problemas de acceso a contidos web. As pautas conteñen unha serie de puntos de verificación que facilitan a detección de posibles problemas de acceso. Cada un dos puntos ten asignado un nivel de prioridade ou peso segundo á importancia ou dificultade de acceso á información que leve asociado. Estes niveis son:

- **Prioridade 1.** Puntos que debe cumprir un sitio web ou doutro xeito non se permitirá que certos grupos de usuarios non poderán acceder aos contidos do sitio.
- **Prioridade 2.** Puntos que debe cumprir un sitio web ou doutro xeito a certos grupos de usuarios seralles moi difícil acceder aos contidos



do sitio.

- **Prioridade 3.** Puntos que debe cumprir un sitio web ou doutro xeito algúns grupos de usuarios poderían ter dificultades á hora de acceder aos contidos do sitio.

En función ao cumprimento destes puntos de verificación establécense tres niveis de conformidade dun sitio web coas pautas de accesibilidade:

- Nivel de Conformidade **A**. Cúmprense todos os puntos de verificación de prioridade 1.
- Nivel de Conformidade **Duplo A**. Cúmprense todos os puntos de verificación de prioridade 1 e 2.
- Nivel de Conformidade **Triplo A**. Cúmprense todos os puntos de verificación de prioridade 1,2 e 3.

Priori dade	Punto de verificación	Cumpri mento
<b>1</b>	Proporcionar texto equivalente a elementos non textuais	S/N/N.A.
<b>1</b>	Asegurar que a información transmitida con cor ten alternativa sen cor	S/N/N.A.
<b>1</b>	Identificación de cambios de idioma	S/N/N.A.
<b>1</b>	Organizar o documento independentemente das follas de estilo	S/N/N.A.
<b>1</b>	Garantir que as alternativas ao contido dinámico actualízanse cando este	S/N/N.A.
<b>1</b>	Non provocar destelos de pantalla	S/N/N.A.
<b>1</b>	Empregar unha linguaxe clara e simple no sitio	S/N/N.A.
<b>1</b>	Proporcionar ligazóns de texto redundantes en mapas de imaxes	S/N/N.A.
<b>1</b>	Nas táboas identificar as cabeceiras de fila e columna	S/N/N.A.
<b>1</b>	Titular cada marco ou <i>frame</i> da páxina	S/N/N.A.



<b>1</b>	Permitir o funcionamento sen linguaxes de scripts de cliente activados	S/N/N.A.
<b>1</b>	Prover alternativas auditivas sincronizadas en contidos multimedia	S/N/N.A.
<b>1</b>	No caso de ser imposible cumprir cos puntos anteriores cómpre crear un documento completo alternativo	S/N/N.A.
<b>2</b>	Garantir contraste de cor de texto e fondo lexible	S/N/N.A.
<b>2</b>	Empregar follas de estilo para maquetación e presentación	S/N/N.A.
<b>2</b>	Empregar documentos validados formalmente	S/N/N.A.
<b>2</b>	Empregar elementos de encabezado	S/N/N.A.
<b>2</b>	Empregar unidades relativas fronte a absolutas	S/N/N.A.
<b>2</b>	Marcar axeitadamente listas, ítems e citas	S/N/N.A.
<b>2</b>	Evitar navegación ou refresco de páxinas automáticos	S/N/N.A.
<b>2</b>	Evitar abrir novas ventás sen informar ao usuario	S/N/N.A.
<b>2</b>	Empregar índices ou mapas dos sitios e metadatos para aportar información semántica	S/N/N.A.
<b>3</b>	Proporcionar unha orde lóxica de tabulación	S/N/N.A.
<b>3</b>	Proporcionar atallos de teclado	S/N/N.A.
<b>3</b>	Incluír caracteres por defecto nos formularios	S/N/N.A.
<b>3</b>	Proporcionar abreviaturas para os encabezados das páxinas	S/N/N.A.
<b>3</b>	Proporcionar resumos para as táboas	S/N/N.A.
<b>3</b>	Proporcionar barras de navegación	S/N/N.A.

***Táboa 2: Exemplo de lista de puntos de verificación.***

Paralelamente desenvolvéronse as **Pautas de accesibilidade para ferramentas de autor** ou ATAG (en inglés *Authoring Tool Accessibility Guidelines* 1.0) co obxectivo de axudar aos desenvolvedores á hora de crear ferramentas de autor destinadas á elaboración de contidos web accesibles; as **Pautas de accesibilidade para XML** ou XAG; e as **Pautas de accesibilidade para axentes de usuario** 1.0 ou UAAG orientadas a definir un marco accesible de soporte a compoñentes software que



acceden a contidos web como navegadores, reprodutores multimedia, procesadores de texto e outras tecnoloxías asistivas. Por último dentro do marco das aplicacións RIA a iniciativa **WAI-ARIA** comeza a dar soporte a requirimentos de accesibilidade para este tipo de tecnoloxías.

Unha das regras de accesibilidade web é cumprir coa validez do documento. Para avaliala unha das etiquetas, DOCTYPE, permite especificar a versión de HTML empregada para elaborar o documento. Ademais de permitir corrixir problemas de compatibilidade con navegadores permite validar que o documento estea ben formado.

Existen tres **definicións do tipo de documento** ou DTD:

- **Strict.** Estrito, respectando escrupulosamente as normas e sintaxe do documento, e evitando parte do conxunto de etiquetas posible total.
- **Transitional.** Con máis liberdade, fai uso de todo o conxunto de etiquetas e permite licenzas na sintaxe.
- **Frameset.** Reservado para documentos que fan uso de marcas para organizar a estrutura do sitio web.

A partir desta especificación existen ferramentas de **validación** en liña e para descarga, que orientadas cara a accesibilidade ou cara a corrección do código permiten avaliar a calidade do documento.

### **30.2.2 HTML PARA DISPOSITIVOS MOBILES**

Así mesmo ao aumentar as posibilidades de conexión dos **dispositivos móbiles**, medrou a demanda de dar soporte web con esta orientación e por tanto de adaptar os documentos ás necesidades específicas deste tipo de equipos, moitas das mesmas moi ligadas ao ámbito da accesibilidade e



o deseño líquido. Os protocolos máis habituais **WAP**, **WAP 2.0** e **WCSS** recollen os DTD máis empregados. O W3C proporciona un subconxunto específico para móbiles o XHTML 1.1 Basic, e a OMA (en inglés *Open Mobile Alliance*) a especificación XHTML Mobile Profile, ambos garanten o soporte dos documentos que cumpran as validacións cos dispositivos máis restrinxidos. No caso de WAP non se atopan soportados os protocolos de Internet, polo que cómpre empregar nodos pasarela. Así mesmo dispón dunha linguaxe de *script* propia, **WMLScript** baseado en ECMAScript e empregado nas páxinas **WML** (en inglés *Wireless Markup Language*) que empregan á súa vez XML. A seguinte versión WAP 2.0 si adopta os protocolos de Internet, o que fai que non sexa imprescindible a pasarela, presentando ademais melloras de rendemento para a comunicación en dispositivos móbiles dentro destes protocolos.

Outras tecnoloxías relacionadas co deseño web para móbiles serían **i-mode**, que presenta unha linguaxe moi similar a HTML para a creación de micropáxinas e situada a niveis de mercado preto do WAP en países como Xapón. Outra tecnoloxía complementaria sería o **J2ME** (en inglés Java2 MicroEdition) versión lixeira, ou subconxunto, destinado a dispositivos móbiles da plataforma Java.

### 30.2.3 HTML 5

O HTML busca dar resposta a moitas das necesidades actuais dos sitios web e que non foron recollidas nas versións anteriores da especificación. Nesta versión ten lugar unha maior integración co DOM, a través do IDL (en inglés *Interface Definition Language*). Dentro das novas características de HTML5 atópanse:

- **Unha nova estrutura de documento.** Con seccións como NAV, bloque de navegación do sitio web, HEADER, FOOTER, ARTICLE ou SECTION que aportan información semántica máis relevante que

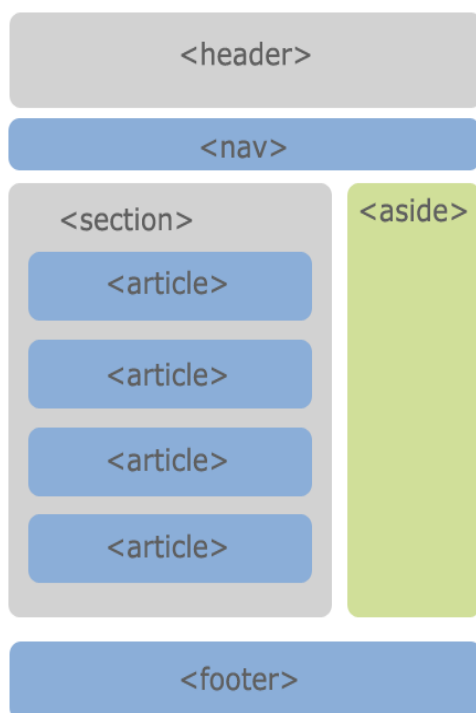


empregar unicamente DIVS ou táboas.

- **Novas etiquetas semánticas.** Como TIME para datas/tempo ou MARK para destacar elementos, METER, PROGRESS, así como incorporación nos documentos de arquivos RDF ou OWL con metadatos.
- **Novas etiquetas multimedia.** Como AUDIO e VIDEO que presentan ademais os códecs necesarios para reproducir os contidos incrustados.
- **Etiquetas para gráficos.** As melloras no compoñente CANVAS permiten a representación de gráficos 2D/3D, como formas, degradados, transformacións e outras opcións gráficas.
- **Novos compoñentes para formularios.** Con novos tipos de datos para correo electrónico, teléfono, números, datas, etc... que permiten a validación no cliente sen empregar Javascript.
- **Compoñentes para operar con grandes conxuntos de datos.** Compoñentes como DATAGRID, MENU ou COMMAND, proporcionan a estes elementos comúns dos sitios web características dinámicas que facilitan o control do comportamento da páxina sen librarías externas.



- **Novas APIS.** Librerías para operacións tan variadas como arrastrar obxectos, funcionamento fóra de liña, xeolocalización, almacenamento persisten en local baseadas en SQL Lite, WebSockets para comunicación entre sitios distribuídos, WebWorkers para traballo



con fíos de execución paralelos e outras posibilidades que se atopan en estudo ou desenvolvemento.

**Figura 1: Exemplo de estrutura dun documento en HTML5.**

### 30.3 AJAX

**AJAX** (en inglés *Asynchronous JavaScript And XML*), Javascript Asíncrono e XML, orixínase para aproveitar que a comunicación entre o usuario e a interface non é fluída permitindo realizar comunicacións asíncronas co servidor e realizar así un maior aproveitamento do ancho de banda e obter unha maior velocidade de resposta. Os datos cárganse nun segundo plano



sen afectar á interface. AJAX non representa unha tecnoloxía en si mesma, senón que se trata da combinación dun grupo de **tecnoloxías** xa existentes:

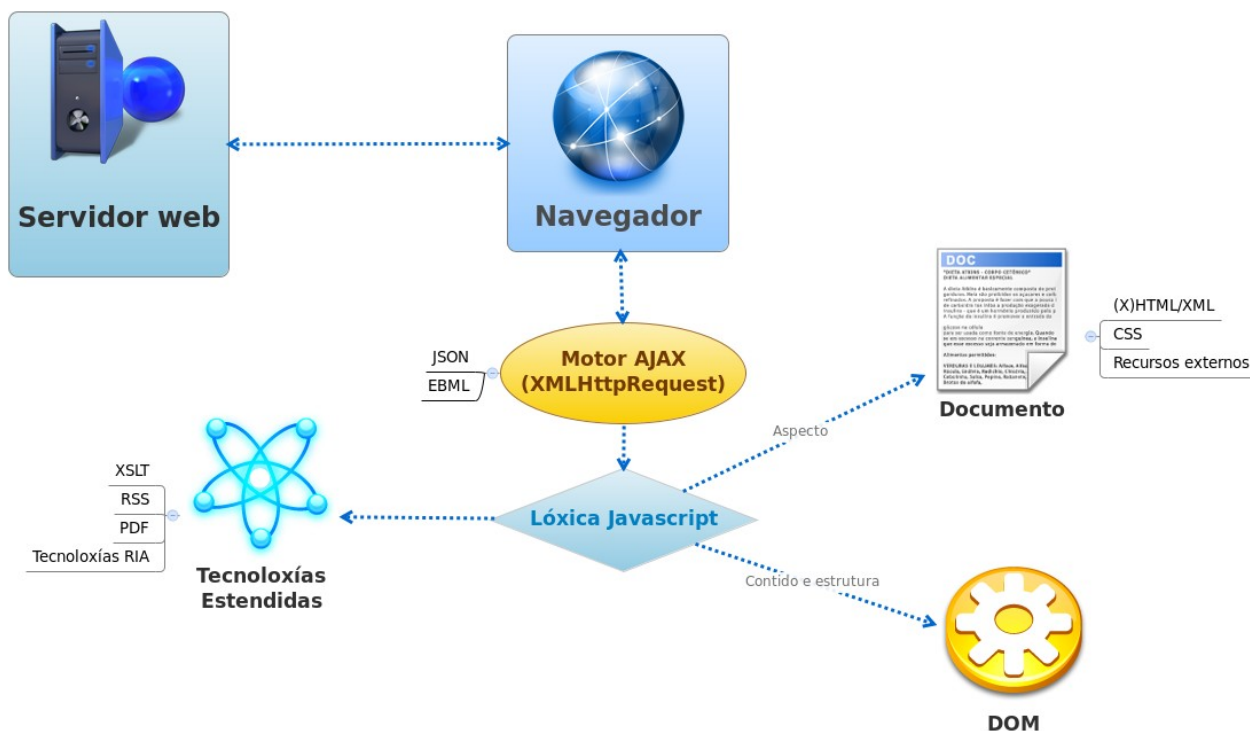
- ✓ (X)HTML e CSS para o deseño das páxinas web.
- ✓ DOM (en inglés *Document Object Model*), ou Modelo de Obxectos do Documento, API que representa un conxunto de obxectos para manipular e modificar dinamicamente documentos (X)HTML e XML a través de linguaxes de Script como Javascript, JScript ou ECMAScript.
- ✓ Obxectos dos tipos *Iframe* ou *XMLHttpRequest* para intercambiar datos de maneira asíncrona co servidor.
- ✓ XML para os formatos de intercambio de datos e comunicacións a través de JSON ou EBML.
  
- ✓ Outras tecnoloxías para facilitar a implantación de solucións específicas como XSLT, RSS, PDF ou outras tecnoloxías RIA.
- ✓ Javascript para proporcionar o nexo común a todo o conxunto.

Como acontece coas tecnoloxías RIA en xeral, é requisito que o navegador teña **soporte** para o conxunto de tecnoloxías AJAX, doutro xeito habería que proporcionar unha alternativa HTML básica. Aínda así cada ves atópase soportado por un maior número de navegadores e non require a instalación de complementos.

O **funcionamento básico** de AJAX baséase no obxecto de comunicación asíncrona, por exemplo o *XMLHttpRequest*, que se instala cunha librería, *framework* ou motor AJAX no lado do cliente. O motor AJAX proverá os métodos para permitir a comunicación asíncrona de datos ademais de definir os compoñentes reutilizables AJAX coa definición do seu comportamento, e o contido e estrutura xeral do documento. O feito de realizar todas estas funcións dende o cliente de maneira asíncrona supoñen a gran mellora de rendemento de AJAX sobre o modelo tradicional de



desenvolvemento web.



**Figura 2: Funcionamento básico de AJAX.**

A **refactorización** de aplicacións tradicionais ao modelo AJAX implica cambios na estrutura interna coa preserva das funcionalidades. A metodoloxía de refactorización habitual implica unha serie de cambios a pequena escala, para o que é ideal a programación orientada a obxectos. Estes cambios van dende:

- ✓ **Refactorización a nivel de método/clase.** En sistemas con pouco acoplamento, evitando por exemplo que se modifiquen os atributos das clases entre obxectos. Estes accesos múltiples poden integrarse nun nivel superior unha vez resoltas as dependencias do método/clase.
- ✓ **Creación de novas clases.** Defínense novas clases especializadas na arquitectura AJAX con responsabilidades e interfaces ben definidos.



- ✓ **Eliminación de clases intermediarias.** Elimínanse as delegacións en exceso entre clases do modelo tradicional en aplicación das consideracións sobre antipatróns de exceso de capas.
- ✓ **Compoñentes e etiquetado.** Diferentes compoñentes sobre todo da vista presentan funcionalidades comúns que se poden factorizar a través de compoñentes AJAX e etiquetados, para facilitar o mantemento e a reutilización.

Na actualidade existen infinidade de **frameworks e librarías AJAX**, incorporando compoñentes da vista, diferentes funcionalidades Javascript, elementos para comunicación asíncrona tanto no cliente coma no servidor e outros elementos para integración con outras tecnoloxías RIA. O obxectivo destes *frameworks* resúmese en facilitar o desenvolvemento de aplicacións web baseadas en AJAX, facendo fincapé en aspectos da capa de vista ou cliente. Os principais *frameworks* en canto ao seu uso máis estendido, son:

- ✓ **Prototype.** Pode ser o *framework* de uso máis estendido, tamén de código aberto dispón da extensión **Scriptaculous** para engadir animacións e efectos nos documentos, e JSON para intercambio de datos. Serve á súa vez a base de outros *frameworks* AJAX. Permite unha grande integración en aplicacións desenvolvidas con *Ruby on Rails* pero tamén pode operar de xeito independente. As súas principais características son:
  - a) **DOM Estendido.** Referencia áxil a obxectos DOM, como por exemplo empregando a función `$()` en lugar de `document.getElementById()`.
  - b) **Scriptaculous.** Aporta ao *framework* un construtor de obxectos DOM (*builder.js*), un repositorio de efectos visuais (*effects.js*, *slider.js*), funcionalidades de control de elementos (X)HTML (*dragdrop.js*, *controls.js*) e métodos para realizar test de



verificación unitarios (*unittest.js*).

- ✓ **Dojo Toolkit.** Proxecto de software libre, actualmente co soporte de IBM e Sun entre outros, que contén varias APIs Javascript modulares e unha ampla coleccións de widgets para uso baixo demanda, agrupados nun sistema de paquetes ao estilo de JEE. Entre as súas principais características están a achega de:
  - a) **Compoñentes empacados en Dijit.** Widgets para a estrutura das páxinas como menús, pestanas; específicos para calendarios, reloxos, gráficos, vectores 2D/3D, ordenación de táboas e paxinación; ademais de elementos para formularios e a súa validación, elementos HTML5 e compoñentes para mellorar a accesibilidade. Así mesmo presenta un editor de texto enriquecido e soporte drag & drop entre os seus compoñentes.
  - b) **Comunicación asíncrona.** Prove dunha capa de abstracción para a comunicación transparente entre o navegador e o servidor web que fai uso de elementos Iframes ocultos para o refresco de datos.
  - c) **Almacenamento no servidor.** Implementa diferentes mecanismos de almacenamento de datos vía CVS, OPML, RDF ou o servizo web Del.icio.us.
  - d) **Soporte para outras tecnoloxías.** Permite integración con tecnoloxías RIA como as aplicacións AIR baseadas en Javascript a través de API e soporte para móbiles.
- ✓ **jQuery.** Outro proxecto de software libre, que neste caso busca simplificar o acceso a documento (X)HTML, o modelo DOM, a manipulación de eventos, utilidades, animacións e efectos. Permite a instalación a través dun paquete básico moi lixeiro (*jquery.js*) que pode ser ampliado a través de plug-ins coma JExpand, para táboas, e JQueryUI para widgets con efectos visuais, entre outros moitos. Entre as súas principais características atópanse:



- a) Integración na plataforma .NET e cos *frameworks* ASP .NET MVC e ASP .NET AJAX.
  - b) Soporte para CSS 3 e XPath.
  - c) Soporte para manipulación de (X)HTML E JSON.
  - d) Fai uso de programación non intrusiva.
  - e) Lixeiro e extensible.
  
- ✓ **Qooxdoo.** Colección de librarías Javascript multipropósito de código aberto, que como as vistas anteriormente permite control áxil a alto nivel de (X)HTML, CSS e DOM, ademais de proporcionar funcionalidades estendidas. A principal diferenza respecto as anteriores é que proporciona widgets de última xeración moi similares aos das aplicacións de escritorio. Entre as súas principais características atópanse:
  - a) **Abstracción do navegador.** Establece unha capa intermedia de abstracción do navegador coas especificacións necesarias para os principais tipos de navegadores definindo así unha interface estándar que mellora a compatibilidade sen necesidade de instalar *plug-ins* adicionais.
  - b) **Administración de eventos.** Prove unha interface propia con métodos para rexistrar e eliminar eventos.
  - c) Ferramenta de desenvolvemento de Interfaces de usuario.
  - d) Soporte de internacionalización i18n e localización l10n que permite o formato de tradución baseado en arquivos .po.
  - e) Prove *frameworks* para depuración de test unitarios e simulacións.
  
- ✓ **Mootools.** *Framework* de código aberto modulas e extensible que permite ao desenvolvedor seleccionar que compoñentes empregar



para minimizar o peso final da librería no cliente. Presenta un compoñente para a incorporación de efectos avanzados e transicións, estreitamente relacionados con Flash sendo un punto forte a súa integración con esta outra tecnoloxía RIA. Prove dos seguintes compoñentes:

- a) **Core.** Núcleo de funcións básicas que empregan todos os demais compoñentes do *framework*.
  - b) **Class.** Librería para instanciación e manipulación de obxectos.
  - c) **Natives.** Extensións de funcións básicas Javascript.
  - d) **Element e Effects/FX.** APIs para manexo de documentos HTML e aplicación de efectos sobre os seus elementos.
  - e) **Remote.** Para intercambio de datos co servidor a través de peticións XmlHttpRequest, JSON ou Cookies.
- ✓ **ExtJS.** Conxunto de librerías derivadas da Yahoo! UI, actualmente emprégase como extensión de JQuery e Prototype incorporando *widgets* especializados, en especial na representación de gráficas e *grids*. Existe ademais unha adaptación específica para GWT denominada ExtGWT, con moitas optimizacións para este contorno. Incorpora unha capa propia dentro dunha arquitectura MVC o que lle permite prover de flexibilidade no tocante aos estilos, facendo uso da extensión SASS (en inglés *Syntactically Awesome Style Sheets*), unha extensión de CSS3. Así mesmo prove de librerías que facilitan a integración con AIR e Spring como backend. Dentro dos compoñentes de datos dispón de varios lectores tanto para XML como JSON. A arquitectura xeral inclúe os paquetes Base e Core coas funcionalidades comúns; os Compoñentes da interface de usuario cos widgets e gadgets; Remoting para a execución de métodos no servidor vía RPC; os Servizos de datos para lectura de vectores, XML e JSON; e o *miniframeframework Drag and drop* para permitir soporte de arrastre entre os compoñentes do *framework*.



- ✓ **Rico.** Baseado en Prototype e orientado cara a Web 2.0 as principais achegas deste *framework* inclúen efectos de animacións que permiten realizar transicións que poden ser interrompidas, pausadas ou reiniciadas, permitindo o solapamento de animacións. Permite a creación de efectos cinematográficos e outros efectos visuais.

Así mesmo proporciona as funcionalidades básicas para soporte AJAX e estende parte do repertorio de Prototype con melloras.

- ✓ **DWR.** (En inglés *Direct Web Remoting*). Framework de código aberto orientado á integración de AJAX con aplicacións JEE a través de mecanismos de RPC como RMI ou SOAP. Permite executar código Java nun servidor de aplicacións como se estivera no navegador do cliente, invocando os obxectos como se foran locais. Consta de dous elementos principais: un *framework* Javascript no cliente e un Servlet no servidor para procesar as peticións e xerar as respostas. O Javascript actuará como proxy das clases Java permitindo que nese código se inclúan as clases Java so servidor. Nunha chamada a un método dunha clase, DWR xera dinamicamente unha versión Javascript da clase AjaxService, invocada a través dun manexador de eventos que xestiona a interacción co servidor. Cando chega a resposta ao cliente invócase unha función *callback* para actualizar o contido do documento. Este método denomínase Reverse AJAX, soportando tres métodos básicos de envío de datos:

- 1) **Polling.** O navegador pregunta ao servidor en intervalos regulares se completou a petición.
- 2) **Piggyback.** O servidor espera á seguinte petición do navegador para darlle a resposta.
- 3) **Comet.** O servidor responde ao navegador de xeito planificado tipo Streaming nunha resposta Http longa.



Así mesmo, prove de dúas opcións de comunicación remota:

- 1) **DWR nativo.** Empregando un superconxunto de JSON onde o motor DW (*engine.js*) manexa as peticións e prepara a execución das chamadas ao servidor.
- 2) **JSON/JSONP.** API para JSON que facilita a integración con outros *frameworks* como Dojo, ExtJS ou JQuery.

No tocante ao tema da seguridade DWR contempla proteccións específicas contra ataques XSS (en inglés *Cross Site Scripting*) e CSRF (en inglés *Cross Site Request Forgery*).

- ✓ **SAJAX.** (En inglés *Simple AJAX Toolkit*). Ferramenta de código aberto que de xeito análogo a DRW permite realizar chamadas a métodos do servidor en PHP, ASP, Coldfusion, Ruby, Perl, Python e outras linguaxes dende Javascript no navegador, sen ter que recargar a páxina.

Prove dunha API para cada linguaxe de servidor, por exemplo *Sajax.php*, que se inclúe no código deste para permitir a integración co Javascript do navegador.

- ✓ **GWT.** (En inglés *Google Web Toolkit*). *Framework* de desenvolvemento AJAX dentro de aplicacións Java. Este contorno permite que ao definir unha interface Java se traduza co compilador GWT de xeito transparente a Javascript e HTML. O principal obxectivo deste *framework* é integrar nun mesmo IDE o desenvolvemento da aplicación e da parte de interfaces de usuario con AJAX pero ademais prove doutras funcionalidades como compoñentes HTML dinámicos e reutilizables, protocolos de transferencia XML e JSON, internacionalización i18n, integración con JUnit incluso nas chamadas RPC e con Javascript a través de JSNI. A arquitectura de GWT estruturase nos seguintes elementos:

1. **Compilador Java a Javascript GWT.** Para aplicacións web



xera automaticamente o código Javascript necesario para a interface Java definida.

2. **Hosted Web Browser.** Motor de execución de aplicacións Java sen traducilas a Javascript a modo de máquina virtual Java.
3. **Librería de Emulación JRE.** Contén os principais paquetes Java de uso común soportadas por GWT.
4. **Librería de clases de Interfaces de Usuario GWT.** Prove dun conxunto de compoñentes para interfaces de usuario.

A maiores destes *frameworks* existen infinidade de alternativas e librarías para temas específicos ou versións mais ou menos simples de propósito xeral, como:

- 1) **AjaxAC.** *Framework* PHP que emprega AJAX no cliente e se orienta á reutilización por dispor de clases moi simples.
- 2) **AJAX .NET Professional.** Librería AJAX para ASP .NET con funcionalidades básicas para controis de usuario e utilidades de uso xeral.
- 3) **ATLAS.** Tamén denominado ASP .NET AJAX, integra nun mesmo *framework* un conxunto de extensións para integrar AJAX en .NET, que inclúe a Microsoft Ajax Library .
- 4) **BAJAX.** Librería Javascript moi lixeira (<6k), para integrar AJAX da forma máis simple posible.
- 5) **Taconite.** *Framework* para desenvolvemento AJAX, que automatiza tarefas para xestionar o obxecto XMLHttpRequest ou a creación de contido dinamicamente.
- 6) **Spry Framework for Ajax.** Librería Javascript de Adobe para a integración de AJAX con orixes de datos XML, JSON e HTML para linguaxes de servidor como Coldfusion, PHP ou ASP .NET. Spry ofrece tres compoñentes principais: Datos, Widgets e Efectos.



- 7) **Tacos**. Librería que proporciona compoñentes, efectos, validacións e funcionamento AJAX para o *framework* Tapestry.
- 8) **XAJAX**. *Framework* AJAX para desenvolvemento en PHP, que permite dende o navegador chamar a funcións do servidor.
- 9) **Zephyr**. *Framework* AJAX para desenvolvemento en PHP5 baixo o modelo MVC. Prove dun motor de modelos, soporte para datos adoDB e outras opcións.
- 10) **ZK**. *Framework* para desenvolvemento de aplicacións Java que busca facer transparente a tecnoloxía Javascript. Os compoñentes da interface de usuario relaciónanse con compoñentes POJO no servidor. Recoméndase a integración con Spring, Toplink ou Hibernate e aporta proteccións contra ataques XSS, CSRF e DoS. Verase polo miúdo máis adiante.

## **30.4 RIA PARA MULTIMEDIA E ANIMACIÓNS**

Conforme mellorou o ancho de banda das conexións foron en aumento as tecnoloxías RIA destinadas a mellorar as funcións multimedia, gráficos vectoriais, animacións e interactividade. A pioneira destas tecnoloxías foi Flash para posteriormente aparecer Flex, AIR, JavaFX, OpenLaszlo e Silverlight como principais alternativas.

### **30.4.1 FLASH RIA**

A plataforma Flash evolucionou de plug in no cliente (Flash Player) para a visualización de imaxes vectoriais e animacións, cara unha arquitectura RIA para dotar de novas funcionalidades ás interfaces web. Representa a primeira tecnoloxía RIA, sendo o marco que engloba diferentes tecnoloxías dentro do que se denomina Flash RIA. Dentro das Flash RIA atópanse as tres tecnoloxías principais soportadas por Adobe, e anteriormente



Macromedia, as cales teñen o uso máis estendido:

1. **Flash.** Empaqueta de aplicación en arquivos SWF a modo de compoñentes.
2. **Flex.** A partir dun servidor de aplicacións JEE realiza a comunicación co SWF permitindo chamar a obxectos do servidor.
3. **AIR.** Para execución de aplicacións Flash no equipo do cliente sen necesidade de navegadores como intermediarios.

Alternativamente, existen outros provedores de tecnoloxías Flash RIA, entre os que se atopan:

1. **OpenLaszlo.** Moi similar a Flash pero cunha linguaxe de programación propia LZX.
2. **SnappMX.** Orientada cara servizos web.
3. **Zulu.** Permite desenvolver aplicacións conxuntamente co estándar XUL e Flash.
4. **XAMLON.** Permite desenvolver aplicacións Flash coa linguaxe de marcado XAML dentro da plataforma .NET.

A orientación principal de Flash a diferenza de AJAX é a de ser un complemento da interface de usuario estendendo determinadas funcionalidades, en especial no campo das animacións e da interacción co usuario. Os contornos de desenvolvemento de Flash están máis orientados cara a edición de animacións que cara o desenvolvemento web, pero foise abrindo camiño no campo da elaboración de widgets, soporte multilinguaxe, efectos 3D, control e validación de formularios. Permite integración con AJAX e Javascript pero dispón dunha linguaxe de *script* propia denominada **ActionScript**. Este *script* de programación orientada a obxectos segue o estándar ECMA-262, implementando E4X (en inglés *ECMAScript for XML*). Opera cun modelo de eventos baseado na



especificación DOM, aínda que non o segue completamente. Lánzase nunha máquina virtual específica AMV2 (en inglés *ActionScript Virtual Machine*) aloxada no contorno de execución Flash Player. Por último destacar que permite conectividade con Servizos web e Bases de datos de maneira remota a través da clase *DataProvider*.

Como acontece na maioría de *frameworks* de desenvolvemento a plataforma Flash pode ampliarse con paquetes de librarías de terceiros, entre estas destacan:

- a) **SPL** (en inglés *Spelling Plus Library*). Para deseño de editores de texto enriquecido con corrección ortográfica.
- b) **Red5**. Servidor Flash de software libre.
- c) **Papervision3D**. Motor de xeración 3D de software libre.
- d) **As3corelib**. Paquete de librarías ActionScript 3 que contén clases e utilidades de uso común. Inclúe codificadores de imaxe, serialización JSON, APIs para datas, Strings e outros tipos de datos e codificación de chaves MD5 e SHA 1.
- e) **SWFObject**. Javascript para incrustar contido Flash en documentos (X)HTML
- f) **Tweener**. Para crear animacións e transicións directamente traballando con ActionScript.
- g) **Gaia**. *Framework* orientado cara a axilización do desenvolvemento Flash.

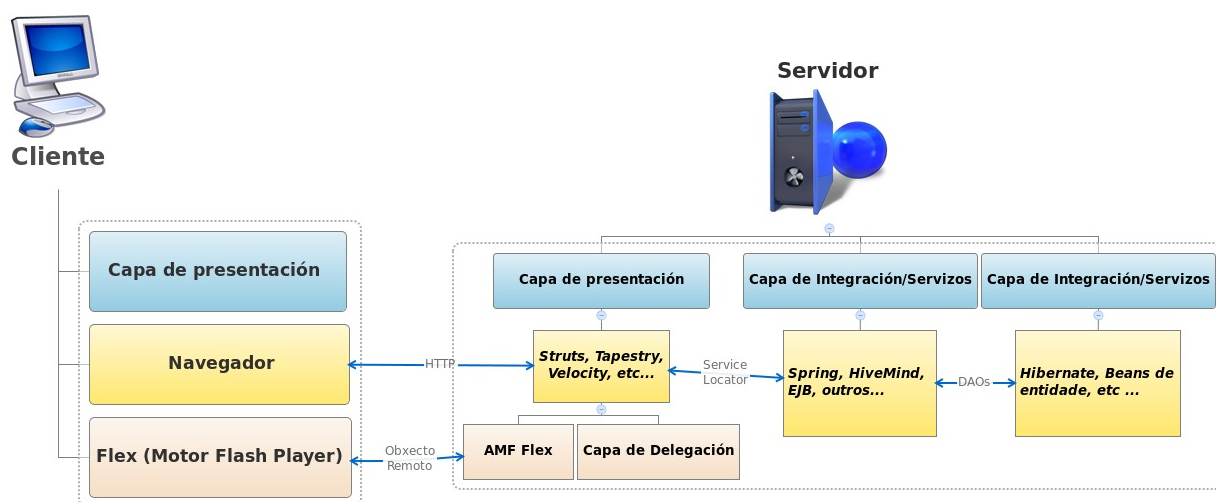
### 30.4.2 FLEX

Flex é a evolución de Flash ampliando o ámbito de desenvolvemento RIA con novas tecnoloxías e formatos. Diferenciase de Flash na súa facilidade de integración con tecnoloxías de linguaxes de servidor o que facilita o uso de patróns de deseño e que emprega MXML (en inglés *Macromedia*



*eXtensible Markup Language*) para definir o aspecto e comportamento das interfaces de usuario. Como Flash soporte a linguaxe ActionScript e a súa plataforma incorpora librarías de compoñentes para interfaces de usuario específicas. Permite integración con outras tecnoloxías do lado do servidor como Servizos Web, REST ou AMF. As aplicacións Flex poden integrarse nun documento HTML de xeito que este pode actualizar dinamicamente a vista e enviar e recibir datos asincronamente co servidor de fondo, de xeito similar a AJAX.

Nas **comunicacións cliente servidor**, Flex no cliente comunícase co servidor vía HTTP, dispoñendo de tres API RPC: HTTPService, WebService e RemoteObject. Non acceden directamente a Bases de datos remotas senón que o fan a través de capas intermedias. A través de HTTPService solicita arquivos JSP ou XML cos datos en formato de variables String, formatos de intercambio XML, E4X ou obxectos ActionScript. No caso de devolver JSON Flex dispón de librarías especializadas para serialización así como para SOAP. A través da API de RemoteObject permite realizar peticións **Flash Remoting** que devolven mensaxes binarias **AMF** (en inglés *Action Message Format*) sobre HTTP. Cando este formato ten aplicación obtense un maior rendemento que noutras tecnoloxías, como JSON ou SOAP.



**Figura 3: Integración de Flex en JEE.**



Así mesmo Flex permite intercambio de datos en tempo real perante dúas vías: **XML Socket** e **Socket Binario**. Co XML Socket créase unha conexión que permanece aberta mentres dure a comunicación, ou é pechada explicitamente. No Socket Binario o funcionamento é similar pero cliente e servidor non precisan intercambiar paquetes XML especificamente senón que envían os datos como información binaria, o que permite conectar con servidores de correo como POP3, SMTP e IMAP ou servidores de novas como NNTP.

No tocante á **seguridade** á hora de integrar Flex nunha aplicación JEE será a arquitectura desta a que impoña o modelo de seguridade, variando dende un *framework* de autenticación/autorización específico a un directorio LDAP, ou arquivos de configuración XML. A información de seguridade debe engadirse aos servizos BlazeDS e LiveCycle Data de xeito que soliciten credenciais nas comunicacións co servidor.

Coma noutras tecnoloxías, en Flex están dispoñibles unha serie de **frameworks** multipropósito. Moitos deles tamén son válidos para AIR. Os máis empregados son:

- a) **Cairngorm**. Microarquitectura que aplica un pequeno conxunto de patróns de deseño (Service Locator, Front-Controller, ...) probados en conxunto. Céntrase en tres áreas chave: manexar accións de usuario, encapsular as interaccións con servidor e a lóxica de negocio e xestionar o estado do cliente representándoo na Interface de usuario.
- b) **Mate**. *Framework* orientado a eventos baseado en etiquetas, implementadas completamente en MXML. Implementa a idea de Inxección de dependencia, construíndo os obxectos para a continuación inxectar nas clases os datos. Os obxectos non solicitan a información pero esta lle é entregada polo sistema.



- c) **PureMVCFramework**. Como Cairngorm representa tamén unha microarquitectura cun pequeno conxunto de patróns de deseño, con MVC e Fachada como núcleos centrais, cada unha a través dun patrón instancia única.
- d) **Swiz**. *Framework* de control de inversión (Ioc), que prove metodoloxías para simplificar o manexo de eventos e as chamadas asíncronas a procedementos remotos. Emprega MVC pero a diferenza dos anteriores só no que respecta á estrutura de clases e non de directorios.
- e) **Parsley**. Conxunto de librarías ActionScript para correspondencia de obxectos e entidades, rexistro de depuración, inxección de dependencia, mensaxería e outras funcionalidades estendidas.

Para a integración con sistemas de información Flex prove do Servizo de xestión de datos dentro do Servizo LiveCycle Data. Neste servizo inclúese sincronización de datos en tempo real entre cliente, servidor e outros clientes, replicación de datos, paxinación baixo demanda, e para aplicacións AIR sincronización de datos locais para conexións ocasionais das aplicacións.

### **30.4.3 SILVERLIGHT/MOONLIGHT**

Complemento para navegadores que permite integrar na mesma extensión elementos multimedia, animacións e interactividade, de xeito similar ao WPF. Atópase baseado en XAML para a definición das interfaces de usuario, a partir das que invoca a métodos do servidor de aplicacións .NET. Permite a carga dinámica de XML co que se pode operar a través de DOM ao xeito de AJAX. Proporciona extensións Javascript e. As principais **características** do *framework* son:



- a) **WPF**. Inclúe un subconxunto de WPF que estende en gran medida elementos da Interface de Usuario.
- b) **XAML**. Definición da Interface de Usuario a través dunha linguaxe de marcado declarativa.
- c) **Integración** con Javascript e ASP .NET AJAX.
- d) Acceso a **obxectos do lado do servidor** .NET.
- e) Conexión con **servizos de rede** WCF, SOAP e ASP .NET AJAX, permitindo orixes de datos JSON, XML e RSS.
- f) Soporta **LINQ** para implementar o acceso a datos

A **arquitectura** de Silverlight se compón de 3 partes fundamentais:

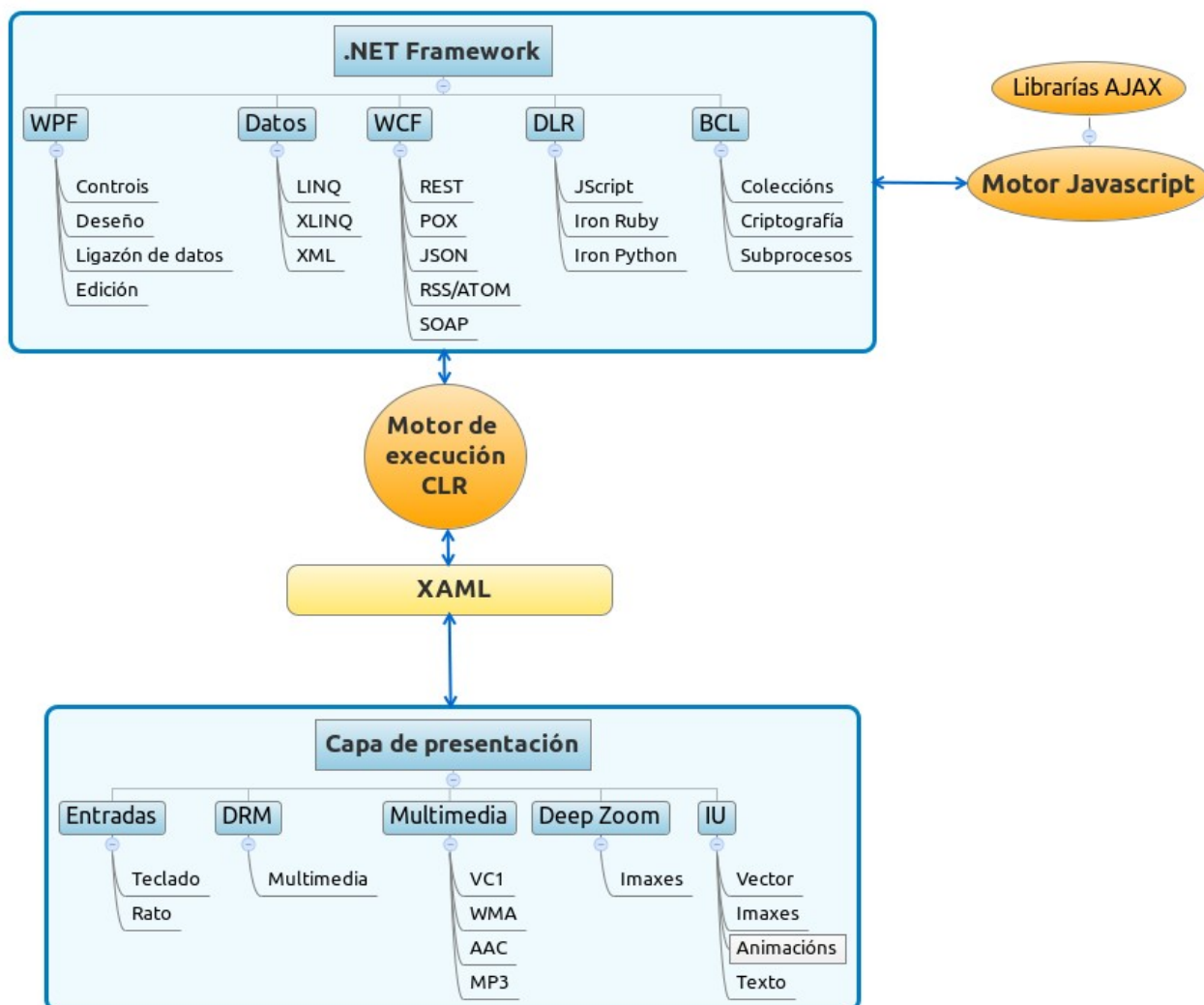
- 1) **Framework de presentación básico**. Compoñentes e servizos orientados á Interface de usuario e a interacción co usuario, elementos multimedia e soporte XAML.
- 2) **.NET Framework para Silverlight**. Subconxunto de .NET Framework para Silverlight que contén compoñentes e librarías, recolector de lixo, WCF e CLR. Así mesmo inclúe os controis da Interface de Usuario, XLINQ, RSS/Atom, serialización XML e DLR (en inglés *Dynamic Language Runtime*)
- 3) **Compoñente de instalación e actualización**. Control de instalación e actualización da extensión.

Mención especial merece o apartado da **seguridade**, como acontecía con outras tecnoloxías RIA Silverlight incorpora políticas de seguridade específicas para:

- a) Seguimento do Ciclo de vida de seguridade de Microsoft SDL (En inglés *Security Development Lifecycle*)
- b) Evitación de ataques XSS.



- c) Illamento de código de arquivos de configuración XAP.
- d) Prover acceso seguro a recursos de rede.
- e) Servizos criptográficos para protección de datos de usuario.
- f) Sinatura dixital das aplicacións.



**Figura 4: Arquitectura Silverlight.**

#### 30.4.4 JAVA FX

Java FX é unha plataforma que se compón de elementos web, multimedia e



scripting xunto con tecnoloxías de servidor JEE para o desenvolvemento de aplicacións multiplataforma. Pode funcionar de maneira independente do navegador sempre que teña o equipo instalada unha máquina virtual Java compatible. Promove o concepto de “Perfil común” para tentar unificar todos os dispositivos que soporten JavaFX, de xeito que o mesmo modelo de desenvolvemento se adapte a calquera contorno.

Os principais **compoñentes** de Java FX son:

- a) **JavaFX Script.** Linguaxe de programación declarativa, con tipos estáticos, que permite invocar métodos de calquera API de Java da plataforma.
- b) **Entorno de execución JavaFX.** Especializado para o dispositivo, Escritorio/Web, Mobile, ou TV.
- c) **Aplicacións JavaFX.** Independentes ou empaquetadas como arquivos JAR.
- d) **Ferramentas de desenvolvemento.** Inclúe o compilador para JavaFX Script, Plug ins para IDEs como Eclipse, e librarías especializadas para gráficos, multimedia ou Servizos web, entre outros.

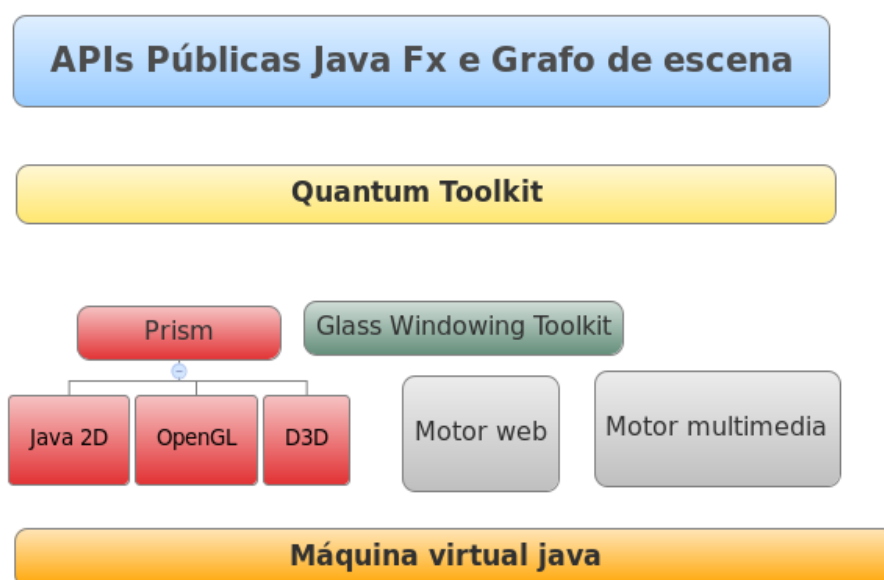
A **arquitectura** de JavaFX presenta nun primeiro nivel:

- 1. As **APIs públicas** de JavaFX. As principais funcionalidades destas APIs son permitir integración con outras linguaxes como JRuby, Groovy e Javascript, funcionalidades xenéricas e extensións para as interfaces de usuario.
- 2. **Grafo de Escena.** Neste grafo, definido na API *javafx.scene* represéntase nunha estrutura en árbore con nodos representando todos os elementos visuais da interface de usuario. Cada nodo pode levar os seus estilos, ademais de efectos, manexadores de eventos e control de estado.



Baixo eles atópase o motor de execución composto polos seguintes compoñentes:

- a) **Prism.** Motor gráfico de alto rendemento que soporta Java2D, OpenGL e DirectX.
- b) **Quantum Toolkit.** Xestiona as regras de procesos para representación gráfica fronte ao manexo de eventos.
- c) **Glass Windowing Toolkit.** Sistema de control de ventás no nivel máis baixo da arquitectura gráfica de JavaFX. Prové dos servizos operativos nativos do sistema ademais de ser responsable de xestionar a cola de eventos.
- d) **Motores web e multimedia.** Inclúen APIs para soporte de medios visuais e de son. Así mesmo o motor web soporta os estándares HTML5, CSS, Javascript, DOM e SVG.



**Figura 5: Arquitectura JavaFX.**

### 30.5 OUTRAS TECNOLOXÍAS RIA



Se ben as tecnoloxías con anterioridade representan as solucións de uso máis estendido actualmente, existen outras multipropósito ou máis específicas que buscan facerse un oco no mundo das RIA. Dentro destas outras tecnoloxías, OpenLaszlo merece mención especial, se ben non representa unha tecnoloxía en si, senón un conxunto de tecnoloxías, con algunhas adaptacións específicas.

### 30.5.1 OPENLASZLO

OpenLaszlo é un *framework* e plataforma de desenvolvemento para aplicacións RIA con licenza GPL, precisando dun servidor propio para o aloxamento das aplicacións desenvolvidas. Unha mesma aplicación definida a través de dunha linguaxe de definición propia pode exportarse a diferentes formatos multinavegador e multiplataforma.

As aplicacións OpenLaszlo poden despregarse no servidor de aplicacións da plataforma, denominado **Despregue en modo proxy**, ou ben en modo **Despregue “SOLO”** con independencia do servidor, por norma xeral empaketada nun arquivo SWF. Outra característica peculiar de funcionamento son as **Librarías dinámicas**, que permiten unha descarga “parcial” da aplicación, para obter unha carga inicial máis rápida, sendo o resto da carga da aplicación baixo demanda. Así mesmo a funcionalidade **Krank** permite cargar as aplicación OpenLaszlo máis rapidamente, realizando un preprocesamento da vista e *scripts* de inicialización.

OpenLaszlo aporta unha linguaxe declarativa propia, o LZX deseñado para describir as interfaces do usuario, ao estilo de XUL e XForms. Esta linguaxe incorpora un *framework* de etiquetas dividido en categorías como: elementos da interface, orixes de datos, efectos multimedia e accións. Emprégase conxuntamente con Javascript/AJAX sendo este último o encargado da interacción co usuario.



A plataforma OpenLaszlo incorpora os seguintes **compoñentes**:

- a) **Compilador.** Permite que unha aplicación definida perante a linguaxe declarativa LZX poida transformarse a Flash (SWF) ou DHTML-AJAX. Así mesmo presenta unha serie de módulos específicos para:
  - 1) **Compilación XML IU.** O compilador transforma as definicións da interface de usuario ao formato de saída especificado para a aplicación.
  - 2) **Compilación ECMAScript.** Clases e instancias LZX tradúcense a ECMAScript e controladores de eventos, transformándoas a Bytecode.
  - 3) **Compilación multimedia.** Codifícanse os arquivos en formatos de imaxe e son, así como as fontes TrueType en ficheiros OBJ para SWF ou XML.
- b) **Servidor.** Fai as funcións de aloxamento da aplicación e proxy, mantendo comunicación bidireccional cos back-ends a través de JAVARPC ou outros protocolos de servizos web. Así mesmo proporciona servizos de transformación de formatos, mensaxería, *streaming*, encriptación SSL e autenticación.
- c) **Contorno de execución ou LCF** (en inglés *Laszlo Foundation Class*). Inclúe compoñentes da interface de usuario, acceso a datos e servizos de rede. A LCF divídese en catro compoñentes principais:
  - 1) **Data Loader/Binder.** O cargador e encargado de asociar e relacionar os datos. Dirixe o tráfico de datos incluíndo o fluxo de datos cara o cliente.
  - 2) **Sistema de eventos.** Permite unha programación baseada en eventos ao recoller os eventos detectados no cliente.
  - 3) **Layout & Animation System.** Sistema de animación e escenario da aplicación que ofrece todos os elementos para a parte gráfica



- da aplicación así como animacións e efectos.
- 4) **Servizos para as aplicacións.** Con funcionalidades extra como contadores, sons, etc...
- d) **Framework.** Prove dunha extensa API para animacións, estrutura de aplicación, acceso a datos, comunicacións co servidor e definición da interface de usuario. Estruturalmente segue un modelo MVC, pero ampliable ás capas que sexan precisas para cada solución.
- e) **Servlet.** Trátase dun compoñente opcional para a aplicación que permite atender e dirixir peticións multimedia ou para servizos web como SOAP, JavaRPC ou XML-RPC. Fai funcións de caché e proxy para priorización e bloqueo de peticións así como de rexistro de trazas e auditoría.

### 30.5.2 ZK FRAMEWORK

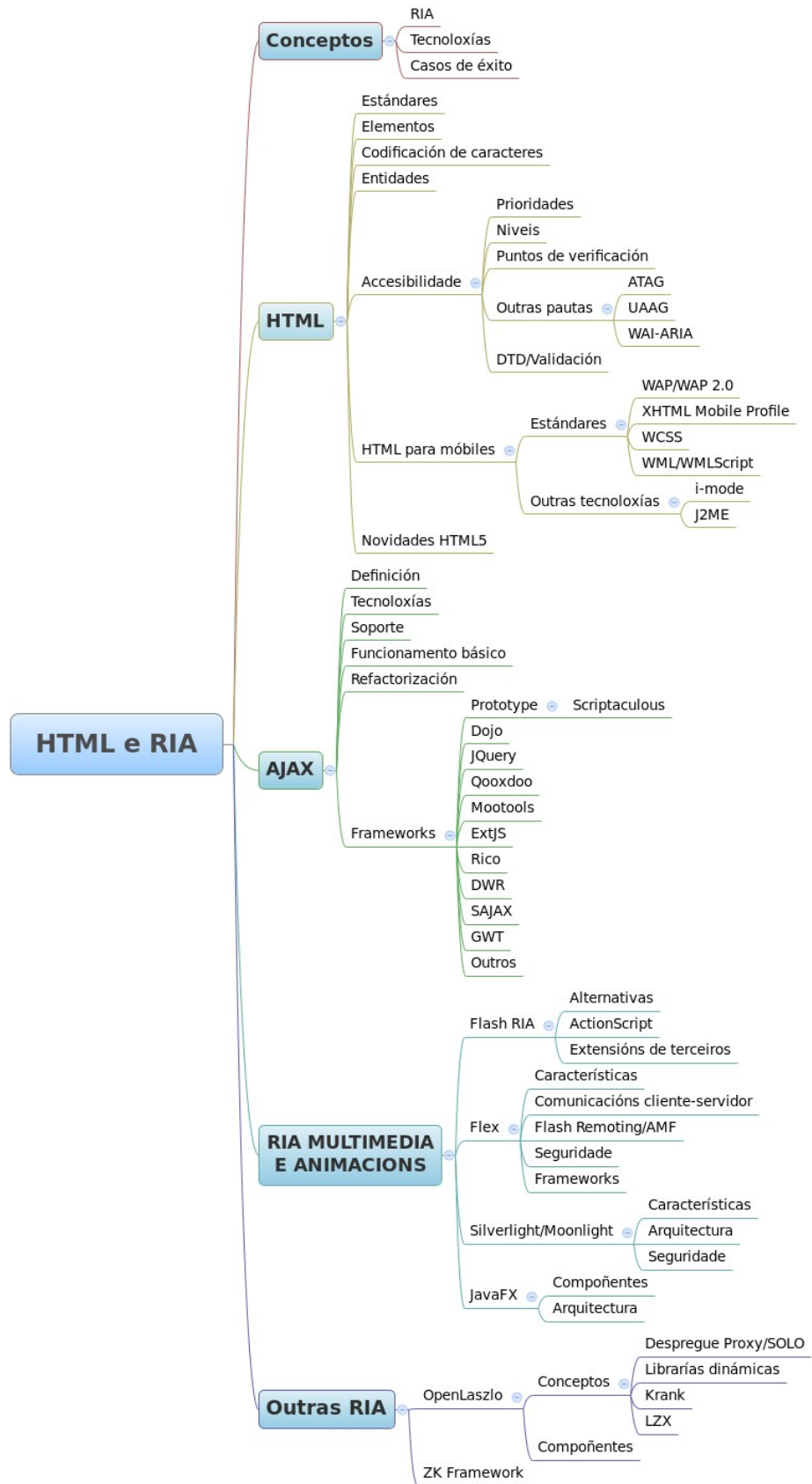
Trátase doutro *framework* orientado a eventos que en esencia poderíase incluír cos *frameworks* AJAX aínda que con algunhas diferenzas. En primeiro lugar emprega unha linguaxe de marcado propia para as interfaces de usuario denominada ZUML que pode mesturarse con outras linguaxes de marcado como XUL e XHTML, ademais de linguaxes de *script* e expresións EL para manipulación de compoñentes e datos. Deseñouse para integración con aplicacións JEE incorporando as capacidades de AJAX en desenvolvementos áxiles e reutilizables.

Prove dun *framework* cunha implementación de ZK Spring, adaptado do *framework* Spring, e librarías con compoñentes e etiquetas JSP con soporte para AJAX. No tocante á seguridade engade protección para ataques XSS, DoS e CSRF, ademais de reforzar a autenticación e os permisos coa incorporación de *frameworks* de terceiros 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 e O. U.**

**Colegiado del CPEIG**



**31. ENXEÑARÍA DO SOFTWARE.  
PROCESO SOFTWARE,  
MODELOS DE PROCESO  
SOFTWARE. PROCESO  
UNIFICADO. CICLOS DE VIDA.  
MODELOS DE CICLO DE VIDA.  
FASES DO CICLO DE VIDA.  
MODELOS DE  
DESENVOLVEMENTO.  
MODELOS ÁXILES.  
METODOLOXÍAS DE  
DESENVOLVEMENTO DE  
SOFTWARE. MÉTRICA  
VERSIÓN 3.**



**Tema 31. Enxeñería do software. Proceso software, modelos de proceso software. Ciclo de vida. Modelos de ciclo de vida. Fases do ciclo de vida. Modelos de desenvolvemento. Modelos áxiles. Metodoloxías de desenvolvemento de software. Métrica versión 3**

**INDICE**

- 31.1 Enxeñería do software
  - 31.1.1 A crise do software
  - 31.1.2 A Enxeñería do 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 e Corrixir
  - 31.4.2 Modelo por Etapas e Modelo en Cadoiro.
  - 31.4.3 Modelos baseados 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 baseados en Transformacións
  - 31.4.6 Desenvolvemento Baseado en Compoñentes
  - 31.4.7 Proceso Unificado de Desenvolvemento de software (PUDS)
  - 31.4.8 Modelo de métodos formais
  - 31.4.9 Programación Extrema (extreme Programming)
- 31.5 Metodoloxías Áxiles
  - 31.5.1 SCRUM
  - 31.5.2 DSDM
  - 31.5.3 Extreme Programming (XP)
  - 31.5.4 Agile Modeling (AM)
  - 31.5.5 FCC
  - 31.5.6 Familia Crystal



31.6 Metodoloxías de desenvolvemento de sistemas de información.

31.7 Métrica Versión 3

### **31.1 Enxeñería do software**

#### *31.1.1 A crise do software.*

A rápida expansión da Informática, sobre todo a partir da segunda xeración de ordenadores nos anos 60, levou á escritura de millóns de liñas de código antes de que se empezasen a presentar de xeito serio metodoloxías para o deseño e a construción de sistemas software e métodos para resolver os problemas de mantemento, fiabilidade, etc. Esta expansión sen control tivo como consecuencia a denominada **crise do software**, que é o nome xenérico que se acuñou para referirse a un conxunto de problemas que se han ir atopando no desenvolvemento do software. Esta problemática non só se limita ao software que non funciona adecuadamente, senón que abrangue outros aspectos como a forma de desenvolver o software, o mantemento dun volume crecente de software existente e a forma de satisfacer a demanda crecente de software.

Os síntomas que fan palpable a aparición da crise do software son, entre outros, os seguintes:

- **Expectativas:** os sistemas non responden ás expectativas que deles teñen os usuarios.
- **Fiabilidade:** os programas fallan demasiado a miúdo.
- **Custo:** os custos do software son moi difíciles de prever e, frecuentemente, son moi superiores ao esperado.
- **Prazos:** o software adóitase entregar tarde e con menos prestacións das ofertadas.
- **Portabilidade:** é difícil cambiar un programa do seu contorno hardware, aínda cando as tarefas a realizar son as mesmas.
- **Mantemento:** a modificación do software é unha tarefa custosa, complexa e propensa a erros.



- **Eficiencia:** os esforzos que se fan para o desenvolvemento do software non fan un aproveitamento óptimo dos recursos dispoñibles (persoas, tempo, diñeiro, ferramentas, etc.).

A solución á crise do software céntrase, pois, en abordar e resolver os seguintes problemas principais:

- A planificación do proxecto software e a estimación dos custos de desenvolvemento, que son moi imprecisos.
- A produtividade das persoas, que non se corresponde coa demanda dos seus servizos.
- A calidade do produto software, que é, en moitos casos, inadecuada.

### 31.1.2 A Enxeñería do Software

A Enxeñería do Software pódese definir como *o establecemento e uso de principios de enxeñería orientados a obter, de xeito económico, software que sexa fiable e funcione eficientemente sobre máquinas reais*.

A Enxeñería do Software abarca tres elementos clave: métodos, ferramentas e procedementos. Os **métodos** proporcionan o xeito de construír tecnicamente o software. Abarcan as tarefas de planificación e estimación de proxectos, análises dos requisitos do sistema e do software, deseño das estruturas de datos, da arquitectura de programas e dos procedementos algorítmicos, e a codificación, probas e mantemento. As **ferramentas** fornecen o soporte automático ou semiautomático para os métodos; isto é, dan soporte ao desenvolvemento do software. Os **procedementos** definen a secuencia na que se aplican os métodos, os controis que axudan a asegurar a calidade e a coordinar os cambios e as guías que lles facilitan aos xestores do software.

O traballo que se asocia á enxeñería do software pódese dividir segundo Pressman en tres **fases xenéricas**, con independencia da área de aplicación, tamaño ou complexidade do proxecto:



- **A fase de definición** céntrase sobre *o que*. É dicir, durante a definición, o que desenvolve o software intenta identificar que información ha de ser procesada, que función e rendemento se desexa, que comportamento do sistema, que interfaces van ser establecidas, que restricións de deseño existen, e que criterios de validación se necesitan para definir un sistema correcto. Xa que logo, hanse identificar os requisitos clave do sistema e do software. Aínda que os métodos aplicados durante a fase de definición variarán dependendo do paradigma de enxeñería do software (ou combinación de paradigmas) que se aplique, dalgún xeito terán lugar tres tarefas principais: enxeñería de sistemas ou de información, planificación do proxecto do software e análise dos requisitos.
- **A fase de desenvolvemento** céntrase no *como*. É dicir, durante o desenvolvemento un enxeñeiro do software intenta definir como han de deseñarse as estruturas de datos, como ha de implementarse a función dentro dunha arquitectura de software, como han de implementarse os detalles procedementais, como han de caracterizarse interfaces, como ha de traducirse o deseño nunha linguaxe de programación (ou linguaxe non procedemental) e como ha de realizarse a proba. Os métodos aplicados durante a fase de desenvolvemento variarán, aínda que sempre teremos: deseño do software, xeración de código e proba do software.
- **A fase de mantemento** céntrase no *cambio* que vai asociado á corrección de erros, ás adaptacións requiridas a medida que evoluciona o contorno do software e a cambios debidos ás melloras producidas polos requisitos cambiantes do cliente. Durante a fase de mantemento atopamos catro tipos de cambios:
  - o **Corrección.** Incluso levando a cabo as mellores actividades de garantía de calidade, é moi probable que o cliente descubra os defectos no software. O mantemento *correctivo* cambia o software para corrixir os defectos.



- o **Adaptación.** Co paso do tempo, é probable que cambie o contorno orixinal para o que se desenvolveu o software. O mantemento *adaptativo* produce modificacións no software para acomodalo aos cambios do seu contorno externo (hardware, sistema operativo, regras de negocio,...).
- o **Mellora.** Conforme se utilice o software, o cliente/usuario pode descubrir funcións adicionais que van producir beneficios. O mantemento *perfectivo* leva o software máis aló dos seus requisitos funcionais orixinais.
- o **Prevención.** O software de computadora deteriórase debido ao cambio. En esencia, o mantemento *preventivo* fai cambios en programas de computadora a fin de que se poidan corrixir, adaptar e mellorar máis facilmente.

Estas fases se complementan cun número de actividades **protectoras** que se aplican ao longo de todo o proceso do software. Entre as actividades típicas desta categoría inclúense: seguimento e control do proxecto de software, revisións técnicas formais, garantía de calidade do software, xestión de configuración do software, preparación e produción de documentos, xestión de reutilización, medicións e xestión de riscos.

### 31.2 Proceso software. Modelos de proceso software

Segundo Sommerville, *un proceso do software é un conxunto de actividades que conducen á creación dun produto software*. Estas actividades poden consistir no desenvolvemento de software desde cero nunha linguaxe de programación estándar como Java ou C. Con todo, cada vez máis desenvólvese novo software ampliando e modificando os sistemas existentes e configurando e integrando software comercial ou compoñentes do sistema.



Para Fugetta, un proceso software é *un conxunto coherente de políticas, estruturas organizacionais, tecnoloxías, procedementos e artefactos que son necesarios para concibir, desenvolver, instalar e manter un produto software.*

Os procesos do software son complexos e, como todos os procesos intelectuais e creativos, dependen das persoas que toman decisións e xuízos. Debido á necesidade de xulgar e crear, os intentos para automatizar estes procesos han ter un éxito limitado. As ferramentas de enxeñería do software asistida por computadora (CASE) poden axudar a algunhas actividades do proceso, pero teñen limitacións. Unha razón pola cal a eficacia das ferramentas CASE está limitada atópase na inmensa diversidade de procesos do software. Non existe un proceso ideal e moitas organizacións desenvolveron o seu propio enfoque para o desenvolvemento do software. Os procesos evolucionaron para explotar as capacidades das persoas dunha organización, así como as características específicas dos sistemas que se están desenvolvendo. Para algúns sistemas, como os sistemas críticos, requírese un proceso de desenvolvemento moi estruturado. Para sistemas de negocio, con requisitos rapidamente cambiantes, un proceso flexible e áxil probablemente sexa máis efectivo.

Aínda que existen moitos procesos diferentes de software, algunhas actividades fundamentais son comúns para todos eles:

1. **Especificación do software** onde os clientes e enxeñeiros definen o software a producir e as restricións sobre a súa operación.
2. **Desenvolvemento do software** onde o software se diseña e programa.
3. **Validación do software** onde o software se valida para asegurar que é o que o cliente require.
4. **Evolución do software** onde o software debe evolucionar para cubrir as necesidades cambiantes do cliente.



Diferentes tipos de sistemas necesitan diferentes procesos de desenvolvemento. Polo tanto, estas actividades xenéricas poden organizarse de diferentes formas e describirse en diferentes niveis de detalle para diferentes tipos de software. O uso dun proceso inadecuado do software pode reducir a calidade ou a utilidade do produto de software que se vai a desenvolver e/ou incrementar os custos de desenvolvemento.

Os procesos do software pódense mellorar coa estandarización. Isto conduce a mellorar a comunicación e a unha redución do tempo de formación, e fai a axuda ao proceso automatizado máis económica. A estandarización tamén é un primeiro paso importante para introducir novos métodos, técnicas e boas prácticas de enxeñería do software.

De todos os xeitos, a existencia dun proceso de software non é garantía de que este será entregado a tempo, de que satisfará as necesidades do cliente, ou de que mostrará as características técnicas que conducirán a características de calidade a longo prazo. O proceso de software debe **avaliarse** para asegurarse de que cumpra unha serie de criterios básicos que demostraron ser esenciais para unha enxeñería de software exitosa.

**CMMI** (Modelo de Capacidade de Madurez Integrado) é un modelo total do proceso, que describe as metas, prácticas e capacidades específicas con que debe contar un proceso de software maduro. O estándar **SPICE (ISO/IEC15504)** define un conxunto de requisitos para a avaliación do proceso de software. O que pretende é axudar ás organizacións no desenvolvemento dunha avaliación obxectiva da eficacia de calquera proceso de software definido.

O ISO **9001:2000 para software** é un estándar xenérico que se aplica a calquera organización que desexe mellorar a calidade xeral dos produtos, sistemas ou servizos que prové. O ISO 9001:2000 subliña a importancia que ten para unha organización identificar, implementar, xestionar e mellorar de xeito continuo a efectividade dos procesos necesarios para o



sistema de administración da calidade e xestionar as interaccións destes procesos para conseguir os obxectivos da organización.

O ISO 9001:2000 adoptou un ciclo de “planear-facer-revisar-actuar” (PDCA Plan-Do-Check-Act en inglés, tamén coñecido como Círculo de Deming) que se aplica aos elementos de xestión de calidade dun proxecto de software. Dentro dun contexto de software, “planear” establece os obxectivos, as actividades e tarefas do proceso necesarios para conseguir un software de alta calidade e unha satisfacción do cliente; “facer” implementa o proceso de software (incluídas as actividades do marco de traballo e as “actividades parasol”); “revisar” monitora e mide o proceso para asegurarse de que todos os requisitos establecidos para a xestión de calidade sexan cumpridos; “actuar” inicia as actividades de melloramento do proceso de software, o cal ten unha continuidade de traballo para mellorar o proceso.

Un **modelo de procesos do software** é unha descrición abstracta e simplificada dun proceso do software que presenta unha visión dese proceso. Cada modelo de proceso representa un proceso desde unha perspectiva particular e así proporciona só información parcial sobre ese proceso. Son abstraccións dos procesos que se poden utilizar para explicar diferentes enfoques para o desenvolvemento de software. Pode pensarse neles como marcos de traballo do proceso que poden ser estendidos e adaptados para crear procesos máis específicos de enxeñería do software. Cada modelo describe unha sucesión de fases e un encadeamento entre elas. Segundo as fases e o modo en que se produza este encadeamento, temos diferentes modelos de proceso. Un modelo é máis adecuado que outro para desenvolver un proxecto dependendo dun conxunto de características do proxecto. Os modelos poden incluír actividades que son parte dos procesos e produtos de software e o papel das persoas involucradas na enxeñería do software. Alternativamente, ás veces úsanse os termos **ciclo de vida, modelo de ciclo de vida e modelo de desenvolvemento**.



A maior parte dos modelos de procesos do software baséanse nun dos tres modelos xerais ou paradigmas de desenvolvemento de software:

1. **O enfoque en cadoiro.** Considera as actividades anteriores e represéntaas como fases de procesos separados, tales como a especificación de requisitos, o deseño do software, a implementación, as probas, etcétera. Despois de que cada etapa queda definida, «asínase» e o desenvolvemento continúa coa seguinte etapa.
2. **Desenvolvemento iterativo.** Este enfoque entrelaza as actividades de especificación, desenvolvemento e validación. Un sistema inicial desenvólvese rapidamente a partir de especificacións moi abstractas. Este refínase baseándose nas peticións do cliente para producir un sistema que satisfaga as necesidades do devandito cliente. O sistema pode entón ser entregado. De forma alternativa, pódese reimplementar utilizando un enfoque máis estruturado para producir un sistema máis sólido e sostible.
3. **Enxeñería do software baseada en compoñentes (CBSE).** Esta técnica supón que as partes do sistema xa existen previamente. O proceso de desenvolvemento do sistema enfócase en integrar estas partes máis que en desenvolverlas desde o principio.

Estes tres modelos de procesos xenéricos utilízanse amplamente na práctica actual da enxeñería do software. Non se exclúen mutuamente e a miúdo utilízanse xuntos, especialmente para o desenvolvemento de sistemas grandes. Os subsistemas dentro dun sistema máis grande poden ser desenvolvidos utilizando enfoques diferentes. Polo tanto, aínda que é conveniente estudar estes modelos separadamente, debe entenderse que, na práctica, a miúdo se combinan.

Pressman separa entre o modelo **persoal** (modelo utilizado por cada desenvolvedor) e **modelo en equipo** (cando o proxecto é dirixido por varios profesionais) para o proceso de software. Ambos destacan a



medición, o planeamento e a autodirección como ingredientes clave para un proceso de software exitoso.

### **31.3 Ciclo de vida do software**

Segundo o estándar ISO-12207 **o ciclo de vida** dun sistema de información é *o marco de referencia que contén os procesos, as actividades e as tarefas involucradas no desenvolvemento, a explotación e o mantemento dun produto de software, abarcando a vida do sistema desde a definición dos requisitos ata a finalización do seu uso*. Tamén se podería definir o ciclo de vida dun sistema de información como *o conxunto de etapas polas que atravesa o sistema desde a súa concepción ata a súa retirada de servizo pasando polo seu desenvolvemento e explotación*.

Non existe un único **modelo de ciclo de vida** que defina os estados polos que pasa calquera produto software. Dado que existe unha gran variedade de aplicacións e que dita variedade supón situacións totalmente distintas, é natural que existan diferentes modelos de ciclo de vida. No entanto, todo modelo de ciclo de vida debe cubrir os seguintes obxectivos básicos:

- Definir as actividades a realizar e en qué orde; é dicir, determinar a orde das fases do proceso software.
- Establecer os criterios de transición para pasar dunha fase á seguinte.
- Proporcionar puntos de control para a xestión do proxecto, é dicir, calendario e organización.
- Asegurar a consistencia co resto dos sistemas de información da organización.

Cada proxecto debe seleccionar o modelo de ciclo de vida que sexa máis apropiado para o seu caso, o cal se elixe considerando unha serie de factores como: a cultura da organización, o desexo de asumir riscos, a área de aplicación, a volatilidade dos requisitos, a comprensión dos devanditos



requisitos, etc. En calquera proxecto software, o modelo de ciclo de vida permite responder as cuestións de qué se fará a continuación? e por canto tempo se fará? Dado que cada modelo de ciclo de vida ten as súas vantaxes e os seus inconvenientes, non se adoitan seguir na práctica os modelos na súa forma pura, senón que de acordo coas peculiaridades do sistema e a experiencia do persoal, pódense adoptar aspectos doutros modelos que sexan máis adecuados ao caso concreto.

É importante non confundir o concepto de ciclo de vida co de metodoloxía. Mentres que o ciclo de vida indica qué actividades cómpre realizar e en qué orde, a **metodoloxía** indica como avanzar na construción do sistema; isto é, con que técnicas, e entre as súas características está a de determinar os recursos a utilizar ou as persoas implicadas en cada actividade.

### **31.4 Modelos de ciclo de vida do software**

Unha posible clasificación sería a que divide os modelos de ciclo de vida en:

- **Modelos tradicionais:** Son os de máis ampla utilización. Dentro deste grupo estarían:
  - o Modelo en cadoiro.
  - o Modelos baseados en prototipos:
    - Modelo de construción de prototipos.
    - Modelo de desenvolvemento incremental.
    - Modelo de prototipado evolutivo.
- **Modelos alternativos**
  - o Modelo en espiral
  - o Modelos baseados en transformacións: A filosofía xeral é chegar a xerar código a partir dunhas especificacións transformándoas por medio de ferramentas. Segundo usemos unhas ou outras ferramentas teremos:



- As que usan técnicas de cuarta xeración (Roger Pressman): linguaxes non procedementais para consultas a BD; xeradores de código, de pantallas, de informes; ferramentas de manipulación de datos; facilidades gráficas de alto nivel.
  - Baseados en modelos de transformación (Carma McClure)  
=> Baseados en ferramentas CASE que permiten, seguindo o MCV clásico, pasar dunha etapa a outra aplicando as transformacións que dan as ferramentas.
- o Desenvolvemento de Software Baseado en Compoñentes (DSBC ou CBSB).

Separadamente destes modelos de ciclo de vida na actualidade existen novas alternativas:

- Proceso unificado de desenvolvemento do software. (PUDS)
- Programación Extrema.

#### *31.4.1 Modelo Codificar e Corrixir (Code and Fix)*

Este é o modelo básico utilizado nos inicios do desenvolvemento de software. Contén dous pasos:

- Escribir código.
- Corrixir problemas no código.

Trátase de primeiro implementar algo de código e logo pensar acerca de requisitos, deseño, validación, e mantemento. Este modelo ten tres problemas principais:

- Logo dun número de correccións, o código pode ter unha moi mala estrutura que fai que os arranxos sexan moi custosos. Isto fixo ver a necesidade dunha fase previa de deseño antes da de codificación.
- Frecuentemente, aínda o software ben deseñado non se axusta ás necesidades do usuario, polo que ben é rexeitado ou ben a súa



reconstrución é moi cara. Isto conduciu á necesidade de introducir unha fase de análise de requisitos antes do deseño.

- O código é difícil de reparar pola súa pobre preparación para probar e modificar. Este problema fixo resaltar a necesidade da planificación e preparación das distintas tarefas en cada fase.

#### *31.4.2 Modelo por Etapas e Modelo en Cadoiro*

Os problemas apuntados do modelo *Code and Fix* levaron á necesidade de realizar o desenvolvemento do software seguindo un modelo de etapas sucesivas, o modelo **por etapas** (Stage Wise) que considera as seguintes: Planificación, Especificacións de operación, Especificacións de codificación, Codificación, Proba de cada unidade, Proba de integración, Eliminación de problemas, e Avaliación do sistema.

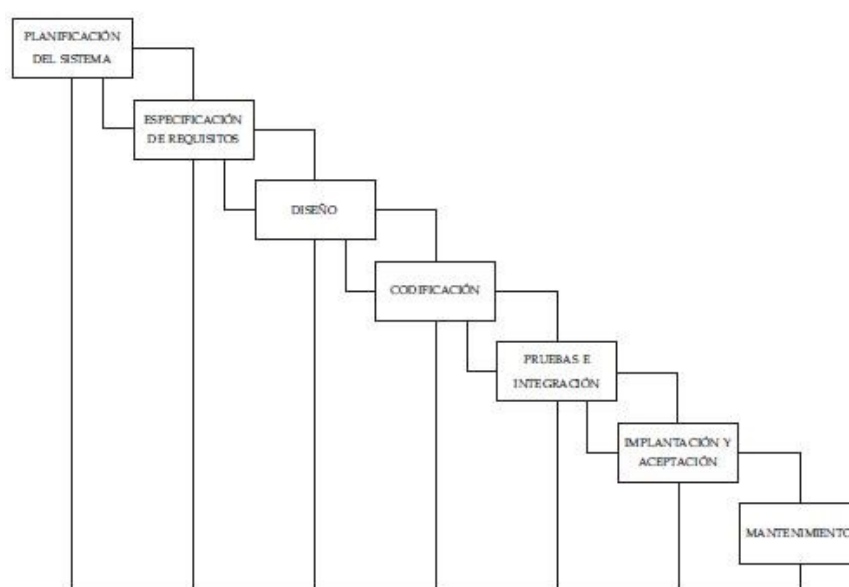
O modelo **en cadoiro** introduce unha serie de melloras respecto ao modelo por etapas, tales como considerar a realización de bucles de realimentación entre etapas, permitindo que se poidan resolver os problemas detectados nunha etapa, na etapa anterior e permitir a incorporación inicial do prototipado a fin de captar as especificacións durante a análise, ou para probar distintas solucións durante o deseño.

O modelo en cadoiro componse dunha serie de fases que se suceden secuencialmente, xerándose en cada unha delas uns resultados que serán necesarios para iniciar a fase seguinte. É dicir, a evolución do produto software prodúcese a través dunha secuencia ordenada de transicións dunha fase á seguinte, segundo unha orde lineal. O número de fases neste modelo é irrelevante, xa que o que o caracteriza é a *secuencialidade* das mesmas e a *necesidade de completar* cada unha delas para pasar á seguinte. O modelo do ciclo de vida en cadoiro está rexido pola documentación, é dicir, a decisión do paso dunha fase á seguinte tómase en función de se a documentación asociada á devandita fase está completa ou non. Con todo, esta forma de proceder non é a máis adecuada para



alguns tipos de software como o que se usa nas aplicacións interactivas e de usuario final.

Desde a súa presentación, o modelo en cadoiro debe ter un papel fundamental no desenvolvemento de proxectos software. Foi, e aínda segue sendo, o máis utilizado, tanto que este modelo coñécese co nome de “ciclo de vida clásico”, aínda que incorporando infinidade de variacións que eliminan o carácter simplista do mesmo. Aínda así, existen unha serie de limitacións que xustifican a necesidade de definir outros modelos.



Como se indicou anteriormente, as fases que comprende o ciclo de vida clásico son irrelevantes, tanto en número, como en cales sexan esas fases sempre que se produzan secuencialmente. Posiblemente, o modelo clásico máis utilizado sexa o modelo de sete fases que son:

- **Planificación do sistema:** Nesta fase é necesario fixar o ámbito do traballo a realizar, os recursos necesarios, as tarefas a realizar, as referencias a ter en conta, o custo estimado do proxecto, a composición do equipo de desenvolvemento e a orde das actividades.
- **Especificación de requisitos:** Nesta fase é preciso analizar, entender e documentar o problema que o usuario trata de resolver co sistema e hanse especificar con detalle as funcións, obxectivos e



restricións do mesmo, a fin de que usuarios e desenvolvedores poidan tomar estas como punto de partida para acometer o resto do sistema. É dicir, na fase de especificación de requisitos trátase de definir **qué** debe facer o sistema e identificar a información a procesar, as funcións a realizar, o rendemento do sistema, as interfaces con outros sistemas e as ligaduras de deseño.

- **Deseño:** Arrinca das especificacións da fase anterior. Na fase de deseño, unha vez elixida a mellor alternativa, débese crear a solución ao problema descrito atendendo a aspectos de interfaces de usuario, estrutura do sistema e decisións sobre a implantación posterior. A fase de deseño trata de definir o *como*.
- **Codificación:** Esta fase consiste en traducir as especificacións e representacións do Deseño a unha linguaxe de programación capaz de ser interpretada e executada polo ordenador.
- **Probas e integración:** Unha vez que se teñen os programas no formato adecuado ao ordenador, hai que levar a cabo as probas necesarias que aseguren a corrección da lóxica interna do programa e que este cobre as funcionalidades previstas. A integración das distintas partes que compoñen a aplicación ou o sistema debe garantir o bo funcionamento do conxunto.
- **Implantación e aceptación do sistema:** O obxectivo desta fase é conseguir a aceptación do sistema por parte dos usuarios do mesmo e levar a cabo as actividades necesarias para a súa posta en produción.
- **Mantemento do sistema:** A fase de mantemento comeza unha vez que o sistema foi entregado ao usuario e continúa mentres permanece activa a súa vida útil. Pode deberse a erros non detectados previamente (correctivo), a modificacións, melloras ou ampliacións solicitadas polos usuarios (perfectivo, ou aumentativo) ou a adaptacións requiridas pola evolución do contorno tecnolóxico ou cambios normativos (mantemento adaptativo).



As principais **críticas** ao modelo céntranse nas súas características básicas, é dicir secuencialidade e utilización dos resultados dunha fase para acometer a seguinte, de maneira que o sistema só se pode validar cando está terminado. En canto ao fluxo secuencial, os proxectos reais raramente seguen o fluxo secuencias que propón o modelo. Sempre ocorren interaccións e nas últimas fases, sobre todo, pódense realizar en paralelo algunhas áreas por exemplo codificación e probas. Unha aplicación do modelo en sentido estrito obrigaría á “conxelación” dos requisitos dos usuarios, suposto este completamente afastado da realidade. O modelo non contempla a posibilidade de realimentación entre fases. Doutra banda, o modelo non prevé revisións ou validacións intermedias por parte do usuario, e así os resultados dos traballos só se ven ao final dunha serie de tarefas e fases de tal forma que se se produciu un erro nas primeiras fases este só se detectará ao final e a súa corrección terá un custo moi elevado, posto que será preciso refacer todo o traballo desde o principio.

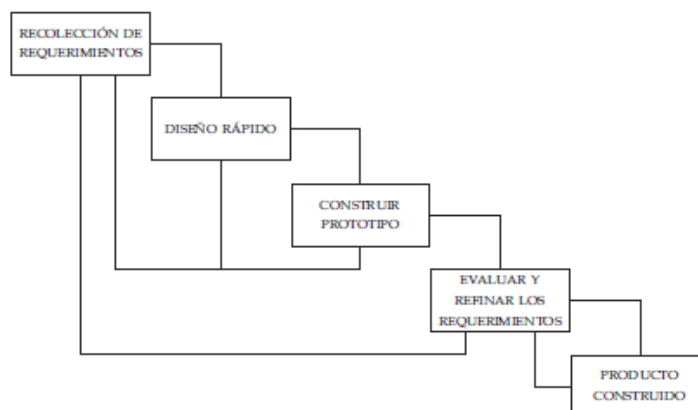
#### *31.4.3 Modelos baseados en prototipos*

Permiten aos desenvolvedores construír rapidamente versións temperás dos sistemas software que poden avaliar os usuarios. Existen varios modelos derivados do uso de prototipos

##### *31.4.3.1 Prototipado rápido*

Os prototipos deben poder construírse con facilidade para avalialos nunha temperá fase do desenvolvemento e ademais han de ser baratos e desenvolvidos en pouco tempo. Tamén se denominan *de usar e tirar*.





O prototipo serve para crear e validar a especificación, e para que o usuario teña unha idea de como será o software antes de que comece o desenvolvemento. É importante precisar que o prototipo se constrúe só para servir como mecanismo de definición dos requisitos funcionais. Posteriormente debe desbotarse e debe construírse o sistema cos criterios normais de calidade e mantemento, seguindo, por exemplo, o ciclo de vida clásico, xa que xeralmente o prototipo construír tomando decisións de implementación contrarias ao bo criterio de desenvolvemento de software. Os obxectivos do prototipo son:

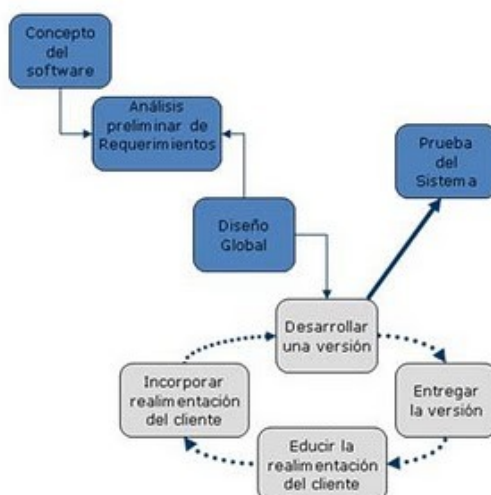
- Reducir o risco de construír un produto que se afaste das necesidades do usuario
- Reducir o custo de desenvolvemento ao diminuír as correccións en etapas avanzadas do mesmo.
- Aumentar as posibilidades de éxito do produto.

O principal problema deste modelo é que o usuario ve no prototipo o que parece ser unha versión de traballo do software, sen saber que coa présa de facer que funcione non se ha ter en conta a calidade do software global ou a facilidade de mantemento a longo prazo. Cando se informa de que o produto se debe construír outra vez para que se poidan manter os niveis altos de calidade, o cliente non o entende e pide que se apliquen uns pequenos axustes que poidan facer do prototipo un produto final.



### 31.4.3.2 Prototipado evolutivo

Neste tipo de ciclo de vida constrúese unha implementación parcial do sistema que satisfai os requisitos coñecidos, a cal é utilizada polo usuario para chegar a comprender mellor a totalidade dos requisitos que desexa.



Desde un punto de vista xenérico, pódese dicir que os modelos evolutivos se encamiñan a conseguir un sistema flexible que se poida expandir, de forma que se poida realizar rapidamente un novo sistema cando cambian os requisitos. Estes modelos consisten en implementar un produto software operativo e facerlle evolucionar de acordo coa propia experiencia operacional. Están especialmente indicados en situacións en que se utilizan linguaxes de cuarta xeración (L4G) e para aquelas outras en que o usuario non pode dicir o que require, pero que o recoñecerá cando o vexa. Os modelos evolutivos danlle ao usuario unha rápida capacidade de operación inicial e unha boa base para determinar melloras do sistema. Está relacionado co concepto de RAD (Rapid Application Development - Desenvolvo Rápido de Aplicacións), que identifica os asistentes, persoais e contornas de fácil e rápida creación de software.



A diferenza fundamental entre o prototipado rápido e o evolutivo estriba en que mentres que no primeiro caso se asume que existen unha serie de requisitos reais, aínda que para establecer o que o usuario quere realmente é necesario establecer unha serie de iteracións antes de que os requisitos se estabilicen ao final, no caso evolutivo asúmese desde o principio que os requisitos cambian continuamente.

No prototipo rápido o lóxico é implementar só aqueles aspectos do sistema que se entenden mal, mentres que no prototipo evolutivo o lóxico é comezar polos aspectos que mellor se comprenden e seguir construíndo apoiados nos puntos fortes e non nos débiles. Como resultado deste xeito de desenvolvemento, a solución software evoluciona, achegándose cada vez máis ás necesidades do usuario. Agora ben, pasado un tempo o sistema software así construído deberá ser refeito ou sufrir unha profunda reestruturación co fin de seguir evolucionando.

O modelo de prototipado evolutivo (Evolutionary Development model) tamén ten as súas **dificultades**. Pódese considerar como unha nova versión, utilizando linguaxes de programación de máis alto nivel, do vello modelo CODE-AND-FIX. Outro inconveniente que presenta é partir da suposición, moitas veces non realista, de que o sistema operacional do usuario final será o suficientemente flexible como para poder incorporar camiños de evolución futuros non planificados con anterioridade.

#### *31.4.3.3 Modelo de Desenvolvemento Incremental*

O modelo de desenvolvemento incremental consiste en desenvolver un sistema que satisfaga unha parte dos requisitos especificados e posteriormente ir creando versións que incorporen os requisitos que faltan ata chegar ao sistema final. Actuando así, preténdese dispoñer pronto dun sistema que aínda que sexa incompleto, sexa utilizable e satisfaga parte dos requisitos, evitando de paso o efecto big-bang, é dicir, que durante un



período longo de tempo non se teña nada e de súpeto haxa unha situación completamente nova. Por outra banda, tamén se logra que o usuario se implique estreitamente na planificación dos pasos seguintes.

O modelo de desenvolvemento incremental tamén se utiliza para evitar a demanda de funcionalidades excesivas ao sistema por parte dos usuarios, xa que como a estes lle resulta difícil definir as súas necesidades reais, tenden a pedir demasiado. Actuando con este modelo aténdese primeiro ás funcionalidades esenciais e as funcionalidades accesorias só se inclúen nas versións sucesivas cando realmente son necesarias.

A diferenza entre o modelo de desenvolvemento evolutivo e o modelo de desenvolvemento incremental radica en dous aspectos:

- O modelo incremental parte da hipótese de que se coñecen todos os requisitos e estes vanse incorporando ao sistema en versións sucesivas. No modelo evolutivo só se coñecen uns poucos requisitos e os restantes vanse descubriendo en sucesivas evolucións do prototipo.
- Cada vez que se desenvolve unha nova versión, no modelo evolutivo é unha versión de todo o sistema, mentres que no incremental é unha versión anterior sen cambios, máis un número de novas funcións.

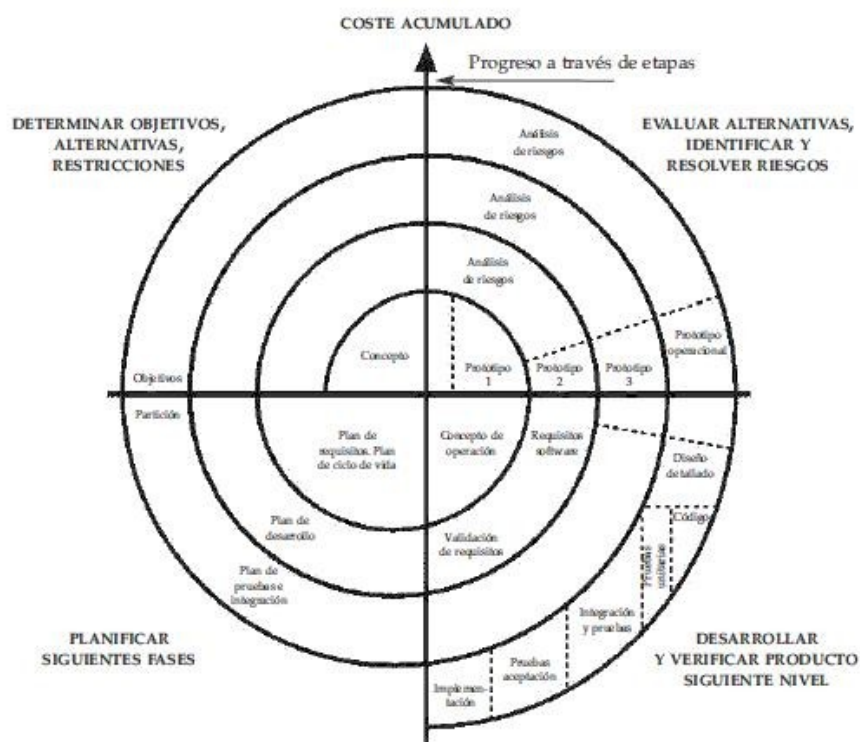
#### *31.4.4 Modelo en Espiral*

O modelo *en espiral*, proposto orixinalmente por Boehm, é un modelo de proceso de software evolutivo que conxuga a natureza iterativa de construción de prototipos cos aspectos controlados e sistemáticos do modelo en cadoiro. Proporciona o potencial para o desenvolvemento rápido de versións incrementais do software. No modelo espiral, o software desenvólvese nunha serie de versións incrementais. Durante as primeiras iteracións, a versión incremental podería ser un modelo en papel ou un



prototipo. Durante as últimas iteracións, prodúcese versións cada vez máis completas do sistema deseñado. As principais diferenzas entre o modelo en espiral e os modelos de ciclo de vida máis tradicionais son:

- No modelo en espiral hai un recoñecemento explícito das diferentes **alternativas** para alcanzar os obxectivos do proxecto.
- O modelo en espiral céntrase na identificación dos **riscos** asociados a cada alternativa e no xeito de resolver os devanditos riscos.
- No modelo en espiral os proxectos divídense en ciclos (ciclos de espiral), avanzándose no desenvolvemento mediante consensos ao final de cada ciclo.
- O modelo en espiral adáptase a calquera tipo de actividade.



O modelo en espiral reflicte a idea de que cada ciclo implica unha progresión no desenvolvemento do produto software que aplica a mesma secuencia de pasos para cada parte do produto e para cada un dos seus niveis de elaboración, desde a concepción global ata a codificación



individual de cada programa. Esta secuencia de pasos, iterativa en cada fase do desenvolvemento, componse das catro actividades seguintes:

- **Planificación:** Este primeiro paso co que comeza cada ciclo de espiral consiste na identificación dos obxectivos da parte do produto que está sendo elaborada (funcionalidade, rendemento, adaptación aos cambios, etc.), identificación das alternativas principais para realizar ou implementar esta parte do produto, e a identificación das restricións impostas (custo, prazo de realización, interfaces, etc.).
- **Análise de riscos:** Comeza coa avaliación de cada alternativa con respecto aos obxectivos e ás restricións. Este proceso de avaliación identificará áreas de incerteza que son fontes significativas de risco no proxecto. Decidirase como resolver os riscos asociados á alternativa elixida.
- **Enxeñería.** Este paso consiste no desenvolvemento e verificación do produto obxecto da fase (ciclo de espiral) en que nos atopemos. Como esta implementación está dirixida polo risco, o desenvolvemento poderá seguir as pautas dun prototipado evolutivo, as do ciclo de vida clásico, as orientadas a transformacións automáticas, ou calquera outro enfoque do desenvolvemento. En definitiva, isto permite ao modelo en espiral acomodarse a calquera mestura de estratexias de desenvolvemento.
- **Avaliación do cliente.** Unha característica importante do modelo en espiral é que cada ciclo de espiral se completa cunha revisión na que participan aqueles que teñen relación co produto (desenvolvedores, usuarios, etc.). Esta revisión inclúe todos os produtos desenvolvidos durante o ciclo, os plans para o seguinte ciclo e os recursos necesarios para levalos a cabo.

Segundo isto, o modelo pódese representar mediante uns ciclos externos de espiral, que representan as fases en que se dividiu o desenvolvemento do proxecto software, normalmente as do modelo clásico, e uns ciclos internos, iterativos para cada fase, nos que se levan a cabo as catro



actividades antes citadas. A dimensión radial indica os custos de desenvolvemento acumulativos, mentres que a dimensión angular indica o progreso feito en cumprimentar cada fase.

A principal vantaxe do modelo en espiral é o amplo rango de opcións a que pode axustarse e que estas permiten utilizar os modelos de proceso de construción de software tradicionais. Por outra banda, a súa orientación ao risco evita, se non elimina, moitas das posibles dificultades. Outras **vantaxes** son:

- Concentra a súa atención en opcións que permiten a reutilización de software xa existente.
- Céntrase na eliminación de erros e alternativas pouco atractivas.
- Non establece procedementos diferentes para o desenvolvemento do software e o mantemento do mesmo.
- Proporciona un marco estable para desenvolvementos integrados hardware-software.
- Permite preparar a evolución do ciclo de vida do produto software, así como o crecemento e cambios deste.
- Permite incorporar obxectivos de calidade no desenvolvemento de produtos software.
- Adáptase moi ben ao deseño e programación orientada a obxectos. Posiblemente con este método é cando obtén mellores resultados.

En canto aos **inconvenientes** que presenta a utilización do modelo en espiral, cabe citar:

- Dificultade para adaptar a súa aplicabilidade ao software contratado, debido á pouca flexibilidade e liberdade deste.
- Dependencia excesiva da experiencia que se teña na identificación e avaliación de riscos.
- Necesidade dunha elaboración adicional dos pasos do modelo, o que depende tamén, en gran medida, da experiencia do persoal.



#### 31.4.5 Modelos baseados en Transformacións

Estes modelos, tamén chamados *ciclo de vida con produción automática de deseño e código*, propuxéronse como solución ao problema que presentan os modelos de desenvolvemento evolutivo de producir software mal estruturado. Baséanse na posibilidade de converter automaticamente unha especificación formal dun produto software nun programa que satisfaga as especificacións, utilizando ferramentas de cuarta xeración. Para iso, os pasos típicos que seguen estes modelos son:

1. Especificación formal do produto tal como o permita a comprensión inicial do problema.
2. Transformación automática da especificación en código.
3. Realizar bucles iterativos para mellorar o rendemento do código resultante.
4. Probar o produto resultante.
5. Reaxustar as especificacións para deixalas en concordancia co resultado da experiencia operativa e volver xerar o código a partir das especificacións, volvendo optimizar e probar o produto.

O modelo de transformación, xa que logo, evita a dificultade de ter que modificar o código pouco estruturado (por pasar por sucesivas reoptimizacións), posto que as modificacións as aplica sobre a especificación de partida. Isto, tamén evita o tempo adicional que se empregaría nos pasos intermedios de deseño, codificación e probas.

A dificultade que presentan estes modelos é que as posibilidades de transformación automática xeralmente só están dispoñibles para produtos relativamente pequenos e aplicados a unhas áreas moi limitadas. Tamén comparte algunhas das dificultades do modelo de desenvolvemento evolutivo tales como, por exemplo, a suposición de que o sistema



operacional do usuario final se prestará a evolucións non planificadas con anterioridade.

Dentro deste tipo de modelos atópanse:

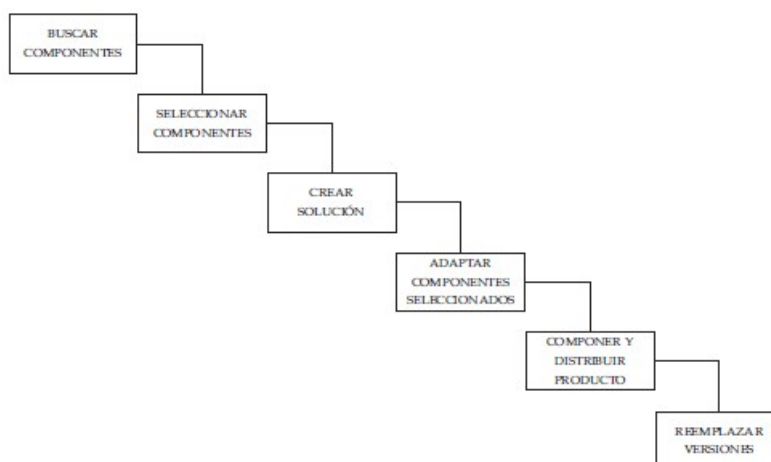
- Os que usan técnicas de cuarta xeración (Roger Pressman): Adoitan estar baseados en ferramentas de cuarta xeración. Estes permiten a xeración de código rápido. Neles indícase qué se quere obter, non cómo.
- Baseados en modelos de transformación (Carma McClure) => Baseados en ferramentas CASE que permiten, seguindo o MCV clásico, pasar dunha etapa a outra aplicando as transformacións que dan as ferramentas.

En ambos casos, a filosofía xeral é chegar a xerar código a partir dunhas especificacións transformándoas por medio de ferramentas.

#### *31.4.6 Desenvolvemento baseado en compoñentes*

A complexidade dos sistemas computacionais actuais levounos a buscar a reutilización do software existente. O desenvolvemento de software baseado en compoñentes permite reutilizar pezas de código preelaborado que permiten realizar diversas tarefas, o que leva aparellados diversos beneficios como as melloras á calidade, a redución do ciclo de desenvolvemento e o maior retorno sobre o investimento. En esencia, un compoñente é unha peza de código preelaborado que encapsula algunha funcionalidade exposta a través de interfaces estándar. Os compoñentes son os "ingredientes das aplicacións", que se xuntan e combinan para levar a cabo unha tarefa. O paradigma de ensamblar compoñentes e escribir código para facer que estes compoñentes funcionen coñécese como “Desenvolvemento de Software Baseado en Compoñentes”.





Os pasos de que consta o ciclo de desenvolvemento para un sistema baseado en compoñentes son:

1. Buscar compoñentes, tanto COTS (Comercial Off-The-Shelf) como non COTS.
2. Seleccionar os compoñentes máis adecuados para o sistema.
3. Crear unha solución composta que integre a solución previa.
4. Adaptar os compoñentes seleccionados de forma que se axusten ao modelo de compoñentes ou aos requisitos da aplicación.
5. Compoñer e distribuír o produto.
6. Substituír versións anteriores ou manter as partes COTS e non COTS do sistema.

Ademais dos problemas inherentes á reutilización do software, os produtos COTS presentan problemas específicos como incompatibilidade, inflexibilidade (non existe código fonte), complexidade (esfuerzo de aprendizaxe) ou cambio de versións, polo que o establecemento de métodos sistemáticos e repetibles para avaliar e seleccionar devanditos compoñentes é un aspecto importante para o desenvolvemento do software baseado en compoñentes e, en xeral, para a Enxeñería do Software Baseada en Compoñentes (ISBC).

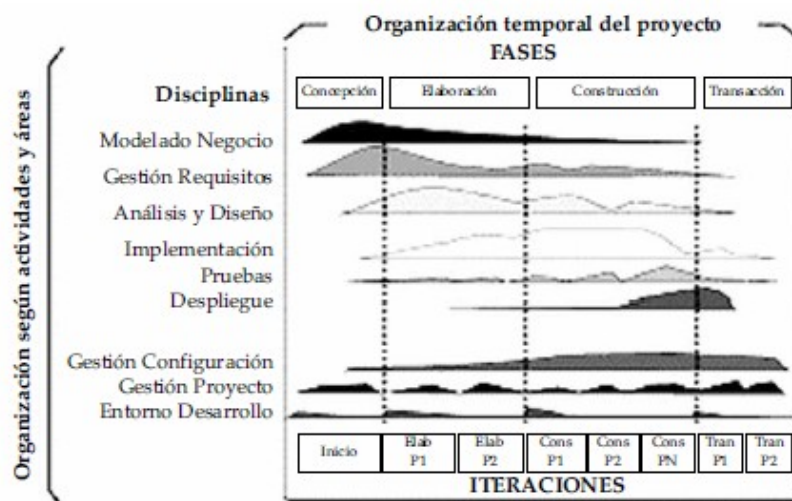
Entre as vantaxes do Desenvolvemento baseado en compoñentes temos que se reducen tempos e custos de desenvolvemento e se aumenta a



fiabilidade. Entre os inconvenientes, teremos a dificultade para recoñecer os compoñentes potencialmente reutilizables, dificultade de catalogación e recuperación e os problemas de xestión de configuración.

### 31.4.7 Proceso Unificado de Desenvolvemento de software (PUDS)

En realidade é unha metodoloxía que propón un modelo de ciclo de vida. Está desenvolvida por tres pais da IS moderna: Yourdon, Booch e Rumbaugh. Propón un modelo de ciclo de vida iterativo e incremental, centrado nunha arquitectura que guía o desenvolvemento do sistema, cuxas actividades están dirixidas por casos de uso e soporta as técnicas orientadas a obxectos. PUDS impulsa un control de calidade e unha xestión de riscos obxectivos e continuos.



O PUDS componse de fases, iteracións e ciclos. Unha fase é o intervalo de tempo entre dous fitos importantes do proceso durante a cal se cumpre un conxunto ben definido de obxectivos, se completan entregables e se toman as decisións sobre se pasar ou non á seguinte fase. As **fases** son:

1. **Iniciación.** Nesta fase establécese a visión do negocio, que inclúe o contexto do negocio, os factores de éxito, e a previsión económica.



Para completar a visión do negocio xérase un plan do proxecto, unha descrición dos posibles riscos e do propio proxecto (requisitos principais do proxecto, restricións e características claves)

2. **Elaboración.** É onde o proxecto comeza a tomar forma. Nesta fase faise a análise do dominio do problema e obtense unha idea básica da arquitectura do sistema, ademais de revisarse os riscos. Nesta fase o proxecto aínda pode cancelarse ou redeseñarse.
3. **Construción.** Nesta fase o enfoque trasládase ao desenvolvemento de compoñentes e outras características do sistema que está sendo deseñado. Aquí realízase o groso das tarefas de codificación. En proxectos grandes pódese separar a fase en varias iteracións para dividir os casos de uso en segmentos manexables que produzan prototipos funcionais.
4. **Transición.** O produto implántase na organización do usuario final. Aquí lévase a cabo a formación dos usuarios finais e as probas de aceptación do sistema para validalo contra as expectativas do usuario.

En cada fase hai unha ou varias iteracións. Unha **iteración** ofrece como resultado un incremento do produto desenvolvido que engade ou mellora as funcionalidades do sistema en desenvolvemento. Cada fase e iteración céntrase en diminuír algún risco e conclúe cun fito ben definido. O paso a través das 4 fases constitúe un **ciclo** de desenvolvemento e produce unha xeración de software. O primeiro ciclo é o inicial e despois serán ciclos de evolución do sistema.

Os fluxos de traballo do proceso son os seguintes:

- Modelado do negocio. O obxectivo é establecer unha mellor comprensión e unha mellor canle de comunicación entre os clientes e os expertos en sistemas.
- Requisitos. O obxectivo é describir o que o sistema debe facer.



- Análise e deseño. Aquí móstrase a forma que terá o sistema na fase de implementación.
- Implementación. Codificar e realizar probas unitarias.
- Probas. Realízanse probas de integración.
- Despregue. Inclúe unha ampla variedade de actividades como a xeración de versións estables ou a distribución e instalación do software.
- Configuración e xestión de cambios.
- Xestión do proxecto. Realízase a 2 niveis, un nivel de gran groso que trata a planificación das fases e outro nivel de gran fino que trata a planificación das iteracións.
- Contorno.

#### *31.4.8 Modelo de métodos formais*

O modelo de métodos formais comprende un conxunto de actividades que conducen á especificación matemática do software de computadora. Os métodos formais permiten que un enxeñeiro de software especifique, desenvolva e verifique un sistema baseado en computadora aplicando unha notación rigorosa e matemática. Algunhas organizacións de desenvolvemento do software actualmente aplican unha variación deste enfoque, chamado **enxeñería do software de sala limpa**.

Cando se utilizan métodos formais a ambigüidade, o incompleto e a inconsistencia descóbrense e corríxense máis facilmente, non mediante unha revisión ad hoc para o caso, senón mediante a aplicación da análise matemática. Cando se utilizan métodos formais durante o deseño, serven como base para a verificación de programas e, por conseguinte, permiten que o enxeñeiro do software descubra e corrixa erros que non se puideron detectar doutro xeito. Os modelos de métodos formais ofrecen a promesa



dun software libre de defectos. Con todo, falouse dunha gran preocupación sobre a súa aplicabilidade nunha contorna de xestión:

1. O desenvolvemento de modelos formais actualmente é bastante caro e leva moito tempo.
2. Requírese un estudo detallado porque poucos responsables do desenvolvemento de software teñen os antecedentes necesarios para aplicar métodos formais.
3. É difícil utilizar os modelos como un mecanismo de comunicación con clientes que non teñen moitos coñecementos técnicos.

No entanto, é posible que o enfoque a través de métodos formais teña máis partidarios entre os desenvolvedores que deben construír software de moita seguridade e robustez.

#### *31.4.9 Programación Extrema (extreme Programming)*

Na programación extrema, todos os requisitos exprésanse como escenarios (chamados historias de usuario), os cales se implementan directamente como unha serie de tarefas. Os programadores traballan en parellas e desenvolven probas para cada tarefa antes de escribiren o código. Todas as probas débense executar satisfactoriamente cando o código novo se integre ao sistema. Existe un pequeno espazo de tempo entre as entregas do sistema. A programación extrema implica varias prácticas que se axustan aos principios dos métodos áxiles:

1. O desenvolvemento incremental lévase a cabo través de entregas do sistema pequenas e frecuentes e por medio dun enfoque para a descrición de requisitos baseado nas historias de cliente ou escenarios que poden ser a base para o proceso de planificación.
2. A participación do cliente lévase a cabo a través do compromiso a tempo completo do cliente no equipo de desenvolvemento. Os



- representantes dos clientes participan no desenvolvemento e son os responsables de definir as probas de aceptación do sistema.
3. O interese nas persoas, no canto de en os procesos, lévase a cabo a través da programación en parellas, a propiedade colectiva do código do sistema, e un proceso de desenvolvemento sustentable que non implique excesivas xornadas de traballo.
  4. O cambio lévase a cabo a través das entregas regulares do sistema, un desenvolvemento previamente probado e a integración continua.
  5. O mantemento da simplicidade lévase a cabo a través da refactorización constante para mellorar a calidade do código e a utilización de deseños sinxelos que non prevén cambios futuros no sistema.

Os clientes do sistema son parte do equipo de desenvolvemento e discuten escenarios con outros membros do equipo. Desenvolven conxuntamente unha «tarxeta de historias» (story card) que recolle as necesidades do cliente. O equipo de desenvolvemento intentará, daquela, implementar ese escenario nunha entrega futura do software. Unha vez que se desenvolveron as tarxetas de historias, o equipo de desenvolvemento divídeas en tarefas e estima o esforzo e recursos requiridos para a súa implementación. O cliente establece entón a prioridade das historias a implementar.

O problema coa implementación de cambios imprevistos é que tenden a degradar a estrutura do software, polo que os cambios se fan cada vez máis difíciles de implementar. A programación extrema aborda este problema suxerindo que se debe refactorizar constantemente o software. Isto significa que o equipo de programación busca posibles melloras do software e as implementa inmediatamente. Polo tanto, o software sempre debe ser fácil de entender e cambiar cando se implementen novas historias.



Outra práctica innovadora que se introduciu é que os programadores traballan en parellas para desenvolver o software. As vantaxes disto son que apoia a idea da propiedade e responsabilidade comúns do sistema, actúa como un proceso de revisión informal do código e axuda na refactorización

### **31.6 Metodoloxías de desenvolvemento de sistemas de información.**

Un proceso de software detallado e completo adoita denominarse “Metodoloxía”. As metodoloxías baséanse nunha combinación dos modelos de proceso xenéricos (cadoiro, evolutivo, incremental, etc.).

Adicionalmente unha metodoloxía debería definir con precisión os produtos, roles e actividades involucrados, xunto con prácticas e técnicas recomendadas, guías de adaptación da metodoloxía ao proxecto, guías para uso de ferramentas de apoio, etc. Habitualmente utilízase o termo “método” para referirse a técnicas, notacións e guías asociadas, que son aplicables a unha (ou algunhas) actividades do proceso de desenvolvemento, por exemplo, adoita falarse de métodos de análises e/ou deseño.

A comparación e/ou clasificación de metodoloxías non é unha tarefa sinxela debido á diversidade de propostas e diferenzas no grado de detalle, información dispoñible e alcance de cada unha delas. A grandes liñas, se tomamos como criterio as notacións utilizadas para especificar produtos producidos en actividades de análises e deseño, podemos clasificar as metodoloxías en dous grupos: Metodoloxías Estruturadas e Metodoloxías Orientadas a Obxectos. Por outra banda, considerando a súa filosofía de desenvolvemento, aquelas metodoloxías con maior énfase na planificación e control do proxecto, en especificación precisa de requisitos e modelado, reciben o apelativo de Metodoloxías Tradicionais (ou pexorativamente denominada Metodoloxías Pesadas, ou Peso Pesado). Outras metodoloxías,



denominadas Metodoloxías Áxiles, están máis orientadas á xeración de código con ciclos moi curtos de desenvolvemento, diríxense a equipos de desenvolvemento pequenos, fan especial fincapé en aspectos humanos asociados ao traballo en equipo e involucran activamente ao cliente no proceso. A continuación revísanse brevemente algunhas destas categorías de metodoloxías.

- **Metodoloxías estruturadas:** Os métodos estruturados comezaron a desenvolverse a finais dos 70 coa Programación Estruturada. A mediados dos 70 apareceron técnicas para o Deseño (por exemplo: o diagrama de Estrutura) primeiro e posteriormente para a Análise (por exemplo: Diagramas de Fluxo de Datos). Estas metodoloxías son particularmente apropiadas en proxectos que utilizan para a implementación linguaxes de 3ª e 4ª xeración. Exemplos de metodoloxías estruturadas de ámbito gobernamental: MERISE (Francia), MÉTRICA (España), SSADM (Reino Unido). Exemplos de propostas de métodos estruturados no ámbito académico: Gane & Sarson, Ward & Mellor, Yourdon & DeMarco e Information Engineering.
- **Metodoloxías orientadas a obxectos:** A súa historia vai unida á evolución das linguaxes de programación orientada a obxectos. A fins dos 80 comezaron a consolidarse algúns métodos Orientados a Obxectos. En 1995 Booch e Rumbaugh propoñen o Método Unificado coa ambiciosa idea de conseguir unha unificación dos seus métodos e notacións, que posteriormente se reorienta a un obxectivo máis modesto, para dar lugar ao Unified Modeling Language (UML), a notación OO máis popular na actualidade. Algúns métodos OO con notacións predecesoras de UML son: OOAD (Booch), OOSE (Jacobson), Coad & Yourdon, Shaler & Mellor e OMT (Rumbaugh). Algunhas metodoloxías orientadas a obxectos que utilizan a notación UML son: Rational Unified Process (RUP), OPEN, MÉTRICA (que tamén soporta a notación estruturada).



- **Metodoloxías tradicionais (non áxiles):** As metodoloxías non áxiles son aquelas que están guiadas por unha forte planificación durante todo o proceso de desenvolvemento; chamadas tamén metodoloxías tradicionais ou clásicas, onde se realiza unha intensa etapa de análise e deseño antes da construción do sistema. Todas as propostas metodolóxicas antes indicadas poden considerarse como metodoloxías tradicionais..
- **Metodoloxías áxiles:** Un proceso é áxil cando o desenvolvemento de software é **incremental** (entregas pequenas de software, con ciclos rápidos), **cooperativo** (cliente e desenvolvedores traballan xuntos constantemente cunha próxima comunicación), **sinxelo** (o método en si mesmo é fácil de aprender e modificar, ben documentado), e **adaptable** (permite realizar cambios de último momento). Algunhas das metodoloxías áxiles identificadas son Extreme Programming, Scrum, Familia de Metodoloxías Crystal, Feature Driven Development, Proceso Unificado Rational, Dynamic Systems Development Method, Adaptive Software Development. Veranse con máis detalle nun apartado posterior.
- **Proceso Unificado de Rational:** É un proceso de desenvolvemento de software e xunto coa Linguaxe Unificada de Modelado UML, constitúe a metodoloxía estándar máis utilizada para a análise, implementación e documentación de sistemas orientados a obxectos. O RUP non é un sistema con pasos firmemente establecidos, senón un conxunto de metodoloxías adaptables ao contexto e necesidades de cada organización. Verase nun apartado posterior.
- **Métrica V3:** A metodoloxía MÉTRICA Versión 3 ofrece ás Organizacións un instrumento útil para a sistematización das actividades que dan soporte ao ciclo de vida do software. Verase nun apartado posterior.
- **Open Source Development Software:** Open Source é software desenvolvido coa falta de coordinación, onde os programadores



colaboran libremente, utilizando o código fonte distribuído e a infraestrutura de comunicacións de Internet. O código aberto baséase na filosofía do software libre, con todo, estende esta ideoloxía lixeiramente para presentar un enfoque máis comercial que inclúe tanto un modelo de negocio como unha metodoloxía de desenvolvemento.

### **31.7 Métrica Versión 3.**

A metodoloxía MÉTRICA Versión 3 ofrece ás Organizacións un instrumento útil para a sistematización das actividades que dan soporte ao ciclo de vida do software dentro do marco que permite alcanzar os seguintes obxectivos:

- Proporcionar ou definir Sistemas de Información que axuden a conseguir os fins da Organización mediante a definición dun marco estratéxico para o desenvolvemento dos mesmos.
- Dotar á organización de produtos software que satisfagan as necesidades dos usuarios dando unha maior importancia á análise de requisitos.
- Mellorar a produtividade dos departamentos de Sistemas e Tecnoloxías da Información e as Comunicacións, permitindo unha maior capacidade de adaptación aos cambios e tendo en conta a reutilización na medida do posible.
- Facilitar a comunicación e entendemento entre os distintos participantes na produción de software ao longo do ciclo de vida do proxecto, tendo en conta o seu papel e responsabilidade, así como as necesidades de todos e cada un deles.
- Facilitar a operación, mantemento e uso dos produtos software obtidos.



Na elaboración de MÉTRICA Versión 3 deberanse ter en conta os métodos de desenvolvemento máis estendidos, así como os últimos estándares de enxeñería do software e calidade, ademais de referencias específicas en canto a seguridade e xestión de proxectos. Nunha única estrutura, a metodoloxía MÉTRICA Versión 3 cobre distintos tipos de desenvolvemento: estruturado e orientado a obxectos, facilitando a través de interfaces a realización dos procesos de apoio ou organizativos: Xestión de Proxectos, Xestión de Configuración, Aseguramento de Calidade e Seguridade.

No que se refire a estándares cómpre en conta como referencia o Modelo de Ciclo de Vida de Desenvolvemento proposto na norma ISO 12.207 *"Information technology - Software life cycle processes"*. Seguindo este modelo elaborouse a estrutura de MÉTRICA Versión 3 na que se distinguen procesos principais (Planificación, Desenvolvemento e Mantemento) e interfaces (Xestión de Proxectos, Aseguramento da Calidade, Seguridade e Xestión de Proxectos) cuxo obxectivo é dar soporte ao proxecto nos aspectos organizativos. Ademais da norma ISO 12.207, entre os estándares de referencia hai que destacar as normas ISO/IEC TR 15.504/SPICE *"Software Process Improvement and Assurance Standards Capability Determination"*, UNE-EN-ISO 9001:2000 *Sistemas de Xestión da Calidade. Requisitos*, UNE-EN-ISO 9000:2000 *Sistemas de Xestión da Calidade. Fundamentos e Vocabulario* e o estándar IEEE 610.12-1.990 *"Standard Glossary of Software Engineering Terminology"*. Igualmente se han ter en conta outras metodoloxías como SSADM, Merise, Information Engineering, *MAGERIT. Metodoloxía de Análise e Xestión de Riscos dos Sistemas de Información* promovida polo Consello Superior de Informática e *EUROMÉTODO*.

Diferenciouse entre a aplicación de Técnicas, como conxunto de heurísticas e procedementos apoiados en estándares que utilizan notacións específicas en termos de sintaxe e semántica, e de Prácticas cuxa



utilización non implica regras preestablecidas coa mesma rixidez. As novas técnicas están amplamente soportadas por ferramentas comerciais.

### *31.7.1 Procesos Principais de Métrica Versión 3*

MÉTRICA Versión 3 ten un enfoque orientado ao proceso e por iso, como xa se dixo, enmarcouse dentro da norma ISO 12.207, que se centra na clasificación e definición dos procesos do ciclo de vida do software. MÉTRICA Versión 3 cobre o Proceso de Desenvolvemento e o Proceso de Mantemento de Sistemas de Información.

MÉTRICA Versión 3 foi concibida para abarcar o desenvolvemento completo de Sistemas de Información sexa cal for a súa complexidade e magnitude, polo cal a súa estrutura responde a desenvolvementos máximos e deberá adaptarse e dimensionarse en cada momento de acordo ás características particulares de cada proxecto. A metodoloxía descompón cada un dos procesos en actividades, e estas á súa vez en tarefas. Para cada tarefa descríbese o seu contido facendo referencia ás súas principais accións, produtos, técnicas, prácticas e participantes. A orde asignada ás actividades non debe interpretarse como secuencia na súa realización, xa que estas poden realizar en orde diferente á súa numeración ou ben en paralelo. Con todo, non se dará por acabado un proceso ata non finalizar todas as actividades do mesmo determinadas ao comezo do proxecto.

Así os procesos da estrutura principal de MÉTRICA Versión 3 son os seguintes:

- PLANIFICACIÓN DE SISTEMAS DE INFORMACIÓN.
- DESENVOLVEMENTO DE SISTEMAS DE INFORMACIÓN.
- MANTEMENTO DE SISTEMAS DE INFORMACIÓN.

### *31.7.2 Planificación de sistemas de información*



O obxectivo dun Plan de Sistemas de Información é proporcionar un marco estratéxico de referencia para os Sistemas de Información dun determinado ámbito da Organización. O resultado do Plan de Sistemas debe, xa que logo, orientar as actuacións en materia de desenvolvemento de Sistemas de Información co obxectivo básico de apoiar a estratexia corporativa, elaborando unha arquitectura de información e un plan de proxectos informáticos para dar apoio aos obxectivos estratéxicos. Por este motivo é necesario un proceso como o de Planificación de Sistemas de Información, no que participen, por unha banda os responsables dos procesos da organización cunha visión estratéxica e por outro, os profesionais de SI capaces de enriquecer dita visión coa acerca de vantaxes competitivas por medio dos sistemas e tecnoloxías da información e comunicacións.

Como produtos finais deste proceso obtéñense os seguintes:

- Catálogo de requisitos de PSI que xorde do estudo da situación actual no caso de que sexa significativo devandito estudo, do diagnóstico que se levou a cabo e das necesidades de información dos procesos da organización afectados polo plan de sistemas.
- Arquitectura de información que se compón dos seguintes produtos: modelo de información, modelo de sistemas de información, arquitectura tecnolóxica, plan de proxectos e plan de mantemento do PSI.

Este novo enfoque de aliñamento dos sistemas de información coa estratexia da organización, a implicación directa da alta dirección e a proposta de solución presenta como vantaxes:

- A implicación da alta dirección facilita que se poida desenvolver cos recursos necesarios e o calendario establecido.



- A perspectiva horizontal dos procesos dentro da Organización facilita que se atenda a intereses globais e non particulares de unidades organizativas que poidan desvirtuar os obxectivos do Plan.
- A prioridade do desenvolvemento dos sistemas de información da organización por obxectivos estratéxicos.
- A proposta de Arquitectura de Información que se fai no plan é máis estratéxica que tecnolóxica.

### *31.7.3 Desenvolvemento de Sistemas de Información*

O proceso de Desenvolvemento de MÉTRICA Versión 3 contén todas as actividades e tarefas que se deben levar a cabo para desenvolver un sistema, cubrindo desde a análise de requisitos ata a instalación do software. Ademais das tarefas relativas á análise, inclúe dous partes no deseño de sistemas: arquitectónico e detallado. Tamén cobre as probas unitarias e de integración do sistema. Este proceso é, sen dúbida, o máis importante dos identificados no ciclo de vida dun sistema e relaciónase con todos os demais.

En MÉTRICA Versión 3 abordáronse os dous tipos de desenvolvemento: estruturado e orientado a obxecto, polo que foi necesario establecer actividades a realizar en función do tipo de desenvolvemento elixido. Definíronse 5 subprocesos para este apartado:

- Estudo de Viabilidade do Sistema (EVS)
- Análise Do Sistema De Información (ASI).
- Deseño Do Sistema De Información (DSI).
- Construción Do Sistema De Información (CSI).
- Implantación E Aceptación Do Sistema (IAS).



#### *31.7.3.1 Estudo de Viabilidade do Sistema (EVS)*

O propósito deste proceso é analizar un conxunto concreto de necesidades, coa idea de propoñer unha solución a curto prazo. Os criterios cos que se fai esta proposta non serán estratéxicos senón tácticos e relacionados con aspectos económicos, técnicos, legais e operativos. Os resultados do Estudo de Viabilidade do Sistema constituirán a base para tomar a decisión de seguir adiante ou abandonar. Se se decide seguir adiante poden xurdir un ou varios proxectos que afecten a un ou varios sistemas de información. Consideraranse alternativas de solución baseadas en solucións "a medida", solucións baseadas na adquisición de produtos software do mercado ou solucións mixtas. Para valorar as alternativas formuladas e determinar unha única solución, estudarase o impacto na organización de cada unha delas, o investimento e os riscos asociados. O resultado final deste proceso son os produtos relacionados coa solución que se propón para cubrir a necesidade concreta que se presentou no proceso, e que depende de se a solución implica desenvolvemento a medida ou non.

#### *31.7.3.2 Análise do Sistema de Información (ASI)*

O propósito deste proceso é conseguir a especificación detallada do sistema de información a través dun catálogo de requisitos e unha serie de modelos que cubran as necesidades de información dos usuarios para os que se desenvolverá o sistema de información e que serán a entrada para o proceso de Deseño do Sistema de Información.

En primeiro lugar descríbese inicialmente o sistema de información, a partir dos produtos xerados no proceso Estudo de Viabilidade do Sistema (EVS). Se delimita o seu alcance, xérase un catálogo de requisitos xerais e descríbese o sistema mediante uns modelos iniciais de alto nivel.



Recóllense de forma detallada os requisitos funcionais que o sistema de información debe cubrir, catalogándoos, o que permite facer a traza ao longo dos procesos de desenvolvemento. Ademais, identifícanse os requisitos non funcionais do sistema, é dicir, as facilidades que debe proporcionar o sistema e as restricións a que estará sometido, en canto a rendemento, frecuencia de tratamento, seguridade, etc. Para facilitar a análise do sistema identifícanse os subsistemas de análises, e elabóranse os modelos de Casos de Uso e de Clases, en desenvolvementos orientados a obxectos, e de Datos e Procesos en desenvolvementos estruturados. Especificaranse todas as interfaces entre o sistema e o usuario, como formatos de pantallas, diálogos, formatos de informes e formularios de entrada. Finalizados os modelos, realízase unha análise de consistencia. Unha vez realizada a devandita análise de consistencia elabórase o produto **Especificación de Requisitos Software**, que constitúe un punto de referencia no desenvolvemento do software e a liña base de referencia para as peticións de cambio sobre os requisitos inicialmente especificados. Neste proceso iníciase tamén a especificación do Plan de Probas, que se completará no proceso Deseño do Sistema de Información (DSI).

Neste proceso é moi importante a participación dos usuarios, a través de técnicas interactivas, como deseño de diálogos e prototipos, que permiten ao usuario familiarizarse co novo sistema e colaborar na construción e perfeccionamento do mesmo.

### *31.7.3.3 Deseño do Sistema de Información (DSI)*

O propósito do Deseño do Sistema de Información (DSI) é obter a definición da arquitectura do sistema e da contorna tecnolóxica que lle vai a dar soporte, xunto coa especificación detallada dos compoñentes do sistema de información. A partir de dita información, xéranse todas as especificacións de construción relativas ao propio sistema, así como a



especificación técnica do plan de probas, a definición dos requisitos de implantación e o deseño dos procedementos de migración e carga inicial, estes últimos cando cumpra.

Este proceso consta dun primeiro bloque de actividades, que se realizan en paralelo, e cuxo obxectivo é obter o deseño de detalle do sistema de información que comprende a partición física do sistema de información, independente dun contorno tecnolóxico concreto, a organización en subsistemas de deseño, a especificación do contorno tecnolóxico sobre o que se despregan os devanditos subsistemas e a definición dos requisitos de operación, administración do sistema, seguridade e control de acceso. No caso de deseño orientado a obxectos, convén sinalar que se contemplou que o deseño da persistencia se leva a cabo sobre bases de datos relacionais.

Un segundo bloque de actividades complementa o deseño do sistema de información, no que se xeran todas as especificacións necesarias para a construción do sistema de información.

#### *31.7.3.4 Construción do Sistema de Información (CSI)*

A construción do Sistema de Información (CSI) ten como obxectivo final a construción e proba dos distintos compoñentes do sistema de información, a partir do conxunto de especificacións lóxicas e físicas do mesmo, obtido no Proceso de Deseño do Sistema de Información (DSI). Desenvólvense os procedementos de operación e seguridade e elabóranse os manuais de usuario final e de explotación, estes últimos cando cumpra.

Para conseguir o devandito obxectivo, recóllese a información relativa ao produto do deseño Especificacións de construción do sistema de información, prepárase o contorno de construción, xérase o código de cada un dos compoñentes do sistema de información e vanse realizando, a



medida que se vaia finalizando a construción, as probas unitarias de cada un deles e as de integración entre subsistemas.

Se fose necesario realizar unha migración de datos, é neste proceso onde se leva a cabo a construción dos compoñentes de migración e procedementos de migración e carga inicial de datos.

#### *31.7.3.5 Implantación e Aceptación do Sistema (IAS)*

Este proceso ten como obxectivo principal, a entrega e aceptación do sistema na súa totalidade, que pode comprender varios sistemas de información desenvolvidos de xeito independente, segundo se estableceu no proceso de Estudo de Viabilidade do Sistema (EVS), e un segundo obxectivo que é levar a cabo as actividades oportunas para o paso a produción do sistema.

Establécese o plan de implantación, unha vez revisada a estratexia de implantación e detállase o equipo que o realizará. Para o inicio deste proceso tómanse como punto de partida os compoñentes do sistema probados de forma unitaria e integrados no proceso Construción do Sistema de Información (CSI), así como a documentación asociada. O Sistema someterase ás Probas de Implantación coa participación do usuario de operación cuxa responsabilidade, entre outros aspectos, é comprobar o comportamento do sistema baixo as condicións máis extremas. Tamén se someterá ás Probas de Aceptación cuxa execución é responsabilidade do usuario final. Neste proceso elabórase o plan de mantemento do sistema de forma que o responsable do mantemento coñeza o sistema antes de que este pase a produción. Tamén se establece o acordo de nivel de servizo requirido unha vez que se inicie a produción.



#### *31.7.4 Mantemento de Sistemas de Información (MSI)*

O obxectivo deste proceso é a obtención dunha nova versión dun sistema de información desenvolvido con MÉTRICA, a partir das peticións de mantemento que os usuarios realizan con motivo dun problema detectado no sistema ou pola necesidade dunha mellora do mesmo. Só se considerarán en MÉTRICA Versión 3 os tipos de Mantemento **Correctivo** e **Evolutivo**. Ante unha petición de cambio dun sistema de información xa en produción, realízase un rexistro das peticións, se diagnostica o tipo de mantemento e decídese se se lle dá resposta ou non, en función do plan de mantemento asociado ao sistema afectado pola petición, e establécese con que prioridade. A definición da solución ao problema ou necesidade formulada polo usuario que realiza o responsable de mantemento, inclúe un estudo do impacto, a valoración do esforzo e custo, as actividades e tarefas do proceso de desenvolvemento a realizar e o plan de probas de regresión.

#### *31.7.5 Interfaces De Métrica Versión 3*

A estrutura de MÉTRICA Versión 3 inclúe tamén un conxunto de interfaces que definen unha serie de actividades de tipo organizativo ou de soporte ao proceso de desenvolvemento e aos produtos, que no caso de existir na organización deberanse aplicar para enriquecer ou influír na execución das actividades dos procesos principais da metodoloxía e que se non existen haberá que realizar para complementar e garantir o éxito do proxecto desenvolvido con MÉTRICA Versión 3. Son catro:

- **Xestión de Proxectos:** Ten como finalidade principal a planificación, o seguimento e control das actividades e dos recursos humanos e materiais que interveñen no desenvolvemento dun Sistema de Información. Como consecuencia deste control é posible coñecer en



todo momento qué problemas se producen e resolvelos ou palialos o máis pronto posible, o cal evitará desviacións temporais e económicas.

As actividades da Interface de Xestión de Proxectos son de tres tipos:

- o *Actividades de Inicio do Proxecto*, que permiten estimar o esforzo e establecer a planificación do proxecto.
  - o *Actividades de Seguimento e Control*, supervisando a realización das tarefas por parte do equipo de proxecto e xestionando as incidencias e cambios nos requisitos que poidan presentarse e afectar á planificación do proxecto.
  - o *Actividades de Finalización do Proxecto*, pechadura e rexistro da documentación de xestión.
- **Seguridade:** A interface de Seguridade fai posible incorporar durante a fase de desenvolvemento as funcións e mecanismos que reforzan a seguridade do novo sistema e do propio proceso de desenvolvemento, asegurando a súa consistencia e seguridade, completando o plan de seguridade vixente na organización ou desenvolvéndoo desde o principio, utilizando MAGERIT como metodoloxía de análise e xestión de riscos no caso de que a organización non dispoña da súa propia metodoloxía. Contempla dous tipos de actividades diferenciadas: as relacionadas coa seguridade intrínseca do sistema de información, e as que velan pola seguridade do propio proceso de desenvolvemento do sistema de información. Ademais faise especial fincapé na formación en materia de seguridade. Ao ser finitos os recursos, non poden asegurarse todos os aspectos do desenvolvemento dos sistemas de información, polo que haberá que aceptar un determinado nivel de risco concentrándose nos aspectos máis comprometidos ou ameazados.
- **Xestión da Configuración:** A interface de xestión da configuración consiste na aplicación de procedementos administrativos e técnicos durante o desenvolvemento do sistema de información e o seu posterior mantemento. A súa finalidade é identificar, definir, proporcionar información e controlar os cambios na configuración do sistema, así



como as modificacións e versións dos mesmos. Este proceso permitirá coñecer o estado de cada un dos produtos que se definiron como elementos de configuración, garantindo que non se realizan cambios incontrolados e que todos os participantes no desenvolvemento do sistema dispoñen da versión adecuada dos produtos que manexan. A xestión de configuración facilita ademais o mantemento do sistema, achegando información precisa para valorar o impacto dos cambios solicitados e reducindo o tempo de implementación dun cambio, tanto evolutivo como correctivo.

- **Aseguramento da Calidade:** O obxectivo da interface de Aseguramento da Calidade é proporcionar un marco común de referencia para a definición e posta en marcha de plans específicos de aseguramento de calidade aplicables a proxectos concretos. As actividades están orientadas a verificar a calidade dos produtos. Son actividades que avalían a calidade e que son realizadas por un grupo de Asesoramento da Calidade independente dos responsables da obtención dos produtos. As actividades contempladas permitirán reducir, eliminar e prever as deficiencias de calidade dos produtos a obter, así como alcanzar unha razoable confianza en que as prestacións e servizos esperados polo cliente ou o usuario queden satisfeitas.

### **31.5 Metodoloxías Áxiles.**

Nas **metodoloxías áxiles**, a creación de valor mediante a adaptación ás necesidades cambiantes aparece nun primeiro plano fronte á tradicional idea de deseñar un plan e cumprir uns calendarios/requisitos estáticos. Os proxectos xestionados con metodoloxías áxiles iníciase sen un detalle pechado do que vai ser construído. A nivel comercial, os proxectos poden ser vendidos como servizos e non como produtos. As características básicas dos proxectos xestionados con metodoloxías áxiles son as seguintes:



- **Incerteza:** a dirección indica a necesidade estratéxica que se desexa cubrir (sen entrar en detalles), ofrecendo máxima liberdade ao equipo de traballo.
- **Equipos auto-organizados:** non existen roles especializados
  - o Autonomía: liberdade para a toma de decisións.
  - o Auto-superación: de forma periódica avalíase o produto que se esta desenvolvendo.
  - o Auto-enriquecemento: transferencia do coñecemento.
- **Fases de desenvolvemento solapadas:** As fases non existen como tal senón que se desenvolven tarefas/actividades en función das necesidades cambiantes durante todo o proxecto. De feito, en moitas ocasións non é posible realizar un deseño técnico detallado antes de empezar a desenvolver e ver algúns resultados. Por outra banda, as fases tradicionais efectuadas por persoas diferentes non favorece o traballo en equipo e poden chegar a xerar máis inconvenientes que vantaxes (por ex. un atraso nunha fase, afecta a todo o proxecto).
- **Control sutil:** establecementos de puntos de control para realizar un seguimento adecuado sen limitar a liberdade e creatividade do equipo. Así mesmo, recoméndase:
  - o Avaliar o ambiente laboral, sendo fundamental a elección de persoas que non xeren conflitos.
  - o Recoñecer os méritos mediante un sistema de avaliación xusto e entender os erros como puntos de mellora e aprendizaxe.
  - o Potenciar a interacción entre o equipo e o negocio, para que poidan coñecer as necesidades de primeira man.
- **Difusión e transferencia do coñecemento:** alta rotación dos membros dos equipos entre diferentes proxectos. Por outra banda, potenciar o acceso libre á información e documentación.

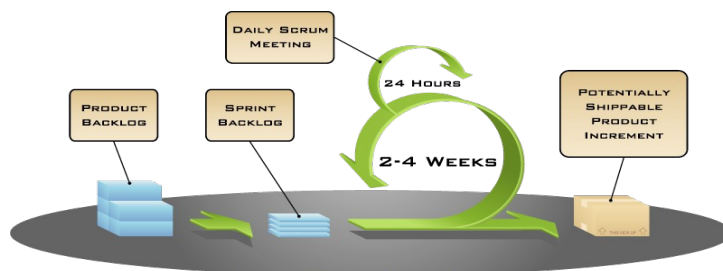


Algunhas das metodoloxías áxiles máis coñecidas verémolas nos apartados seguintes.

### *31.5.1 Scrum*

É un proceso de desenvolvemento de software iterativo e incremental utilizado comunmente en contornos baseados na metodoloxía Agile de desenvolvemento de software. É un proceso marco que inclúe un conxunto de prácticas e roles predefinidos. Os roles principais en Scrum son o ScrumMaster, que mantén os procesos e traballa de forma similar ao director de proxecto, o ProductOwner, que representa aos stakeholders (clientes externos ou internos), e o Team que inclúe aos desenvolvedores. Durante cada sprint, un período entre 15 e 30 días (a lonxitude é definida polo equipo), o equipo crea un incremento de software potencialmente entregable (utilizable). O conxunto de características que forma parte de cada sprint vén do **product backlog**, que é un conxunto de requisitos de alto nivel priorizados que dan forma ao traballo a realizar. Os elementos do backlog que forman parte do sprint determínanse durante a reunión de sprint **planning**. Durante esta reunión, o Product Owner informa ao equipo dos elementos no product backlog que quere ver completados. O equipo, nese momento, determina a cantidade dese traballo que pode comprometerse a completar durante o seguinte sprint. Durante o sprint, ninguén pode cambiar o sprint backlog, o que significa que os requisitos están conxelados durante o sprint. Existen varias implementacións de sistemas para xestionar o proceso de Scrum, que van desde notas amarelas "post-it" e encerados ata paquetes de software. Unha das maiores vantaxes de Scrum é que é moi fácil de aprender, e require moi pouco esforzo para comezarse a utilizar.





### 31.5.2 Dynamic Systems Development Method (DSDM)

Provê un framework para o [desenvolvemento áxil de software](#), apoiado pola continua implicación do usuario nun [desenvolvemento iterativo e crecente](#). DSDM foi desenvolvido no [Reino Unido](#) nos [anos 90](#). Como extensión do [Desenvolvemento rápido de aplicacións](#) (RAD), DSDM céntrase nos proxectos de sistemas de información que son caracterizados por orzamentos e axendas apertadas. DSDM trata os problemas que ocorren con frecuencia no desenvolvemento dos sistemas de información no que respecta a pasar sobre tempo e orzamento e outras razóns comúns para a falta no proxecto tal como falta de implicación do usuario e da comisión superior da xerencia. DSDM consiste en 3 fases: fase do pre-proxecto, fase do ciclo de vida do proxecto, e fase do post-proxecto. A fase do ciclo de vida do proxecto subdivídese en 5 etapas: estudo de viabilidade, estudo da empresa, iteración do modelo funcional, deseño e iteración da estrutura, e implementación. Ten 9 principios fundamentais:

- **Involucrar o cliente** é a clave para levar un proxecto eficiente e efectivo, onde ambos, cliente e desenvolvedores, comparten un contorno de traballo para que as decisións poidan ser tomadas con precisión.
- **O equipo do proxecto debe ter o poder** para tomar decisións que son importantes para o progreso do proxecto, sen esperar aprobación de niveis superiores.
- DSDM céntrase na **entrega frecuente de produtos**, asumindo que entregar algo cedo é sempre mellor que entregar todo ao final. Ao entregar o produto frecuentemente desde unha etapa temperá do



proxecto, o produto pode ser verificado e revisado alí onde a documentación de rexistro e revisión pode ser tida en conta na seguinte fase ou iteración.

- O principal **criterio de aceptación** de entregables reside en entregar un sistema que satisfai as actuais necesidades de negocio.
- O desenvolvemento é **iterativo e incremental**, guiado pola realimentación dos usuarios para converxer nunha solución de negocio precisa.
- Todos os **cambios** durante o desenvolvemento son reversibles.
- Requisitos globais antes de comezar o proxecto.
- As probas son realizadas durante todo o ciclo vital do proxecto.
- A comunicación e cooperación entre todas as partes interesadas.

### *31.5.3 Extreme Programming (XP)*

A programación extrema diferénciase das metodoloxías tradicionais principalmente en que pon máis énfase na adaptabilidade que na previsibilidade. Os defensores de XP consideran que os cambios de requisitos sobre a marcha son un aspecto natural, inevitable e ata desexable do desenvolvemento de proxectos. Creen que ser capaz de adaptarse aos cambios de requisitos en calquera punto da vida do proxecto é unha aproximación mellor e máis realista que intentar definir todos os requisitos ao comezo do proxecto e investir esforzos despois en controlar os cambios nos requisitos. XP constrúe un proceso de deseño evolutivo que se basea en refactorizar un sistema simple en cada iteración. Todo o deseño céntrase na iteración actual e non se fai nada anticipadamente para necesidades futuras. Os Valores orixinais da programación extrema son: **simplicidade** (de deseño, código e documentación), **comunicación** (a comunicación co cliente é fluída xa que o cliente forma parte do equipo de desenvolvemento. O cliente decide qué características teñen prioridade e sempre debe estar dispoñible para solucionar dúbidas),



**retroalimentación** (*feedback*. Ao estar o cliente integrado no proxecto, a súa opinión sobre o estado do proxecto coñécese en tempo real.), e **coraxe** (Requírese coraxe para implementar as características que o cliente quere agora sen caer na tentación de optar por un enfoque máis flexible que permita futuras modificacións). Un quinto valor, **respecto** (os membros do equipo respectan o traballo do resto non facendo menos a outros, senón orientándoos a realizalo mellor, obtendo como resultado unha mellor autoestima no equipo e elevando o ritmo de produción no equipo), foi engadido posteriormente.

As características fundamentais desta metodoloxía son:

- Desenvolvemento iterativo e incremental
- Probas unitarias continuas
- Programación en parellas
- Integración do equipo de programación co cliente
- Corrección de todos os erros
- Refactorización do código
- Propiedade do código compartida
- Simplicidade no código

#### 31.5.4 Agile Modeling (AM)

Pódese describir como unha metodoloxía baseada na práctica para o modelado efectivo de sistemas de software. Non define procedementos detallados de como crear un tipo de modelo dado. En lugar diso, suxire prácticas para que os modelos e documentación sexan efectivos. O seu segredo non está nas técnicas de modelado a usar, senón en como se aplican. Non é un desenvolvemento de software completo xa que non cobre actividades de programación, proba, xestión de proxectos, implementación, soporte ou outros elementos da realización de proxectos que non sexan a documentación e o modelado. É necesario, polo tanto,



combinalo con outras metodoloxías como poden ser XP, DSDM, SCRUM ou RUP. Os valores desta metodoloxía son a **comunicación** (entre participantes do equipo de traballo, desenvolvedores e analistas, etc.), **simplicidade, coraxe** (para tomar decisións importantes e ser capaces de cambiar de dirección cando o camiño tomado non é o correcto) e **humildade** (todos os interesados no proxecto poden contribuír en algo para a mellor realización).

#### 31.5.5 *Feature Driven Development (FDD)*:

Baséase nun proceso iterativo con iteracións curtas que producen un software funcional que o cliente e a dirección da empresa poden ver e monitorar. As iteracións decídense en base a funcionalidades, que son pequenas partes do software con significado para o cliente. Non cobre todo o ciclo de vida, senón só as fases de deseño e construción. Non require un modelo específico de proceso e se complementa con outras metodoloxías. FDD consiste en cinco procesos secuenciais durante os cales se diseña e constrúe o sistema:

- **Desenvolvemento dun modelo xeral:** Cando comeza esta fase, os expertos do dominio xa teñen unha idea do contexto e os requisitos do sistema. O dominio global é dividido en diferentes áreas e realízase informe detallado para cada unha delas por parte dos expertos do dominio.
- **Construción da lista de funcionalidades** Os ensaios, modelos de obxectos e documentación de requisitos proporcionan a base para construír unha ampla lista de funcionalidades. Estas funcionalidades son pequenos ítems útiles aos ollos do cliente. A lista de funcionalidades é revisada polos usuarios e patrocinadores para asegurar a súa validez. As funcionalidades que requiran de máis de dez días descompóñense noutras máis pequenas.



- **Formulación por funcionalidades:** Nesta etapa inclúese a creación dun plan de alto nivel, no cal a lista de funcionalidades é ordenada baseándose na prioridade e a dependencia entre cada funcionalidade. Ademais, as clases identificadas na primeira etapa son asignadas a cada programador.
- **Deseño e construción por funcionalidades:** O deseño e construción da funcionalidade é un proceso iterativo durante o cal as funcionalidades seleccionadas son producidas. Unha iteración pode levar desde uns poucos días a un máximo de dúas semanas. Este proceso iterativo inclúe tarefas como inspección do deseño, codificación, probas unitarias, integración e inspección do código.

#### *31.5.6 Familia Cristal*

Alistair Cockburn é o propulsor detrás da serie de metodoloxías Crystal. É unha familia porque el cre que os tipos diferentes de proxectos requiren tipos diferentes de metodoloxías. El mira esta variación ao longo de dous eixes: o número de persoas no proxecto, e as consecuencias dos erros. Dispón un código de cor para marcar a complexidade de cada metodoloxía. Comparte coa XP unha orientación humana, pero esta centralización na xente faise dun xeito diferente. Alistair considera que as persoas atopan difícil seguir un proceso disciplinado, así que máis que seguir a alta disciplina da XP, Alistair explora a metodoloxía menos disciplinada que aínda podería ter éxito, intercambiando conscientemente produtividade por facilidade de execución. El considera que aínda que Cristal é menos produtivo que a XP, máis persoas serán capaces de seguilo. Alistair tamén pon moito peso nas revisións ao final da iteración, animando ao proceso a ser “automellorable”. Defende que o desenvolvemento iterativo está para atopar os problemas cedo, e entón permitir corrixilos. Isto pon máis énfase na xente supervisando o seu proceso e afinándoo conforme o desenvolven.



## **Bibliografía**

- Ingeniería del Software. Un enfoque práctico. ROGER S. PRESSMAN. Ed. McGraw Hill
- Ingeniería de Software 7ª Edición - Ian Sommerville
- Metodologías para la gestión y desarrollo de software - ¿?
- <http://msdn.microsoft.com/es-es/library/bb972268.aspx> Desenvolvemento baseado en Compoñentes.
- <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 Enxeñería de Software. Luis Calabria. Universidad ORT Uruguay
- <http://www.programacionextrema.org/articulos/newMethodology.es.html> La Nueva Metodología. [Martin Fowler](#)

Autor: Hernán Vila Pérez  
Xefe do Servizo de Informática. Instituto Galego de Vivenda e Solo  
Vicepresidente do CPEIG





**32. ANÁLISE ESTRUTURADA.  
ANÁLISE ORIENTADA A  
OBXECTOS. LINGUAXE  
UNIFICADA DE MODELAXE  
(UML). DESEÑO. DESEÑO  
ESTRUTURADO. DESEÑO DE  
DATOS. DESEÑO DA  
INTERFACE DE USUARIO.  
DESEÑO PROCEDEMENTAL.  
DESEÑO ORIENTADO A  
OBXECTOS.**



**Tema 32: Análise estruturada. Análise orientada a obxectos. Linguaxe unificada de modelado (UML). Deseño. Deseño estruturado. Deseño de datos. Deseño da interface de usuario. Deseño procedemental. Deseño orientado a obxectos.**

### 32.1 Análise do sistema de información.

#### 32.1.1 Análise estruturada.

#### 32.1.2 Análise orientada a obxectos.

#### 32.1.3 Linguaxe unificada de modelado.

### 32.2 Deseño dun sistema de información.

#### 32.2.1 Deseño estruturado.

#### 32.2.2 Deseño de datos.

#### 32.2.3 Deseño da interface de usuario.

#### 32.2.4 Deseño procedemental.

#### 32.2.5 Deseño orientado a obxectos.

### **32.1 Análise do sistema de información.**

A enxeñería do software empeza cunha serie de tarefas de modelado que levan a unha especificación completa dos requisitos e a unha representación do deseño xeral do software a construír. O *modelo de análise*, realmente un conxunto de modelos, é a primeira representación técnica dun sistema. Cos anos propuxéronse moitos métodos para o modelado da análise, pero o panorama actual está dominado por dúas tendencias: a **análise estruturada**, método de modelado clásico, cada vez máis en desuso, e a **análise orientada a obxectos**, que é a tendencia que domina hoxe en día.

#### **32.1.1 Análise estruturada.**

O termo análise estruturada orixinalmente acuñado por Douglas Ross, foi popularizado por Tom DeMarco no seu libro “Análise estruturada”, que foi o



primeiro en establecer os principios polos cales debe guiarse un proceso de análise de software. Baséase principalmente na análise dos fluxos de información dun sistema, permitíndolle ao analista coñecer un sistema ou proceso dunha forma lóxica e manexable, ao mesmo tempo que proporciona a base para asegurar que non se omite ningún detalle pertinente.

Segundo Pressman, o modelo de análise debe lograr tres obxectivos primarios:

1. Describir o que require o cliente.
2. Establecer unha base para a creación dun deseño de software
3. Definir un conxunto de requisitos que se poida validar unha vez que se constrúe o software.

Para lograr estes obxectivos, o modelo de análise extraída durante a análise estruturada toma a forma ilustrada na seguinte figura:



No centro do modelo atópase o **diccionario de datos** (un almacén que contén definicións de todos os obxectos de datos consumidos e producidos polo software). Tres diagramas diferentes rodean o núcleo.

- O **diagrama de entidade-relación** (DER) representa as relacións entre os obxectos de datos. O DER é a notación que se usa para





realizar a actividade de modelado de datos. Os atributos de cada obxecto de datos sinalados no DER pódese describir mediante unha descrición de obxectos de datos.

- O diagrama **de fluxo de datos** (DFD) serve para dous propósitos:
  - a) Proporcionar unha indicación de como se transforman os datos a medida que se avanza no sistema
  - b) Representar as funcións (e subfuncións) que transforman o fluxo de datos.

O DFD proporciona información adicional que se usa durante a análise do dominio de información e serve como base para o modelado de función. Nunha especificación de proceso (EP) atópase unha descrición de cada función presentada no DFD.

- O diagrama **de transición de estados** (DTE) indica como se comporta o sistema como consecuencia de sucesos externos. Para lograr isto, o DTE representa os diferentes modos de comportamento (chamados estados) do sistema e o xeito en que se fan as transicións de estado a estado. O DTE serve como a base do modelado de comportamento. Dentro da especificación de control (EC) atópase máis información sobre os aspectos de control do software.

Estes tres diagramas permiten modelizar completamente o sistema desde tres puntos de vista:

- **Punto de Vista Funcional.** Descríbese como se transforman os datos dentro do sistema. Para describir este modelo funcional créase un modelo de procesos, o cal se constrúe usando a técnica dos Diagramas de Fluxo de Datos.
- **Punto de Vista Estático.** Describe os obxectos de información do sistema e como se relacionan estes obxectos entre eles. Para crear este modelo estático constrúese o modelo de datos, o cal se pode construír usando a técnica do diagrama Entidade - Relación.
- **Punto de Vista Dinámico.** Describe os aspectos do sistema que cambian ao longo do tempo. Descríbense os estados polos que vai



pasando o sistema e as condicións e eventos que fan que o sistema pase dun estado a outro. Unha posible técnica para construír o modelo dinámico é a do Diagrama de Transición de Estados.

A metodoloxía de Análise estruturada **Moderno** é unha proposta de Edward Yourdon que establece un novo modelo para a análise estruturada e incorpora as ideas, criterios e ferramentas que o precederon. Baséase na construción do **modelo esencial** do sistema, que indica o que o sistema debe facer para satisfacer os requisitos do usuario e debe mencionar o mínimo posible como o sistema o levará a cabo. Está composto por dous compoñentes principais:

- **O Modelo Ambiental:** define a fronteira entre o sistema e o resto do mundo (é dicir, o ambiente onde o sistema reside). O modelo ambiental está composto dun Diagrama de Contexto, unha Lista de Eventos e unha descrición pequena do propósito do sistema: a Declaración de Obxectivos.
- **O Modelo de Comportamento:** describe a conduta, do interior do sistema, necesaria para interactuar exitosamente co ambiente. Esta composto de Diagramas de Fluxo de Datos, Lista de Eventos, Diagramas de Entidades e Relacións, Diagramas de Transición de Estados, Dicionario de Datos e Especificación de Procesos.

### **31.1.2 Análise orientada a obxectos.**

Os **obxectos** son entidades que teñen atributos (datos) e formas de comportamento (procedementos ou métodos) particulares. As **clases** describen un conxunto de obxectos diferentes con propiedades (atributos) similares e un comportamento común. As clases son un concepto estático definido no programa fonte, son unha abstracción da esencia dun obxecto, mentres que os obxectos son entes dinámicos que existen en tempo e espazo e que ocupan memoria na execución dun programa. Os obxectos interaccionan entre si mediante **mensaxes**, que non son máis que unha



solicitude que fai un obxecto a outro pedíndolle que se comporte dalgunha forma determinada.

Os principios do modelo orientado a obxectos son:

- **Identidade:** Cada obxecto ten a súa propia identidade inherente, é dicir, dous obxectos son distintos aínda que teñan todas as súas propiedades iguais.
- **Clasificación.** Refírese a que os obxectos que teñen a mesma estrutura de datos (atributos), e o mesmo comportamento (operacións), están agrupados nunha clase.
- **Herdanza.** A herdanza é o mecanismo mediante o cal unha clase (subclase) adquire as propiedades doutra clase xerarquicamente superior (superclase, clase base). A herdanza proporciona o mecanismo para compartir automaticamente métodos e datos entre clases e subclases.
- **Abstracción:** Trátase de abstraer os datos e métodos comúns a un conxunto de obxectos para almacenalos nunha clase.
- **Encapsulación:** É o termo que se utiliza para expresar que os datos dun obxecto só poden ser manipulados mediante as mensaxes e métodos predefinidos.
- **Polimorfismo:** É a propiedade pola cal un mesmo mensaxe pode orixinar condutas diferentes ao ser recibido por obxectos diferentes. Tamén falamos de polimorfismo cando temos distintos métodos que teñen un comportamento distinto en función do número ou tipo de parámetros que reciben.
- **Reusabilidade:** É a capacidade de producir compoñentes reutilizables para outros deseños ou aplicacións.
- **Persistencia:** A persistencia é a calidade que se refire á permanencia do obxecto, é dicir, ao tempo durante o cal sae, se lle asigna espazo e permanece accesible na memoria do ordenador (principal ou secundaria).



- **Extensibilidade:** É a capacidade dun programa para ser facilmente alterado de forma que poida tratar con novas clases de entrada.

A Análise Orientada a Obxectos é *un método de análise que examina os requisitos desde a perspectiva das clases e obxectos atopados no vocabulario do dominio do problema* (Booch). Debe ser comprendido polo cliente e servir de axuda para atopar os verdadeiros requisitos do sistema. Firesmith describe a **análise do dominio** como “*a identificación, análise e especificación de requisitos comúns nun dominio de aplicación específico, normalmente para a súa reutilización en múltiples proxectos dentro do mesmo dominio de aplicación. A análise orientada a obxectos do dominio é a identificación, análise e especificación de capacidades comúns e reutilizables dentro dun dominio de aplicación específico, en termos de obxectos, clases, submontaxes e marcos de traballo comúns*”. Do mesmo xeito que nos métodos estruturados tradicionais, na etapa Análise hai que establecer que é o que debe facerse, deixando para etapas posteriores os detalles. O resultado da análise debe ser unha completa comprensión do problema. As dúas grandes etapas de que consta a Análise son as seguintes:

1. A descrición ou especificación do problema. Esta descrición non debe considerarse inmutable, senón máis ben como a base para ir refinando as especificacións reais. A especificación do problema debe comprender o establecer o ámbito do problema, describir as necesidades e requisitos, o contexto da aplicación, os supostos de que se parte ou as necesidades de rendemento do sistema. Nestas especificacións, o usuario do sistema debe indicar cales son obrigadas e cales se poden considerar opcionais. Así mesmo, outros puntos a tratar poden ser os estándares de Enxeñería do Software, deseño das probas a efectuar, previsión de futuras extensións, etc.
2. A modelización da Análise: As características esenciais deben abstraerse nun modelo. As especificacións expresadas en linguaxe natural tenden a ser ambiguas, incompletas e inconsistentes; con



todo, o Modelo de Análise é unha representación precisa e concisa do problema, que permite construír unha solución. A etapa seguinte de deseño remitirase a este modelo, en lugar de ás vagas especificacións iniciais. O Modelo de Análise constrúese identificando as clases e obxectos do dominio do problema (estrutura estática), as interaccións entre os obxectos e o seu secuenciamento (estrutura dinámica), e as accións a realizar polo sistema que producen un resultado observable e valioso para os usuarios (estrutura funcional).

### **32.1.3 Linguaxe unificada de modelado (UML).**

UML (Unified Modeling Language) é unha linguaxe que permite modelar, construír e documentar os elementos que forman un sistema software orientado a obxectos. Converteuse no estándar de facto da industria, debido a que foi impulsado polos autores dos tres métodos máis usados de orientación a obxectos: Grady Booch, Ivar Jacobson e Jim Rumbaugh. É o estándar actual do chamado Object Management Group (OMG).

UML serve para *especificar*, modelos concretos, non ambiguos e completos. Un modelo de UML representa un sistema software desde unha perspectiva específica. Cada modelo permítenos fixarnos nun aspecto distinto do sistema. Debido á súa estandarización, e aínda que non sexa unha linguaxe de programación, UML pódese conectar de xeito directo a linguaxes de programación como Java, C++ ou Visual Basic.

UML exprésase a través de *elementos de construción*, de relacións e de diagramas que conteñen elementos e relacións

Os **elementos de construción** son os bloques básicos de construción, de catro tipos:

1. **Elementos estruturais.** Son as partes estáticas do modelo. Son sete:





- a) **Clase:** descrición dun conxunto de obxectos que comparten os mesmos atributos, operacións, relacións e semántica.
  - b) **Interface:** colección de operacións que especifican un servizo dunha clase ou compoñente, mostrando o comportamento visible externamente dese elemento.
  - c) **Colaboración:** define unha interacción e representa un conxunto de elementos do modelo no que colaboran para proporcionar un comportamento cooperativo maior que a suma dos comportamentos dos seus elementos.
  - d) **Caso de uso:** descrición dun conxunto de secuencias de accións que un sistema executa e que produce un resultado observable de interese para un usuario particular.
  - e) **Clase activa:** Clase cuxos obxectos teñen un ou máis procesos ou fíos de execución, e polo tanto poden dar orixe a actividades de control.
  - f) **Compoñente:** Parte física e substituíble dun sistema que representa tipicamente o empaketamento físico de diferentes elementos lóxicos, como clases, interfaces e colaboracións.
  - g) **Nodo:** Elemento físico que existe en tempo de execución e representa un recurso computacional que xeralmente dispón de memoria e capacidade de procesamento. (Impresoras, PCs, ...)
2. **Elementos de comportamento:** Partes dinámicas dos modelos. Son dous:
- a) **Interaccións:** unha interacción é un comportamento que comprende un conxunto de mensaxes intercambiadas entre un conxunto de obxectos, dentro dun contexto particular para alcanzar un propósito específico. O comportamento dunha sociedade de obxectos ou unha operación individual pode especificarse mediante unha interacción.



- b) **Máquinas de estados:** unha máquina de estados especifica as secuencias de estados polas que pasa un obxecto ou unha interacción durante a súa vida en resposta a eventos.
3. **Elementos de agrupación:** Son as partes organizativas dos modelos de UML. Só existe unha: o paquete, que é que é un mecanismo para organizar os elementos en grupos.
4. **Elementos de anotación:** Son comentarios que se poden aplicar para describir, clarificar e facer observacións sobre calquera elemento dun modelo. O principal elemento de anotación é a nota, que é simplemente un símbolo para mostrar restricións e comentarios xunto a un elemento ou unha colección de elementos.

As **relacións** entre os elementos estruturais son de catro tipos:

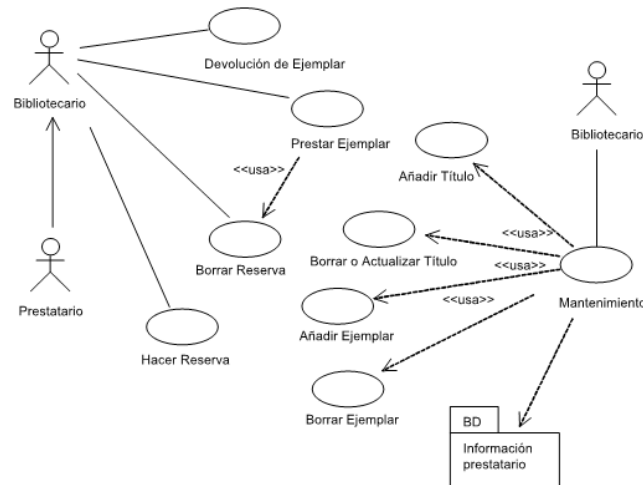
- **Dependencia:** é unha relación semántica entre dous elementos, na cal un cambio a un elemento (o elemento independente) pode afectar á semántica do outro elemento (o elemento dependente).
- **Asociación:** Unha **asociación** é unha relación estrutural que describe un conxunto de enlaces, os cales son conexións entre obxectos. A **agregación** é un tipo especial de asociación, que representa unha relación estrutural entre un todo e os seus partes. A **composición** é un tipo de agregación no que cada parte só pode pertencer a un todo e non pode existir a parte sen o todo.
- **Xeneralización:** Unha xeneralización é unha relación de especialización /xeneralización na cal os obxectos do elemento especializado (o fillo) poden substituír aos obxectos do elemento xeral (o pai). Desta forma, o fillo comparte a estrutura e o comportamento do pai.
- **Realización:** É unha relación semántica entre elementos, onde un elemento especifica un contrato que outro elemento garante que cumprirá. Poden ser entre interfaces e as clases e compoñentes que as realizan, e entre os casos de uso e as colaboracións que os realizan.



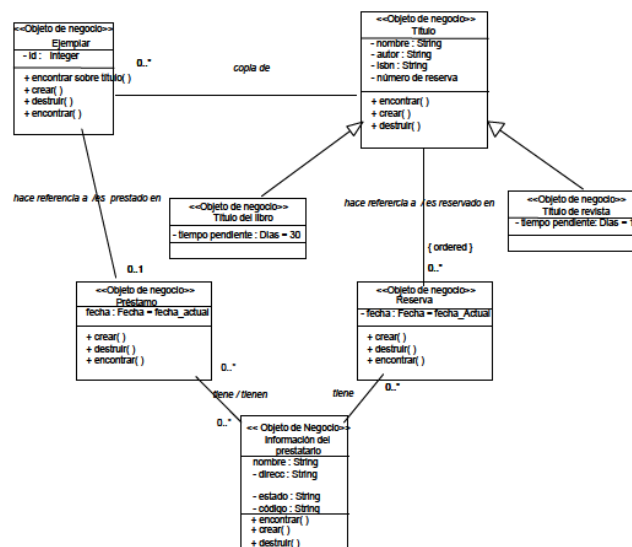
Para rematar, os **diagramas** son a representación gráfica dun conxunto de elementos, en xeral visualizado como un grafo conexo de nodos (elementos) e arcos (relacións). Os diagramas débúxanse para visualizar un sistema desde diferentes perspectivas. Pódense agrupar en dous bloques en función de se vemos o modelo de forma estática (estrutural) ou de forma dinámica (comportamento). A primeira inclúe os diagramas de despregue, compoñentes, clases e obxectos, mentres que a segunda inclúe os diagramas de estados, actividades, secuencia, colaboración e casos de uso.

1. **Diagrama de casos de uso.** Un caso de uso é unha secuencia de accións realizadas polo sistema, que producen un resultado observable e valioso para un usuario en particular, é dicir, representa o comportamento do sistema co fin de dar respostas aos usuarios. Os escenarios son os distintos camiños polos que pode evolucionar un caso de uso, dependendo das condicións que se van dando na súa realización. Os diagramas están formados por dous elementos: **actores**, que é algo ou alguén que se atopa fóra do sistema e que interactúa con el, e **casos de uso**, que representa o comportamento que ofrece o sistema de información desde o punto de vista do usuario. Entre estes elementos pódense dar tres tipos de relacións: comunica (é a relación entre un actor e un caso de uso, que denota a participación do actor en dito caso de uso), usa (relación de dependencia entre dous casos de uso que denota a inclusión do comportamento dun escenario noutro. Úsase cando se quere reflectir un comportamento común en varios casos de uso) e esténdese (relación de dependencia entre dous casos de uso no que un é unha especialización do outro).



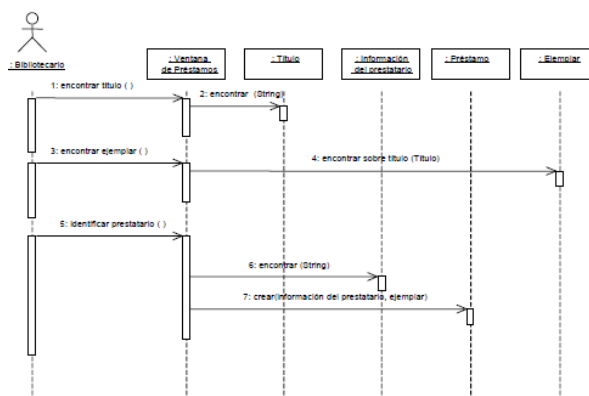


2. **Diagrama de clases:** Representa os aspectos estáticos do sistema, utilizando diversos mecanismos de abstracción (clasificación, xeneralización, agregación). Recolle as clases de obxectos e as súas asociacións. Neste diagrama represéntase a estrutura e o comportamento de cada un dos obxectos do sistema e as súas relacións cos demais obxectos. O modelo está formado por: **clases** (represéntase como unha caixa dividida en tres zonas. Na zona superior ponse o nome da clase, no centro colócanse os atributos (características) da clase e na zona inferior inclúese unha lista coas operacións que proporciona a clase), **relacións** (asociacións, agregacións, composicións, dependencias ou herdanza), **interfaces e paquetes**.





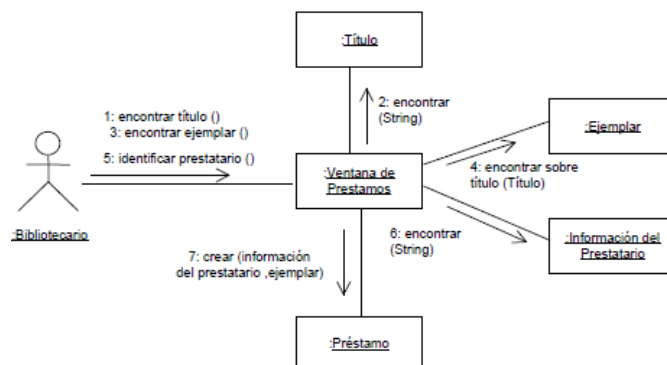
3. **Diagrama de obxectos:** Modelan as instancias de elementos contidos nos diagramas de clases. Mostra un conxunto de obxectos e as súas relacións nun momento concreto. A representación gráfica dun obxecto en UML é igual que a dunha clase pero co nome subliñado.
4. **Diagramas de Interacción.** Utilízanse para modelar os aspectos dinámicos do sistema. Existen dous tipos de diagramas de interacción: **secuencia** e **colaboración**. Ambos son equivalentes. A diferenza entre eles está nos aspectos que resaltan. Os diagramas de secuencia destacan a orde temporal das mensaxes, mentres que os diagramas de colaboración destacan a organización estrutural dos obxectos.
- a. **Diagramas de secuencia:** Mostran as interaccións entre obxectos ordenadas en secuencia **temporal**. Mostra os obxectos que se atopan no escenario e a secuencia de mensaxes intercambiadas entre eles para levar a cabo a funcionalidade descrita polo escenario.



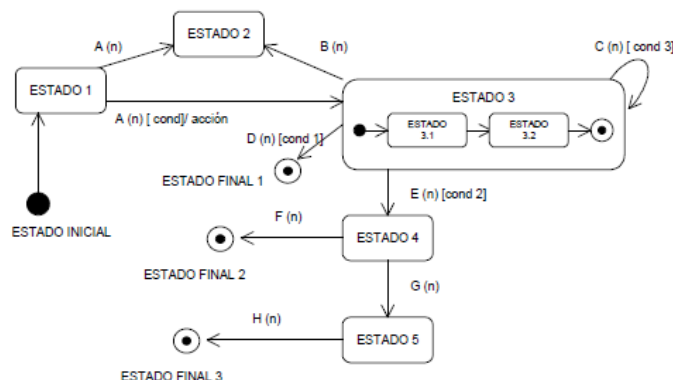
- b. **Diagramas de colaboración:** É unha forma alternativa ao diagrama de secuencia para mostrar un escenario. Mostra a mesma información pero de forma diferente. Nos diagramas de colaboración non existe unha secuencia temporal. Este diagrama resalta a **organización estrutural** dos obxectos que envían e reciben as mensaxes. Este tipo de diagrama mostra



un conxunto de obxectos, enlaces entre eles e as mensaxes que intercambian.

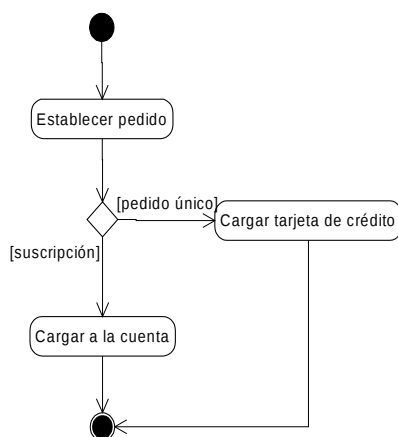


5. **Diagramas de estados:** Representan os estados que pode tomar un compoñente ou un sistema e mostra os eventos que implican o cambio dun estado a outro. Os dous elementos principais nestes diagramas son os **estados** (representa algún comportamento que é observable externamente e que perdura durante un período de tempo finito. Vén dado polo valor dun ou varios atributos que o caracterizan nun momento dado) e as posibles **transicións** entre eles. Unha transición é un cambio de estado producido por un evento e reflicte os posibles camiños para chegar a un estado final desde un estado inicial. Un sistema só pode ter un **estado inicial**, que representa o punto de partida. O estado **final** representa que un compoñente deixou de ter calquera interacción ou actividade. Pode haber varios estados finais nun diagrama.



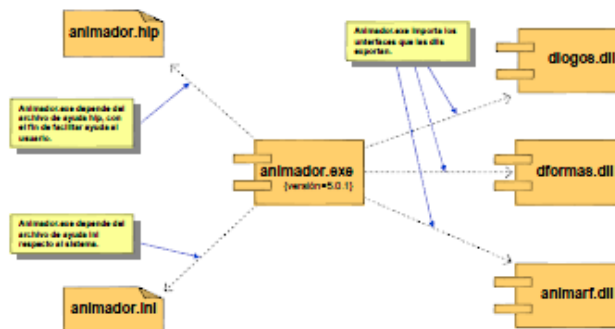


6. **Diagrama de actividades.** Pode considerarse un caso especial do diagrama de estados no cal case todos os estados son estados acción (identifican unha acción que se executa ao entrar nel) e case todas as transicións evolucionan ao termo de dita acción. Adóitanse utilizar para modelar os pasos dun algoritmo.

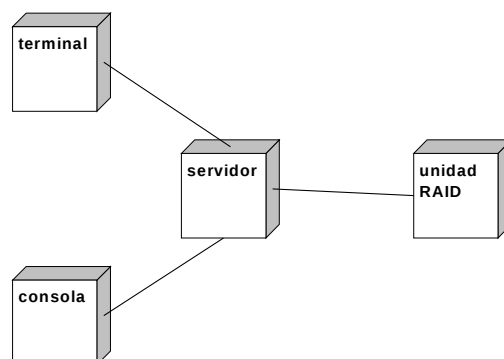


7. **Diagramas de implementación.** Un diagrama de implementación mostra as dependencias entre as partes de código do sistema (diagrama de compoñentes) ou a estrutura do sistema en execución (diagrama de despregue). Os diagramas de compoñentes utilízanse para modelar a vista de implementación estática dun sistema, mentres que os diagramas de despregue se utilizan para modelar a vista de despregue estática.
8. **Diagrama de compoñentes:** Mostra as organizacións e dependencias lóxicas entre compoñentes software, sexan estes compoñentes de código fonte, binarios, documentos, arquivos ou executables. Os diagramas de compoñentes poden conter paquetes para organizar os elementos.





9. **Diagrama de despregue.** Mostra as relacións físicas entre os compoñentes hardware e software no sistema final, é dicir, a configuración dos elementos de procesamento en tempo de execución e os compoñentes software (proceso e obxectos que se executan neles). Cobre principalmente a distribución, entrega e instalación das partes que configuran o sistema físico, e adóitanse utilizar para modelar sistemas encaixados, sistemas cliente-servidor e sistemas distribuídos.



## 32.2 Deseño dun sistema de información

Defínese o Deseño **do Sistema** como o *proceso de definición da arquitectura software: compoñentes, módulos, interfaces, procedementos de proba e datos dun sistema que se crean para satisfacer uns requisitos especificados e cos suficientes detalles como para permitir a súa implementación física*. Segundo a Metodoloxía Métrica V3, o obxectivo do proceso de Deseño do Sistema de Información é a definición da arquitectura do sistema e do contorno tecnolóxico que lle vai a dar soporte,



xunto coa especificación detallada dos compoñentes do sistema de información.

### **32.2.1 Deseño estruturado.**

O deseño estruturado ocúpase da identificación, selección e organización dos módulos e as súas relacións. Decídese que compoñentes son necesarios e a interconexión entre os mesmos para solucionar un problema ben especificado. Comézase coa especificación resultante do proceso de análise e realízase unha descomposición do sistema en módulos estruturados en xerarquías de tal modo que permitan a implementación dun sistema que non requira elevados custos de mantemento.

Os principios que se consideran xeralmente aceptados para as técnicas de deseño estruturado son os seguintes:

- **Modularización.** Perséguese que a Arquitectura técnica do sistema se fundamente en módulos de pequeno tamaño.
- **Independencia modular.** Cada un dos módulos do sistema debe orientarse a que realice unha función ben definida e independente, o cal permita “illar” a orixe dun problema ou a área do sistema a modificar en caso preciso.
- **Modelización conceptual.** A estruturación do sistema en módulos non debe atender a un criterio exclusivamente técnico en sentido reducido, senón que debe reflectir conceptualmente a estrutura da organización á que serve, de modo que se facilite a súa comprensión.
- **Principio de “caixa negra”.** Unha correcta definición das interfaces permitirá ignorar a estrutura interna dos módulos (é dicir, consideralos como unha “caixa negra”) e centrar a atención exclusivamente nos resultados que devolve.

A técnica máis importante dentro do Deseño Estruturado é o Diagrama de Estruturas (Structure chart), tamén denominada Diagrama de Estrutura de Cadros Os **diagramas de estrutura** son unha ferramenta para modelar os



módulos dun sistema e as súas relacións e, xunto coas especificacións de funcionalidade dos módulos e as estruturas de datos, compoñen un deseño inicial que deberá ser analizado e mellorado. Permiten modelar un programa como unha xerarquía de módulos. Cada nivel da xerarquía representa unha descomposición máis detallada do módulo do nivel superior. A notación usada componse basicamente de tres símbolos:

- **Módulos:** un conxunto de instrucións que executan algunha actividade e se presenta como unha caixa negra. Desde un punto de vista práctico, un módulo é unha colección de instrucións dun programa con catro características básicas: *Entradas e Saídas*, é dicir, o que un módulo recibe nunha invocación e que necesita e o que retorna como resultado; *Función* que son as actividades que un módulo fai coa entrada para producir a saída; a *Lóxica Interna* pola cal se executa a función; e *Estado Interno* que é a súa área de datos privada, datos para os cales só o módulo fai referencia. As entradas e saídas e as funcións provén unha visión externa do módulo, mentres que a lóxica interna representa a visión interna do módulo. Un módulo preséntase como unha caixa, cun nome no interior que representa a súa función e as entradas e saídas representadas por pequenas frechas que entran e saen do módulo.
- **Invocacións:** os diagramas de estrutura mostran as invocacións que un módulo fai a outros módulos. Estas invocacións son deseñadas como unha frecha que sae do módulo chamador e apunta ao módulo chamado. Os diagramas de estrutura non teñen especificada a orde de invocación dos módulos invocados.
- **Cuplas:** Son as comunicacións entre módulos. Existen varios tipos, baseados no que elas poden producir no módulo receptor:
  - o Cupla de Datos. Transporta datos a un módulo. Son tamén coñecidos como parámetros de entrada do módulo.
  - o Cupla Modificada. Cunha frecha dobre (apuntando ao modulo chamador e ao módulo chamado) especifícase un argumento enviado a un



módulo que deberá modificar o seu valor e do volver este valor modificado.  
Parámetros de entrada/saída

- o                      Cupla de Resultados. Existen módulos que retornan valores sen a necesidade de que estean inicializados no momento que se invocan.

Como dixemos ao principio ao falar dos diagramas de estruturas, estes constitúen unha aproximación inicial que debe ser analizada e mellorada. Entre os criterios que se utilizan para mellorar o deseño temos:

- **Acoplamento:** O acoplamento entre módulos clasifica o grado de independencia entre pares de módulos. O obxectivo é minimizar o acoplamento, é dicir, maximizar a independencia entre módulos.
- **Cohesión:** é a medida de intensidade de asociación funcional dos elementos dun módulo. O obxectivo do deseño estruturado é obter módulos altamente cohesivos, cuxos elementos estean forte e xenuinamente relacionados uns con outros. Doutra banda, os elementos dun módulo non deberían estar fortemente relacionados con elementos doutros módulos, porque iso levaría a un forte acoplamento entre eles.
- **Descomposición** (Factoring): A descomposición é a separación dunha función contida nun módulo, para un novo módulo. Pode realizarse para reducir o tamaño do módulo, para facer o sistema máis claro, para minimizar a duplicación de código, para separar o traballo da coordinación de traballos ou para crear módulos máis xerais e facilmente reutilizables.
- **Fan-Out:** O fan-out dun módulo é usado como unha medida de complexidade. É o número de subordinados inmediatos dun módulo (cantidade de módulos invocados).
- **Fan-In:** O fan-in dun módulo é usado como unha medida de reusabilidade. É o número de superiores inmediatos dun módulo (a cantidade de módulos que o invocan).

### **32.2.2 Deseño de datos**



O deseño de datos (ás veces chamado arquitectura de datos) crea un modelo de datos e/ou información que se representa cun alto nivel de abstracción (a visión de datos do cliente/usuario). Este modelo de datos é entón refinado en progresivas representacións específicas da implementación, que poden ser procesadas por un sistema baseado en computadora. A estrutura de datos foi sempre unha parte importante do deseño de software.

Os obxectos de datos definidos durante a análise de requisitos do software son modelados utilizando diagramas entidade-relación e o dicionario de datos. A actividade de deseño de datos traduce eses elementos do modelo de requisitos en estruturas de datos a nivel dos compoñentes do software e, cando é necesario, a arquitectura de base de datos a nivel de aplicación.

O deseño de datos a nivel de compoñentes céntrase na representación de estruturas de datos ás que se accede directamente a través dun ou máis compoñentes do software. En realidade, o deseño de datos empeza durante a creación do modelo de análise. Wasserman propuxo un conxunto de principios que poden empregarse para especificar e deseñar dita estrutura de datos, que poden integrarse nas fases de análises e deseño:

- 1) Os principios da análise sistemática aplicados á función e ao comportamento deberían aplicarse tamén aos datos. Investimos moito tempo e esforzo en obter, revisar e especificar os requisitos funcionais e o deseño preliminar. As representacións de fluxo de datos e contido deberían desenvolverse e revisarse; as de obxectos de datos deberían identificarse, deberíanse estudar as organizacións alternativas de datos e debería avaliarse o impacto do modelado de datos sobre o deseño do software. Unha organización de datos alternativa poderíanos levar a obter mellores resultados.
- 2) Todas as estruturas de datos e as operacións a levar a cabo en cada unha delas deberían estar claramente identificadas. O deseño dunha estrutura de datos eficaz debe ter en conta as operacións que se van



a levar a cabo sobre devandita estrutura de datos. A especificación dos tipos de datos abstractos pode simplificar considerablemente o deseño do software.

- 3) Deberíase establecer un dicionario de datos e usalo para definir o deseño dos datos e do programa. Un dicionario de datos representa explicitamente as relacións entre os obxectos de datos e as restricións dos elementos dunha estrutura de datos. Os algoritmos que deben aproveitarse destas relacións específicas poden definirse máis facilmente se existe unha especificación de datos tipo dicionario.
- 4) As decisións de deseño de datos de baixo nivel deberían deixarse para o final do proceso de deseño. Pódese usar un proceso de refinamento paso a paso para o deseño de datos. É dicir, a organización xeral de datos pode definirse durante a análise de requisitos; refinarse durante os traballos de deseño de datos e especificarse en detalle durante o deseño a nivel de compoñentes. O enfoque descendente do deseño de datos proporciona vantaxes análogas ao enfoque descendente do deseño de software: deséñanse e avalíanse primeiramente os atributos estruturais principais de maneira que se poida establecer a arquitectura dos datos.
- 5) A representación das estruturas de datos deberían coñecerla só aqueles módulos que deban facer uso directo dos datos contidos dentro da estrutura. O concepto de ocultación de información e o concepto relacionado de acoplamento proporcionan unha importante visión da calidade do deseño do software. Ademais, débese separar a visión lóxica dun obxecto de datos da súa visión física.
- 6) Debería desenvolverse unha biblioteca de estruturas de datos útiles e das operacións que se lles poden aplicar. As estruturas de datos e as operacións deberían considerarse como recursos para o deseño do software. As estruturas de datos poden deseñarse para que se poidan reutilizar. Unha biblioteca de persoais de estruturas de datos (tipos



abstractos de datos) pode reducir o esforzo do deseño e da especificación de datos.

- 7) Un deseño do software e unha linguaxe de programación debería soportar a especificación e realización dos tipos abstractos de datos. A implementación dunha estrutura de datos sofisticada pode facerse excesivamente difícil se non existen os medios de especificación directa da estrutura na linguaxe de programación escollida para dita implementación.

### **32.2.3 Deseño da interface de usuario.**

Todo equipo, e dentro deste toda aplicación software, necesita poder comunicarse co usuario dalgunha forma. A idea fundamental no concepto de interface é a de mediación entre home e máquina. Podemos definir entón a Interface de Usuario (en diante “IU”), como o conxunto de elementos hardware e software dun equipo que presentan información ao usuario e permítenlle interactuar coa información e co equipo. É dicir, é a cara que vai presentar o equipo (e dentro desta cada aplicación software) ao usuario para poder comunicarse con el.

Dentro da interface de usuario teremos dous elementos:

- **Parte Hardware.** É o referente á parte física do ordenador, os dispositivos utilizados para introducir, procesar e entregar os datos, basicamente: teclado, rato e monitor.
- **Parte Software.** É na que adoita pensarse cando se fala de Interface de Usuario. É a relativa a presentación que mostra, normalmente a través do monitor, unha aplicación software.

A evolución das interfaces vai en paralelo ao sistema operativo, podendo ter os seguintes tipos de interfaces:

- **Interface de liña de comandos:** o usuario escribe ordes usando unha linguaxe formal con vocabulario e sintaxe propia. Úsase un



teclado, tipicamente, e as ordes están encamiñadas a realizar unha acción e mostrar os resultados ao usuario.

- **Interface a través de Menús:** Un Menú é unha lista de opcións que se mostran en pantalla ou nunha fiestra da pantalla para que os usuarios elixan a opción que desexen. Temos distintos tipos: Menús de pantalla completa, menús de barra, en cadoiro, paleta de ferramentas ou menús contextuais.
- **Interfaces gráficos de usuario.** A idea básica ou paradigma das interfaces gráficas é: *O que ves é o que podes conseguir* (en inglés, a coñecida frase: *What you see is what you get*, WYSIWYG) e isto conséguese fundamentalmente coa manipulación directa do Interface de usuario a través de iconas e gráficos en xeral. De feito, a interface de usuario gráfica dun sistema xunto cos seus dispositivos da entrada ás veces é chamado: *look-and-feel* (mirar e sentir). Un GUI é unha representación gráfica na pantalla do ordenador dos programas, datos e obxectos, así como da interacción con eles. Proporciona ao usuario as ferramentas para realizar as súas operacións, máis que unha lista das posibles operacións. Aínda que o uso dos GUI é máis sinxelo que o das interfaces de liña de comandos, a introdución de instrucións é máis lenta, polo que as GUI adoitan ter a opción de empregar un sistema equivalente ao de liña de instrucións como alternativa rápida para os usuarios máis expertos.

Existen tres puntos de vista distintos, tamén chamados modelos, no deseño de IU:

- Modelo do **Usuario**: O usuario ten a súa visión persoal do sistema, e espera que este se comporte dunha certa forma.
- Modelo do **Programador**: O programador non ten que considerar a parte visual de interacción co usuario, senón o funcionamento interno da aplicación. Os obxectos que manipula o programador, son distintos dos que trata o usuario.





- Modelo do **Deseñador**: O deseñador debe ter un punto de vista que mesture as necesidades, ideas e desexos do usuario cos materiais de que dispón o programador para deseñar un produto de software . É un intermediario entre ambos. O modelo do deseñador describe os obxectos que utiliza o usuario, a súa presentación ao mesmo e as técnicas de interacción para a súa manipulación. Consta de tres partes: **presentación, interacción e relacións** entre os obxectos.

Existen varios principios relevantes para o deseño e implementación de IU, sobre todo para os IU gráficas, sendo algúns dos máis importantes son os seguintes

- **Anticipación.** As aplicacións deberían intentar anticiparse ás necesidades do usuario e non esperar a que o usuario teña que buscar a información, recompilala ou invocar as ferramentas que vai utilizar.
- **Autonomía.** O ordenador, a IU e o contorno de traballo deben estar a disposición do usuario. Débese dar ao usuario o ambiente flexible para que poida aprender rapidamente a usar a aplicación. Con todo, está comprobado que o contorno de traballo debe ter certas cotas, é dicir, ser explorable pero non azaroso.
- **Percepción da cor.** Aínda que se utilicen convencións de cor na IU, deberíanse usar outros mecanismos secundarios para prover a información a aqueles usuarios con problemas na visualización de cores
- **Valores por defecto.** Os valores por defecto deberían ser opcións intelixentes e sensatas. Ademais, os mesmos teñen que ser fáciles de modificar.
- **Consistencia.** Para lograr unha maior consistencia na IU requírese profundar en diferentes aspectos que están catalogados en niveis. Realízase un ordenamento de maior a menor consistencia:
  - o **Interpretación do comportamento do usuario:** A IU debe comprender o significado que lle atribúe un usuario a cada





requisito. Exemplo: manter o significado dos comandos abreviados (shortcutkeys) definidos polo usuario.

- o **Estruturas invisibles:** requírese unha definición clara das mesmas, xa que, doutra maneira, o usuario nunca podería chegar a descubrir o seu uso. Exemplo: ampliación de fiestras mediante a extensión dos seus bordos.
- o **Pequenas estruturas visibles:** pódese establecer un conxunto de obxectos visibles capaces de ser controlados polo usuario que permitan aforrar tempo na execución de tarefas específicas. Exemplo: icona e/ou botón para impresión.
- o **Unha soa aplicación ou servizo:** a IU permite visualizar á aplicación ou servizo utilizado como un compoñente único. Exemplo: A IU desprega un único menú, podendo ademais acceder ao mesmo mediante comandos abreviados.
- o **Un conxunto de aplicacións ou servizos:** a IU visualiza á aplicación ou servizo utilizado como un conxunto de compoñentes. Exemplo: A IU preséntase como un conxunto de barras de comandos despregadas en diferentes lugares da pantalla, podendo ser desactivadas en forma independente.
- o **Consistencia do ambiente:** a IU mantense en concordancia co ambiente de traballo. Exemplo: A IU utiliza obxectos de control como menús, botóns de comandos de xeito análogo a outras IU que se usen no ambiente de traballo.
- o **Consistencia da plataforma:** A IU é concordante coa plataforma. Exemplo: A IU ten un esquema baseado en fiestras, o cal é acorde ao manexo do sistema operativo utilizado.
- **Eficiencia do usuario.** Débese considerar a produtividade do usuario antes que a produtividade da máquina. Se o usuario debe esperar a resposta do sistema por un período prolongado, estas perdas de tempo pódense converter en perdas económicas para a organización. As mensaxes de axuda deben ser sinxelas e prover respostas aos problemas.



Os menús e etiquetas de botóns deberían ter as palabras claves do proceso.

- **Lei de Fitt.** Segundo di a lei de Fitt: “O tempo para alcanzar un obxectivo é unha función da distancia e tamaño do obxectivo”. É por iso que é conveniente usar obxectos grandes para as funcións importantes.
- **Interfaces explorables.** Sempre que sexa posible débese permitir que o usuario poida saír axilmente da IU, deixando unha marca do estado de avance do seu traballo, para que poida continualo noutra oportunidade. Para aqueles usuarios que sexan novatos no uso da aplicación, deberase prover de guías para realizar tarefas que non sexan habituais. Sempre se debe contar cun comando “desfacer”. Leste suprimirá a necesidade de ter que contar con diálogos de confirmación para cada acción que realice en sistema. O usuario debe sentirse seguro de poder saír do sistema cando o desexe. A IU debe ter un obxecto fácil de accionar co cal poder finalizar a aplicación.
- **Uso de metáforas.** As boas metáforas crean figuras mentais fáciles de recordar. A IU pode conter obxectos asociados ao modelo conceptual en forma visual, con son ou outra característica perceptible polo usuario que axude a simplificar o uso do sistema.
- **Curva de aprendizaxe.** A aprendizaxe dun produto e a súa usabilidade non son mutuamente excluíntes. O ideal é que a curva de aprendizaxe sexa nula, e que o usuario principiante poida alcanzar o dominio total da aplicación sen esforzo.
- **Redución de latencia.** Sempre que sexa posible, o uso de tramas (multi-threading) permite colocar a latencia en segundo plano (background ). As técnicas de traballo multitarefa posibilitan o traballo ininterrompido do usuario, realizando as tarefas de transmisión e computación de datos en segundo plano.
- **Protección do Traballo.** Débese poder asegurar que o usuario nunca perda o seu traballo, xa sexa por erro do seu parte, problemas de transmisión de datos, de enerxía, ou algunha outra razón inevitable.



- **Lexibilidade.** Para que a IU favoreza a usabilidade do sistema de software, a información que se exhiba nela debe ser fácil de situar e ler. Para lograr obter este resultado débense ter en conta algunhas premisas como: o texto que apareza na IU debería ter un alto contraste, débese utilizar combinacións de cores como o texto en negro sobre fondo branco ou amarelo suave. O tamaño das fontes ten que ser o suficientemente grande como para poder ser lido en monitores estándar. É importante facer clara a presentación visual (colocación/agrupación de obxectos, evitar a presentación de excesiva información).

#### **32.2.4 Deseño procedemental.**

O deseño **procedemental** realízase despois de que se estableceu a estrutura do programa e dos datos. Transforma os elementos estruturais nunha descrición dos elementos do software. Define os algoritmos de procesamento necesarios. Debe especificar os detalles dos procedementos sen ambigüidade e ademais debe facilitar a codificación, de forma que o código se obteña de forma natural a partir do deseño.

A finais dos 60 propúxose a utilización dun conxunto de construcións lóxicas coas que podía formarse calquera programa. Estas construcións serían a **secuencia**, a **condición** e a **repetición**, e calquera programa podería construírse utilizando só este tipo de construcións. Estas son un dos fundamentos da programación estruturada. As construcións estruturadas propuxéronse para limitar o deseño procedemental do software a un número pequeno de operacións predicibles, o que facilita a lexibilidade, proba e mantemento do software.

Para a especificación procedemental que define os algoritmos poderíase utilizar a linguaxe natural, pero debido á cantidade de ambigüidades que esta linguaxe implica, é necesario utilizar unha forma máis restrinxida de representación. E entre estas formas de representación temos:

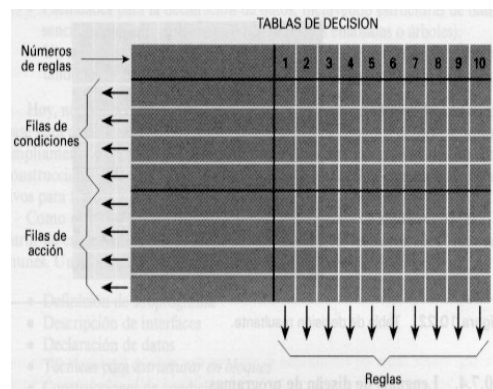
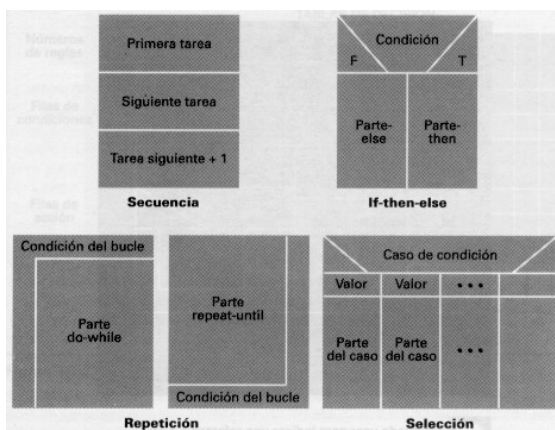
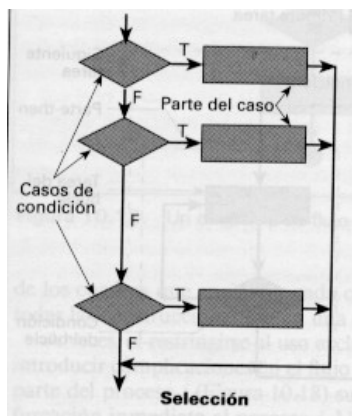




- **Diagrama de fluxo:** É a representación gráfica que máis se utiliza no deseño procedemental. Para representar un paso de procesamento utilízase un cadro, para representar unha condición utilízase un rombo, e para representar o fluxo de control utilízanse frechas.
- **Diagrama de caixas (N-S):** Esta notación xurdiu do desexo de desenvolver unha representación para o deseño procedemental que non permitise a violación de construcións estruturadas. Estes diagramas foron desenvolvidos por Nassi e Schneiderman e perfeccionados por Chapin, e teñen as características seguintes:
  - o O ámbito funcional está ben definido e é claramente visible.
  - o A transferencia de control arbitraria é imposible.
  - o É fácil determinar o ámbito dos datos locais e globais.
  - o A recursividade é fácil de representar
- **Táboas de decisión.** Constitúen unha notación que traduce as accións e condicións a unha forma tabular. Os pasos para a creación dunha táboa de decisión son os seguintes:
  - o Listar todas as accións que se poden asociar a un módulo.
  - o Listar todas as condicións necesarias para a execución dun procedemento.
  - o Asociar conxuntos de condicións específicas a accións específicas, eliminando combinacións imposibles. Desenvolver as posibles combinacións de condicións.
  - o Definir regras indicando as accións que ocorren para unha serie de condicións.
- **Linguaxe de deseño de programas.** Unha linguaxe de deseño de programas (LDP), tamén coñecida como linguaxe estruturada ou pseudocódigo é unha linguaxe que utiliza o vocabulario dunha linguaxe e a sintaxe doutra. Independentemente da súa orixe, unha LDP ten que ter as características seguintes:



- o Unha sintaxe fixa de palabras clave que permitan construír todas as construcións estruturadas, declarar datos e establecer características de modularidade.
- o Unha sintaxe libre en linguaxe natural para describir as características de procesamento.
- o Facilidades para a declaración de datos, incluíndo estruturas de datos simples e complexas.
- o Un mecanismo de definición de subprogramas e de chamada a estes.



**32.2.5**

**Deseño**

### ***orientado a obxectos.***

O deseño orientado a obxectos transforma o modelo de análise creada usando análise orientada a obxectos, nun modelo de deseño que serve para a construción de software que poderá ser implementado nunha linguaxe de programación. Débense identificar os obxectos pertinentes, clasificalos dentro das clases na granularidade correcta, definir interfaces de clases e xerarquías de herdanza e establecer relacións clave entre eles, de tal forma que se obteña a solución ao problema do usuario. O deseño debe ser específico ao problema que se ten entre mans, pero suficientemente xeral para adaptarse a problemas e requisitos futuros.

Os métodos estruturados e os seus correspondentes notacións definen un sistema como un conxunto secuencial de módulos interdependentes que



comparten datos. En cambio, os métodos orientados a obxectos definen un conxunto de módulos independentes relacionados e con visibilidade limitada entre eles.

Esencialmente, o DOO consta dos tres seguintes pasos:

- **Identificación e definición de obxectos e clases.** O principal problema do desenvolvemento dun sistema orientado a obxectos é atopar os obxectos na fase de AOO e DOO. Un posible método para atopalos é o método gramatical proposto por Booch que analiza a descrición textual do sistema e propón os substantivos como posibles identificadores das clases e os verbos como posibles métodos das clases. A listaxe resultante de clases (nomes) e métodos (verbos) utilizarase para comezar o proceso de deseño. Unha vez identificados os obxectos, detállanse os atributos de cada un. Co **método de Booch** para atopar clases é difícil conseguir un resultado de alta calidade, pois a abstracción de clases que consigamos depende dunha estruturación intelixente da descrición do problema en elementos independentes e intuitivamente correctos. Outro modelo que utiliza o método gramatical para a identificación de clases e obxectos é o modelo **CRC** (Clases - Responsabilidades - Colaboracións). Un modelo CRC é unha colección de tarxetas que representan clases. As tarxetas están divididas en tres seccións. Ao longo da cabeceira da tarxeta escríbese o nome da clase. No corpo lístanse as responsabilidades da clase á esquerda e á dereita os colaboradores. As **responsabilidades** estarían formadas polos atributos e operacións das clases. As clases cumpren coas súas responsabilidades de dúas formas: Ou ben unha clase pode usar as súas propias operacións para manipular os seus propios atributos, cumprindo polo tanto cunha responsabilidade particular, ou pode colaborar con outras clases.

Definíronse unha serie de directrices que facilitan a identificación e definición de clases e métodos:





- o Modelar con clases as entidades que ocorren de forma natural no problema.
- o Diseñar métodos cunha soa finalidade.
- o Diseñar un método novo antes de ampliar un existente.
- o Evitar métodos extensos.
- o Gardar como instancia os datos necesitados por máis dun método ou por unha subclase.
- o O deseñador debe traballar para obter unha biblioteca de clases, e non para el mesmo, nin para desenvolver o sistema actual.

Ademais, para evitar a creación de clases innecesarias ou declaración de clases que non o sexan, unha clase debe ofrecer unha serie de servizos a obxectos dun tipo determinado. Unha clase deberíase crear cando:

- o A nova clase represente unha abstracción significativa do problema
  - o É probable que os servizos que proporciona sexan utilizados por outras clases.
  - o O seu comportamento sexa complexo.
  - o Se se representase como un método doutra clase, poucos usuarios desta o invocarían.
- **Organización de relacións entre clases.** Unha vez identificadas as clases deberíamos atopar as relacións entre estas. Para iso poderíamos usar distintas alternativas, como o modelo Clases-Responsabilidades-Colaboracións (CRC) visto anteriormente. Cada tarxeta do modelo CRC contén unha clase cunha lista de responsabilidades. O seguinte paso é definir aquelas clases colaboradoras que axudan na realización de cada responsabilidade. Isto establece as **conexións ou relacións** entre clases. As **relacións** poden derivarse a partir do exame dos verbos na



descripción do problema. Unha vez conectadas as clases cos seus colaboradores, etiquetamos cada unha destas conexións, engadímoslles unha dirección, en función de que clase chama a que outra e para rematar avalíase cada extremo da conexión para determinar a cardinalidade. Este modelo de clases conectadas dá lugar ao **modelo Obxecto-Relación**.

- **Extracción de estruturas nunha xerarquía de clases.** Unha vez definidos os obxectos e as súas relacións, o paso seguinte consiste en observar características comúns para crear abstraccións a nivel de clase, pero non existe ningunha metodoloxía formal para a realización destas abstraccións. Un dos obxectos dun deseño orientado a obxectos é gardar propiedades e métodos de clases no nivel de abstracción máis alto posible, de forma que os compartan a maioría de clases e se fomente a reutilización. Para este paso empréganse os diagramas de xeneralización ou herdanza.

Aínda que desde os inicios do desenvolvemento de software orientado a obxectos existiron múltiples metodoloxías, por exemplo 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, a Linguaxe Unificada de modelado (UML) puxo fin á guerra de metodoloxías. Actualmente na industria do desenvolvemento de software temos a UML como un estándar para o modelado de sistemas orientados a obxectos que é usado tanto para a fase de análise como desenvolvemento.

Durante o proceso de DOO, un enxeñeiro do software debe observar cada oportunidade na que poida reutilizar patróns de deseño existentes (cando cumpren as necesidades do deseño), no canto de crear outros novos. Os patróns de deseño son a base para a procura de solucións a problemas comúns no desenvolvemento de software e outros ámbitos referentes ao



diseño de interacción ou interfaces. Un **patrón de diseño** é unha solución a un problema de diseño. Para que unha solución sexa considerada un patrón debe posuír certas características. Unha delas é que debe comprobar a súa efectividade resolvendo problemas similares en ocasións anteriores. Outra é que debe ser reutilizable, o que significa que é 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 dirixido pola introdución á ambigüidade. 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**

**Xefe do Servizo de Informática. Instituto Galego de Vivenda e Solo**

**Vicepresidente do CPEIG**



# **33. ENXEÑARÍA DE REQUISITOS. VERIFICACIÓN. VALIDACIÓN. ESPECIFICACIÓN DE REQUISITOS. XESTIÓN DE REQUISITOS.**



## **Tema 33. Enxeñería de requisitos. Verificación. Validación. Especificación de requisitos. Xestión de requisitos.**

### **INDICE**

- 33.1 Enxeñería de requisitos
  - 33.1.1 Tipos de requisitos
- 33.2 Identificación dos requisitos do software
  - 33.2.1 Entrevistas
  - 33.2.2 JAD (Joint Application Design)
  - 33.2.3 Prototipos
  - 33.2.4 Análise de factores críticos de éxito
  - 33.2.5 Brainstorming
  - 33.2.6 Escenarios e 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 Xestión de requisitos
  - 33.5.1 Planificación da xestión requisitos.
  - 33.5.2 Xestión do cambio.

### **35.1 Enxeñería de requisitos**

Na enxeñería de software, a Enxeñería de requisitos comprende todas as tarefas relacionadas coa determinación das necesidades ou das condicións a satisfacer para un software novo ou modificado, tomando en conta os diversos requisitos dos investidores. Así, os requisitos xéranse a partir da interacción entre os usuarios e os enxeñeiros do software, representando



as características do sistema a construír, é dicir, as necesidades dos usuarios.

Hai múltiples definicións do termo requisito, algunhas delas son:

- Segundo IEEE un **requisito** é:
  - o Unha condición ou capacidade necesitada por un usuario para resolver un problema ou alcanzar un obxectivo.
  - o Unha condición ou capacidade que debe cumprir ou posuír un sistema ou un compoñente do mesmo para satisfacer un contrato, un estándar, unha especificación, ou outro documento imposto dun xeito formal.
  - o Unha representación documentada dunha condición ou capacidade tal como as expresadas nos dous puntos anteriores.
- Desde o punto de vista do usuario, os requisitos son as condicións ou capacidades necesarias para que un usuario poida resolver un problema ou alcanzar un obxectivo
- Desde o punto de vista do equipo de desenvolvemento, os requisitos son as capacidades ou condicións que debe reunir un sistema para satisfacer un contrato, estándar ou calquera outro documento imposto formalmente.

En resumo, os requisitos son as características que debe cumprir o sistema para que cubra as necesidades dos usuarios. Moitas veces fálase de “requirimentos” no canto de requisitos. Isto débese a unha mala tradución do inglés. A palabra *requirement* debe ser traducida como requisito, mentres que requirimento tradúcese ao inglés como *request*.

Sommerville divide o proceso de enxeñería de requisitos en catro subprocesos que son a avaliación de se o sistema é útil para o negocio (**estudo de viabilidade**); o **descubrimento de requisitos** (obtención e análises); a transformación destes requisitos en formularios estándar



(**especificación**), e a verificación de que os requisitos realmente definen o sistema que quere o cliente (**validación**).

Para Thayer, “A enxeñería de requisitos proporciona o mecanismo apropiado para entender o que o cliente quere, analizar as necesidades, avaliar a factibilidade, negociar unha solución razoable, especificar a solución sen ambigüidades, validar a especificación e administrar os requisitos conforme se transforman nun sistema operacional” e establece as seguintes fases para o proceso:

- **Inicio.** Establécese unha comprensión básica do problema por parte dos analistas.
- **Obtención.** Obtéñense os requisitos do software mediante a interacción entre os analistas e o cliente.
- **Elaboración.** Refínase a información obtida no paso anterior e enfócase á construción dun modelo de análise que represente o sistema a construír.
- **Negociación.** Hai requisitos que non son implementables ou son difíciles de facerse, por esta razón os analistas negocian estes requisitos para chegar a un entendemento e lograr un sistema factible de desenvolverse nun prazo e custo.
- **Especificación.** Confecciónase un conxunto de documentos (descricións en linguaxe natural, diagramas, etc.) que definan o que o sistema debe facer.
- **Validación.** Examínase a especificación para asegurar que todos os requisitos software establecéronse dun xeito preciso, que non hai inconsistencias, omisións nin erros, ademais de cumprirse os estándares de calidade establecidos para o proxecto.
- **Xestión de Requisitos.** Esta actividade permite tratar co inevitable problema dos cambios de especificacións, identificando, controlando e determinando o impacto do cambio de requisitos no resto.

Outros autores modifican o número de etapas e divídenas en:



- **Edución de Requisitos.** Nela os analistas obteñen as necesidades do cliente a partir de todas as fontes de información que teñen dispoñibles (documentación, entrevistas, estudo dos procesos da organización, etc.). Termos equivalentes usados polos enxeñeiros de software para esta actividade son Extracción de Requisitos, Identificación de Requisitos, Determinación de Requisitos, etc. Vemos que estas se corresponderían coas dúas primeiras do modelo anterior.
- **Análise de Requisitos:** procédese a traballar sobre os requisitos educidos no paso anterior. Estúdanse os requisitos en busca de conflitos e incoherencias, implicacións, información non obtida e aspectos non resoltos. Despois clasifícanse, avalía a súa viabilidade e intégranse os novos requisitos cos xa existentes. O obxectivo final é lograr unha listaxe de requisitos que defina as necesidades do cliente. Corresponderían coas fases de Elaboración e Negociación do modelo anterior.
- **Representación dos Requisitos.** Actividade na que se representan os requisitos dunha ou máis formas, utilizando para iso diferentes técnicas, por exemplo a linguaxe formal, a linguaxe natural, representacións gráficas, etc. Para a representación existen múltiples técnicas; as máis usadas son, entre outras, os Diagramas de Fluxo de Datos, Modelo Entidade Relación, Casos de Uso ou Diagramas de Clases. Unha vez que están os requisitos representados, é necesario que se reúnan os diversos participantes no desenvolvemento para revisalos e aprobalos. O produto final co que culmina esta fase é a Especificación de Requisitos Software, onde está descrito con exactitude todo o que o sistema debe facer. Corresponde ás fases de Especificación e Validación do modelo anterior.
- **Validación de Requisitos** Procédese a definir unha serie de criterios e técnicas que permitirán, cando o sistema estea construído, comprobar que este cumpre os requisitos.



Independentemente do modelo que usemos, o importante é ter claro que o obxectivo da enxeñería de requisitos é determinar con claridade e precisión que é o que hai que facer, e para iso será necesario identificar os requisitos clave.

Para pensar na importancia dos requisitos na Enxeñería do Software, algúns estudos reflicten que o 45 por 100 dos erros teñen a súa orixe nos requisitos e no deseño preliminar, outros sinalan que o 56 por 100 dos erros que teñen lugar nun proxecto software débense a unha mala especificación de requisitos, etc.

Os factores principais que conducen ao fracaso nos proxectos software son: a falta de comunicación cos usuarios, os requisitos incompletos e os cambios nos requisitos; e a evidencia demostra que os requisitos conteñen demasiados erros, que moitos destes erros non se detectan ao principio, pero poderían ser detectados, e que non detectar estes erros incrementa grandemente os custos do proxecto e a súa duración. A consecuencia é que o sistema non satisfará os usuarios, que se producirán desacordos entre usuarios e desenvolvedores, e se gastará tempo e diñeiro en construír un sistema equivocado.

Quizais non sexa posible elaborar os requisitos con absoluta perfección, pero débese intentar minimizar o impacto dos erros nos requisitos e organizar mellor as tarefas relacionadas con eles. Para remediar esta situación é para o que xorde a **Enxeñería de Requisitos**.

### *33.1 Tipos de requisitos*

A maioría dos autores distinguen entre:

- **Requisitos funcionais:** Son declaracións dos servizos que debe proporcionar o sistema, do xeito en que este debe reaccionar a entradas particulares e de como se debe comportar en situacións concretas. Nalgúns



casos, os requisitos funcionais dos sistemas tamén poden declarar explicitamente o que o sistema non debe facer. Os requisitos funcionais dun sistema describen o que o sistema debe facer.

- **Requisitos non funcionais:** Son restricións dos servizos ou funcións ofrecidos polo sistema. Inclúen restricións de tempo, sobre o proceso de desenvolvemento e estándares. Como o seu nome suxire, non se refiren directamente ás funcións específicas que proporciona o sistema, senón ás propiedades deste como a fiabilidade, o tempo de resposta e a capacidade de almacenamento. Poden vir das características requiridas do software (requisitos do produto), da organización que desenvolve o software (requisitos organizacionais) ou de fontes externas.
- o **Requisitos do produto.** Estes requisitos especifican o comportamento do produto. Algúns exemplos son os requisitos de rendemento na rapidez de execución do sistema e cantos memoria se precisa; os requisitos de fiabilidade que fixan a taxa de fallos para que o sistema sexa aceptable; os requisitos de portabilidade, e os requisitos de usabilidade.
- o **Requisitos organizacionais.** Estes requisitos derívanse de políticas e procedementos existentes na organización do cliente e na do desenvolvedor. Algúns exemplos son os estándares nos procesos que deben utilizarse; os requisitos de implementación, como as linguaxes de programación ou o método de deseño a utilizar, e os requisitos de entrega que especifican cando se entregará o produto e a súa documentación.
- o **Requisitos externos.** Este grande apartado inclúe todos os requisitos que se derivan dos factores externos ao sistema e do seu proceso de desenvolvemento. Estes poden incluír os requisitos de interoperabilidade que definen o xeito en que o sistema interactúa con sistemas doutras organizacións; os requisitos legislativos que deben seguirse para asegurar que o sistema funcione dentro da lei, e os requisitos éticos. Estes últimos son postos nun sistema para asegurar que será aceptado polos seus usuarios e polo público en xeral.



Sommerville distingue entre **requisitos do usuario**, que son declaracións, en linguaxe natural e en diagramas, dos servizos que se espera que o sistema proporcione e das restricións baixo as cales debe funcionar, e **requisitos do sistema**, que establecen con detalle as funcións, servizos e restricións operativas do sistema. Estes diferentes niveis de especificación serían de utilidade debido a que comunican a información do sistema a diferentes tipos de lectores.

### **33.2 Identificación dos requisitos do software**

A educación, identificación ou determinación de requisitos é o paso durante o cal os requisitos do software son obtidos de fontes tales como: a xente implicada (usuarios, clientes, expertos na materia, etc.), as necesidades que ha de satisfacer o sistema, o contorno físico que rodea o sistema, o contorno organizacional, etc.

Os problemas da obtención de requisitos pódense agrupar en tres categorías:

- 1- Problemas de alcance, xa que os requisitos poden implicar demasiada ou moi pouca información.
- 2- Problemas de comprensión, como consecuencia dunha pobre comunicación entre usuario e analista. Neste caso os requisitos obtidos son ambiguos, incompletos, inconsistentes e incorrectos, porque non responden ás verdadeiras necesidades dos usuarios.
- 3- Problemas de volatilidade, xa que os requisitos evolucionan co tempo. En efecto, a medida que avanza o desenvolvemento do sistema, as necesidades do usuario poden madurar a causa do coñecemento adicional froito do desenvolvemento, ou de necesidades do contorno ou da organización non previstas.

A solución ao primeiro problema pasa por determinar claramente o contexto do sistema, é dicir, os límites e obxectivos do mesmo. Se non se



contempla o contexto onde vai funcionar o sistema pódese chegar a requisitos incompletos, non verificables, innecesarios e non utilizables. A solución do segundo é que exista unha boa comunicación entre usuarios, desenvolvedores e clientes, a fin de que os requisitos se poidan escribir de maneira que permitan, tanto que o desenvolvidor poida distinguir se os devanditos requisitos se poden implementar, como que o persoal de control poida comprobar se a implementación cumpre cos requisitos. Para rematar a solución ao terceiro problema é incorporar estes cambios aos requisitos orixinais, pois do contrario, estes serán incompletos e inconsistentes coa nova situación.

Son numerosas as estratexias e técnicas que se desenvolveron para a obtención dos requisitos. As máis importantes verémolas a continuación.

### *33.2.1 Entrevista*

Enténdese por entrevista o encontro que se realiza “cara a cara” entre un usuario e a persoa responsable de obter a información. Para realizar a entrevista só é necesario designar ás persoas que deben participar nela e determinar o lugar no que poder levala a cabo. É importante identificar a que tipo de perfil vai dirixida a entrevista, a quen se vai a entrevistar e cal é o momento máis oportuno, co fin de evitar situacións embarazosas e conseguir que a entrevista sexa eficaz e produtiva.

Como paso previo á realización da entrevista débense ter en conta unha serie de regras xerais ou directrices básicas:

- Desenvolver un plan global da entrevista.
- Asegurarse de que se conta coa aprobación para falar cos usuarios.
- Preparar a entrevista previamente.
- Realizar a entrevista.
- Consolidar o resultado da entrevista.



Ademais, é conveniente planificar as entrevistas estudando a secuencia en que se van levar a cabo, en función dos distintos perfís implicados e as relacións existentes entre os entrevistados. Segundo a información a obter e dependendo das distintas fontes que poden proporcionala, poida que cumpra realizar unha entrevista conxunta con varias persoas. Durante a preparación da entrevista é imprescindible remitirlle ao usuario un guión previo sobre os puntos a tratar, para que poida estudalo con tempo e solicitar a información que estime conveniente para a entrevista. Débese pensar ben o tipo de guión, segundo o perfil e as responsabilidades do entrevistado e a súa extensión, de forma que se poida conseguir a suficiente información, sen provocar rexeitamento no entrevistado. Se se considera apropiado, pódense utilizar ferramentas automatizadas.

Unha vez que se dispón da aprobación para falar cos usuarios, faise a convocatoria da entrevista enviando a información oportuna e fixando os obxectivos, o método de traballo que se vai seguir e o tempo do que se dispón.

Para realizar a entrevista, é importante facer un resumo xeral dos temas a tratar, utilizar un estilo apropiado e crear desde o seu inicio un clima de confianza entre os asistentes. É posible que o entrevistado se resista a facilitar información, sendo útil nestes casos utilizar técnicas específicas de comunicación.

Antes de finalizar a entrevista é importante que o entrevistador sintetice as conclusións e comprobe que todos os asistentes están de acordo, deixando sempre aberta a posibilidade de volver contactar para aclarar temas que xurdan ao estudar a información recompilada.

Finalmente, o responsable depura e consolida o resultado das entrevistas, elaborando un informe de conclusións. Nalgúns casos pode ser conveniente elaborar un acta que reflicta estas conclusións e remitirla aos



entrevistados co obxectivo de asegurar que se comprenderon ben as especificacións dadas.

### *33.2.2 JAD (Joint Application Design)*

As características dunha sesión de traballo tipo JAD pódense resumir nos seguintes puntos:

- Establécese un equipo de traballo cuxos compoñentes e responsabilidades están perfectamente identificados e o seu fin é conseguir o consenso entre as necesidades dos usuarios e os servizos do sistema en produción.
- Lévanse a cabo poucas reunións, de longa duración e moi ben preparadas.
- Durante a propia sesión elabóranse os modelos empregando diagramas fáciles de entender e manter, directamente sobre ferramentas CASE.
- Ao finalizar a sesión obtéñense un conxunto de modelos que deberán ser aprobados polos participantes.

É importante definir claramente o perfil e as responsabilidades dos participantes dunha sesión JAD. Pódense distinguir os seguintes perfís:

- Moderador (líder) con amplos coñecementos da metodoloxía de traballo, dinámica de grupos, psicoloxía do comportamento, así como dos procesos da organización obxecto do estudo.
- Promotor, persoa que impulsou o desenvolvemento.
- Xefe de proxecto, responsable da implantación do proxecto.
- Especialista en modelización, responsable da elaboración dos modelos no transcurso da sesión.
- Desenvolvedores, aseguran que os modelos son correctos e responden aos requisitos especificados.



- Usuarios, responsables de definir os requisitos do sistema e validalos.

Para levar a cabo unha sesión JAD, é necesario realizar unha serie de actividades antes do seu inicio, durante o desenvolvemento e logo da súa finalización. Estas actividades detállanse a continuación:

- **Inicio:** defínese o ámbito e a estrutura do proxecto, os produtos a obter, prepárase o material necesario para a sesión, determínase o lugar onde se van levar a cabo, selecciónanse os participantes e suxírese unha axenda de traballo.
- **Desenvolvemento:** identifícanse as saídas do proxecto e débese conseguir o consenso entre os participantes de modo que se materialice nos modelos.
- **Finalización:** valídase a información da sesión e xéranse os produtos da metodoloxía de traballo proposta. Se fose necesario intégranse os produtos de saída.

### *33.2.3 Prototipado*

Os prototipos son sistemas software que só implementan unha parte do sistema. Normalmente, os prototipos empréganse para obter requisitos do usuario cando estes non están completamente claros. É moito máis fácil que o usuario entenda e aprobe un sistema, que aínda que é reducido pero tamén é operativo e pode interaccionar con el, a revisar unha longa lista de texto cos requisitos que describen o devandito sistema. Despois de todo, como se adoita dicir, o prototipo permite salvar a situación seguinte: “Non sei o que quero, pero heino saber cando o vexa”. E en efecto, será moito máis fácil para un usuario saber se o que quere é o que ten diante e que pode interaccionar con el, que se é unha chea de follas con todos os requisitos escritos un a un.

A aplicación da técnica do prototipado consta dos seguintes pasos:



- 1- Estudo preliminar dos requisitos do usuario.
- 2- Proceso iterativo consistente en:
  - a. Construír un prototipo.
  - b. Avalialo cos usuarios.
  - c. Formular novos requisitos.
  - d. Desbotar o prototipo.

Os aspectos clave no deseño de prototipos son: a identificación dos usuarios aos que vai dirixido, tendo en conta que debe responder a diferentes individualidades, con distintos coñecementos e habilidades, qué funcións teñen asignadas, e que tipo de información requirirán para levar a cabo ditas funcións.

#### *33.2.4 Análise de factores críticos de éxito.*

Esta técnica consistente en identificar e concentrarse nun pequeno conxunto de factores críticos dos que depende o éxito e efectividade do sistema. Aínda que a identificación dos factores críticos correctos é, ás veces, un labor complexo, esta técnica é bastante útil cando o sistema é tecnicamente complexo.

#### *33.2.5 Brainstorming*

Esta técnica úsase principalmente para a xeración de ideas nos casos nos que a xeración destas non é obvia. Trátase dunha técnica sinxela na que se reúnen en grupo de 4 a 10 persoas para xerar ideas, sen restricións, nun ambiente libre de críticas. Un dos principais puntos fortes da técnica é que ideas que nun principio poden parecer absurdas, tamén teñen cabida. Primeiro propóñanse as ideas e nunha segunda volta refínanse. O Brainstorming permite achegar á obtención de requisitos os seguintes aspectos:



- **Xera múltiples puntos de vista dun problema.** Cada un dá a súa visión do problema, que non ten porqué ser a mesma. Unha vez vistos todos os puntos de vista pódese atacar o problema dunha forma máis efectiva.
- **Formular un problema de distintas formas.** Cada participante pode ver o problema desde unha óptica distinta, polo que á hora de formulalo o fará dun xeito distinto. Estas diferenzas poden achegar gran valor á hora do estudo do problema, identificando os puntos clave do problema de cada participante.

A técnica de Brainstorming, para que sexa efectiva, debe ser previamente preparada. As fases polas que pasa a execución da técnica son:

- 1- **Fase de Preparación:** Identifícanse os participantes da sesión (clientes, usuarios, analistas, etc.), désígnase un líder que leva a sesión, planifícase a sesión e búscase unha sala adecuada.
- 2- **Fase de Xeración:** Comézase expoñendo as ideas de forma libre, por quendas ou espontaneamente. As ideas vanse apuntando nun encerado que todos poidan ver.
- 3- **Fase de Consolidación:** Revísanse as ideas obtidas no paso anterior, clarifícanse se non están claras, reescríbense se é necesario, descártanse as que non son utilizables, discútense as restantes e priorízanse.

### *33.2.6 Escenarios e Casos de uso*

Normalmente, as persoas atopan máis fácil dar exemplos da vida real que descrições abstractas. Os **escenarios** poden ser especialmente útiles para agregar detalles a un esbozo da descripción de requisitos. Son descrições de exemplos das sesións de interacción. Cada escenario abarca unha ou máis posibles interaccións. O escenario comeza cun esbozo da interacción



e, durante a obtención, agréganse detalles para crear unha descrición completa desta interacción. De forma xeral, un escenario pode incluír:

- 1- Unha descrición do que esperan o sistema e os usuarios cando o escenario comeza.
- 2- Unha descrición do fluxo normal de eventos no escenario.
- 3- Unha descrición do que pode ir mal e como manexalo.
- 4- Información doutras actividades que se poderían levar a cabo ao mesmo tempo.
- 5- Unha descrición do estado do sistema cando o escenario termina.

Os escenarios pódense redactar como texto, complementados por diagramas, fotografías das pantallas, etcétera. De forma alternativa, pódese adoptar un enfoque máis estruturado, coma os escenarios de evento ou os casos de uso.

Os casos de uso son unha técnica que se basea en escenarios para a obtención de requisitos. Na súa forma máis simple, un caso de uso identifica o tipo de interacción e os actores involucrados. Os actores no proceso represéntanse como figuras delineadas, e cada clase de interacción represéntase como unha elipse co seu nome. O conxunto de casos de uso representa todas as posibles interaccións a representar nos requisitos do sistema. Actualmente convertéronse nunha característica fundamental da notación de UML, que se utiliza para describir modelos de sistemas orientados a obxectos.

Os escenarios e os casos de uso son técnicas eficaces para obter requisitos para os puntos de vista dos interactuadores, onde cada tipo de interacción se pode representar como un caso de uso. Con todo, debido a que se centran nas interaccións, non son tan eficaces para obter restricións e requisitos de negocio e non funcionais de alto nivel de puntos de vista indirectos ou para descubrir requisitos do dominio.



### *33.2.7 Observación directa e investigación contextual (Etnografía)*

A **etnografía** é unha técnica de observación que se pode utilizar para entender os requisitos sociais e organizacionais. Un analista mergúllase por si só no contorno laboral onde se utilizará o sistema. Observa o traballo diario e anota as tarefas reais nas que os participantes están involucrados. O valor da etnografía é que axuda aos analistas a descubrir os requisitos implícitos que reflicten os procesos reais máis que os formais nos que a xente está involucrada.

A etnografía é especialmente efectiva para descubrir dous tipos de requisitos:

- Os requisitos que se derivan da forma na que a xente traballa realmente máis que da forma na que as definicións dos procesos establecen que debería traballar.
- Os requisitos que se derivan da cooperación e coñecemento das actividades dos demais.

Os estudos etnográficos poden revelar os detalles dos procesos críticos que outras técnicas de obtención de requisitos a miúdo esquecen. Con todo, posto que se centran no usuario final, este enfoque non é apropiado para descubrir os requisitos organizacionais ou do dominio. Os estudos etnográficos non sempre poden identificar novas propiedades que se deban agregar ao sistema. Polo tanto, a etnografía non é un enfoque completo para a obtención de requisitos por si mesmo, e debe utilizarse para complementar outros enfoques, como a análise de casos de uso.

## **33.3 Verificación - Validación**

A **validación** de requisitos trata de mostrar que estes realmente definen o sistema que o cliente desexa. A validación de requisitos é importante debido a que os erros nos requisitos poden conducir a importantes custos



ao repetir o traballo cando son descubertos durante o desenvolvemento ou despois de que o sistema estea en uso. O custo de arranxar un problema nos requisitos facendo un cambio no sistema é moito maior que reparar os erros de deseño ou os de codificación.

A razón disto é que un cambio nos requisitos normalmente significa que o deseño e a implementación do sistema tamén deben cambiar e que este debe probarse novamente.

Durante o proceso de validación de requisitos, débense levar a cabo as seguintes **verificacións**:

- 1- **Verificacións de validez.** Un usuario pode pensar que se necesita un sistema para levar a cabo certas funcións. Con todo, o razoamento e a análise poden identificar que se requiren funcións adicionais ou diferentes. Tamén se deben ter en conta que nun mesmo sistema adoita haber diferentes usuarios, con diferentes puntos de vista, algunhas veces contrapostos, e que estes deben chegar a un compromiso á hora de definir os requisitos do sistema.
- 2- **Verificacións de consistencia.** Os requisitos non deben contradicirse. Isto é, non debe haber restricións ou descricións contraditorias da mesma función do sistema.
- 3- **Verificacións de completitude.** Os requisitos deben definir todas as funcións e restricións propostas polo usuario do sistema.
- 4- **Verificacións de realismo.** Utilizando o coñecemento da tecnoloxía existente, os requisitos deben verificarse para asegurar que se poden implementar. Estas verificacións tamén deben ter en conta o orzamento e a confección de axendas para o desenvolvemento do sistema.
- 5- **Verificabilidade.** Para reducir a posibilidade de discusións entre o cliente e o desenvolvedor, os requisitos do sistema sempre deben redactarse de tal forma que sexan verificables. Isto significa que



debe poder escribir un conxunto de probas que demostren que o sistema a entregar cumpre cada un dos requisitos especificados.

Poden utilizarse, en conxunto ou de forma individual, varias técnicas de validación de requisitos:

- 1- **Revisións de requisitos.** Os requisitos son analizados manual e sistematicamente por un equipo de revisores formado por persoas tanto da organización do cliente como da desenvolvedora. Verifícanse os requisitos en canto a anomalías e omisións. Poden ser informais ou formais. As informais sinxelamente implican que os desenvolvedores deben tratar os requisitos con tantos usuarios do sistema como sexa posible. Na revisión formal de requisitos, o equipo de desenvolvemento debe «conducir» ao cliente a través dos requisitos do sistema, explicándolle as implicacións de cada requisito. Os revisores deben comprobar a:
  - a. **Verificabilidade.** Pode probarse o requisito de modo realista?
  - b. **Comprensibilidade.** As persoas que adquiren o sistema ou os usuarios finais comprenden correctamente o requisito?
  - c. **Rastrexabilidade.** Está claramente establecida a orixe do requisito? Pode ter que volver á fonte do requisito para avaliar o impacto do cambio. A rastrexabilidade é importante xa que permite avaliar o impacto do cambio no resto do sistema.
  - d. **Adaptabilidade.** É adaptable o requisito? É dicir, pode cambiarse o requisito sen causar efectos de grande escala nos outros requisitos do sistema?

Os conflitos, contradicións, erros e omisións nos requisitos deben ser sinalados polos revisores e rexistrarse formalmente no informe de revisión. Tócalles aos usuarios, á persoa que adquire o sistema e ao desenvolvedor deste negociar unha solución para estes problemas identificados.



- 2- **Construción de prototipos** Neste enfoque de validación, móstrase un modelo executable do sistema aos usuarios finais e aos clientes. Estes poden experimentar con este modelo para ver se cumpre as súas necesidades reais.
- 3- **Xeración de casos de proba.** Os requisitos deben poder probarse. Se as probas para estes se conciben como parte do proceso de validación, a miúdo revela os problemas nos requisitos. Se unha proba é difícil ou imposible de deseñar, normalmente significa que os requisitos serán difíciles de implementar e deberían ser considerados novamente.

É difícil demostrar que un conxunto de requisitos cumpre as necesidades do usuario. Como consecuencia, de cando en cando atópanse todos os problemas nos requisitos durante o proceso de validación destes. É inevitable que haxa cambios adicionais de requisitos para corrixir as omisións e as malas interpretacións despois de que o documento de requisitos sexa aprobado.

### **33.4 Especificación de requisitos**

O documento **de requisitos do software** (algunhas veces denominado *especificación de requisitos do software* ou **ERS**) é a declaración oficial de que deben implementar os desenvolvedores do sistema. Ten un conxunto diverso de usuarios que vai desde os altos cargos da organización que pagan polo sistema, ata os enxeñeiros responsables de desenvolver o software. Os obxectivos que pretende o ERS son os seguintes:

- Proporcionar os medios de comunicación entre todas as partes implicadas no sistema: clientes, usuarios, analistas e deseñadores.
- Servir como base para as actividades de proba e verificación.
- Axudar ao control da evolución do sistema software.



O documento ERS debe incluír unha descrición completa e concisa de toda a interface externa do sistema co seu contorno, incluído o resto do software, portos de comunicación, hardware e usuarios. É dicir, debe incluír tanto os requisitos de comportamento do sistema (funcionais), que son aqueles que definen o que fai este e a información que manexa, como os requisitos que non son de comportamento, isto é, aqueles que definen os atributos do sistema segundo realiza o seu traballo (eficiencia, fiabilidade, seguridade, etc.).

Por contra, un documento ERS non debe incluír os elementos de xestión do proxecto (planificacións, fitos, etc.), nin o deseño, nin os plans de control do produto (xestión de configuración, garantía de calidade, etc.). Un documento ERS debe reunir as seguintes características:

- **Correcto.** Cada requisito establecido debe representar algo requirido para o sistema.
- **Non Ambiguo.** Cada requisito establecido ten unha soa interpretación.
- **Completo.** Debe incluír todo o que o software ten que facer.
- **Verificable.** Deberase poder comprobar, mediante un proceso efectivo e de custo limitado, que o produto reúne cada requisito establecido.
- **Consistente.** Cada requisito non pode estar en conflito con outros requisitos.
- **Modificable.** A estrutura e estilo do documento debe facer fáciles os cambios.
- **Conciso,** comprensible polo usuario e organizado.
- **Referenciado.** Cada requisito debe estar cualificado e debidamente referenciado.

#### 33.4.1 IEEE/ANSI 830-1998



Varias organizacións grandes, como o Departamento de Defensa dos Estados Unidos e o IEEE, definiron estándares para os documentos de requisitos. O estándar máis amplamente coñecido é o IEEE/ANSI 830-1998 (IEEE, 1998). Este estándar IEEE suxire a seguinte estrutura para os documentos de requisitos:

## **1. Introducción**

- 1.1 Propósito do documento de requisitos: Propósito e a quen vai dirixido.
- 1.2 Alcance do produto
- 1.3 Definicións, acrónicos e abreviaturas
- 1.4 Referencias
- 1.5 Descrición do resto do documento

## **2. Descrición xeral**

- 2.1 Perspectiva do produto
- 2.2 Funcións do produto
- 2.3 Características do usuario
- 2.4 Restricións xerais
- 2.5 Suposicións e dependencias

## **3. Requisitos específicos.** Este será o groso do documento.

**3.1. Interfaces Externas:** Describíranse os requisitos que afecten á interface de usuario, interface con outros sistemas (hardware e software) e interfaces de comunicacións.

**3.2. Funcións:** Esta subsección (quizá a máis longa do documento) deberá especificar todas aquelas accións (funcións) que deberá levar a cabo o software. Se se considera necesario, poderán utilizarse notacións gráficas e táboas, pero sempre supeditadas á linguaxe natural, e non ao revés. O estándar permite organizar esta subsección de múltiples formas, e suxire, entre outras, as seguintes:

- **Por tipos de usuario:** Distintos usuarios posúen distintos requisitos. Para cada clase de usuario que exista na



organización, especificaranse os requisitos funcionais que lle afecten ou teñan maior relación coas súas tarefas.

- **Por obxectos:** Os obxectos son entidades do mundo real que serán reflectidas no sistema. Para cada obxecto, detallaranse os seus atributos e as súas funcións. Os obxectos poden agruparse en clases. Esta organización da ERS non quere dicir que o deseño do sistema siga o paradigma de Orientación a Obxectos.
- **Por obxectivos:** Un obxectivo é un servizo que se desexa que ofrezca o sistema e que require unha determinada entrada para obter a súa resultado. Para cada obxectivo ou subobxectivo que se persiga co sistema, hanse detallar as funcións que permitan levalo a cabo.
- **Por estímulos:** Especificaranse os posibles estímulos que recibe o sistema e as funcións relacionadas co devandito estímulo.
- **Por xerarquía funcional:** Se ningunha das anteriores alternativas resulta de axuda, a funcionalidade do sistema especificarase como unha xerarquía de funcións que comparten entradas, saídas ou datos internos. Detallaranse as funcións (entrada, proceso, saída) e as subfuncións do sistema. Isto non implica que o deseño do sistema deba realizarse segundo o paradigma de Deseño Estruturado.

**3.3. Requisitos de Rendemento:** Detallaríanse os requisitos relacionados coa carga que se espera teña que soportar o sistema. Por exemplo, o número de terminais, o número esperado de usuarios simultaneamente conectados, número de transaccións por segundo que deberá soportar o sistema, etc. Tamén, se é necesario, especificaranse o requisitos de datos, é dicir, aqueles requisitos que afecten á información que se gardará na base de datos. Por exemplo, a frecuencia de uso, as



capacidades de acceso e a cantidade de rexistros que se espera almacenar (decenas, centos, miles ou millóns).

**3.4. Restricións de Deseño:** Todo aquilo que restrinja as decisións relativas ao deseño da aplicación: Restricións doutros estándares, limitacións do hardware, etc.

**3.5. Atributos do Sistema:** Detallaranse os atributos de calidade do sistema: Fiabilidade, mantenibilidade, portabilidade, e, moi importante, a seguridade. Debería especificarse que tipos de usuario están autorizados, ou non, a realizar certas tarefas, e como se implementarán os mecanismos de seguridade

**3.6. Outros Requisitos**

**4. Apéndices**

**5. Índice**

O estándar IEEE é un marco xeral que se pode transformar e adaptar para definir un estándar axustado ás necesidades dunha organización en particular. A información que se inclúa nun documento de requisitos debe depender do tipo de software a desenvolver e do enfoque de desenvolvemento que se utilice.

O documento de requisitos é fundamental cando un desenvolvedor exterior está construíndo o sistema software. Con todo, os métodos de desenvolvemento áxiles sosteñen que os requisitos cambian tan rapidamente que un documento de requisitos queda desfasado en canto se redacta, polo que o esforzo en gran parte, malgástase.

### **33. 5 Xestión de requisitos**

Os requisitos para sistemas software grandes son sempre cambiantes. Debido a que o problema non pode definirse completamente, é moi probable que os requisitos do software sexan incompletos. Durante o proceso do software, a comprensión do problema por parte dos clientes



cambia, e os requisitos deben, daquela, evolucionar para reflectir isto. Ademais, unha vez que un sistema se instalou, inevitablemente xorden novos requisitos. Cando os usuarios finais traballan cun sistema, descubren novas necesidades e prioridades debidas a:

- Normalmente, os sistemas grandes dispoñen dunha comunidade de usuarios diversa onde os usuarios teñen diferentes requisitos e prioridades. Os que se usaron para definir o sistema puideron non ser os mellores.
- As persoas que pagan polo sistema e os usuarios deste raramente son a mesma persoa. Os clientes do sistema impoñen requisitos debido ás restricións organizacionais e de orzamento. Estes poden estar en conflito cos requisitos dos usuarios finais e, logo da entrega, poden ter que engadirse novas características de apoio ao usuario se o sistema ten que cumprir os seus obxectivos.
- O contorno de negocios e técnico do sistema cambia logo da instalación, e estes cambios débense reflectir no sistema. Pódese introducir novo hardware, pode ser necesario que o sistema interactúe con outros sistemas, as prioridades de negocio poden cambiar con modificacións consecuentes na axuda ao sistema, e pode haber unha nova lexislación e regulacións que deben ser implementadas polo sistema.

A **xestión de requisitos** é o proceso de comprender e controlar os cambios nos requisitos do sistema. É necesario poder avaliar o impacto dos cambios nos requisitos. Hai que establecer un proceso formal para implementar as propostas de cambios. O proceso de xestión de requisitos debería empezar en canto estea dispoñible unha versión preliminar do documento de requisitos, pero debería empezar a planificar cómo xestionar os requisitos que cambian durante o proceso de obtención de requisitos.



Existen **requisitos duradeiros**, que son requisitos relativamente estables que se derivan da actividade principal da organización e que están relacionados directamente co dominio do sistema. Estes requisitos pódense derivar dos modelos do dominio que mostran as entidades e relacións que caracterizan un dominio de aplicación. Doutra banda, existen **requisitos volátiles**, que son requisitos que probablemente cambian durante o proceso de desenvolvemento do sistema ou despois de que este se puxo en funcionamento, debido a cambios da contorna, lexislación,...

#### *33.5.1 Planificación da xestión requisitos.*

Para cada proxecto, a etapa de planificación establece o nivel de detalle necesario na xestión de requisitos. Haberá que decidir sobre:

1. **A identificación de requisitos.** Cada requisito débese identificar de forma única de tal forma que poidan ser remitidos por outros requisitos, de xeito que poida utilizarse nas avaliacións de rastrexo.
2. **Un proceso de xestión do cambio.** Este é o conxunto de actividades que avalían o impacto e custo dos cambios. Verémolo nun apartado posterior.
3. **Políticas de rastrexo ou trazabilidade.** Estas políticas definen as relacións entre os requisitos, e entre estes e o deseño do sistema que se debe rexistrar e o xeito en que estes rexistros débense manter.
4. **Selección de ferramentas CASE.** A xestión de requisitos comprende o procesamento de grandes cantidades de información sobre os requisitos. As ferramentas que se poden utilizar van desde sistemas de xestión de requisitos especializados ata follas de cálculo e sistemas sinxelos de bases de datos.

O concepto de trazabilidade fai referencia á posibilidade de determinar como se chegou a un certo elemento do software a partir doutros. Para poder levar a cabo esta trazabilidade é especialmente importante o feito de



poder relacionar uns requisitos con outros e tamén relacionar requisitos con elementos do sistema aos que dá lugar. Fálase de trazabilidade *cara adiante* cando, partindo dun requisito, se chega a todos os elementos que materializan o devandito requisito, ou *cara atrás*, cando partindo dun elemento do sistema, se chega ao requisito que o xerou.

Un requisito débese poder relacionar con outros requisitos similares para así evitar repeticións nos mesmos. Ademais, cando se fai un cambio nun, é máis fácil localizar aqueles requisitos nos que hai relación, para ver o impacto do cambio. Doutra banda, é útil poder relacionar un requisito cos elementos do sistema aos que este dá lugar, por se hai que cambiar un elemento do sistema, ou simplemente, para poder saber que requisitos deron lugar a un determinado compoñente do sistema. Esta relación non só debe ser entre requisitos, senón tamén entre calquera elemento que se derivou posteriormente, xa sexa da análise, do deseño, da construción, de probas, etc.

A xestión de requisitos necesita axuda automatizada. As ferramentas CASE para isto deben elixirse durante a fase de planificación. Precísanse ferramentas de axuda para:

1. **Almacenar requisitos.** Os requisitos deben manterse nun almacén de datos seguro e administrado que sexa accesible a todos os que estean implicados no proceso de enxeñería de requisitos.
2. **Xestionar o cambio.**
3. **Xestionar o rastrexo.** As ferramentas de axuda para o rastrexo permiten que se descubran requisitos relacionados. Algunhas ferramentas utilizan técnicas de procesamento da linguaxe natural para axudarlle a descubrir posibles relacións entre os requisitos.

Para sistemas pequenos, é posible que non sexa necesario utilizar ferramentas de xestión de requisitos especializadas, pero si é moi conveniente para sistemas grandes.



### 33.5.2 Xestión do cambio

A vantaxe de utilizar un proceso formal para xestionar o cambio é que todos os cambios propostos son tratados de forma consistente e que os cambios nos requisitos fanse de forma controlada. Existen varias etapas principais nun proceso de xestión de cambio:

- **Proposta de Cambios.** O afectado por un requisito que non lle satisfai debe reencher un formulario de proposta de cambio indicando cal é o cambio que cómpre realizar. Este cambio remitirase ao equipo de xestión de requisitos para que o estude.
- **Análise de Impactos.** A proposta de cambio avalíase para determinar o impacto do cambio no resto dos requisitos. Hai propostas cuxo impacto é mínimo e outras cuxo impacto fai que se deba modificar unha parte substancial do sistema. Ademais do impacto, deberase estudar a oportunidade ou necesidade do cambio. Ocorre moitas veces que a un usuario non lle parece ben un determinado aspecto do sistema pero a outros si lle satisfai. Nestes casos débese chegar a un consenso entre todas as partes. Outras veces o cambio é interesante pero considérase que non é oportuno facelo no devandito momento, quizais porque se queira lanzar o sistema canto antes ou porque o cambio é considerable e sería mellor atacalo nunha fase posterior de desenvolvemento.
- **Toma de Decisións.** Dependendo do impacto, da necesidade ou oportunidade do cambio e doutras cuestións específicas que poidan xurdir ao redor do cambio, débese determinar se o cambio debe ser realizado ou non. Se se decide realizalo débense facer as modificacións pertinentes no sistema para integrar o cambio co resto, empezando polos requisitos e rematando polas partes máis avanzadas do sistema. Se pola contra, se decide non facelo pódese optar por dúas solucións:
  - o Rexeitar o cambio, porque se considera que non ten sentido.



- o Pospoñer a súa realización para un futuro considérase o cambio necesario, pero o momento non é oportuno. Posponse para cando se poida acometer.
- **Comunicación.** Tanto se se acepta como se non, débese notificar os efectos da proposta de cambio a todos os afectados.
  - **Incorporación.** Fanse as modificacións pertinentes que se identificaron na análise de impacto nos diferentes elementos afectados polo cambio.
  - **Medición da Estabilidade dos Requisitos.** Avalíanse os parámetros que definen a estabilidade dos requisitos tralas modificacións que se realizaron.

### **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 segundo o estándar de IEEE 830

<http://www.fdi.ucm.es/profesor/gmendez/docs/is0809/ieee830.pdf>

**Autor: Hernán Vila Pérez**

**Xefe do Servizo de Informática. Instituto Galego de Vivenda e Solo**

**Vicepresidente do CPEIG**





**34. TÉCNICAS DE  
PROGRAMACIÓN.  
PROGRAMACIÓN  
ESTRUTURADA.  
PROGRAMACIÓN ORIENTADA A  
OBJECTOS. ENXEÑARÍA  
INVERSA E REENXEÑARÍA.**



Tema 34. Técnicas de programación. Programación Estructurada.  
Programación orientada a obxectos. Enxeñaría inversa e reenxeñaría.

## ÍNDICE

- 34.1 Técnicas de programación
  - 34.1.1 Clasificación e evolución das linguaxes de programación
- 34.2 Programación estruturada
  - 34.2.1 Recursos abstractos
  - 34.2.2 Deseño descendente
  - 34.2.3 Estructuras básicas
- 34.3 Programación orientada a obxectos
- 34.4 Enxeñaría inversa
  - 34.4.1 Modelo cíclico
  - 34.4.2 Modelo de ferradura
  - 34.4.3 Modelo do IEEE
  - 34.4.4 Método Análise de Opcións para Reenxeñaría (OAR)

### 34.1 Técnicas de programación

Unha **linguaxe de programación** é unha linguaxe artificial deseñada para representar expresións e instrucións de xeito que as entenda un ordenador. Inda que máis sinxelas que as linguaxes naturais, teñen un alfabeto (símbolos básicos) co que se constrúe o vocabulario (*tokens*, palabras reservadas), que se combina segundo unhas regras sintácticas formando expresións e sentenzas, cuxo significado vén dado pola semántica da linguaxe. Chámasele **programa** ao conxunto de instrucións escritas nunha linguaxe de programación, destinadas a realizar unha tarefa. De forma xeral, tomará uns datos de entrada e devolverá uns datos de saída. Dise que o que fai o programa é poñer en práctica un **algoritmo**, que é un conxunto finito de instrucións que se deben seguir para realizar unha determinada tarefa.

O deseño dun programa que se realiza sen seguir unha metodoloxía pode funcionar, pero cómpre ter en conta que co tempo converterase nun conxunto de instrucións difíciles de tocar e manter. Isto fará que os programas sexan difíciles de adaptar a unha nova configuración, que se



perda moito tempo coa corrección de erros, que sexan difíciles de entender para outros programadores, etc.

É de suma importancia poder previr as modificacións que se poidan realizar no futuro, así como tamén ter a documentación sempre actualizada. A creación de programas debe ter a flexibilidade suficiente para seren modificables no momento en que se requira. Estes deben ser claros, simples, co fin de poder ser lidos e interpretados de forma fácil. Para a programación deberanse asumir certas normas que permitan a estandarización, o que implica unha diminución en custos, independencia do programador e seguridade.

Entendemos pois a **programación** como *a planificación, proxección, desenvolvemento e posta en práctica da resolución dun problema*, a que abarca obviamente a creación do algoritmo. Un profesional na programación debe encarar a solución do problema de xeito que o seu produto sexa útil agora e no futuro, estando ou non el no centro de desenvolvemento. Para lograr isto debe ter moi presentes as futuras posibles modificacións deste.

As características, entón, que debe ter un programa son:

- 1) **Claridade algorítmica:** Que sexa claro significa que a súa resolución algorítmica sexa sinxela, que estea correctamente estruturado, resultando de fácil comprensión.
- 2) **Lexibilidade:** Que sexa lexible significa que, cando se codificou, se elixiron ben os nomes dos obxectos utilizados, se agregaron comentarios para indicar o que se ía facendo e se estruturou ben o código para resaltar o contido semántico sobre o sintáctico.
- 3) **Modificabilidade:** Que sexa facilmente modificable implica que calquera modificación do problema que xere un engadido, supresión ou cambio dalgunha das súas partes, non debe obrigar a cambiar todo o programa, senón só unha parte.



Nos seus inicios a programación non seguía ningunha metodoloxía, e tiñamos centos ou miles de liñas de código, feitas por un programador, e que só el se atrevía a tocar. Nesta época os profesionais programadores adoitaban estar atados á empresa e non era frecuente que a abandonasen. Os programas non tiñan estrutura definida. As modificacións no programa supoñían un custo importante, xa que había que revisar o código case liña a liña para buscar onde facer as modificacións. Ademais, estes cambios non deixaban de ser un risco importante debido aos posibles efectos colaterais. Isto acabou no que se denominou crise do software.

Para tratar de lle dar resposta á crise do software, nos anos sesenta xurdiron técnicas de programación como a **programación modular** e a **programación estruturada**. Ambas as técnicas parten da idea do deseño descendente (tamén chamado refinamento sucesivo), que consiste en dividir un problema complexo en varios problemas menores, máis sinxelos de resolver. Comeza formulando o problema e a súa solución a un nivel alto, superior, e logo séguese descompoñendo o problema xeral noutros problemas máis sinxelos de resolver. Este proceso continúa ata que chegamos a pequenos problemas, facilmente implementables en código nos chamados módulos. Un módulo é un conxunto de accións (un bloque do código do algoritmo total) xunto a un conxunto de especificacións de datos para realizar unha tarefa específica. Cando se utiliza esta técnica, a solución queda dividida en varias partes: un algoritmo principal e un ou varios subalgoritmos (os módulos). A execución iníciase no algoritmo principal e desde este invócanse os módulos. A programación estruturada fai uso da programación modular, ademais doutras características que se verán nun apartado posterior.

Tamén como resposta á crise do software xurdiron nos anos sesenta os conceptos da orientación a obxectos, que tiñan como principal obxectivo reducir a complexidade do desenvolvemento e mantemento do software. Dado que a tecnoloxía non estaba acorde coa súa implantación, mantívose



só como concepto ata a súa grande expansión a finais dos oitenta. A **programación orientada a obxectos** non supón unha ruptura radical fronte ao paradigma da programación estruturada / imperativa predominante ata a súa aparición, senón que supón unha evolución. Fronte á programación estruturada, cuxos programas separan datos e estruturas de datos de funcións e procedementos, a programación orientada a obxectos encapsula nunha mesma abstracción estes dous elementos clásicos dos sistemas de información: os datos e os procesos. Isto faino a través dunha nova abstracción: o obxecto. A orientación a obxectos consiste nunha visión dos obxectos como entidades activas que executan accións sobre os seus propios datos en resposta a peticións externas. Non considera uns e outros (datos e procesos) como realidades illadas susceptibles de se analizar e implantar por separado. Os obxectos tratan de abstraer características comúns que se poderán compartir entre varias aplicacións e reutilizar todo o posible. A creación dun novo sistema consiste esencialmente nun labor de ensamblaxe de obxectos preexistentes, completado co desenvolvemento dunha porcentaxe reducida de novos obxectos, que, pola súa vez, alimentasen as correspondentes librerías para poder ser utilizados nos vindeiros sistemas.

O paradigma de ensamblar compoñentes e escribir código para facer que estes compoñentes funcionen coñécese como **Desenvolvemento de Software Baseado en Compoñentes**. Un compoñente é unha peza de código preelaborado que encapsula algunha funcionalidade exposta a través de interfaces estándar. Cada compoñente é deseñado para axustarse perfectamente cos seus pares, as conexións son estándar e o protocolo de comunicación está xa preestablecido. Ao unírense as partes, obtemos un sistema completo.

Para rematar, a **programación orientada a aspectos** xorde para resolver o problema da "dispersión de código" existente na programación orientada a obxectos para aqueles aspectos dun programa que non teñen



que ver directamente co dominio de negocio do problema: xestión de conexións, logs, rastros de mensaxes, sincronización e concorrencia, manexo de erros e excepcións, etc. Este código está disperso ao longo de diferentes obxectos do programa (sendo boa parte das veces o mesmo código repetido), o que dificulta o seu mantemento. Esta é unha nova metodoloxía de programación que intenta separar os compoñentes e os aspectos uns dos outros, proporcionando mecanismos que fagan posible abstraelos e compoñelos para formar todo o sistema. É un desenvolvemento que segue o paradigma da orientación a obxectos e, como tal, soporta a descomposición orientada a obxectos, ademais da procedemental e a descomposición funcional. Pero, malia isto, non se pode considerar como unha extensión da POO, xa que se pode utilizar cos diferentes estilos de programación xa mencionados.

#### *34.1.1 Clasificación e evolución das linguaxes de programación*

Para explicar a evolución das linguaxes de programación falamos de xeracións:

- **Primeira xeración.** Linguaxe máquina. Os programas están feitos de secuencias de uns e ceros que o computador é capaz de interpretar. É o único que é directamente entendible polo ordenador, sen necesidade de tradución.
- **Segunda xeración.** Ensamblador. Úsanse mnemotécnicos que substitúen os uns e ceros. Seguen sendo dependentes da máquina. O código fonte é traducido á linguaxe máquina mediante tradutores.
- **Terceira xeración.** Linguaxes de alto nivel. Linguaxes estruturadas con comandos próximos á linguaxe natural. Foron creados para facilitar o proceso de programación. Os comandos aseméllanse a palabras de uso común.
- **Cuarta xeración.** Nacen as linguaxes 4G. Son linguaxes de propósito especial, orientados a resolver problemas específicos, como xeración



de informes, pantallas, consultas de bases de datos... Caracterízanse por ter unha maior facilidade de uso en comparación cos de terceira xeración, permitindo a creación de prototipos rapidamente. Xeran o código fonte automaticamente a través de asistentes, modelos, etc.

- **Quinta xeración.** Linguaxes naturais. Son linguaxes orientadas á intelixencia artificial e aos sistemas expertos. En lugar de só executar ordes, o obxectivo dos sistemas é anticipar as necesidades dos usuarios. Están aínda en desenvolvemento.

As linguaxes de programación pódense clasificar segundo moitos criterios:

- Segundo o **nivel de abstracción**:
  - o **Linguaxes máquina e de baixo nivel**: a linguaxe ou código máquina está formada por cadeas binarias entendibles directamente pola máquina. Non necesitan tradución e son moi rápidas. A linguaxe ensambladora permite polo menos escribir as instrucións utilizando unha notación simbólica, utilizar enderezos de memoria relativos e non absolutos, inserir comentarios. Necesitan dun tradutor para converter a código máquina, tradución que resulta case trivial debido ao seu baixo nivel. Estas linguaxes son completamente dependentes da máquina.
  - o **Linguaxes de medio nivel**: aquí atoparíase a linguaxe C. A linguaxe C permite un manexo abstracto e independente da máquina (o que a achegaría ao alto nivel), pero sen perder a súa característica de potencia, eficiencia e proximidade á máquina (uso directo de punteiros, etc.)
  - o **Linguaxes de alto nivel**: son independentes da arquitectura da máquina, dispoñen de gran diversidade de instrucións potentes, e usan unha sintaxe similar á linguaxe natural, o que facilita a súa comprensión. Aquí estarían basicamente o resto das linguaxes coñecidas.
- Segundo a **forma de execución**:



- o **Compiladas:** unha vez escrito o programa, este tradúcese a partir do seu código fonte por medio dun compilador nun ficheiro executable para unha determinada plataforma. O compilador le o código fonte e almacena o executable resultado da tradución para posibles execucións futuras. Exemplos: C, C++, Pascal, Kilix, Delphi...
- o **Interpretadas:** necesitan dun intérprete para executar o código escrito nos programas. As instrucións tradúcense ou interprétanse unha a unha en tempo de execución a unha linguaxe intermedia ou linguaxe máquina. Na execución dunha linguaxe interpretada vanse lendo as liñas do código fonte e vanse traducindo a medida que vai sendo necesario. Cada vez que se le unha instrución interprétase e execútase, xerando o seu código máquina “ao voo”. Exemplos: Java, JavaScript, PHP, linguaxes de .NET...
- Segundo o **paradigma de programación:** Con isto referímonos ao enfoque á hora de afrontar o problema que tratamos de programar.
  - o **Imperativas:** fundaméntanse na utilización de variables para almacenar valores e na realización de instrucións para operar cos datos almacenados. Aparece o concepto de algoritmo, o COMO conseguir o obxectivo. Dentro das linguaxes imperativas, podemos clasificalas en:
    - **«Procedurais» ou procedementais:** agrupan código en subprogramas que se poden chamar desde distintos puntos do programa. Ex.: C, Pascal, Fortran, Basic, etc.
    - **Orientadas a obxectos:** ven o programa como unha colección de obxectos que interactúan os uns cos outros a través de mensaxes. É o máis utilizado na actualidade. Ex.: Smalltalk, C++, Java, C#, Visual Basic.NET, Eiffel, etc.

Na actualidade, case todas as novas versións das linguaxes incorporan características de orientación a obxectos. De feito,



moitos autores propoñen o paradigma orientado a obxectos como paradigma propio, evolución do paradigma imperativo.

- o **Declarativas**: decláranse ou descríbense as propiedades ou características do problema, deixando que a máquina atope a solución. Realmente decláranse condicións, proposicións, feitos, regras, afirmacións, restricións, ecuacións, transformacións, etc. que debe cumprir o conxunto de valores que constitúen a solución. Dentro desta temos tres tipos:
  - **Funcional ou aplicativa**: todas as construcións son funcións matemáticas. Non hai instrucións, e tampouco hai instrución de asignación. O programa defínese por composición de funcións máis simples. Para executalo, chámase unha función cos datos de entrada e obtense un resultado. Ex.: Haskell, LISP, CAML, etc.
  - **Lóxica**: baséase na definición de regras lóxicas para logo interrogar o sistema e resolver problemas. A programación lóxica trata con relacións (predicados) entre obxectos (datos), en lugar de facelo con funcións. Ex.: Prolog.
  - **Alxébricas ou relacionais**: algúns autores falan destoutro paradigma para clasificar SQL, que é a linguaxe utilizada para interrogar BBDD relacionais. Especificamos o resultado que queremos e SQL proporciónanolo.

Hai que dicir que algunhas linguaxes son **multiparadigma**. Por exemplo, C++ é unha linguaxe orientada a obxectos, pero tamén a podemos usar como linguaxe imperativa, sen facer uso da orientación a obxectos, ou Prolog, que é lóxica, mais tamén conta con estruturas repetitivas propias do paradigma imperativo.

## **34.2 Programación Estruturada.**



A programación estruturada é un concepto que xorde como reposta á crise do software dos anos sesenta, cando o custo no desenvolvemento de programas era cada vez maior, e o seu mantemento se facía cada vez menos manexable.

Foi desenvolvida nos seus inicios por Edsger W. Dijkstra nos seus “Notes on Structured Programming” e baséase no denominado Teorema da Estrutura ou de Böhm-Jacopini en honor de Corrado Böhm e Giuseppe Jacopini.

Edsger Dijkstra definiu a programación estruturada como *aquela que utiliza recursos abstractos, se basea no deseño descendente e respecta un conxunto de estruturas básicas chamadas Estruturas de Control: estrutura secuencial, estruturas selectivas e estruturas repetitivas.*

#### 34.2.1 Recursos abstractos

Todas as linguaxes de programación posúen un conxunto de recursos que poderíamos denominar recursos concretos: instrucións, palabras reservadas, tipos de datos, funcións, regras, etc. Porén, estes recursos non abundan para escribir programas para distintas aplicacións, xa que logo, o programador deberase valer de artificios para executar os seus algoritmos utilizando só os recursos concretos. Estes artificios denomínanse recursos abstractos.

Segundo Dijkstra, *escribir un programa en termos de recursos abstractos consiste en descompoñer as accións complexas en accións simples, capaces de seren executadas por unha máquina.* Isto significa que é posible escribir programas complexos empregando un conxunto limitado de instrucións moi simples.

#### 34.2.2 Deseño descendente

Existen dúas técnicas para escribir os algoritmos: o deseño ascendente e o deseño descendente.



Cando se utiliza deseño ascendente, pártese desde os detalles particulares de implantación da solución cara á solución xeral do problema. A solución do problema lógrase coa integración de todas as solucións particulares formuladas ao principio. Isto significa que primeiro se resolven partes particulares do problema e para rematar o problema en si. Esta técnica presenta dúas dificultades: lograr que as solucións particulares funcionen en conxunto e lograr que varios programadores traballen en coordinación.

O deseño descendente consiste en comprender o problema que cómpre solucionar e logo descompoñelo nun conxunto de problemas menores. Cando se usa esta técnica, primeiro formúlase a solución de forma xeral para logo pasar aos detalles particulares. Isto realízase en varios pasos chamados refinamentos. Poden existir varios niveis de refinamento ata chegar aos detalles de implantación da solución (subalgoritmos independentes chamados módulos). Unha vantaxe desta técnica é que permite a división de tarefas entre varios programadores onde cada un poderá escribir un algoritmo que obteña unha parte da solución.

A programación estruturada fai pois uso da **programación modular**. Un módulo é un conxunto de accións que realiza unha tarefa específica. Pode realizar as mesmas accións ca un programa: aceptar datos, realizar cálculos e devolver resultados. Con todo, os módulos utilízanse para un fin específico. Cando se utiliza esta técnica, toda a solución (o algoritmo) queda dividida en varias partes: un algoritmo principal e un ou varios subalgoritmos (os módulos). A execución iníciase no algoritmo principal e desde este invócanse os módulos. Os módulos tamén poden ser invocados desde outros módulos e o control da execución sempre se retorna ao momento desde onde o invocou por última vez.



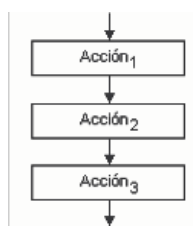
### 34.2.3 Estruturas básicas

Böhm e Jacopini demostraron que é posible resolver problemas escribindo programas denominados **propios**. Un programa defínese como propio se cumpre con:

- ten un só punto de entrada e un só de saída (inicio e fin),
- todas as accións do algoritmo son accesibles, é dicir, para cada acción existe un conxunto de datos que fará que esta sexa accesible desde o inicio e ademais poderase chegar ao fin,
- non posúe lazos ou bucles infinitos.

Pois ben, o **teorema da estrutura** establece que *calquera programa propio pode ser escrito utilizando só as seguintes estruturas lóxicas de control: secuencia, selección e iteración*. Un algoritmo correctamente executado posuirá un único punto de entrada, un único punto de saída, e permitirá a execución de todas as súas instrucións en función dos parámetros de entrada. A programación estruturada baséase no concepto de programa propio e utiliza só estas tres estruturas básicas:

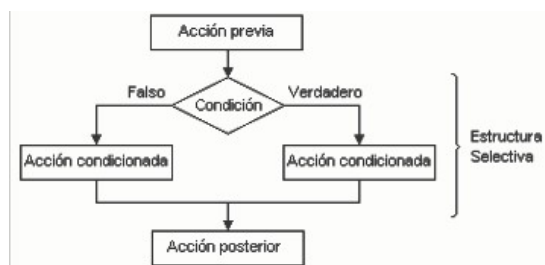
- **Estrutura secuencial:** caracterízase porque unha acción se escribe e executa a continuación doutra. Está representada por unha sucesión de operacións que se executan de xeito secuencial.



- **Estrutura selectiva:** caracterízase porque existen dúas ou máis secuencias alternativas de accións no programa. A selección dunha ou doutra realízase de acordo cunha *condición* que se debe cumprir para que o conxunto de accións sexa executado. Dise que as estruturas selectivas *bifurcan* a execución do programa. Unha instrución de bifurcación avalía unha *condición* e, en función do



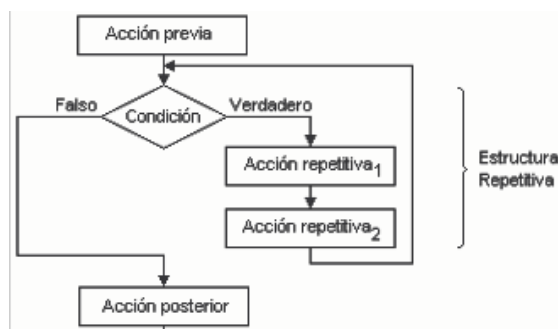
resultado desa avaliación, a execución bifúrcase a un determinado punto. Unha *condición* constrúese con *expresións lóxicas*.



TRADUCCIÓN FIGURA: Acción previa, Falso, Condición, Verdadeiro, Acción condicionada, Acción posterior, Estrutura Selectiva

- **Estrutura repetitiva ou bucle:** caracterízase porque existe un conxunto de accións cuxa execución se debe repetir un determinado número de veces chamado bucle (ou lazo). A execución do bucle debe ser tal que sexa posible acceder a el e tamén saír del. Non deben existir lazos de execución infinita. Para que un bucle cumpra coas condicións mencionadas (poder acceder e poder saír del) é necesario o uso dunha condición que o controle. En cada ciclo de execución do bucle deberase avaliar a condición para decidir se se volve executar ou se se sae del, así, todo bucle deberá posuír un controlador (un contador ou un sentinela), unha decisión (unha condición para decidir a repetición ou a saída do bucle) e un corpo (conxunto de accións que se repiten). Un contador é unha variable numérica enteira cuxo contido se incrementa ou decrece sucesivamente cun valor constante. Nun proceso de introdución de datos, o sentinela é o valor cuxa lectura como dato indica a finalización do bucle.





TRADUCCIÓN FIGURA: Acción previa, Falso, Condición, Verdadeiro, Acción repetitiva, Acción posterior, Estrutura Repetitiva.

Como exemplos de linguaxes de programación estruturadas temos FORTRAN, PASCAL, MODULA, ADA, C, etc.

### 34.3 Programación orientada a obxectos

A programación orientada a obxectos é o que se coñece como un paradigma ou modelo de programación. Isto significa que non é unha linguaxe específica, ou unha tecnoloxía, senón unha forma de programar, un xeito considerar a programación. O que caracteriza a programación orientada a obxectos é que intenta levar ao mundo do código o mesmo que atopamos no mundo real. Cando miramos ao noso redor, que vemos? Pois cousas, **obxectos**, e podemos recoñecer estes obxectos porque **cada obxecto pertence a unha clase** de obxecto. Iso permítenos distinguir, por exemplo, un can dun coche (porque son de clases diferentes) e tamén un TV doutro (porque, malia que sexan iguais, cada un é un obxecto distinto). Este é o modelo que a programación orientada a obxectos intenta seguir para estruturar un sistema.

A construción de software orientado a obxectos é o método de desenvolvemento de software que basea a arquitectura de calquera sistema software en módulos deducidos dos tipos de obxectos que manipula (en lugar de basearse na función ou funcións que o sistema está



destinado a asegurar). Os desenvolvedores analizarán os tipos dos obxectos do sistema. O deseño irá pasando polas sucesivas melloras da súa comprensión destas clases de obxectos. É un proceso ascendente, fronte á programación estruturada, que era descendente, de construción de solucións robustas e extensibles para certas partes do problema, e de combinación desas solucións en montaxes cada vez máis potentes, ata que a montaxe final dá unha solución do problema orixinal; ademais, os mesmos compoñentes ensamblados de xeito diferente e combinados posiblemente con outros, deberán ser o suficientemente xerais para produciren tamén outros subprodutos software, é dicir, deben ser reutilizables.

Así, un sistema concíbese como un conxunto de obxectos, pertencentes a algunha clase, que se comunican entre eles mediante mensaxes. Os obxectos distínguense por:

- Posuír un estado que pode cambiar,
- ter unha identidade única,
- soportar interrelacións con outros obxectos,
- posuír un comportamento,
- poder recibir e emitir mensaxes.

Un **obxecto** descríbese polas súas propiedades, tamén chamadas atributos (definen a estrutura do obxecto), e polos servizos (métodos ou operacións) que pode proporcionar (comportamento do obxecto). O estado dun obxecto vén determinado polos valores que toman os seus atributos. O obxecto é a unidade fundamental de descomposición na programación orientada a obxectos.

Ademais, todo obxecto é unha instancia dalgunha **clase**. Todos os obxectos se crean de acordo a unha definición de clase de obxectos. Esta clase inclúe declaracións de todos os atributos e operacións que se deberían asociar cun obxecto da devandita clase. Unha clase é unha



abstracción das propiedades comúns e comportamento dun conxunto de obxectos.

Así pois, un programa orientado a obxectos consta dunha ou máis clases interdependentes. As clases permiten describir as propiedades (atributos ou campos) e habilidades (métodos) dos obxectos cos que o programa ten que tratar. O conxunto de atributos e métodos dunha clase reciben tamén o nome de membros desa clase.

A definición dos membros pode ir precedida dun **modificador de acceso**, que é algo que indica desde qué puntos do código é accesible ese membro. Os tres principais modificadores de acceso son:

- **Public:** pódese acceder desde calquera punto.
- **Protected:** é accesible desde os métodos da súa clase e das clases que dela se puidesen derivar.
- **Private:** só se pode utilizar desde os métodos da propia clase.

Estes son os modificadores de acceso máis comúns, inda que dependendo da linguaxe poderíamos ter outros modificadores que varían os accesos en función de se as clases están no mesmo paquete...

O **comportamento** dun obxecto vén determinado polo conxunto de métodos que este proporciona. Referirémonos indistintamente a eles como **métodos, servizos ou operacións**. Un método está formado por un conxunto de sentenzas (accións) que se deben levar a cabo para facer efectivo un comportamento dun obxecto. Un obxecto proporciona un conxunto de métodos que se poden utilizar desde outros obxectos. Este conxunto de métodos constitúe o que se denomina **interface** do obxecto. Mediante a interface dun obxecto poderemos acceder aos valores dos seus atributos (estado) para a súa consulta e/ou modificación e activar un comportamento deste. Cando se fai unha chamada a un método da interface dun obxecto dicimos que lle estamos enviando unha mensaxe a este obxecto. Así, unha **mensaxe** pódese definir como unha petición a un obxecto para a obtención dalgún comportamento desexado deste. Unha



chamada a un método pode conter **parámetros**, ou conxuntos de datos de entrada do método. Os métodos poden devolver valores, obxectos ou nada (*void*).

Dise que a información acerca dun obxecto está **encapsulada** polo seu comportamento. A un obxecto débenselle pedir os seus datos, ou pedir que os cambie cunha mensaxe. Ao encapsular ou ocultar información sepáranse os aspectos externos dun obxecto (os accesibles para todos) dos detalles de implementación (os accesibles para ninguén). Con isto trátase de lograr que ao ter algún cambio na implementación dun obxecto non se teñan que modificar os programas que utilizan tal obxecto.

Unha das propiedades máis características da orientación a obxectos é a **herdanza**, que é a capacidade de crear novas clases (subclases) a partir doutra clase xa existente, especificando só as diferenzas coa clase pai. Inda que algunhas linguaxes permiten a herdanza múltiple (unha clase que deriva de máis dunha clase), en xeral, a maioría delas non o fan e só permiten que se herde dunha clase. Para simular a herdanza múltiple utilízanse as interfaces, que si son soportadas por todas as linguaxes orientadas a obxectos. As interfaces especifican (non implementan) conxuntos de métodos e atributos para que as clases os implementen. Unha clase que implementa unha interface está obrigada a ter unha implementación para cada método definido na interface.

Cando se diseña un modelo orientado a obxectos, é útil introducir clases a certo nivel que poden non existir en realidade, pero que permiten actuar como un depósito de métodos e atributos compartidos para as subclases de nivel inferior. Estas clases denomínanse **clases abstractas**, e non poden ter ningunha instancia. Non se poden crear obxectos desta clase. Tamén temos **métodos abstractos**, que son aqueles en que non se especifica ningunha implementación. Son as clases derivadas as que o teñen que facer. Unha **clase** dise que é **selada** cando non pode ter clases



derivadas. Igualmente, un **método selado** é aquel que non pode ser redefinido nas clases derivadas.

Os atributos de obxecto son aqueles que almacenan un valor para cada obxecto da clase. Os atributos de clase son aqueles que almacenan un valor accesible para todos os obxectos da clase. Os métodos de obxecto son aqueles que acceden ao estado dun obxecto concreto, mentres que os métodos de clases non necesitan acceder a ningún atributo de ningún obxecto. Os membros (atributos e métodos) de clase tamén son chamados estáticos ou compartidos.

Un **construtor** é un método especial que se executa automaticamente cando se crea o obxecto. O seu propósito é inicializar o obxecto con, polo menos, os datos mínimos que necesita. Igual que para o resto dos métodos, pode haber varios construtores sobrecargados. É dicir, podemos crear obxectos de distintas formas. Chámaselle construtor por defecto a aquel que non ten argumentos. Un **destrutor** é un método especial que adoita liberar a memoria e outros recursos cando un obxecto deixa de ser usado. Pode ser chamado automaticamente, liberando así da responsabilidade ao programador.

Para rematar falaremos doutra das características fundamentais da programación orientada a obxectos, o **polimorfismo**. O polimorfismo refírese a dous aspectos diferentes: por unha banda a sobrecarga de métodos e operadores (métodos polimórficos) e por outra banda, a ligadura dinámica. O primeiro é a capacidade de ter distintos métodos na mesma clase co mesmo nome, inda que con distinta sinatura (número de argumentos, tipos dos argumentos, orde). O que non poden é devolver un resultado de distinto tipo. O outro aspecto, a ligadura dinámica, refírese a que cando se lle envía unha mensaxe a un obxecto, o código que se chama non se determina ata o momento da execución. O compilador asegura que a función existe, mais non coñece o código exacto que cómpre executar. Para iso, o compilador insire un código especial en lugar dunha chamada



absoluta. Este código calcula en tempo de execución o enderezo real do método que hai que executar utilizando a información almacenada no propio obxecto. Esta ligadura dinámica está relacionada coa herdanza. En termos prácticos, o polimorfismo permite definir unha referencia a unha clase pai, que en tempo de execución apuntará a un obxecto concreto dalgunha clase filla (non coñecida en tempo de compilación). A chamada aos métodos virtuais (os métodos da clase pai que serán redefinidos na clase filla) a través da antedita referencia executaranse correctamente, pois durante a execución verifícase o tipo do obxecto almacenado e chámase a versión correcta do método.

Exemplos de linguaxes de programación orientadas a obxectos 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 Enxeñaría inversa e reenxeñaría**

A enxeñaría inversa ocúpase de estudar un sistema de información na orde inversa establecida no ciclo de vida habitual; isto é, partindo do código fonte, trátanse de identificar os compoñentes do sistema e as relacións existentes entre eles. Ata a súa chegada, o ciclo de vida do software ía, en teoría, nunha soa dirección; agora, pódese falar de dúas direccións: *forward* ou cara a adiante, que é a tradicional, e *reverse* ou cara atrás, que é a da enxeñaría inversa. A enxeñaría inversa tamén é coñecida como modernización de caixa branca (*White-Box Modernization*).

Daremos un par de definicións de enxeñaría inversa:

“A análise dun sistema para identificar os seus compoñentes actuais e as dependencias que existen entre eles, para extraer e crear abstraccións do devandito sistema e información do seu deseño” [Chikofsky, 1990].



“O proceso de analizar o código, documentación e comportamento dun sistema para identificar os seus compoñentes actuais e as súas dependencias para extraer e crear unha abstracción do sistema e información de deseño. O sistema en estudo non é alterado, senón que se produce coñecemento adicional acerca do sistema” [SEI, 2004].

A partir destas definicións podemos declarar que a enxeñaría inversa ten a misión de desentrañar os misterios e segredos dos sistemas en uso. Consiste principalmente en recuperar o deseño dunha aplicación a partir do código. Isto realízase principalmente mediante ferramentas que extraen información dos datos, procedementos e arquitectura do sistema existente. O obxectivo primordial é proporcionar unha base para o mantemento e futuros desenvolvementos. Este obxectivo xeral pódese traducir nos seguintes obxectivos parciais:

- Reducir os erros e os custos do mantemento.
- Facer os sistemas máis fáciles de entender, cambiar e probar.
- Protexer e estender a vida do sistema.
- Facilitar a reutilización de compoñentes do sistema existentes.
- Proporcionar documentación que non existe, ou actualizar a existente.
- Migrar a outra plataforma hardware ou software, cando sexa necesario.
- Levar o sistema baixo o control dun contorno CASE.

En vista destes obxectivos, os sistemas candidatos a lles aplicar a enxeñaría inversa reúnen algunhas das seguintes características:

- As especificacións de deseño e a documentación, non existen ou están incompletas.
- O código non é estruturado.
- Inexistencia de documentación interna nos programas, ou ben esta é incomprendible ou está desfasada.



- O sistema necesita un excesivo mantemento correctivo.
- Algúns módulos fixéronse excesivamente complexos debido aos sucesivos cambios realizados neles.
- Necesítase unha migración cara a unha nova plataforma de hardware ou de software.
- A aplicación está suxeita a cambios frecuentes, que poden afectar a parte do deseño.
- Prevese que a aplicación poida ter aínda longa vida.

Aplicar a enxeñaría inversa supón un enorme esforzo e, xa que logo, faise necesario avaliar exhaustivamente e sendo moi realistas os casos en que é rendible a súa aplicación. O seu resultado varía fortemente en función dos seguintes elementos:

- **O nivel de abstracción** do proceso de enxeñaría inversa, e das ferramentas que se usen. Isto alude á sofisticación da información de deseño que se pode extraer do código fonte. O nivel de abstracción ideal deberá ser o máis alto posible. Isto é, o proceso de enxeñaría inversa deberá ser capaz de derivar as súas representacións de deseño de procedementos (cun baixo nivel de abstracción); e a información das estruturas de datos e de programas (un nivel de abstracción lixeiramente máis elevado); modelos de fluxo de datos e de control (un nivel de abstracción relativamente alto); e modelos de entidades e de relacións (un elevado nivel de abstracción). A medida que crece o nivel de abstracción proporciónaselle ao enxeñeiro do software información que lle permitirá comprender máis facilmente estes programas.
- **A completitude** do proceso. A completitude dun proceso de enxeñaría inversa alude ao nivel de detalle que se proporciona nun determinado nivel de abstracción. Na maioría dos casos, a completitude decrece a medida que aumenta o nivel de abstracción. Por exemplo, dado un listado do código fonte, é relativamente sinxelo



desenvolver unha representación de deseño de procedementos completa. Tamén se poden derivar representacións sinxelas do fluxo de datos, pero é moito máis difícil desenvolver un conxunto completo de diagramas de fluxo de datos ou un diagrama de transición de estados. A completitude mellora en proporción directa á cantidade de análise realizada pola persoa que está efectuando a enxeñaría inversa.

- **A interactividade do proceso.** A interactividade alude ao grao co cal o ser humano se "integra" coas ferramentas automatizadas para crear un proceso de enxeñaría inversa efectivo. Na maioría dos casos, a medida que crece o nivel de abstracción, a interactividade deberase incrementar, ou senón a completitude verase reducida.
- **A direccionalidade do proceso.** Se a direccionalidade do proceso de enxeñaría inversa é monodireccional, toda a información extraída do código fonte proporcionaráselle á enxeñaría do software, que poderá entón utilizala durante a actividade de mantemento. Se a direccionalidade é bidireccional, entón a información subministraráselle a unha ferramenta de reenxeñaría que intentará reestruturar ou rexenerar o vello programa.

A enxeñaría inversa non implica a modificación do sistema, nin a xeración de novos sistemas, con todo, existen unha serie de técnicas intrinsecamente relacionadas con ela:

- **Redocumentación:** é a produción dunha representación semántica dun sistema a calquera nivel de abstracción que se requira. As ferramentas usadas parten do código fonte existente, para producir diagramas de fluxo de datos, modelos de datos, etc. Se a redocumentación toma a forma de modificación de comentarios no código fonte, pode ser considerada unha forma suave de reestruturación. Se se pensa nela como unha transformación desde o



código fonte a pseudocódigo e/ou prosa, esta última é considerada como de máis alto nivel de abstracción que a primeira.

- **Recuperación do deseño:** é un subconxunto da enxeñaría inversa, no cal, separadamente das observacións do sistema, se engaden coñecementos sobre o seu dominio de aplicación, información externa, e procesos dedutivos co obxecto de identificar abstraccións significativas a un maior nivel.
- **Reestruturación:** a transformación desde unha forma de representación a outra no mesmo nivel de abstracción, preservando as características externas do sistema (funcionalidade e semántica) [Chikofsky, 1990]. A reestruturación do software modifica o código fonte e/ou os datos nun intento de adecualo a futuros cambios. En xeral, a reestruturación non modifica a arquitectura global do programa. Tende a centrarse nos detalles de deseño de módulos individuais e en estruturas de datos locais definidas dentro dos módulos. Se o esforzo da reestruturación se estende máis aló dos límites dos módulos e abarca a arquitectura do software, a reestruturación pasa a ser enxeñaría directa (*forward engineering*). Arnold indica que os beneficios que se poden lograr coa reestruturación do software son o obter programas de maior calidade, mellorar a produtividade dos enxeñeiros do software, reducir o esforzo requirido para levar a cabo actividades de mantemento, e facer que o software sexa máis sinxelo de comprobar e de depurar.
- **Reenxeñaría:** A reenxeñaría parte dos resultados obtidos na enxeñaría inversa para reconstruír o sistema mediante enxeñaría cara a adiante. A reenxeñaría non só recupera a información de deseño dun software existente, senón que usa esta para alterar ou reconstruír o sistema existente, nun esforzo por mellorar a calidade xeral.



Chikofsky define a **reenxeñaría** como o *exame e alteración dun sistema para reconstruílo dunha nova forma e a subseguinte implementación desta nova forma*. Arnold, pola súa banda, sinala que *reenxeñaría é calquera actividade que mellore o noso entendemento sobre o software e prepare ou mellore o propio software, normalmente para a súa facilidade de mantemento, reutilización ou evolución*. A definición dada polo Reengineering Center do Software Engineering Institute é *a transformación sistemática dun sistema existente a unha nova forma para realizar melloras da calidade en operación, capacidade do sistema, funcionalidade, rendemento ou capacidade de evolución a baixo custo, cun plan de desenvolvemento curto e con baixo risco para o cliente*.

A importancia das técnicas de reenxeñaría do software estriban en que reducen os riscos evolutivos dunha organización, axudan as organizacións a recuperaren os seus investimentos en software, fan o software máis facilmente modificable, amplían a capacidade das ferramentas CASE e son un catalizador para a automatización do mantemento do software.

Para algúns autores, a reenxeñaría de sistemas pódese clasificar segundo os niveis de coñecementos requiridos para levar a cabo o proxecto. A reenxeñaría que require coñecementos a baixos niveis de abstracción (código fonte) chámase **enxeñaría inversa ou modernización de caixa branca** e aquela que só require o coñecemento das interfaces do sistema chámase **reenxeñaría** propiamente dita ou **modernización de caixa negra**.

Para realizar a reenxeñaría nos sistemas existentes (tamén chamados **legacy systems** ou **sistemas herdados**) empréganse técnicas métricas, de visualización de programas, de abstracción e reformulación do código. Tanto para reenxeñaría como para enxeñaría inversa, créanse patróns para a resolución de problemas relacionados con estas técnicas. En función do coñecemento do sistema, os datos, as funcionalidades e as interfaces,



desenvólvense novas técnicas de reenxeñaría non baseadas no coñecemento do código senón no exame do comportamento das entradas e saídas do sistema, desenvolvendo novos patróns de reenxeñaría e sentando as bases da reenxeñaría baseada en wrapping. Idealmente, **wrapping** é unha reenxeñaría onde só se analizan as interfaces (as entradas e saídas) do sistema existente ignorando os detalles internos. Esta solución non é aplicable sempre e ás veces require o concurso da enxeñaría inversa para o coñecemento interno do sistema.

Veremos nos seguintes apartados distintos modelos de reenxeñaría:

#### *34.4.1 Modelo cíclico*

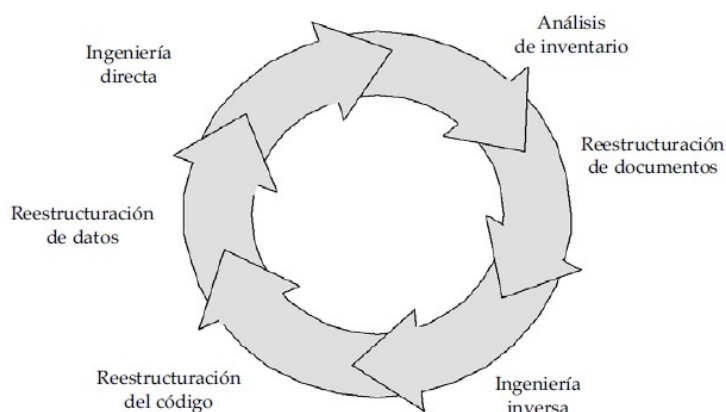
O modelo cíclico, debido a Pressman, concibe a reenxeñaría como un proceso composto por seis actividades que se producen, en xeral, de forma secuencial e lineal. As actividades que se definen no modelo cíclico son:

- **Análise de inventario.** Todas as organizacións de software deberán dispoñer dun inventario de todas as súas aplicacións, co fin de poderen identificar os candidatos á reenxeñaría.
- **Reestruturación de documentos.** Unha documentación escasa é a marca de moitos sistemas de información herdados. Ante iso será necesario, ou ben crear a documentación, ou actualizar a existente ou facela nova por completo.
- **Enxeñaría inversa.** A enxeñaría inversa do software é o proceso de análise dun programa co fin de crear unha representación de programa cun nivel de abstracción máis elevado que o código fonte. Na enxeñaría inversa extraerase do programa existente información do deseño arquitectónico e de proceso, e información dos datos.
- **Reestruturación do código.** O tipo máis común de reenxeñaría é a reestruturación do código. Algúns sistemas herdados teñen unha arquitectura de programa relativamente sólida, pero os módulos individuais foron codificados dunha forma que fai difícil



comprendelos, comprobalos e mantelos. Nestes casos, pódese reestruturar o código situado dentro dos módulos sospeitosos.

- **Reestruturación de datos.** Un programa que posúa unha estrutura de datos débil será difícil de adaptar e de mellorar. De feito, para moitas aplicacións, a arquitectura de datos ten máis que ver coa viabilidade a longo prazo do programa que o propio código fonte. A diferenza da reestruturación de código, que se produce nun nivel relativamente baixo de abstracción, a estruturación de datos é unha actividade de reenxeñaría a grande escala.
- **Enxeñaría directa (*forward engineering*).** A enxeñaría directa non só recupera a información de deseño dun software xa existente, senón que, ademais, utiliza esta información nun esforzo por mellorar a súa calidade global. Na maioría dos casos, o software procedente dunha reenxeñaría volve implementar a funcionalidade do sistema existente, e engade ademais novas funcións e/ou mellora o rendemento global.



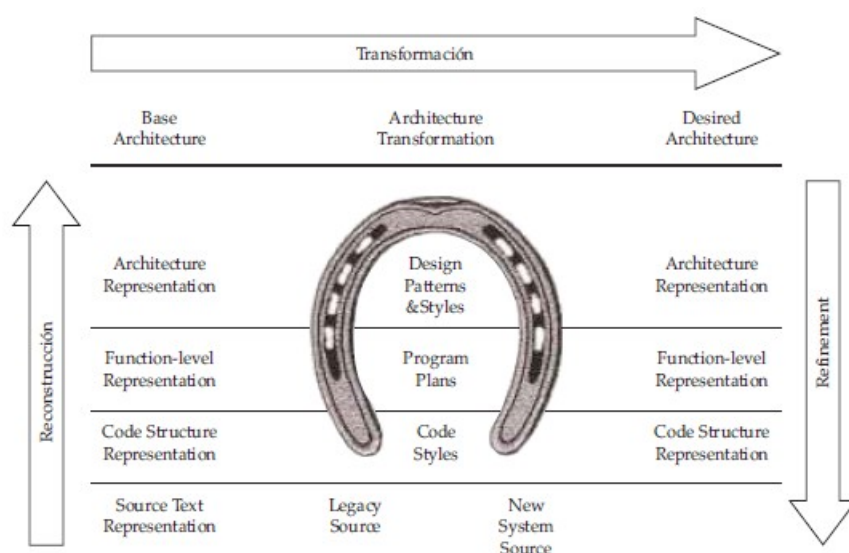
TRADUCCIÓN FIGURA: Enxeñaría directa, Análise de inventario, Reestruturación de datos, Reestruturación de documentos, Reestruturación do código, Enxeñaría inversa.

#### 34.4.2 O modelo de ferradura

O modelo de ferradura fundaméntase en considerar tres niveis de abstracción en todo sistema e que a reenxeñaría está formada por tres procesos básicos:



- **Análise dun sistema existente:** sobe o extremo esquerdo da ferradura, e recupera a arquitectura por medio da extracción de artefactos desde o código fonte.
- **Transformación lóxica:** cruza a parte superior e é a transformación de arquitectura. A arquitectura antes construída é recuperada e reenxeñarízase para facer a nova arquitectura desexable.
- **Desenvolvemento dun novo sistema:** baixa polo extremo dereito da ferradura, e constrúe a nova arquitectura desexable.



TRADUCCIÓN FIGURA: Reconstrucción.

A riqueza do modelo de ferradura son os tres niveis de abstracción que poden ser adoptados para as descrições lóxicas, que poden ser artefactos tan concretos e simples como o código fonte do sistema ou tan complexos e abstractos como a arquitectura do sistema. Os tres niveis que adopta o modelo de ferradura son:

- **Representación da estrutura de código,** o cal inclúe código fonte e artefactos tales como árbores de sintaxe abstractas e diagramas de fluxo.
- **Representación do nivel funcional,** o cal describe a relación entre as funcións do programa (chamadas), datos (funcións e relacións de datos), e arquivos (agrupamento de funcións e datos).



- **Nivel conceptual**, o cal representa grupo tanto de funcións e artefactos do nivel de código que son ensamblados dentro de subsistemas de compoñentes relacionados ou conceptos.

#### 34.4.3 O modelo do IEEE

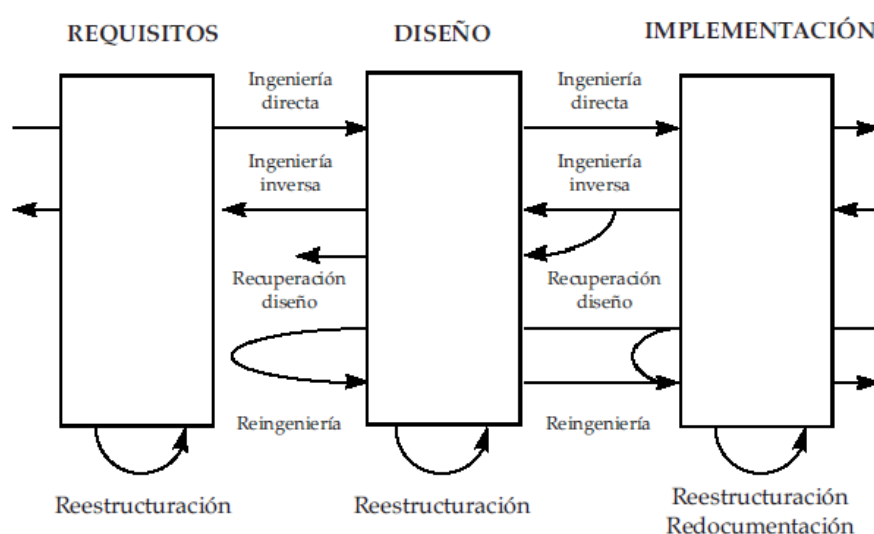
Este modelo baséase en considerar tres niveis de abstracción en todo sistema, o nivel requisitos, o nivel deseño e o nivel implementación, e en fixar unha terminoloxía. Os conceptos que define o IEEE, baseados nas definicións de Chikofsky e Cross, son os seguintes:

- **Enxeñaría inversa (*Reverse Engineering*)**: é o proceso de analizar un sistema para identificar os compoñentes e as interrelacións entre eles, creando representacións do sistema noutra forma distinta á orixinal ou ben a un nivel superior de abstracción.
- **Reenxeñaría (*Reengineering*)**: é o exame e modificación dun sistema para ser reconstruído dunha forma nova e ademais realizar a implantación derivada desa nova forma. A reenxeñaría normalmente inclúe algunha forma de enxeñaría inversa e vai seguida dalgunha forma de enxeñaría «cara a adiante» ou tamén dunha reestruturación.
- **Reestruturación (*Restructuring*)**: é a transformación dunha forma de representación do sistema noutra distinta, pero do mesmo nivel de abstracción, sen modificar o comportamento externo do sistema.
- **Enxeñaría cara a adiante (*Forward Engineering*)**: é o proceso que vai desde un alto nivel de abstracción, que é independente da implementación concreta, ata a propia implementación física do sistema. É dicir, é a enxeñaría do software na súa vertente restrinxida ao novo desenvolvemento.
- **Reenxeñaría de empresas (*Business Process Reengineering*)**: é a aplicación do concepto de reenxeñaría ao campo económico, e



desenvólvese ao redor de tres actividades clave: redeseñando os procesos básicos de traballo para alcanzar os obxectivos do negocio; utilizando as novas tecnoloxías para concibir, deseñar e poñer en marcha novas actividades; e cambiando a forma en que traballan os empregados.

As relacións entre as definicións e os niveis de abstracción son as que se ven na figura.



TRADUCCIÓN FIGURA: REQUISITOS, DESEÑO, IMPLEMENTACIÓN, Enxeñaría directa, Enxeñaría inversa, Recuperación deseño, Reenxeñaría, Reestruturación.

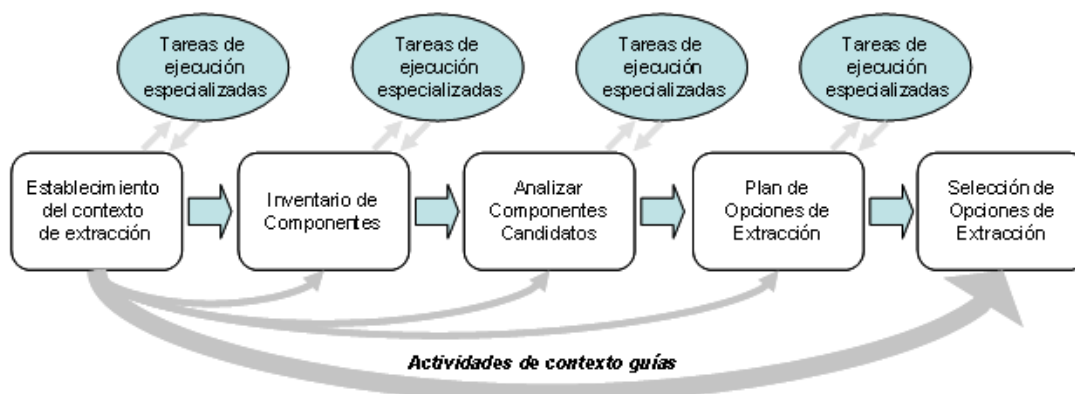
#### 34.4.4 O método *Análise de Opcións para Reenxeñaría* ("Options Analysis for Reengineering" (OAR))

É un método sistemático para a identificación e extracción de compoñentes dentro de grandes e complexos sistemas de software. OAR identifica compoñentes de arquitectura potencialmente relevantes e analiza os cambios requiridos para usalos nunha liña de produción de software ou novas arquitecturas de software. En esencia, OAR proporciona un conxunto de opcións de extracción xunto con estimación de custos, esforzo e riscos asociados a estas opcións. O método OAR consiste en cinco actividades principais:



- **Establecemento do contexto de extracción:** consiste en entrevistar os accionistas e estudar a liña de produción da organización ou novos requirimentos de sistema, base herdada e expectativas para a extracción de compoñentes herdados. Estes esforzos establecen unha liña base dun conxunto de metas, expectativas e necesidades de compoñentes.
- **Inventario de compoñentes:** o equipo OAR identifica os compoñentes do sistema herdado que potencialmente poden ser extraídos para usalos nunha liña de produción ou nunha nova arquitectura de software. Esta actividade resulta nun inventario dos compoñentes herdados candidatos.
- **Análise de compoñentes candidatos:** o seguinte paso dos membros do equipo é analizar o conxunto de candidatos de compoñentes herdados para extraer os tipos de cambios que son requiridos.
- **Plan de opcións de extracción:** Dado o conxunto de compoñentes candidatos analizados, o equipo desenvolverá alternativas para a extracción baseada en consideracións de calendario, custo, esforzo, risco e recursos. O equipo OAR tamén filtra unha vez máis os compoñentes candidatos e analiza o impacto de agregación de diferentes compoñentes.
- **Selección de opcións de extracción:** Finalmente, os membros do equipo seleccionan a mellor opción de extracción ou combinación de opcións. Logo de avaliaren cada opción de extracción, eles preparan un resumo onde presentan e xustifican as súas eleccións.





TRADUCCIÓN FIGURA: Tarefas de execución especializadas, Establecemento do contexto de extracción, Inventario de Compoñentes, Analizar Compoñentes Candidatos, Plan de Opcións de Extracción, Selección de Opcións de Extracción.

## Bibliografía

- Ingeniería del Software. Un enfoque práctico.
- Algoritmos y estructuras de datos.
- Reverse engineering and design recovery: A taxonomy. E. Chikofsky e 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**

**Xefe do Servizo de Informática. Instituto Galego de Vivenda e Solo**

**Vicepresidente do CPEIG**







# **35. MÉTODOS DE PROBA DO SOFTWARE. FUNDAMENTOS. ESTRATEXIA DE PROBA DO SOFTWARE: VERIFICACIÓN E VALIDACIÓN. CAIXA NEGRA E CAIXA BRANCA. PROBAS DE CONTORNO E APLICACIÓNS ESPECIALIZADAS. PROBAS FUNCIONAIS. PROBAS DE INTEGRACIÓN. PROBAS DE REGRESIÓN. PROBAS DE VALIDACIÓN.**



Tema 35. Métodos de proba do software. Fundamentos. Estratexia de proba do software. Verificación e validación. Caixa negra e caixa branca. Probas de contorno e aplicacións especializadas. Probas funcionais. Probas de integración. Probas de regresión. Probas de validación.

### 35.1 Métodos de proba do software. Fundamentos. Verificación e validación

#### 35.1.1 Principios das probas

### 35.2 Estratexias de proba do software

#### 35.2.1 Probas de unidade

#### 35.2.2 Probas de integración

#### 35.2.3 Probas de regresión

#### 35.2.4 Probas do sistema

#### 35.2.5 Probas de implantación

#### 35.2.6 Probas de validación

### 35.3 Probas de caixa negra e caixa branca

#### 35.3.1 Probas de caixa branca

##### 35.3.1.1 Proba do camiño básico

##### 35.3.1.2 Proba da estrutura de control

##### Proba de condicións

##### Proba de fluxo de datos

##### Proba de bucles

#### 35.3.2 Probas de caixa negra

##### 35.3.2.1 Partición equivalente

##### 35.3.2.2 Análise de valores límite

##### 35.3.2.3 Valores típicos de erro e valores imposibles

##### 35.3.2.4 Baseados en grafos

##### 35.3.2.5 Táboa ortogonal

##### 35.3.2.6 Proba de comparación

### 35.4 Probas de contorno e aplicacións especializadas

#### 35.4.1 Proba de interfaces gráficas de usuario

#### 35.4.2 Proba de arquitecturas cliente/servidor

#### 35.4.3 Proba da documentación e facilidades de axuda



#### 35.4.4 Proba de sistemas de tempo-real

### 35.5 Probas funcionais

## **35.1 Fundamentos das probas do software. Verificación e validación**

A proba do software é un elemento dun tema máis amplo que, a miúdo, é coñecido como **verificación** e **validación** (V&V). A **verificación** refírese ao conxunto de actividades que aseguran que o software executa correctamente unha función específica. A **validación** refírese a un conxunto diferente de actividades que aseguran que o software construído se axusta aos requisitos do cliente.

- **Verificación:** Estamos construíndo o produto correctamente?
- **Validación:** Estamos construíndo o produto correcto?

A verificación e a validación abarcan unha ampla lista de actividades de Garantía de Calidade do Software (SQA, en inglés) que inclúe: revisións técnicas formais, auditorías de calidade e de configuración, monitorización de rendementos, simulación, estudos de factibilidade, revisión da documentación, revisión da base de datos, análise algorítmica, probas de desenvolvemento, probas de validación e probas de instalación. Malia que as actividades de proba teñen un papel moi importante na verificación e validación, moitas outras actividades son tamén necesarias. A aplicación adecuada dos métodos e das ferramentas, as revisións técnicas formais efectivas e unha sólida xestión e medición conducen á calidade, que se confirma durante as probas.

A proba presenta unha anomalía para o enxeñeiro do software, xa que, mentres nas fases anteriores de definición e desenvolvemento, o enxeñeiro intenta construír, ao chegar as probas, o enxeñeiro diseña unha serie de



casos de proba que pretenden “demoler” o construído. Por iso os autores din que a proba se pode ver como destrutiva, en lugar de construtiva.

O proceso de probas do software ten dous obxectivos distintos:

- Para demostrarlles ao desenvolvedor e ao cliente que o software satisfai os seus requirimentos. Para o software a medida, isto significa que debería haber polo menos unha proba para cada requirimento dos documentos de requirimentos do sistema e do usuario. Para produtos de software xenéricos, significa que debería haber probas para todas as características do sistema que se incorporarán na entrega do produto.
- Para descubrir erros no software en que o comportamento deste é incorrecto, non desexable ou non cumpre a súa especificación. A proba de erros está relacionada coa eliminación de todos os tipos de comportamentos do sistema non desexables, tales como caídas do sistema, interaccións non permitidas con outros sistemas, cálculos incorrectos e corrupción de datos.

Das moitas definicións válidas da “proba do software” podémonos quedar con:

- A proba é o proceso de execución dun programa coa intención de descubrir erros (Glenford Myers).
- A proba é calquera actividade dirixida a avaliar a capacidade dun programa e determinar que alcanza os resultados requiridos.

### **35.1.1 Principios das probas**

Coñecida a definición de proba, algúns dos principios que lle afectan son:

- A proba é o proceso de executar un programa coa intención de descubrir erros, polo que un bo caso de proba é aquel que ten unha alta probabilidade de descubrir un erro non atopado ata entón.
- As probas débense planificar antes de que se empece a codificar. Así, a planificación comeza co modelo de requisitos, e a definición



detallada dos casos de proba unha vez que o deseño do sistema está consolidado.

- O 80% dos erros estarán localizados no 20% dos módulos.
- A proba completa é imposible.
- Para seren máis eficaces, as probas deberían ser feitas por un equipo independente.
- A probabilidade da existencia de máis erros nunha parte do software é proporcional ao número de erros xa atopados nesa parte.

### **35.2 Estratexias de proba do software.**

Unha estratexia de proba do software integra as técnicas de deseño de casos de proba nunha serie de pasos ben planificados que dan como resultado unha correcta construción do software. A estratexia proporciona un mapa que describe os pasos que hai que levar a cabo como parte da proba, cando se deben planificar e realizar eses pasos, e canto esforzo, tempo e recursos se van requirir. Xa que logo, calquera estratexia de proba debe incorporar a planificación da proba, o deseño de casos de proba, a execución das probas e a agrupación e avaliación dos datos resultantes.

Unha estratexia de proba do software debe ser suficientemente flexible para promover a creatividade e a adaptabilidade necesarias para adecuar a proba a todos os grandes sistemas baseados en software. Ao mesmo tempo, a estratexia debe ser suficientemente ríxida para promover un seguimento razoable da planificación e a xestión a medida que progresa o proxecto.

Propuxéronse varias estratexias de proba do software en distintos libros. Todas lle proporcionan ao enxeñeiro do software un persoal para a proba e todas teñen as seguintes características xerais:

- As probas comezan a nivel de módulo e traballan «cara a fóra», cara á integración de todo o sistema.





- Segundo o momento, son apropiadas diferentes técnicas de proba.
- A proba lévaa a cabo o responsable do desenvolvemento do software e (para grandes proxectos) un grupo independente de probas.
- A proba e a depuración son actividades diferentes, pero a depuración débese incluír en calquera estratexia de proba.

Unha estratexia de proba do software debe incluír probas de baixo nivel que verifiquen que todos os pequenos segmentos do código fonte se implementaron correctamente, así como probas de alto nivel que validen as principais funcións do sistema fronte aos requisitos do cliente. Unha estratexia débelle proporcionar unha guía ao profesional e proporcionar un conxunto de puntos clave para o xefe de proxecto.

Débense abordar os seguintes puntos se se desexa levar a cabo con éxito unha estratexia de proba do software:

- Especificar os requisitos do produto de xeito cuantificable moito antes de que comecen as probas.
- Establecer os obxectivos da proba de xeito explícito.
- Comprender qué usuarios van manexar o software e desenvolver un perfil para cada categoría de usuario.
- Desenvolver un plan de proba que faga fincapé na «proba de ciclo rápido».
- Construír un software «robusto» deseñado para probarse a si mesmo.
- Usar revisións técnicas formais efectivas como filtro antes da proba.
- Levar a cabo revisións técnicas formais para avaliar a estratexia de proba e os propios casos de proba.
- Desenvolver un enfoque de mellora continua ao proceso de proba.

Se consideramos o proceso desde o punto de vista procedemental, a proba, no contexto da enxeñaría do software, realmente é unha serie de cinco pasos que se levan a cabo de forma secuencial. Inicialmente, a proba céntrase en cada módulo individualmente, asegurando que funcionan



adecuadamente como unha unidade. De aí o nome de *proba de unidade*. A proba de unidade fai un uso intensivo das técnicas de proba de caixa branca (veranse con posterioridade noutro apartado), exercitando camiños específicos da estrutura de control do módulo para asegurar un alcance completo e unha detección máxima de erros. A seguir, débense ensamblar ou integrar os módulos para formar o paquete de software completo. A *proba de integración* diríxese a todos os aspectos asociados co dobre problema de verificación e de construción do programa. Durante a integración, as técnicas que máis prevalecen son as de deseño de casos de proba de caixa negra (veranse con posterioridade noutro apartado), inda que se poden levar a cabo algunhas probas de caixa branca co fin de asegurar que se cobren os principais camiños de control. Despois de integrar (construír) o software, diríxense un conxunto de *probos de alto nivel*. Débese comprobar que o software, ao combinalo con outros elementos do sistema (por exemplo, hardware, xente, bases de datos), cada elemento encaixa de forma adecuada e que se alcanza a funcionalidade e o rendemento esixido. Esta é a base da *proba de sistemas*. Posteriormente asegurárase mediante a *proba de implantación* o funcionamento correcto do sistema integrado de hardware e software no contorno de operación, e permitirlle ao usuario que, desde o punto de vista de operación, realice a aceptación do sistema unha vez instalado no seu contorno real e en función do cumprimento dos requisitos non funcionais especificados. A *proba de aceptación ou de validación* proporciona unha seguridade final de que o software satisfai todos os requisitos funcionais, de comportamento e de rendemento. Durante as últimas tres probas (*sistemas, implantación e aceptación*) úsanse exclusivamente técnicas de proba de caixa negra.

### **35.2.1 Probas de unidade**



As probas unitarias teñen como obxectivo verificar a funcionalidade e estrutura de cada compoñente individualmente unha vez que foi codificado.

A **proba de unidade** é un proceso para probar os subprogramas, as subrutinas, os procedementos individuais ou as clases nun programa. É dicir, é mellor probar primeiro os bloques desenvolvidos máis pequenos do programa, que inicialmente probar o software na súa totalidade. As motivacións para facer isto son tres. Primeira, as probas de unidade son un xeito de manexar os elementos de proba combinados, xa que se centra a atención inicialmente en unidades máis pequenas do programa. En segundo lugar, a proba dunha unidade facilita a tarefa de eliminar erros (o proceso de establecer claramente e de corrixir un erro descuberto), xa que, cando se atopa un erro, sábese que existe nun módulo particular. Finalmente, as probas de unidade introducen paralelismo no proceso de probas do software presentándose a oportunidade de probar os múltiples módulos simultaneamente.

Necesítanse dous tipos de información ao deseñar os casos de proba para unha proba de unidade: a especificación para o módulo e o código fonte do módulo. A especificación define tipicamente os parámetros de entrada e de saída do módulo e a súa función.

As probas de unidade son en gran parte orientadas a caixa branca. Unha razón é que como en probas de entidades máis grandes tales como programas enteiros (é o caso para os procesos de proba subsecuentes), a proba de caixa branca chega a ser menos factible. Unha segunda razón é que os procesos de proba subsecuentes están orientados a atopar diversos tipos de erros. Polo tanto, o procedemento para o deseño de casos de proba para unha proba de unidade é o seguinte: analizar a lóxica do módulo usando un ou máis dos métodos de caixa branca e despois completar os casos de proba aplicándolle métodos de caixa negra á especificación do módulo.



### **35.2.2 Probas de integración**

O obxectivo das **probas de integración** é verificar a correcta ensamblaxe entre os distintos compoñentes unha vez que foron probados unitariamente co fin de comprobar que interactúan correctamente a través das súas interfaces, tanto internas como externas, cobren a funcionalidade establecida e se axustan aos requisitos non funcionais especificados nas verificacións correspondentes.

Nas probas de integración examínanse as interfaces entre grupos de compoñentes ou subsistemas para asegurar que son chamados cando é necesario e que os datos ou mensaxes que se transmiten son os requiridos. Debido a que nas probas unitarias é necesario crear módulos auxiliares que simulen as accións dos compoñentes invocados polo que se está probando e a que se teñen que crear compoñentes "condutores" para establecer as precondicións necesarias, chamar o compoñente obxecto da proba e examinar os resultados da proba, a miúdo se combinan os tipos de proba unitarias e de integración.

Os tipos fundamentais de integración son os seguintes:

- *Integración incremental*: combínase o seguinte compoñente que se debe probar co conxunto de compoñentes que xa están probados e vaise incrementando progresivamente o número de compoñentes que cómpre probar. Co tipo de proba incremental o máis probable é que os problemas que xurdan ao incorporar un novo compoñente ou un grupo de compoñentes previamente probado, sexan debidos a este último ou ás interfaces entre este e os outros compoñentes.
- *Integración non incremental*: próbase cada compoñente por separado e posteriormente intégranse todos dunha vez realizando as probas pertinentes. Este tipo de integración denomínase tamén **Big-Bang**.



Dentro da *integración incremental*, temos tres tipos de estratexias:

- *Estratexia descendente (top-down)*: o primeiro compoñente que se proba é o primeiro da xerarquía. Os compoñentes de nivel máis baixo substitúense por compoñentes auxiliares chamados **resgardos** para simular os compoñentes invocados. Logo vanse substituíndo os resgardos subordinados polos compoñentes reais. Vantaxes: as interfaces entre os distintos compoñentes próbanse nunha fase temperá e con frecuencia. Verifica os puntos de decisión ou de control principais ao principio do proceso de proba.
- *Estratexia ascendente (bottom-up)*: neste caso créanse primeiro os compoñentes de máis baixo nivel e créanse compoñentes **condutores ou controladores** para simular os compoñentes que os chaman. A seguir substitúense os controladores polos módulos desenvolvidos de máis alto nivel e próbanse.
- *Estratexia combinada*: a miúdo é útil aplicar as estratexias anteriores conxuntamente. Deste xeito, próbanse as partes principais do sistema cun enfoque **top-down**, mentres que as partes de nivel máis baixo se proban seguindo un enfoque **bottom-up**.

### **35.2.3 Probas de regresión**

Cada vez que se engade un novo módulo como parte dunha proba de integración, o software cambia. Establécense novos camiños de fluxo de datos, poden ocorrer novas E/S e invócase unha nova lóxica de control. Estes cambios poden causar problemas con funcións que antes traballaban perfectamente. Neste contexto, a **proba de regresión** consiste en volver executar un subconxunto de probas que se levaron a cabo anteriormente para asegurarse de que os cambios non propagaron efectos colaterais non desexados.



Nun contexto máis amplo, as probas con éxito (de calquera tipo) dan como resultado o descubrimento de erros, e os erros hai que corrixilos. Cando se corrixo o software, cámbiase algún aspecto da configuración do software (o programa, a súa documentación ou os datos que o soportan). A proba de regresión é a actividade que axuda a asegurar que os cambios (debidos ás probas ou por outros motivos) non introducen un comportamento non desexado ou erros adicionais. A proba de regresión pódese facer manualmente, volvendo realizar un subconxunto de todos os casos de proba ou utilizando ferramentas automáticas de reprodución de captura. As ferramentas de reprodución de captura permítenlle ao enxeñeiro do software capturar casos de proba e os resultados para a subseguinte reprodución e comparación.

#### **35.2.4 Probas de sistema**

As probas do sistema teñen como obxectivo exercitar profundamente o sistema comprobando a integración do sistema de información globalmente, verificando o funcionamento correcto das interfaces entre os distintos subsistemas que o compoñen e co resto de sistemas de información cos que se comunica.

Unha vez que se probaron os compoñentes individuais e se integraron, próbase o sistema de forma global. Nesta etapa pódense distinguir os seguintes tipos de probas, cada un cun obxectivo claramente diferenciado:

- **Probas funcionais.** Dirixidas a asegurar que o sistema de información realiza correctamente todas as funcións que se detallaron nas especificacións dadas polo usuario do sistema.
- **Probas de comunicacións.** Determinan que as interfaces entre os compoñentes do sistema funcionan adecuadamente, tanto a través de dispositivos remotos, como locais. Así mesmo, débense probar as interfaces home/máquina.



- **Probas de rendemento.** Consisten en determinar que os tempos de resposta están dentro dos intervalos establecidos nas especificacións do sistema.
- **Probas de volume.** Consisten en examinar o funcionamento do sistema cando está traballando con grandes volumes de datos, simulando as cargas de traballo esperadas.
- **Probas de sobrecarga.** Consisten en comprobar o funcionamento do sistema na fronteira límite dos recursos, someténdoo a cargas masivas. O obxectivo é establecer os puntos extremos nos cales o sistema empeza a operar por baixo dos requisitos establecidos.
- **Probas de dispoñibilidade de datos.** Consisten en demostrar que o sistema se pode recuperar ante fallos, tanto de equipo físico como lóxico, sen comprometer a integridade dos datos.
- **Probas de facilidade de uso.** Consisten en comprobar a adaptabilidade do sistema ás necesidades dos usuarios, tanto para asegurar que se acomoda ao seu modo habitual de traballo, como para determinar as facilidades que proporciona ao introducir datos no sistema e obter os resultados.
- **Probas de operación.** Consisten en comprobar a correcta implantación dos procedementos de operación, incluíndo a planificación e control de traballos, arranque e rearranque do sistema, etc.
- **Probas de contorno.** Consisten en verificar as interaccións do sistema con outros sistemas dentro do mesmo contorno.
- **Probas de seguridade.** Consisten en verificar os mecanismos de control de acceso ao sistema para evitar alteracións indebidas nos datos.
- **Probas de configuración.** Programas tales como sistemas operativos, sistemas de xerencia de base de datos, e programas de conmutación de mensaxes soportan unha variedade de configuracións de hardware,



incluíndo varios tipos e números de dispositivos de entrada-saída e liñas de comunicacións, ou diversos tamaños de memoria. A miúdo o número de configuracións posibles é demasiado grande para probar cada un, pero no posible, débese probar o programa con cada tipo de dispositivo de hardware e coa configuración mínima e máxima. Se o programa por si mesmo se pode configurar para omitir compoñentes, ou se pode funcionar en diversos ordenadores, cada configuración posible deste debe ser probada.

- **Probos de instalación.** Ao funcionar incorrectamente o programa de instalación podería evitar que o usuario teña unha experiencia acertada co sistema. A primeira experiencia dun usuario é cando el ou ela instala a aplicación. Se esta fase se realiza mal, entón o usuario/o cliente pode buscar outro produto ou ter pouca confianza na validez da aplicación.
- **Probos de documentación.** A documentación do usuario debe ser o tema dunha inspección, comprobándoa para saber se hai exactitude e claridade. Calquera dos exemplos ilustrados na documentación se deben probar e facer parte dos casos e alimentalos ao programa.

### ***35.2.5 Proba de implantación***

O obxectivo das **probos de implantación** é comprobar o funcionamento correcto do sistema integrado de hardware e software no contorno de operación, e permitirlle ao usuario que, desde o punto de vista de operación, realice a aceptación do sistema unha vez instalado no seu contorno real e en función do cumprimento dos requisitos non funcionais especificados.

Unha vez que realizadas as probas do sistema no contorno de desenvolvemento, lévanse a cabo as verificacións necesarias para asegurar que o sistema funcionará correctamente no contorno de operación. Débese comprobar que responde satisfactoriamente os requisitos de rendemento,



seguridade, operación e coexistencia co resto dos sistemas da instalación para conseguir a aceptación do usuario de operación.

As probas de seguridade van dirixidas a verificar que os mecanismos de protección incorporados ao sistema cumpren o seu obxectivo; as de rendemento a asegurar que o sistema responde satisfactoriamente nas marxes establecidas en canto tempos de resposta, de execución e de utilización de recursos, así como os volumes de espazo en disco e capacidade; para rematar coas probas de operación compróbase que a planificación e control de traballos do sistema se realiza de acordo aos procedementos establecidos, considerando a xestión e control das comunicacións e asegurando a dispoñibilidade dos distintos recursos.

Así mesmo, tamén son levadas a cabo as probas de xestión de copias de seguridade e recuperación, co obxectivo de verificar que o sistema non ve comprometido o seu funcionamento ao existir un control e seguimento dos procedementos de salvagarda e de recuperación da información, en caso de caídas nos servizos ou nalgúns dos seus compoñentes. Para comprobar estes últimos, provócase o fallo do sistema, verificando se a recuperación se leva a cabo de forma apropiada. No caso de realizarse de forma automática, avalíanse a inicialización, os mecanismos de recuperación do estado do sistema, os datos e todos aqueles recursos que se vexan implicados.

As verificacións das probas de implantación e as probas do sistema teñen moitos puntos en común ao compartir algunhas das fontes para o seu deseño como poden ser os casos para probar o rendemento (probas de sobrecarga ou de *stress*).

O responsable de implantación xunto ao equipo de desenvolvemento determina as verificacións necesarias para realizar as probas así como os criterios de aceptación do sistema. Estas probas realízaas o equipo de



operación, integrado polos técnicos de sistemas e de operación que recibiron previamente a formación necesaria para levalas a cabo.

### **35.2.6 Probas de validación**

O obxectivo das **probas de validación**, tamén coñecidas como **probas de aceptación** é validar que un sistema cumpre co funcionamento agardado e permitirlle ao usuario deste sistema que determine a súa aceptación desde o punto de vista da súa funcionalidade e rendemento.

As probas de validación son definidas polo usuario do sistema e preparadas polo equipo de desenvolvemento, inda que a execución e aprobación final correspóndenlle ao usuario. Estas probas van dirixidas a comprobar que o sistema cumpre os requisitos de funcionamento agardado, recollidos no catálogo de requisitos e nos criterios de aceptación do sistema de información, e conseguir así a aceptación final do sistema por parte do usuario.

O responsable de usuarios debe revisar os criterios de aceptación que se especificaron previamente no plan de probas do sistema e, posteriormente, dirixir as probas de aceptación final. A validación do sistema conséguese mediante a realización de probas de caixa negra que demostran a conformidade cos requisitos e que se recollen no plan de probas, que define as verificacións que cómpre realizar e os casos de proba asociados. Este plan está deseñado para asegurar que se satisfán todos os requisitos funcionais especificados polo usuario tendo en conta tamén os requisitos non funcionais relacionados co rendemento, seguridade de acceso ao sistema, aos datos e procesos, así como aos distintos recursos do sistema.

A formalidade destas probas dependerá en maior ou menor medida de cada organización, e virá dada pola criticidade do sistema, o número de



usuarios implicados nelas e o tempo do que se dispoña para levalas a cabo, entre outros.

Se o software se desenvolve como un produto que vai ser usado por moitos clientes, non é práctico realizar probas de aceptación formais para cada un deles. A maioría dos desenvolvedores de produtos de software levan a cabo un proceso denominado proba alfa e beta para descubrir erros que pareza que só o usuario final pode descubrir.

A **proba alfa** lévase a cabo, por un cliente, no lugar de desenvolvemento. Úsase o software de forma natural co desenvolvedor como observador do usuario e rexistrando os erros e os problemas de uso. As probas alfa lévanse a cabo nun contorno controlado.

A **proba beta** lévase a cabo polos usuarios finais do software nos lugares de traballo dos clientes. A diferenza da proba alfa, o desenvolvedor non está presente normalmente. Así, a proba beta é unha aplicación «en vivo» do software nun contorno que non pode ser controlado polo desenvolvedor. O cliente rexistra todos os problemas (reais ou imaxinarios) que atopa durante a proba beta e informa a intervalos regulares ao desenvolvedor. Como resultado dos problemas informados durante a proba beta, o desenvolvedor do software leva a cabo modificacións e así prepara unha versión do produto de software para toda a clase de clientes.

### **35.3 Probas de caixa negra e caixa branca.**

Calquera produto de enxeñaría se pode probar dunha destas dúas formas:

- Coñecendo a función específica para a que foi deseñado o produto, pódense levar a cabo probas que demostren que cada función é completamente operativa.
- Coñecendo o funcionamento do produto, pódense desenvolver probas que aseguren que «todas as pezas encaixan», ou sexa, que a



operación interna se axusta ás especificacións e que todos os compoñentes internos se comprobaron de forma adecuada.

O primeiro enfoque de proba denomínase proba de **caixa negra** e o segundo, proba de **caixa branca**.

A proba de **caixa negra** refírese ás probas que se levan a cabo sobre a interface do software. Ou sexa, os casos de proba pretenden demostrar que as funcións do software son operativas, que a entrada se acepta de forma adecuada e que se produce un resultado correcto, así como que a integridade da información externa (por exemplo, arquivos de datos) se mantén. Unha proba de caixa negra examina algúns aspectos do modelo fundamental do sistema sen ter moito en conta a estrutura lóxica interna do software.

A proba de **caixa branca** do software baséase no minucioso exame dos detalles dos procedementos. Compróbanse os camiños lóxicos do software propoñendo casos de proba que exerciten conxuntos específicos de condicións e/ou bucles. Pódese examinar o «estado do programa» en varios puntos para determinar se o estado real coincide co esperado ou mencionado.

### **35.3.1 Probas de caixa branca**

O enfoque da estratexia de probas coñecido polo nome de método de «**caixa branca**», ou tamén «**conducido pola lóxica**» ou «**logic driven**», céntrase en probar o comportamento interno e a estrutura do programa, examinando a súa lóxica interna e sen considerar os aspectos de rendemento. O obxectivo deste enfoque é executar, polo menos unha vez, todas as sentenzas e executar todas as condicións tanto na súa vertente verdadeira como falsa, tendo en conta que a única información de entrada con que se conta é o deseño do programa e o código fonte. As técnicas específicas máis usuais que seguen o método de «caixa branca» son:

- Proba do camiño básico



- Proba da estrutura de control
  - o Proba de condicións
  - o Proba de fluxo de datos
  - o Proba de bucles

#### **35.3.1.1. Proba do camiño básico**

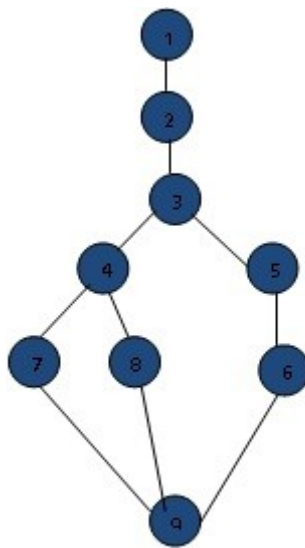
Foi proposta inicialmente por Tom McCabe. Permítelle ao deseñador de casos de proba obter unha medida da complexidade lóxica dun deseño procedemental e usar esa medida como guía para a definición dun conxunto básico de camiños de execución. Os casos de proba obtidos do conxunto básico garanten que durante a proba se executa polo menos unha vez cada sentenza do programa.

Da técnica do camiño básico derívanse dúas probas complementarias e non excluentes:

- *Proba de cobertura de sentenzas.* Consiste en xerar casos de proba que permitan probar todas e cada unha das sentenzas dun módulo unha vez. Esta proba é necesaria pero non é suficiente.
- *Proba de cobertura de condicións.* Consiste en deseñar xogos de proba que consideren todos os valores posibles de cada unha das condicións. Esta proba tamén é necesaria pero non é suficiente, xa que non garante que todos os camiños sexan cubertos, polo que debe ser complementada coa anterior.

A técnica de proba do camiño básico utiliza unha convención de representación denominada grafos de fluxo para a determinación de camiños e para iso apóiase no concepto de complexidade ciclomática proposto por Tom McCabe.





A notación de **grafo de fluxo**, proposta por McCabe, utilízase para simplificar o desenvolvemento do conxunto básico de camiños de execución. O seu obxectivo é exemplificar o fluxo de control do módulo que se está probando e para iso utiliza tres elementos:

- **Nodos** ou tarefas de procesamento (N). Representan cero, unha ou varias sentenzas procedementais. Cada nodo comprende como máximo unha sentenza de decisión (bifurcación). Cada nodo que contén unha condición denomínase **nodo predicado** e está caracterizado porque dúas ou máis arestas emerxen del.
- **Arestas**, fluxo de control ou conexións (A). Unen dous nodos, mesmo inda que o nodo non represente ningunha sentenza procedemental.
- **Rexións** (R). Son as áreas delimitadas polas arestas e nodos. Cando se contabilizan as rexións débese incluír a área externa como unha rexión máis.

A **complexidade ciclomática** é unha métrica do software baseada na teoría de grafos, que proporciona unha medición cuantitativa da complexidade lóxica dun programa. Cando se usa no contexto do método de proba do camiño básico, o valor calculado como complexidade



ciclomática define o número de camiños independentes do conxunto básico dun programa e dámos un límite superior para o número de probas que se deben realizar para asegurar que se executa cada sentenza polo menos unha vez. Un camiño independente é calquera camiño do programa que introduce, polo menos, un novo conxunto de sentenzas de proceso ou unha nova condición. En termos do grafo de fluxo, un camiño independente está constituído polo menos por unha aresta que non sexa percorrida con anterioridade á definición do camiño. O valor da complexidade pódese obter de tres formas:

- 1- O número de rexións do grafo. (R)
- 2- O número de arestas menos o número de nodos + 2. ( $A - N + 2$ )
- 3- Número de nodos predicado + 1. ( $P + 1$ )

A proba do camiño básico permítenos obter os casos de proba da seguinte maneira:

- 1- Usando o deseño ou o código como base, debuxamos o correspondente grafo de fluxo.
- 2- Determinamos a complexidade ciclomática do grafo de fluxo resultante.
- 3- Determinamos un conxunto básico de camiños linealmente independentes.
- 4- Preparamos os casos de proba que forzarán a execución de cada camiño do conxunto básico.

### **35.3.1.2 Probas da estrutura de control**

Dentro deste tipo de proba contéplase o método do camiño básico mencionado anteriormente pero ademais existen outras probas asociadas que permiten ampliar a cobertura da proba e mellorar a súa calidade.



#### **35.3.1.2.1 Proba de condición.**

É un método de deseño de casos de proba que exercita as condicións lóxicas contidas no módulo dun programa. Algúns conceptos empregados ao redor desta proba son os seguintes:

- *Condición simple:* é unha variable lóxica ou unha expresión relacional ( $E_1 < \text{operador} - \text{relacional} > E_2$ ).
- *Condición composta:* esta formada por dúas ou máis condicións simples, operadores lóxicos e parénteses.

En xeral, os tipos de erros que se buscan nunha proba de condición son os seguintes:

- *Erro en operador lóxico* (existencia de operadores lóxicos incorrectos, desaparecidos, sobrantes).
- *Erro en variable lóxica.*
- *Erro en paréntese lóxica.*
- *Erro en operador relacional.*
- *Erro en expresión aritmética.*

#### **35.3.1.2.2 Proba do fluxo de datos.**

Selecciona camiños de proba dun programa de acordo coa localización das definicións e cos usos das variables do programa.

#### **35.3.1.2.3. Proba de bucles.**

A proba de bucles é unha técnica de proba de caixa branca que se centra exclusivamente na validez das construcións de bucles. Pódense definir catro clases diferentes de bucles:



- **Bucles simples.** Aos bucles simples débeseles aplicar o seguinte conxunto de probas, onde  $n$  é o número máximo de pasos permitidos polo bucle:
  1. pasar por alto totalmente o bucle
  2. pasar unha soa vez polo bucle
  3. pasar dúas veces polo bucle
  4. facer  $m$  pasos polo bucle con  $m < n$
  5. facer  $n - 1$ ,  $n$  e  $n + 1$  pasos polo bucle
- **Bucles aniñados.** Se estendésemos o enfoque de proba dos bucles simples aos bucles aniñados, o número de posibles probas aumentaría xeometricamente a medida que aumenta o nivel de aniñamento. Isto levaría un número impracticable de probas. Suxírese un enfoque que axuda a reducir o número de probas:
  1. Comezar polo bucle máis interior. Establecer ou configurar os demais bucles cos seus valores mínimos.
  2. Levar a cabo as probas de bucles simples para o bucle máis interior, mentres se manteñen os parámetros de iteración (por exemplo, contador do bucle) dos bucles externos nos seus valores mínimos. Engadir outras probas para valores fóra de rango ou excluídos.
  3. Progresar cara a fóra, levando a cabo probas para o seguinte bucle, pero mantendo todos os bucles externos nos seus valores mínimos e os demais bucles aniñados nos seus valores «típicos».
  4. Continuar ata que se probaron todos os bucles.
- **Bucles concatenados.** Os bucles concatenados pódense probar mediante o enfoque antes definido para os bucles simples, mentres cada un dos bucles sexa independente do resto. Con todo, se hai dous bucles concatenados e se usa o controlador do bucle 1 como valor inicial do bucle 2, entón os bucles non son independentes.



Cando os bucles non son independentes, recoméndase usar o enfoque aplicado para os bucles aniñados.

- **Bucles non estruturados.** Neste caso, ante a complexidade que pode representar a comprensión do fluxo de control, é máis práctico redeseñar o módulo que se vai probar de xeito que se codifique mediante bucles estruturados.

### **35.3.2 Probas de caixa negra.**

O enfoque da estratexia de probas coñecido polo nome de método de «**caixa negra**», ou tamén «**conducido polos datos**» («***data driven***») ou «**conducido pola entrada/saída**» («***input-output driven***»), ou tamén **probas de comportamento**, non considera o detalle procedemental dos programas e céntrase en buscar situacións onde o programa non se axusta á súa especificación, utilizando esta como entrada para derivar os casos de proba.

Se polas especificacións funcionais se sabe o que ten que facer un módulo, é máis sinxelo comprobalo que esmiuzalo e examinalo internamente en todas as circunstancias posibles. Por iso, as probas de «caixa negra» son o enfoque máis simple de proba do software, e nela deseñaremos os casos de proba a partir das especificacións funcionais.

Neste tipo de probas os casos de proba consisten en conxuntos de datos de entrada que deberán xerar unha saída acorde coa especificación. A atención céntrase, pois, nos datos de entrada e saída ignorando intencionadamente o coñecemento do código do programa. Se con esta técnica se quixesen atopar todos os erros do programa, habería que recorrer a probar todas as posibles combinacións de casos de entrada, o que supoñería xerar todas as posibles combinacións de valores para todas as posibles variables de entrada, e iso, na realidade, é imposible. Desta



imposibilidade pódese extraer a conclusión de que mediante o método da «caixa negra» non é posible asegurar que un programa estea libre de erros.

Dado que non se poden probar todos os casos posibles, o método de «caixa negra» contempla unha serie de técnicas encamiñadas a simplificar os casos de proba. Estas son:

- Partición equivalente.
- A análise de valores límite.
- Os valores típicos de erro e os valores imposibles.
- Métodos de proba baseados en grafos.
- Proba da táboa ortogonal.
- As probas de comparación.

#### ***35.3.2.1 Partición equivalente.***

A **partición equivalente** é un método de proba de caixa negra que divide o campo de entrada dun programa en clases de datos dos que se poden derivar casos de proba. Un caso de proba ideal descobre de forma inmediata unha clase de erros (por exemplo, proceso incorrecto de todos os datos de carácter) que, doutro xeito, requirirían a execución de moitos casos antes de detectar o erro xenérico. A partición equivalente diríxese á definición de casos de proba que descubran clases de erros, reducindo así o número total de casos de proba que hai que desenvolver.

O deseño de casos de proba para a partición equivalente baséase nunha avaliación das clases de equivalencia para unha condición de entrada. Unha **clase de equivalencia** representa un conxunto de estados válidos ou non válidos para condicións de entrada. Tipicamente, unha condición de entrada é un valor numérico específico, un rango de valores, un conxunto



de valores relacionados ou unha condición lóxica. As clases de equivalencia pódense definir de acordo coas seguintes directrices:

- 1- Se unha condición de entrada especifica un rango, defínese unha clase de equivalencia válida e dúas non válidas.
- 2- Se unha condición de entrada require un valor específico, defínese unha clase de equivalencia válida e dúas non válidas.
- 3- Se unha condición de entrada especifica un membro dun conxunto, defínese unha clase de equivalencia válida e unha non válida.
- 4- Se unha condición de entrada é lóxica, defínese unha clase de equivalencia válida e unha non válida.

### **35.3.2.2 Análise de valores límite.**

A **análise de valores límite** é unha técnica de deseño de casos de proba que complementa a partición de equivalencia e se xustifica na constatación de que para unha condición de entrada que admite un rango de valores, é máis fácil que existan erros nos límites que no centro. Xa que logo, a diferenza entre esta técnica e a partición de equivalencia estriba en que na análise de valores límite non se selecciona un elemento representativo da clase de equivalencia, senón que se seleccionan un ou máis elementos de maneira que os límites de cada clase de equivalencia son obxecto de proba. Outra diferenza é que con esta técnica tamén se derivan casos de proba para as condicións de saída.

Do mesmo xeito que no caso anterior, a técnica de análise de valores límite non asegura a proba completa, xa que é imposible probar exhaustivamente todos os conxuntos de datos de entrada tanto na súa vertente válida como inválida. Con todo, a vantaxe que presenta esta técnica é que maximiza o número de erros atopados co menor número de casos de proba posibles, o que rendibiliza o investimento efectuado na proba.



As directrices de AVL son semellantes en moitos aspectos ás que proporciona a partición equivalente:

- 1- Se unha condición de entrada especifica un rango delimitado polos valores 'a' e 'b', débense deseñar casos de proba para os valores 'a' e 'b', e para os valores xusto por baixo e xusto por riba de 'a' e 'b', respectivamente.
- 2- Se unha condición de entrada especifica un número de valores, débense desenvolver casos de proba que exerciten os valores máximo e mínimo. Tamén se deben probar os valores xusto por riba e xusto por baixo do máximo e do mínimo.
- 3- Aplicarlles as directrices 1 e 2 ás condicións de saída.
- 4- Se as estruturas de datos internas teñen límites preestablecidos (por exemplo, unha matriz que teña un límite definido de 100 entradas), haise que asegurar de deseñar un caso de proba que exercite a estrutura de datos nos seus límites.

### ***35.3.2.3. Valores típicos de erro e valores imposibles.***

Un bo complemento das dúas técnicas fundamentais de probas tipo «caixa negra» (particións de equivalencia e análise de valores límite) consiste en incluír nos casos de proba certos valores dos datos de entrada susceptibles de causaren problemas, isto é, valores típicos de erro, e valores especificados como non posibles, é dicir, valores imposibles.

A determinación dos valores típicos de erro realízase en función da natureza e funcionalidade do programa que se vai probar, polo que depende en boa medida da experiencia do deseñador da proba.

Así mesmo, dentro das especificacións do sistema ou do programa que se vai probar pode haber valores de datos especificados como non posibles. O feito de probar estes valores imposibles débese a que estes valores poderían ser xerados internamente polo sistema ou polo programa



provocando un mal funcionamento deste. A proba de valores imposibles débese realizar sempre que estes valores poidan ser detectados e o programa os poida manexar adecuadamente sen provocar erros irreparables.

#### **35.3.2.4 Métodos de proba baseados en grafos.**

Neste método débense entender os obxectos (obxectos de datos, obxectos de programa tales como módulos ou coleccións de sentenzas da linguaxe de programación) que se modelan no software e as relacións que conectan estes obxectos. Unha vez que se levou a cabo isto, o seguinte paso é definir unha serie de probas que verifiquen que todos os obxectos teñen entre eles as relacións agardadas. Neste método:

1. Créase un grafo de obxectos importantes e as súas relacións.
2. Deséñase unha serie de probas que cubran o grafo de maneira que se exerciten todos os obxectos e as súas relacións para descubrir erros.

Boris Beizer describe un número de modelados para probas de comportamento que poden facer uso dos grafos:

- *Modelado do fluxo de transacción.* Os nodos representan os pasos dalgunha transacción (por exemplo, os pasos necesarios para unha reserva nunha liña aérea usando un servizo en liña), e os enlaces representan as conexións lóxicas entre os pasos (por exemplo, voo.información.entrada é seguida de validación /dispoñibilidade.procesamento).
- *Modelado de estado finito.* Os nodos representan diferentes estados do software observables polo usuario (por exemplo, cada unha das pantallas que aparecen cando un telefonista colle unha petición por teléfono), e os enlaces representan as transicións que ocorren para moverse de estado a estado (por exemplo, petición-información





verifícase durante inventario-dispoñibilidade-procura e é seguido por cliente-factura-información-entrada).

- *Modelado de fluxo de datos.* Os nodos obxectos de datos e os enlaces son as transformacións que ocorren para converter un obxecto de datos noutro.
- *Modelado de planificación.* Os nodos son obxectos de programa e os enlaces son as conexións secuenciais entre eses obxectos. Os pesos de enlace úsanse para especificar os tempos de execución requiridos ao executarse o programa.
- *Gráfica causa-efecto.* A gráfica causa-efecto representa unha axuda gráfica ao seleccionar, dun xeito sistemático, un gran conxunto de casos de proba. Ten un efecto secundario beneficioso en precisar estados incompletos e ambigüidades na especificación. Un gráfico de causa-efecto é unha linguaxe formal á que se traduce unha especificación. O gráfico é realmente un circuíto de lóxica dixital (unha rede combinatoria de lóxica), pero no canto da notación estándar da electrónica, utilízase unha notación algo máis simple. Non hai necesidade de ter coñecemento de electrónica, con excepción dunha comprensión da lóxica booleana (entendendo os operadores da lóxica 'e', 'ou', e 'non').

#### **35.3.2.5 Proba da táboa ortogonal.**

Hai aplicacións onde o número de parámetros de entrada é pequeno e os valores de cada un dos parámetros están claramente delimitados. Cando estes números son moi pequenos (por exemplo, 3 parámetros de entrada tomando 3 valores diferentes), é posible considerar cada permutación de entrada e comprobar exhaustivamente o proceso do dominio de entrada. En calquera caso, cando o número de valores de entrada crece e o número de valores diferentes para cada elemento de dato se incrementa, a proba exhaustiva faise impracticable.



A proba da **táboa ortogonal** pódese aplicar a problemas en que o dominio de entrada é relativamente pequeno pero demasiado grande para posibilitar probas exhaustivas. O método de proba da táboa ortogonal é particularmente útil ao atopar erros asociados con fallos localizados —unha categoría de erro asociada con defectos da lóxica dentro dun compoñente software—. A proba de táboa ortogonal permite proporcionar unha boa cobertura de probas con bastantes menos casos de proba que na estratexia exhaustiva.

#### **35.3.2.6 Proba de comparación.**

Hai situacións en que a fiabilidade do software é algo absolutamente crítico. Nese tipo de aplicacións, a miúdo utilízase hardware e software redundante para minimizar a posibilidade de erro. Cando se desenvolve software redundante, varios equipos de enxeñaría do software separados desenvolven versións independentes dunha aplicación, usando as mesmas especificacións. Nesas situacións, débense probar todas as versións cos mesmos datos de proba, para asegurar que todas proporcionan unha saída idéntica. Logo, execútanse todas as versións en paralelo e faise unha comparación en tempo real dos resultados, para garantir a consistencia.

Esas versións independentes son a base dunha técnica de proba de caixa negra denominada **proba de comparación ou proba man a man**.

### **35.4 Probas de contorno e aplicacións especializadas**

A medida que o software de computadora se fixo máis complexo, creceu tamén a necesidade de enfoques de probas especializados. Os métodos de proba de caixa branca e de caixa negra son aplicables a todos os contornos, arquitecturas e aplicacións, pero ás veces precísanse unhas directrices e enfoques máis específicos para as probas.



#### ***35.4.1. Proba de interfaces gráficas de usuario (IGU)***

Co paso do tempo a complexidade das IGU aumentou, orixinando máis dificultade no deseño e execución dos casos de proba. Dado que as IGU modernas teñen a mesma aparencia e filosofía, pódense obter unha serie de probas estándar. Os grafos de modelado de estado finito poden ser utilizados para realizar probas que vaian dirixidas sobre datos específicos e programas obxecto que sexan relevantes para as IGU. Considerando o gran número de permutacións asociadas coas operacións IGU, sería necesario utilizar ferramentas automáticas para realizar as probas.

#### ***35.4.2 Proba de arquitecturas cliente/servidor***

A natureza distribuída dos contornos cliente/servidor, os aspectos de rendemento asociados co proceso de transaccións, a presenza potencial de diferentes plataformas hardware, as complexidades das comunicacións de rede, a necesidade de servir a múltiples clientes desde unha base de datos centralizada (ou nalgúns casos, distribuída) e os requisitos de coordinación impostos ao servidor combínanse todos para facer as probas da arquitectura **C/S**, e o software residente nelas, considerablemente máis difíciles que a proba de aplicacións individuais.

#### ***35.4.3 Proba da documentación e facilidades de axuda***

Os erros na documentación poden ser tan destrutivos para a aceptación do programa como os erros nos datos ou no código fonte. Nada é máis frustrante que seguir fielmente o manual de usuario e obter resultados ou comportamentos que non coinciden cos anticipados polo documento. A proba da documentación pódese enfocar en dúas fases. A primeira fase, a revisión e inspección, examina o documento para comprobar a claridade editorial. A segunda fase, a proba en vivo, utiliza a documentación xunto ao uso do programa real.

#### ***35.4.4 Proba de sistemas de tempo-real***



A natureza asíncrona e dependente do tempo de moitas aplicacións de tempo real engádelle un novo e potencialmente difícil elemento á complexidade das probas (o tempo). O responsable do deseño de casos de proba non só ten que considerar os casos de proba de caixa branca e de caixa negra, senón tamén o tratamento de sucesos (por exemplo, procesamento de interrupcións), a temporización dos datos e o paralelismo das tarefas (procesos) que manexan os datos. En moitas situacións, os datos de proba proporcionados ao sistema de tempo real cando se atopa nun determinado estado darán un proceso correcto, mentres que ao llos proporcionar noutro estado levarán a un erro. Ademais, a estreita relación que existe entre o software de tempo real e o seu contorno de hardware tamén pode introducir problemas na proba. Pódese propoñer unha estratexia en catro pasos como método de proba para sistemas de tempo real:

- **Proba de tarefas.** O primeiro paso da proba de sistemas de tempo real consiste en probar cada tarefa de xeito independente.
- **Proba de comportamento.** Utilizando modelos do sistema creados con ferramentas CASE, é posible simular o comportamento do sistema en tempo real e examinar o seu comportamento como consecuencia de sucesos externos. Estas actividades de análise poden servir como base do deseño de casos de proba que se levan a cabo cando se construíu o software de tempo real.
- **Proba intertarefas.** Unha vez que se illaron os erros nas tarefas individuais e no comportamento do sistema, a proba diríxese cara aos erros relativos ao tempo. Próbanse as tarefas asíncronas que se sabe que comunican con outras, con diferentes taxas de datos e cargas de proceso para determinar se se producen erros de sincronismo entre as tarefas. Ademais, próbanse as tarefas que se comunican mediante colas de mensaxes ou almacéns de datos, para detectar erros no tamaño desas áreas de almacenamento de datos.



- **Proba do sistema.** O software e o hardware están integrados, polo que se leva a cabo unha serie de probas completas do sistema para intentar descubrir erros na interface software/hardware.

### **35.5 Probas funcionais**

Nas probas funcionais faise unha verificación dinámica do comportamento dun sistema, baseada na observación dun conxunto seleccionado de execucións controladas ou casos de proba. As probas funcionais son aquelas que se lle aplican ao produto final, e permiten detectar en que puntos o produto non cumpre as súas especificacións, é dicir, comprobar a súa funcionalidade. Para realizalas débese facer unha planificación que consiste en definir os aspectos que se van examinar e a forma de verificar o seu correcto funcionamento, punto onde adquiren sentido os casos de proba. Úsanse sobre todo no campo da orientación a obxectos.

As formas actuais usadas para derivar casos de proba funcionais son:

- A metodoloxía SCENT permite derivar casos de proba tomando como partida a definición de escenarios e actores que interactúan co sistema, para logo definir prioridades, pasando pola elaboración de diagramas de dependencia, diagramas de estados e para rematar xerar casos de proba.
- Heumann desenvolve un método para xerar casos de proba tomando como base casos de uso, e identificando dentro de cada un os posibles escenarios, ou camiños de execución, e para rematar definir os valores que cómpre probar de cada caso de proba. Finalmente obtense unha lista de casos de proba, cos valores que deben probar e os resultados agardados para cada caso.
- A proposta de Riebisch está centrada na transformación automática dun modelo de casos de uso a un modelo de uso que serve como entrada para realizar probas estatísticas automáticas, que melloran o



nivel de cobertura, partindo de que diferentes partes dun software non necesitan seren probadas coa mesma minuciosidade. O método comeza co refinamento dos casos de uso ampliándoos con precondicións e poscondicións, alternativas ao camiño de execución principal e referencia a outros casos de uso relacionados. Despois tradúcense a diagramas de estado e elabórase o modelo de uso onde se indica a probabilidade de que ocorra unha transición e identifícanse os camiños de execución máis frecuentes. Para rematar, extraíense os modelos de proba a partir dos modelos de uso e xéranse percorridos aleatorios sobre cada modelo de uso. Cada camiño aleatorio será un caso de proba.

- Para rematar, Hartman propón unha metodoloxía centrada en dous produtos: o primeiro composto por un modelo do sistema escrito na linguaxe de modelado IF e un conxunto de diagramas UML de clases e estados que van permitir a xeración automática do conxunto de probas. O segundo, conformado por un conxunto de obxectos de casos de proba executables tanto no modelo do sistema como na implantación, o que permite comparar os resultados agardados e os obtidos. Para obter os produtos enunciados, en primeiro lugar constrúese un modelo de comportamento do sistema a partir das súas especificacións. Este modelo está composto por diagramas UML de clases e un diagrama UML de estados por cada clase que describe o comportamento dos obxectos desta clase. A seguir elabóranse os obxectivos das probas (probas de casos de uso con datos concretos, probas de carga do sistema, etc.) e tradúcense a un conxunto de directivas de xeración e execución de probas. No seguinte paso, unha ferramenta xera automaticamente unha serie de probas que satisfán os obxectivos de proba anteriores e execútase automaticamente. Para rematar, analízanse os resultados e repítense os pasos ata que se alcanzan os obxectivos desexados.



## Bibliografía

- Ingeniería del Software. Un enfoque práctico. Roger Pressman. Ed. McGraw 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 González Palacio.

**Autor: Hernán Vila Pérez**

**Xefe do Servizo de Informática. Instituto Galego de Vivenda e Solo**  
**Vicepresidente do CPEIG**



**36. SEGURIDADE DA  
INFORMACIÓN.  
CONFIDENCIALIDADE,  
INTEGRIDADE E  
DISPOÑIBILIDADE. MEDIDAS DE  
SEGURIDADE FÍSICAS,  
TÉCNICAS, ORGANIZATIVAS E  
LEGAIS. IDENTIFICACIÓN E  
AUTENTICACIÓN. CONTROL DE  
ACCESOS FÍSICOS E LÓXICOS.  
CONTROL DE FLUXO DE  
DATOS.**



**Tema 36. Seguridade da información. Confidencialidade, integridade e dispoñibilidade. Medidas de seguridade físicas, técnicas, organizativas e legais. Identificación e autenticación. Control de accesos físicos e lóxicos. Control de fluxo de datos.**

**ÍNDICE**

1.1. Introducción.....	2
1.2. As dimensións da seguridade.....	3
1.2. Ameazas á seguridade.....	5
1.2.1. Persoas.....	6
1.2.2. Ameazas lóxicas.....	6
1.2.3. Catástrofes.....	8
2. MEDIDAS DE SEGURIDADE FÍSICAS, TÉCNICAS, ORGANIZATIVAS E LEGAIS.	
CONTROL DE ACCESO FÍSICO E LÓXICO.....	8
2.1. Plan de seguridade e plan de continxencias.....	11
2.1.1. O plan de seguridade.....	11
2.1.2. O plan de continxencias.....	12
2.2. Seguridade Física.....	13
2.3. Seguridade Lóxica.....	15
2.3.1. Ataques contra a seguridade lóxica .....	16
2.3.2. A protección da seguridade lóxica.....	22
3. IDENTIFICACIÓN E AUTENTICACIÓN.....	24
4. CONTROL DO FLUXO DE DATOS.....	26
4.1. Técnicas de control de fluxo de datos.....	26
4.1.1. Paridade simple (paridade horizontal).....	27
4.1.2. Paridade cruzada (paridade horizontal-vertical).....	27
4.1.3. Códigos de redundancia cíclica (CRC):.....	28
4.1.4. Suma de comprobación (checksum).....	29
4.1.5. Distancia de Hamming.....	30
5. REFERENCIAS.....	31



SEGURIDADE DA INFORMACIÓN: AUTENTICIDADE, INTEGRIDADE, CONFIDENCIALIDADE, DISPOÑIBILIDADE E TRAZABILIDADE.

## **1.1. Introducción**

Definimos un sistema informático como o conxunto formado por recursos físicos denominados *hardware*, recursos lóxicos denominados *software* e recursos humanos, que interactúan entre si co obxectivo de obter, almacenar e procesar a información.

Neste contexto, a seguridade informática será unha característica dos sistemas informáticos que indicará se o sistema está libre de perigo, dano ou risco. Con frecuencia, a seguridade absoluta é difícil, ou mesmo imposible, de acadar e os sistemas informáticos non son unha excepción.

Nun sistema informático os esforzos centraranse en protexer o software, o hardware e os datos. Estes adoitan ser o elemento máis valioso polo que requiren dun maior investimento en seguridade ao tratarse do recurso máis ameazado e o máis difícil de recuperar.

Non obstante, tendo en conta que a seguridade absoluta dos sistemas de información é inalcanzable, cómpre buscar sempre un compromiso entre o nivel de risco asumido e o custo das medidas de seguridade, de tal xeito que o devandito custo non supere nunca o valor do que se pretende protexer.

A seguridade nos sistemas de información afecta a todos os membros da organización. A súa natureza dinámica fai que deba ser planificada, deseñada, executada e mellorada de forma continua, xa que todos os días xorden novas ameazas. Para protexernos contra elas non é suficiente con implantar medidas técnicas, senón que teremos que preparar unha política de seguridade da organización, plans de actuación, medidas de seguridade física, formación, concienciación, etc.



## 1.2. As dimensións da seguridade

As dimensións da seguridade, tamén denominadas servizos ou factores de seguridade, fan referencia ás propiedades que a información debe cumprir para considerar un sistema como seguro. Un sistema informático encóntrase en óptimas condicións de seguridade para operar cando é capaz de garantir a seguridade de todos os seus compoñentes en cinco dimensións:

- **Confidencialidade:** é a propiedade da información pola que se garante que unicamente está accesible para os usuarios e procesos.

Se a información se atopa almacenada nun sistema propio, a confidencialidade baséase en primeiro lugar en garantir a autenticidade de calquera usuario que intente acceder a ela. Unha vez autenticado, debe controlarse que o usuario está autorizado, é dicir, que conta cos permisos para acceder a esa información en concreto.

Porén, cando a información se transmite entre un emisor e un receptor a través dun medio externo e por tanto inseguro, a confidencialidade debe garantirse mediante o uso de técnicas de cifrado, co fin de evitar que un terceiro que poida interceptar a mensaxe sexa quen de extraer dela calquera información intelixible.

- **Integridade:** é a propiedade da información pola que se garante que esta non foi manipulada intencionada ou accidentalmente por usuarios non autorizados.

A integridade pode protexerse mediante técnicas de control do fluxo de datos como sumas de validación (*checksum*), uso de bits de paridade, revisións de redundancia cíclica (CRC), algoritmos de





resumo (SHA1, MD5, etc.) e calquera outro algoritmo destinado a detectar calquera cambio non autorizado na información.

- **Dispoñibilidade:** é a propiedade da información pola que se garante que esta se atopará a disposición das persoas, procesos ou aplicacións que deban acceder a ela e dispoñan de autorización para iso.

A dispoñibilidade implica que o sistema, tanto no que se refire ao hardware coma ao software, debe manterse funcionando eficientemente e que é capaz de recuperarse rapidamente en caso de fallo. Isto conséguese deseñando os sistemas de xeito apropiado en termos de capacidade e escalabilidade para poder crecer de forma ordenada en caso de ser necesario. Ademais disto, existen técnicas destinadas a evitar problemas de dispoñibilidade como o balanceo de carga entre servidores, virtualización de servidores, uso de almacenamentos redundantes (RAID) ou os sistemas de respaldo.

- **Autenticidade:** é a propiedade da información pola que se garante a súa xenuinidade e que permite identificar o seu autor ou xerador.

Existe unha propiedade relacionada coa autenticidade coñecida como imposibilidade de rexeitamento ou 'non repudio'. Esta propiedade permite demostrar ante terceiros que unha información foi enviada ou consignada por unha entidade sen que esta poida negalo.

A autenticidade implica a necesidade de dar probas de identidade dos participantes nunha comunicación. O modo en que un usuario pode demostrar a súa identidade recae nun ou máis dos seguintes factores: algo que o usuario sabe (por exemplo, unha clave de acceso), algo que ten (por exemplo, unha tarxeta de acceso) ou



algo que é parte do propio usuario (por exemplo, medidas biométricas como as impresións dixitais). Considérase que para que unha autenticación sexa realmente segura debe incluír polo menos elementos de dous dos tres (se non dos tres) factores.

Por exemplo, a arquitectura de sinatura electrónica con certificados emitidos por entidades fiables (Autoridades de Certificación) é un sistema de autenticación multifactor, xa que primeiro debemos posuír un documento que nos identifique para poder obter un certificado avalado por un terceiro e coñecer un código persoal para realizar as sinaturas.

- **Trazabilidade:** a trazabilidade nun sistema de información fai referencia á súa capacidade para seguir e conservar a secuencia de todas as accións (ou polo menos daquelas que poidan afectar á seguridade do sistema) que acontezan no sistema para determinar a súa orixe. Un sistema con procesos trazables facilitará a resposta a incidencias na seguridade. Tamén será de grande axuda á hora de realizar auditorías de seguridade e deseñar sistemas de reposta baseados no estudo de patróns de situacións, como os sistemas de detección de intrusos (IDS).

## 1.2. Ameazas á seguridade

Os sistemas informáticos atópanse expostos a diversos axentes externos ou internos capaces de causarlles dano. Dependendo da súa orixe, distinguiremos tres tipos de ameazas: persoas, ameazas lóxicas e catástrofes.



### 1.2.1. Persoas

Tanto o propio cadro de persoal da organización coma curiosos ou intrusos supoñen unha ameaza para a seguridade da información. Os primeiros, malia que normalmente o farán de forma non intencionada, poden causar danos ao sistema durante o exercicio da súa actividade diaria. Estes danos serán proporcionais ao nivel de exposición da información, xa que non todos os usuarios terán o mesmo nivel de acceso á información. Pola súa banda, os intrusos supoñen a miúdo unha ameaza maior, xa que atacan de forma consciente coa intención de facer o maior dano posible. Dentro deste grupo podemos incluír exempregados, *crackers*, terroristas, intrusos remunerados, etc.

### 1.2.2. Ameazas lóxicas

Nesta categoría inclúense todo tipo de programas que, dun xeito ou doutro, poden danar o noso sistema, xa sexa intencionadamente ou por erro. Podemos enumerar nesta categoría as seguintes ameazas:

- **Software incorrecto:** programas que, debido a defectos no seu código, poden causar danos no sistema ou facilitar ataques desde o exterior.
- **Ferramentas de seguridade:** poden ser unha excelente ferramenta de axuda para xestionar a seguridade e configurar os nosos sistemas correctamente; pero constitúen unha arma de dobre fío, xa que tamén son útiles para descubrir vulnerabilidades no noso sistema. Exemplos deste tipo de ferramentas son os *sniffers*<sup>1</sup> (como por exemplo, *Wireshark*) ou as ferramentas de escaneo de portos (como por exemplo, *nmap*)

---

<sup>1</sup> Programas que permiten rexistrar e analizar toda a información transmitida a través dunha rede.



- **Bombas lóxicas:** son partes do código de certos programas que permanecen inactivas ata que se cumpre certa condición (ausencia ou presenza de certos ficheiros, datas, etc.). Cando esta condición se activa, execútase o código malicioso que ataca ao sistema.
- **Canles de comunicación ocultas:** permiten que un proceso transfira información, violando a política de seguridade.
- **Portas traseiras:** son 'atallos' deixados polos programadores para acelerar os procesos de probas do software, pero que se non son corrixidos antes do paso a produción se poden converter en perigosos buracos na seguridade.
- **Virus:** son secuencias de código que se insiren nalgún lugar do sistema (habitualmente en arquivos executables, aínda que poden residir noutros lugares) para realizar a súa tarefa maliciosa e tentar estenderse a outras partes do mesmo sistema ou incluso a outros sistemas.
- **Vermes:** programas capaces de executarse e propagarse por si mesmos a través de redes.
- **Troianos:** son instrucións de código escondidas en programas que se aloxan no sistema e que realizan funcións ocultas, normalmente destinadas a tomar o control dun sistema (Por exemplo, os rootkit).
- **Programas coello** (tamén coñecidos como programas bacteria): programas que non fan nada fóra de reproducirse ata esgotar os recursos do sistema e conseguir bloquealo.
- **Técnicas Salami:** roubo automatizado e sistemático de pequenas cantidades de bens dunha gran cantidade orixinal.



### **1.2.3. Catástrofes**

As catástrofes, sexan naturais ou artificiais, constitúen a ameaza menos probable pero tamén a máis desastrosa para os sistemas de información. A defensa contra elas baséase en recursos físicos e nun deseño adecuado da sede física dos nosos sistemas.

- **Incendios:** O centro de proceso de datos debe contar cun sistema de extinción de incendios axeitado que permita sufocar calquera conato sen que a propia extinción lles cause danos aos sistemas.
- **Inundacións:** Ademais de buscar un emprazamento co mínimo risco de inundación, é recomendable instalar sensores que detecten posibles fugas de auga e alerten sobre elas antes de que poidan causar dano.
- **Terremotos:** : Nas zonas con actividade sísmica é un risco que cómpre ter en conta.
- **Tormentas eléctricas:** A instalación eléctrica debe contar con todas as garantías proporcionando alimentación ininterrompida e neutralizando os picos de tensión que se puidesen producir.
- **Temperaturas extremas:** Os equipos informáticos, como aparatos electrónicos que son, necesitan unhas condicións de temperatura e humidade axeitadas para un funcionamento óptimo. Estas condicións adoitan conseguirse mediante a instalación de equipos de climatización.

## **2. MEDIDAS DE SEGURIDADE FÍSICAS, TÉCNICAS, ORGANIZATIVAS E LEGAIS. CONTROL DE ACCESO FÍSICO E LÓXICO**



A seguridade dun sistema de información require accións en diferentes ámbitos, que podemos diferenciar nos ámbitos físico, técnico (ou lóxico), organizativo e legal.

As medidas de seguridade no aspecto físico pasan por controlar os accesos físicos ao sistema de información, deseñar as infraestruturas axeitadamente conforme ás actividades da organización e ao ámbito físico da súa situación.

As medidas técnicas son todas aquelas destinadas a garantir a seguridade dos datos e os equipos dende o punto de vista do software e as comunicacións, entendidas estas últimas como os protocolos e os datos, e non como os medios de transmisión físicos (equipos de transmisión, cableado, etc.).

A seguridade informática non é un problema que deba ser abordado en solitario polo departamento de sistemas de información. O ámbito organizativo debe participar activamente na definición das políticas de seguridade da organización. Trátase de definir procedementos, políticas e normas destinadas a garantir a seguridade.

Algunhas das accións organizativas serven en realidade para axudar a deseñar as medidas físicas e técnicas. Toda organización debería contar con:

- **Análise de riscos:** a análise de riscos debe realizarse sempre como paso inicial no deseño da seguridade dos nosos sistemas de información. Supón identificar os activos a protexer e o dano que sufriría a organización en caso de ser afectados por un ataque. Veremos este punto en profundidade no vindeiro capítulo.
- **Identificación de ameazas:** consiste en identificar cada unha das ameazas e vulnerabilidades que pode afectar aos recursos do sistema.



- **Políticas de seguridade:** é fundamental contar cunha política de seguridade, deseñada a medida para a organización e coñecida por todos os empregados e/ou usuarios. Neste sentido, o RFC 2196 *Site Security Handbook* é unha guía para o desenvolvemento de políticas e procedementos de seguridade aplicables a sistemas de información. Está destinado principalmente a sistemas que traballan no ámbito da internet, pero tamén a aqueles sistemas que simplemente se comunican con outros. E en forma xeral tamén pode ser utilizado en sistemas illados. O contido inclúe políticas, conceptos de seguridade en redes e sistemas, e respostas aos incidentes de seguridade.
- **Estratexia de seguridade:** unha estratexia axeitada debe ser concibida de modo que abranca varios niveis de seguridade: seguridade física, seguridade lóxica, o persoal da organización e a interacción que existe entre estes factores. O plan de seguridade é un documento fundamental na organización que debe incluír unha estratexia de previsión de ataques para minimizar os puntos vulnerables existentes na directiva de seguridade. Debe tamén desenvolver plans de continxencias. Estes servirán como unha estratexia reactiva de resposta ao ataque que axude ao persoal de seguridade a avaliar o dano causado e a recuperar os niveis de servizo necesarios.
- **Equipos de resposta a incidencias:** é aconsellable formar un equipo de respostas a incidencias que dea apoio ao responsable de seguridade e que actúe seguindo os plans de continxencia en caso de ser necesario.

O nivel de medidas legais vén evidentemente establecido polas administracións autonómicas, estatais e europeas competentes na materia. Ademais da lexislación xeral, relacionada con violacións de privacidade,



roubos, etc., existe certa lexislación específica intimamente ligada coa seguridade da información.

Podemos citar a Lei orgánica 15/1999 de protección de datos de carácter persoal (LOPD) e o Real decreto 1720/2007 que a desenvolve; a Lei 59/2003, do 19 de decembro, da sinatura electrónica; a Lei 11/2007, do 22 de xuño, de acceso electrónico dos cidadáns aos servizos públicos, que introduce no seu artigo 42 o Esquema Nacional de Seguridade, desenvolvido no Real decreto 3/2010. No ámbito autonómico galego, desenvolveuse o Plan Director de Seguridade da Información e o Decreto 230/2008 de boas prácticas.

## **2.1. Plan de seguridade e plan de continxencias**

A seguridade informática debe ser unha cuestión que abranca todos os ámbitos da organización, cun grande compoñente técnico baseado en tecnoloxías da información, pero tamén político, de concienciación e formación de todo o persoal.

### **2.1.1. O plan de seguridade**

O Plan de Seguridade Informática establece os principios organizativos e funcionais da actividade de seguridade informática nunha organización. Recolle claramente as políticas de seguridade e as responsabilidades de cada un dos participantes no proceso informático, así como as medidas e procedementos que permitan previr, detectar e responder ás ameazas que gravitan sobre el.

Logo de aprobarse a política de seguridade, debe poñerse á disposición de todos os membros da organización xa que serán eles os responsables finais do seu éxito. As políticas deben ser revisadas e actualizadas anualmente (ou se é posible cada seis meses) para reflectir os cambios na organización.



Non debería haber dúas políticas de seguridade iguais, xa que cada empresa é diferente e os detalles da política dependen das necesidades exclusivas de cada unha. Sen embargo, pódese comezar cun conxunto xeral de políticas de seguridade e despois personalizalo de acordo cos requisitos específicos, limitacións de financiamento e infraestrutura existente.

Un plan de seguridade informática completo é un recurso valioso que xustifica a dedicación de tempo e esforzo á súa elaboración.

### **2.1.2. O plan de continxencias**

Un plan de continxencias é un conxunto de procedementos alternativos á orde normal da organización, cuxo fin é permitir o normal funcionamento desta, mesmo cando algunha das súas funcións for danada por un fallo de seguridade.

Que unha organización prepare os seus plans de continxencias non significa que recoñeza a ineficacia do seu plan de seguridade, senón que supón un avance para superar calquera eventualidade que poida carrexar perdas importantes.

Os plans de continxencias débense facer de cara a futuros acontecementos para os que cómpre estar preparado.

A función principal dun plan de continxencias é a continuidade das operacións da empresa. Dividimos a súa elaboración en catro etapas:

1. Avaliación.
2. Planificación.
3. Probas de viabilidade.
4. Execución.



As tres primeiras fan referencia ao compoñente preventivo e a última á execución do plan logo de acontecido o sinistro.

Cómpre crear plans de continxencias para todos os riscos de seguridade coñecidos para previr a aparición de novas ameazas ou vulnerabilidades descoñecidas que danen os nosos sistemas.

Os plans de continxencias de seguridade deben especificar claramente as accións que se realizarán de producirse un incidente para minimizar as consecuencias e a repercusión nos activos da organización. O ideal é contar con plans de resposta a incidentes e plans de continuación de negocio para garantir unha reacción eficaz durante e despois dun ataque.

## **2.2. Seguridade Física**

A seguridade física encárgase da protección contra ameazas ás instalacións, equipamentos, sistemas de comunicacións e persoal que forman parte dun sistema de información. Consiste na aplicación de barreiras físicas e procedementos de control ante ameazas aos recursos do sistema.

Resulta habitual que as organizacións se centren na seguridade lóxica, descoidando ás veces, a seguridade física, o que pode ser fonte de problemas, xa que a seguridade física afecta aos tres tipos de recursos — hardware, software e datos— e os ataques con acceso físico exitosos poden facer moito mal. Por exemplo, un acceso non autorizado que ten como resultado a desaparición dun equipo con datos sensibles da organización.

As ameazas á seguridade física poden clasificarse atendendo a diversos aspectos. Establecemos aquí tres tipos:

- Ameazas ocasionadas involuntariamente por persoas: trátase de accidentes causados por persoas de forma fortuíta por accidente ou neglixencia no uso do sistema. Estas persoas non teñen por



que ser exclusivamente traballadores técnicos. Inclúe tamén persoal de limpeza e mantemento, visitantes, etc. Na práctica tradúcese en derrames de líquidos, desconexións bruscas da rede, caídas e roturas de material, etc.

- Accións hostís deliberadas: trátase de ameazas que poden ser realmente perigosas en función da capacidade e intencións do atacante. Son accións deliberadas e usualmente planificadas contra o noso sistema ou as persoas que traballan nel. Inclúe accesos non autorizados ás instalacións, roubo, secuestros, fraudes, sabotaxes, etc.
- Desastres naturais, incendios, humidade e inundacións: son axentes externos que, de afectaren aos nosos sistemas, adoitan causar graves danos. Aínda que son infrecuentes, é necesario prevelos e telos en conta ao longo de todo o deseño e a vida útil do noso sistema.

As medidas de seguridade física permiten limitar o alcance das ameazas citadas mediante o uso de controis e procedementos de seguridade. O estándar TIA-942, aprobado pola *Telecommunications Industry Association* e por ANSI<sup>2</sup>, proporciona unha guía de recomendacións e normas para o deseño e instalación de infraestruturas de centros de procesamento de datos (CPD) que contribúen a un considerable aumento da seguridade física do sistema de información. Establece catro niveis (*tiers*) de dispoñibilidade implantados mediante medidas de carácter arquitectónico, de telecomunicacións, eléctricas e mecánicas.<sup>3</sup>

---

<sup>2</sup> Siglas en inglés de American National Standards Institute.

<sup>3</sup> Deixamos para o tema 56, dedicado ao deseño de centros de procesamento de datos, o desenvolvemento das medidas de seguridade física.



### **2.3. Seguridade Lóxica**

A seguridade lóxica fai referencia ao conxunto de operacións e técnicas destinadas á protección da información, procesos e programas contra a destrución, a modificación, a divulgación indebida ou o atraso na súa xestión..

É pouco frecuente que un ataque lóxico afecte ao hardware, aínda que existe a posibilidade de que algún ataque deste tipo poida chegar a danar algún compoñente.

Nun sistema de información, os datos constitúen un dos recursos máis importantes e valiosos. A gran variedade dos ataques, a súa posible orixe remota e as repercusións que poden ter obríganos a establecer medidas de protección semellantes ou superiores ás asumidas na seguridade física.

A seguridade lóxica suscita unha serie de requisitos intimamente relacionados coas precitadas dimensións da seguridade, entre os que podemos mencionar os seguintes:

- Asegurar que os operadores poidan traballar sen necesidade dunha supervisión minuciosa e que non poidan modificar os programas nin os arquivos que non lles correspondan.
- Asegurar que se estean utilizando os datos e os programas correctos polo procedemento correcto.
- Restringir o acceso ao software e aos datos.
- Que a información transmitida sexa recibida polo destinatario ao que lle foi enviada e non por outro.
- Prover técnicas que permitan garantir que a información non resulta alterada durante un proceso de transmisión.



- Procurar que os sistemas de comunicación entre os distintos compoñentes do sistema de información estean redundados.
- Manter un rexistro das operacións levadas a cabo no sistema xunto co usuario que as realiza para poder realizar o seu seguimento en caso de ser necesario.

### 2.3.1. Ataques contra a seguridade lóxica

A seguridade lóxica pode ser vulnerada por medio de ataques moi heteroxéneos e que, ademais, evolucionan constantemente. Dada a gran diversidade de sistemas operativos, aplicacións e protocolos, é imposible determinar o seu número, que segue a aumentar día a día.

Por outra banda, a transmisión de información a través de sistemas de alleos á organización, como internet, resulta cada vez máis frecuente e o volume de información transmitido aumenta tamén dun xeito notable. Estas operacións de transmisión de información a través dunha canle non controlada pola organización e compartida con terceiros son especialmente vulnerables aos ataques destes.

Podemos clasificar estes ataques atendendo ao seu obxectivo da seguinte maneira:

1. **Ataques Pasivos:** a información non resulta alterada, senón que o atacante unicamente a captura ou monitorea para obter os datos que están a ser transmitidos.

Os obxectivos destes ataques son a interceptación de datos e a análise de tráfico coas seguintes finalidades:

- o Recompilación de datos sobre contas de usuarios e claves de acceso para utilizar máis tarde en ataques activos.



- o Obtención da orixe e destinatario da comunicación a través da lectura das cabeceiras dos paquetes monitorados.
  - o Monitorización do volume de tráfico intercambiado entre as entidades obxecto do ataque, conseguindo así información acerca de actividade ou inactividades inusuais.
  - o Monitorización das horas habituais de intercambio de datos entre as entidades da comunicación para extraer información acerca dos períodos de actividade.
2. **Ataques activos:** ataques implican algún tipo de modificación dos datos ou a creación de datos falsos. Adoitan ser intencionados e realizados por persoas con coñecementos e consciencia do que están facendo.

Pódense subdividir en varias categorías atendendo ás accións que se realizan durante o ataque:

- o **Interceptación:** un elemento non autorizado consegue un acceso a un determinado obxecto do sistema, pero este non é modificado en ningún modo. Se se trata dunha comunicación, esta chegará ao seu destino sen constancia da interceptación.
- o **Destrucción:** algúns autores consideran un caso especial da modificación a destrución, entendéndoa como unha modificación que inutiliza o obxecto afectado.
- o **Modificación:** Se ademais de lograr unha interceptación, o ataque consegue modificar o obxecto de datos.
- o **Interrupción:** un ataque clasifícase como interrupción se fai que un obxecto do sistema se perda, quede inutilizable ou non dispoñible.



- o **Fabricación:** modificación destinada a suplantar o obxecto real.

Como dicíamos, existe un elevado número de tipos de ataques contra a seguridade lóxica. Citamos deseguido algúns deles:<sup>4</sup>

- **Enxeñería Social:** consiste en manipular as persoas do contorno para obter acceso aos sistemas ou a información confidencial que facilite outros ataques técnicos.
- **Enxeñería Social Inversa:** o intruso dá a coñecer dalgún xeito que é capaz de brindarlles axuda aos usuarios, e estes chámanos ante algún imprevisto. O intruso aproveitará a oportunidade para pedir información necesaria para solucionar o problema, conseguindo de paso información útil para realizar ataques.
- **Shoulder-surfing<sup>5</sup>:** consiste en espiar fisicamente aos usuarios namentres introducen o seu nome de usuario e clave de acceso correspondente, ou cando están accedendo a información restrinxida.
- **Piggybacking:** relacionado co anterior. Hoxe en día fai referencia ao uso de redes *wireless* alleas, pero o seu significado orixinal era o de “coarse” nun lugar detrás doutra persoa.
- **Masquerading/spoofing** (Suplantación): suplantación electrónica ou física de persoas autorizadas para acceder ao sistema ou obter información relevante sobre el. Dependendo do obxecto da suplantación, poderíamos falar de *IP spoofing*, *DNS spoofing*, *web spoofing*, etc.

---

<sup>4</sup> O RFC 4949 (*Internet Security Glossary, Version 2*) proporciona unha listaxe máis extensa e descritiva.

<sup>5</sup> Expresión en inglés que se refire a mirar por enriba do ombro.





- **Scavenging:** paradoxalmente, un dos ataques máis efectivos. Consiste en inspeccionar os refugallos en papeleiras, colectores, etc., en busca de información sensible.
- **Exploits:** aproveitamento de erros coñecidos (*bugs*) en determinadas versións do software instalado no sistema de información que poden ser usados como porta de entrada de ataques.
- **Escaneo de portos:** técnicas de ataque pasivas que analizan as máquinas dun sistema para determinar cal é o estado dos portos (aberto/pechado). Nalgúns casos permiten coñecer mesmo cal é o sistema operativo da máquina e incluso que software e que versión do mesmo se encontra escoitando en cada porto.
- **Wiretapping:** un tipo de ataque que intercepta e accede á información que se transmite por unha canle de comunicacións. O nome do termo (que poderíamos traducir como “intervir cables”) ten como orixe a intervención de teléfonos que se facía de forma mecánica. Hoxe en día fai referencia á captura de datos mediante calquera técnica, con ou sen cables polo medio.
- **Eavesdropping-packet sniffing:** un tipo de ataque *wiretapping* pasivo. É a interceptación pasiva do tráfico de rede. Realízase con aplicacións chamadas *sniffers*, que son programas que capturan paquetes de datos que circulan pola rede. Pódese facer colocando o *sniffer* nun dos equipos pertencentes á rede ou conseguindo conectar un equipo externo á ela. Este último caso é a forma máis habitual en redes *wireless*.
- **Man-in-the-middle:** un tipo de ataque *wiretapping* activo. O atacante intercepta e modifica selectivamente os paquetes de



datos capturados para simular ser un dos participantes nunha comunicación allea.

- **Denegación de servizo:** coñecido habitualmente polas súas siglas en inglés, DoS (*Denial of Service*); este tipo de ataques consiste en conseguir que o obxectivo do ataque deixe de realizar a súa función de modo temporal ou definitivo. O obxectivo do ataque pode ser unha máquina, unha rede de comunicacións, un servizo concreto, etc. É dun dos tipos de ataques máis comúns e efectivos. Na práctica existen múltiples variedades deste ataque, das que podemos citar: *Flooding*, *ICMP flood*, *Syn flood*, *ping of death*, *land*, *smurf*, *teardrop*, etc.
- **Denegación de Servizo Distribuída:** un caso especial do anterior, tamén coñecido polas súas siglas en inglés, DDoS (*Distributed Denial of Service*). Neste caso o ataque de DoS non provén dun único atacante, senón que provén de moitos á vez de forma coordinada. O conxunto de atacantes pode non ser realmente un conxunto de persoas, xa que é habitual o uso de máquinas secuestradas (“máquinas zombis”) que participan no ataque, moitas veces sen que o seu lexítimo dono teña coñecemento.
- **Ataques de secuestro (*Hijack*):** céntranse no secuestro dalgún elemento nunha comunicación previamente establecida pola vítima ou dun recurso vital dunha máquina. Os obxectivos do secuestro poden ser a sesión dun usuario xa autenticado nun sistema, o propio navegador da vítima ou ata unha páxina ofrecida por un servidor para modificala e facer que os datos inseridos nela lle sexan enviados a unha máquina baixo o control do atacante, por exemplo.



- **Tamper:** ataque consistente en realizar modificacións na configuración dunha máquina ou sistema obxectivo que degraden o nivel de seguridade destes.
- **Phishing:** é, en realidade, un tipo de ataque de tipo *masquerading* que creceu en frecuencia nos últimos anos. É un ataque que tenta adquirir información sensible do obxectivo (número de contas bancarias, nomes de usuario e claves de acceso, etc.) mediante unha solicitude fraudulenta nun correo electrónico ou páxina web que o atacante construíu para simular ser unha entidade ou persoa de confianza do obxectivo.
- **SQL injection:** é un tipo de ataque por inserción de código dirixido á base de datos en linguaxe SQL. A técnica consiste en inserir secuencias concretas que son sintacticamente correctas en SQL en campos de texto de aplicacións (usualmente web) para executar consultas fraudulentas. Aínda que é un ataque perigoso, é facilmente contrarrestable filtrando adecuadamente as entradas.
- **Cross-site request forgery:** abreviado habitualmente como CSRF ou XSRF, é tamén coñecido como “ataque dun click”. Baséase en tirar partido da confianza que un sitio web ten no navegador web dun usuario. A vítima é enganada para usar un hiperenlace manipulado polo atacante. A manipulación consiste na construción dun obxecto de petición (*request*) que realiza directamente unha acción fraudulenta e involuntaria para o usuario que a executa. A clave reside en que o sitio web debe confiar no usuario (porque xa se autenticou previamente, por exemplo) e executar a petición directamente; de aí o nome de “ataque nun click”.



- **Cross site scripting:** abreviado habitualmente como XSS. Trátase de calquera ataque que permita executar código de *scripting* (*VBScript, Javascript, etc.*) inserido polo atacante no contexto dun sitio web.

### 2.3.2. A protección da seguridade lóxica

Existen medidas de protección que poden e deben ser utilizadas para aumentar a seguridade dos nosos sistemas de información.

Máis adiante veremos que os niveis de seguridade poden ser controlados e xestionados mediante a análise de riscos, polo que neste apartado nos limitaremos a enunciar as medidas organizativas e técnicas que podemos aplicar para aumentar a nosa seguridade.

Os ataques á seguridade lóxica adoitan basearse en realidade en fallos de deseño inherentes a internet (ou aos seus protocolos), ou aos sistemas operativos utilizados. A continua aparición de novas tecnoloxías fai que o número de tipos de ataques tamén aumente.

Polo tanto, os responsables de seguridade e administradores dos sistemas deben manterse actualizados respecto dos novos ataques e de como protexerse contra eles. E, por suposto, é de vital importancia que manteñan actualizado o sistema operativo e todo o software utilizado nas máquinas do sistema.

Unha máquina que conteña información que non sexa considerada valiosa, debe terse en conta igualmente á hora de definir as políticas de seguridade xa que pode resultar útil para un atacante á hora de empregala nun ataque de denegación de servizo distribuído (DDoS) ou de utilizala como paso intermedio para ocultar o verdadeiro enderezo do atacante

Cómpre realizar auditorías de seguridade de forma periódica e valorar a posibilidade de implantar sistemas de xestión da seguridade da



información (SXI) que garantan a calidade dos nosos sistemas e medidas de seguridade.

Un dos ataques con maior taxa de éxito de todos os citados é o da enxeñería social. De nada serve que teñamos o noso sistema perfectamente actualizado e que contemos coas medidas de seguridade físicas e lóxicas máis avanzadas se mediante unha chamada telefónica un atacante pode conseguir que un usuario lle transmita o seu usuario e clave de acceso.

A defensa contra ataques de enxeñería social pasa por manter aos usuarios do sistema informados e formados en materia de seguridade. Cómpre mantelos alerta e inculcar neles un espírito de desconfianza que permita evitar ataques baseados na enxeñería social ou o *phishing*

Algunhas ferramentas e técnicas de protección da seguridade que podemos aplicar nos nosos sistemas de información son:

- **Uso de sistemas operativos seguros:** o mercado ofrece diferentes sistemas operativos que poden ser usados nos nosos sistemas. Como é de supoñer, non todos eles ofrecen o mesmo nivel de seguridade. É conveniente usar sistemas operativos con niveis de seguridade acordes ás necesidades da nosa estratexia de seguridade. Existe unha catalogación dos niveis de seguridade que ofrece un sistema operativo determinada polo estándar unificado *Common Criteria for Information Technology Security Evaluation* (simplificado habitualmente como *Common Criteria*).
- **Copias de seguridade (backups):** o *backup* de datos permite ter dispoñible unha copia íntegra dos datos en caso de perda ou corrupción destes tras un ataque ou un accidente. É conveniente deseñar coidadosamente a política e o procedemento de creación, transporte e almacenamento das copias de seguridade dos datos e tamén dos programas usados na organización.



- **Devasas** (*firewalls*): son sistemas, hardware ou software, destinados a filtrar o tráfico que circula polos sistemas de comunicacións. É habitual utilizalos para controlar o tráfico que entra e sae da nosa organización para evitar así algúns tipos de ataques baseados en manipulación de paquetes. Ademais, permiten tamén controlar as aplicacións e protocolos que son utilizados polos empregados.
- **Sistemas de Detección de Intrusos**: coñecidos polas súas siglas en inglés, IDS. Un IDS é capaz de recoller e utilizar información dos eventos ocorridos no sistema para detectar patróns de ataques e alertar ao administrador de posibles ataques. Algúns tipos de IDS son mesmo capaces de levar a cabo accións reactivas destinadas a abortar os ataques detectados.
- **Programas antivirus**: son programas destinados á detección e eliminación de virus informáticos. Convén contar con antivirus en todos os equipos do sistema de información. Para que os antivirus sexan efectivos é vital que estean actualizados.
- **Ferramentas de seguridade**: como xa se comentou, poden ser de grande utilidade para xestionar a seguridade e configurar os nosos sistemas correctamente, aumentando así a súa protección.
- **Encriptación da información**: dado que por moitas medidas de seguridade que tomemos seguiremos correndo o risco de sufrir accesos non autorizados á nosa información, é unha boa práctica usar a encriptación para aumentar a protección da información sensible. A encriptación débesele aplicar tanto á información que se transmite a través de sistemas de comunicación como á aquela de especial importancia, aínda que esta non saia do noso sistema.

### 3. IDENTIFICACIÓN E AUTENTICACIÓN



Un aspecto importante da seguridade informática é a identificación de calquera persoa, máquina ou servizo que pretenda acceder ao sistema restrinxindo os privilexios de acceso e execución a aqueles que previamente fosen definidos polo equipo de administración.

Podemos dividir o proceso de autenticación nun sistema informático en dúas fases:

- A **identificación** proporcióname unha identidade a cada usuario do sistema asignándolle un identificador único. No momento de ingresar no sistema, buscaranse entre os identificadores de usuario rexistrados os privilexios de acceso do que pretende identificarse.
- A **autenticación** é o procedemento de comprobación da identidade dun usuario que deberá presentar probas de que é quen di ser.

O modo en que un usuario pode demostrar a súa identidade recae nun ou máis dos seguintes factores:

- o algo que o usuario sabe (por exemplo unha clave de acceso)
- o algo que ten (por exemplo unha tarxeta de acceso)
- o algo que é parte do propio usuario (por exemplo medidas biométricas como a impresión dactilar).

Considérase que para que unha autenticación sexa realmente segura debe involucrar polo menos elementos de dous dos tres (se non dos tres) factores.

Por exemplo, a arquitectura de sinatura electrónica con certificados emitidos por entidades fiables (Autoridades de Certificación) é un sistema de autenticación multifactor, xa que primeiro debemos posuír



un documento que nos identifique para poder obter un certificado avalado por un terceiro e coñecer un código persoal para realizar as sinaturas. O uso de contrasinais asociados a nomes de usuario considérase xuridicamente válido para os efectos de cumprir cos requisitos de identificación e autenticación.

A autorización ou control de acceso é o proceso polo cal unha identidade debidamente autenticada recibe os privilexios de acceso concretos e asociados ao sistema no que se autenticou.

Os chamados sistemas *Single Sign-On* (SSO) permiten que unha identidade que se autenticou correctamente poida ser compartida por varios sistemas, de modo que o usuario poida pasar a traballar dun a outro sen ter que autenticarse varias veces mentres a súa sesión continúe aberta. Exemplos deste tipo de sistemas son o sistema *Kerberos*, o proxecto *OpenID* ou o proxecto *CAS* (*Central Authentication Service*).

#### **4. CONTROL DO FLUXO DE DATOS**

Ao transmitir información a través dunha canle de comunicacións prodúcese un fluxo de datos que pode ser alterado por interferencias ou ataques que fagan que os datos recibidos nun extremo difiran dos emitidos desde o outro.

Os mecanismos de control del fluxo de datos permiten detectar e corrixir estas alteracións mantendo así a integridade da mensaxe transmitida.

##### **4.1. Técnicas de control de fluxo de datos**

O control de fluxo de datos pasa por aumentar o tamaño da información enviada engadindo datos adicionais calculados a partir dos orixinais que permitan realizar comprobacións no destino. Existen dúas estratexias básicas para realizar este control:



- **Códigos de detección de erros:** a información transmitida codifícase incluíndo só a información redundante necesaria en cada bloque de datos para detectar os erros. Neste caso o número de bits de redundancia é menor.
- **Códigos de corrección de erros:** a información codifícase engadindo bits de redundancia en cada bloque que, non só permiten detectar erros, senón tamén corríxilos.

#### 4.1.1. Paridade simple (paridade horizontal)

Consiste en engadir un bit a cada bloque que queremos enviar que tomará o valor 0 ou 1 dependendo de se o número de bits con valor 1 é par ou impar, respectivamente. O bit engadido denomínase bit de paridade.

O receptor, ao recibir o bloque, contará o número de bits con valor 1 e comprobará se coincide co valor de bit de paridade asumindo que se coincide significa que non houbo erro. Este sistema detectará unicamente erros que afecten a un número impar de bits.

#### 4.1.2. Paridade cruzada (paridade horizontal-vertical)

Este método trata de mellorar o anterior enviando un bloque de paridade cada N bloques de datos. O bloque de paridade serve para comprobar a paridade dos n bloque de datos enviados inmediatamente antes ca el. A técnica consiste en calcular a paridade dos bits que ocupan a posición P en cada un dos bloques e realizar a comprobación de paridade contra o bit que ocupa a posición P do bloque de paridade.

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):

Os códigos cíclicos aparecen intentando mellorar os códigos que só controlan a paridade a nivel de bit. Estes códigos utilizan a aritmética modular para detectar unha maior cantidade de erros. Úsanse operacións en módulo 2 e as sumas e restas realízanse sen carrexo (converténdose en operacións de tipo Or-Exclusivo ou XOR).

Ademais, para facilitar os cálculos trabállase, aínda que só teoricamente, con polinomios. A finalidade deste método é crear unha parte de redundancia que se engade ao final do código que se vai transmitir, que sendo a máis pequena posible, detecte o maior número de erros.

O método debe ser sistemático, é dicir, que cun mesmo código para transmitir (e un mesmo polinomio xerador) se xere sempre o mesmo código final. O polinomio xerador é un polinomio elixido previamente e que ten como propiedade minimizar a redundancia. Adoita ter unha lonxitude de 16 bits, para mensaxes de 128 bytes, o que indica que a eficiencia é boa xa que só incrementa a lonxitude nun aproximado 1,6%.

O polinomio usado habitualmente nas redes WAN é  $g(x) = x^{16} + x^{12} + x^5 + x$ , e os cálculos que realiza o equipo transmisor para calcular o seu CRC son:

1. Engadirille tantos ceros pola dereita á mensaxe orixinal como o grao do polinomio xerador.
2. Divide a mensaxe cos ceros incluídos entre o polinomio xerador.
3. O resto que se obtén da división súmase á mensaxe cos ceros incluídos.



4. Envíase o resultado obtido.

O receptor da mensaxe realiza as seguintes operacións:

5. Mediante o protocolo usado determina o polinomio xerador utilizado.
6. Divide o código recibido entre o polinomio.
7. Comproba o resto desa operación. Se o resto é cero, non se produciron erros. En caso de que o resto sexa un, o código recibido contén erros.

Este método pode chegar a detectar gran cantidade de erros:

- o Erros simples: todos
- o Erros dobres: todos
- o Erros nas posicións impares dos bits: todos
- o Erros en refachos cunha lonxitude menor que o grao do polinomio xerador: todos
- o Outros refachos: unha porcentaxe elevada e próxima ao 100%

#### **4.1.4. Suma de comprobación (*checksum*)**

É un método sinxelo, pero eficiente só con cadeas de palabras dunha lonxitude pequena. É por iso que se adoita utilizar en cabeceiras de tramas ou outras cadeas importantes e en combinación con outros métodos. Consiste en agrupar a mensaxe a transmitir en cadeas dunha lonxitude determinada  $L$  non moi grande, de por exemplo 16 bits. Considerando cada cadea como un número enteiro numerado segundo o sistema de numeración  $2^L - 1$ . A continuación súmase o valor de todas as palabras nas que se divide a mensaxe, e engádese o resultado á mensaxe que se vai



transmitir, pero cambiado de signo. Con isto, o receptor o único que ten que facer é sumar todas as cadeas, e se o resultado é 0 non hai erros.

#### **4.1.5. Distancia de Hamming**

Os códigos de Hamming baséanse como os anteriores en incluír información redundante que permite detectar erros e mesmo corrixir algúns deles.

A distancia de Hamming consiste en asignar valores ás operacións (substitución, borrado, engadir, etc.) necesarias para transformar un conxunto de bits (unha palabra) noutro.

Se queremos detectar  $d$  bits erróneos nunha palabra de  $n$  bits, podemos engadir a cada palabra de  $n$  bits  $d+1$  bits predeterminados ao final, de forma que quede unha palabra de  $n+d+1$  bits cunha distancia mínima de Hamming de  $d+1$ . Deste xeito, se un recibe unha palabra de  $n+d+1$  bits que non encaixa con ningunha palabra do código (cunha distancia de Hamming  $x \leq d+1$  a palabra non pertence ao código) detéctase correctamente que é unha palabra errónea. Aínda máis,  $d$  ou menos erros nunca se converterán nunha palabra válida debido a que a distancia de Hamming entre cada palabra válida é de polo menos  $d+1$ , e tales erros conducen soamente ás palabras inválidas que se detectan correctamente. Dado un conxunto de  $m*n$  bits, podemos detectar  $x \leq d$  bits erróneos correctamente usando o mesmo método en todas as palabras de  $n$  bits. De feito, podemos detectar un máximo de  $m*d$  erros se todas as palabras de  $n$  bits son transmitidas cun máximo de  $d$  erros.

Outros códigos detectores e correctores de erros son a *corrección de erros cara a diante* (en inglés, *Forward Error Correction* ou *FEC*) ou o *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**

**Enxeñeiro en Informática no Concello de Ourense**

**Colexiado do CPEIG**

---

<sup>6</sup> Todas as ligazóns foron verificadas en novembro do 2011.





## **37. SEGURIDADE EN REDES: REDES PERIMETRAIS. DMZ. CORTALUMES. INTRUSIÓNS. ACCESOS NON AUTORIZADOS. TÉCNICAS DE SEGURIDADE PREVENTIVAS E REACTIVAS.**





**Tema 37. Seguridade en redes: redes perimetrais. DMZ. Devasas. Intrusións. Accesos non autorizados. Técnicas de seguridade preventivas e reactivas.**

ÍNDICE

<b>1 SEGURIDADE EN REDES: REDES PERIMETRAIS. DMZ.....</b>	<b>2</b>
1.1 Arquitectura de seguridade no modelo OSI.....	3
1.2 Redes perimetrais.....	8
<b>2 TÉCNICAS DE SEGURIDADE PREVENTIVAS E REACTIVAS. DEVASAS. INTRUSIÓN. ACCESOS NON AUTORIZADOS.....</b>	<b>15</b>
2.1 Devasas.....	16
2.2 Sistemas de detección de intrusións (IDS/IPS).....	20
2.3 Redes privadas virtuais (RPV).....	22
<b>3 REFERENCIAS.....</b>	<b>28</b>



## **1 SEGURIDADE EN REDES: REDES PERIMETRAIS. DMZ**

A irrupción dos sistemas informáticos nas organizacións provocou un cambio importante no modo de garantir a **seguridade da información**. Antes do uso estendido dos sistemas informáticos, a documentación sensible atopábase en soporte papel, polo que a seguridade da información para a organización era provista nun sentido físico e administrativo. Un claro exemplo disto eran os armarios con pechadura de combinación utilizados para almacenar os documentos con información sensible e os procesos de selección utilizados para incorporar novo persoal.

A implantación dos sistemas informáticos trouxo consigo a necesidade de ferramentas automatizadas para a protección de arquivos e outra información almacenada nos sistemas. Na actualidade, a información está almacenada en sistemas compartidos e distribuídos que ofrecen a posibilidade de acceder a ela a través dunha rede privada de ordenadores ou a través de internet. As **medidas de seguridade na rede** son o conxunto de medidas adoptadas para protexer os datos durante a transmisión a través de redes non seguras. Con todo, debido a que a práctica totalidade das empresas, gobernos e organizacións académicas teñen interconectados os seus sistemas informáticos cunha colección de redes que, pola súa vez, están interconectadas entre elas, dando lugar ao que coñecemos como internet, é máis frecuente o uso do termo **seguridade en internet**. A seguridade en internet abarca a prevención, detección e corrección de violacións da seguridade que poidan afectar á transmisión de información.

O *Computer Security Handbook* do NIST [NIST95] define o termo **computer security** como a protección conferida a un sistema informático co fin de alcanzar os obxectivos de preservar a integridade, dispoñibilidade e confidencialidade dos recursos de información do sistema (incluíndo



hardware, software, firmware, información/datos e telecomunicacións). Esta definición abrangue os obxectivos centrais da seguridade dos sistemas de información: confidencialidade, integridade, dispoñibilidade, autenticación e trazabilidade.

### 1.1 **Arquitectura de seguridade no modelo OSI**

Se a seguridade en contornos de procesamento de datos pechados é complexa, o uso de redes de área local e extensa aumenta esa complexidade de forma considerable. Por iso, o administrador responsable da seguridade dunha organización necesita algunha metodoloxía que lle permita definir os requisitos de seguridade e identificar os mecanismos que contribúan a cumprilos. Esta metodoloxía deberá facilitar a cobertura efectiva das necesidades de seguridade da organización así como a avaliación e posterior elección dos distintos produtos e políticas.

UIT-T<sup>1</sup> Recomendación X.800, *Security Architecture for OSI*, describe os servizos de seguridade básicos que poden ser aplicados cando é necesario protexer a comunicación entre sistemas. Aínda que se trata dun modelo xenérico definido nos anos noventa, os seus conceptos e definicións aínda seguen vixentes no día a día dos administradores da seguridade. A arquitectura de seguridade do modelo OSI é útil para os administradores, xa que establece un protocolo para organizar a tarefa de proporcionar seguridade. Ademais, como esta arquitectura foi desenvolvida como un estándar internacional, os fabricantes de computadoras e sistemas de comunicación engadíronlles características de seguridade aos seus produtos e servizos que se relacionan con esta definición estruturada de mecanismos e servizos. O modelo OSI de arquitectura da seguridade céntrase nos seguintes conceptos: **mecanismos de seguridade**,

---

<sup>1</sup> UIT-T son as siglas da Unión Internacional das Telecomunicacións, Sector de Estandarización das Telecomunicacións, que é unha axencia patrocinada polas Nacións Unidas que desenvolve estándares, chamadas “Recomendacións”, relacionadas coas telecomunicacións.



**servizos de seguridade e ataques contra a seguridade.** Pódense definir de maneira resumida:

- **Mecanismos de seguridade:** son os procesos que permiten detectar, previr, ou recuperarse fronte a un ataque contra a seguridade.
- **Servizos de seguridade:** un servizo de comunicación ou procesamento que incrementa a seguridade dos sistemas de información e as transferencias de datos realizadas por unha organización. Os servizos tentan previr os ataques contra a seguridade facendo uso dun ou varios mecanismos de seguridade.
- **Ataques contra a seguridade:** calquera acción que atenta contra a seguridade da información da organización.

#### 1.1.1 Mecanismos de seguridade

Os mecanismos de seguridade en X.800 divídense nos que se aplican a unha capa de protocolo específico e nos que non son específicos dunha capa de protocolo ou servizo de seguridade (coñecidos tamén como mecanismos de seguridade persistentes).

**Mecanismos específicos de seguridade:** poden ser incorporados nunha das capas do protocolo co fin de proporcionar algún dos servizos de seguridade OSI:

- **Autenticación:** corrobora que unha entidade, ben sexa orixe ou destino da información, é a desexada; por exemplo, A envía un número aleatorio cifrado coa clave pública de B, B descifra coa súa clave privada e reenvíallo a A, demostrando así que é quen pretende ser. Por suposto, hai que ser coidadoso á hora de deseñar estes protocolos, xa que existen ataques para desbaratalos.
- **Control de accesos:** esforzo para que só aqueles usuarios autorizados accedan aos recursos do sistema ou á rede, por exemplo, mediante os contrasinais de acceso.



- **Sinatura dixital:** consiste en achegar unha serie de datos nunha mensaxe ou realizar unha transformación criptográfica que permita que o receptor comprobe a orixe dunha mensaxe e verifique a súa integridade.
- **Cifrado:** consiste na transformación da información por medio de algoritmos matemáticos a un formato que non é intelixible. A transformación e recuperación da información depende dun algoritmo de cifrado e/ou do uso de claves de cifrado.
- **Notarización:** o uso dun terceiro de confianza para garantir certas propiedades nun intercambio de datos.
- **Integridade:** conxunto de mecanismos para garantir que unha unidade de datos ou un fluxo de datos son correctos e completos.
- **Tráfico de recheo:** consiste en enviar tráfico espurio xunto cos datos válidos para que o atacante non saiba se se está a enviar información, nin qué cantidade de datos útiles se está a transmitir.
- **Control de encamiñamento:** permite a selección, para certos datos, de rutas físicas seguras e a variación destas, especialmente cando se sospeita dunha violación da seguridade.

Dentro dos **mecanismos de seguridade persistentes** temos:

- **Rexistro de auditoría de seguridade:** datos recollidos e potencialmente utilizables para realizar unha auditoría de seguridade.
- **Etiquetas de seguridade:** os atributos ou propiedades de seguridade asociadas a un recurso ou unidade de datos.
- **Funcionalidade de confianza:** o que se debe percibir como correcto con respecto a algún criterio (por exemplo, segundo o establecido por unha política de seguridade).



- **Detección de eventos:** detección de eventos relevantes de seguridade.
- **Recuperación da seguridade:** ocúpase das peticións dos mecanismos, tales como o manexo de eventos e a xestión de funcións, e leva a cabo tarefas de recuperación.

### 1.1.2 Servizos de Seguridade

X.800 define un servizo de seguridade como un servizo provisto por unha capa do protocolo de comunicación e que garante a adecuada seguridade dos sistemas ou transferencias de datos. Se cadra atopamos unha definición máis clara na RFC 4949, que presenta a seguinte definición: un servizo de información ou comunicación que é proporcionado por un sistema para unha clase específica de protección de recursos informáticos. X.800 divide estes servizos en cinco categorías:

- **Confidencialidade:** require que a información sexa accesible unicamente polas entidades autorizadas. A confidencialidade de datos aplícase a todos os datos intercambiados polas entidades autorizadas ou se cadra só a porcións ou segmentos seleccionados dos datos, por exemplo, mediante cifrado. A confidencialidade de fluxo de tráfico protexe a identidade da orixe e destino(s) da mensaxe —por exemplo, enviando os datos confidenciais a moitos destinos ademais do verdadeiro—, así como o volume e o momento de tráfico intercambiado —por exemplo, producindo unha cantidade de tráfico constante ao engadir tráfico espurio ao significativo—, de forma que sexan indistinguibles para un intruso. A desvantaxe destes métodos é que incrementan drasticamente o volume de tráfico intercambiado, repercutindo negativamente na dispoñibilidade do ancho de banda baixo demanda.
- **Servizo de autenticación:** require unha identificación correcta da orixe da mensaxe, asegurando que a entidade non é falsa.



Distínguense dous tipos: de entidade, que asegura a identidade das entidades participantes na comunicación mediante biométrica (pegadas dactilares, identificación de iris, etc.), tarxetas de banda magnética, contrasinais, ou procedementos semellantes; e de orixe de información, que garante que unha unidade de información procede de certa entidade, sendo a sinatura dixital o mecanismo máis estendido.

- **Control de accesos:** no contexto de seguridade informática, o control de acceso é a capacidade de controlar e limitar o acceso a través da rede aos sistemas e aplicacións. Para logralo, cada entidade que tenta conseguir acceso debe ser autenticada, polo que os dereitos de acceso se poden adaptar a cada usuario.
- **Integridade:** igual que a confidencialidade, a integridade pódese aplicar a un fluxo de mensaxes, a unha única mensaxe, ou a un conxunto de campos seleccionados dunha mensaxe. De novo, o enfoque máis sinxelo e útil é a protección total do fluxo de comunicación. Un servizo de integridade **orientado a conexión** traballa con fluxos de mensaxes e garante que as mensaxes son recibidas tal e como son enviadas, sen ser duplicadas, modificadas, reordenadas ou repetidas.
- **Non repudio:** o non repudio evita que o emisor ou receptor dunha mensaxe poida negar a transmisión. Así, cando se envía a mensaxe, o receptor pode probar que o suposto emisor fixo o envío. Do mesmo xeito, cando unha mensaxe é recibida, o emisor pode probar o feito de que a mensaxe efectivamente foi recibida.

Tanto X.800 como RFC 4949 definen a **dispoñibilidade** como a propiedade dun sistema ou recurso de ser usado e estar dispoñible baixo un sistema de autorización de entidade, de acordo coas especificacións de rendemento para o sistema. É dicir, o sistema está dispoñible se



proporciona os seus servizos de acordo co deseño do sistema cada vez que os usuarios o solicitan. Unha gran variedade de ataques poden producir a perda ou redución da dispoñibilidade. Algúns deses ataques poden ser evitados con medidas automáticas, tales como a autenticación e o cifrado, mentres que outros precisan dalgún tipo de acción física para previr ou recuperarse dunha perda de dispoñibilidade nos elementos dun sistema distribuído.

A *Táboa 1* indica a relación entre os servizos de seguridade e os mecanismos de seguridade.

### **1.1.3 Ataques contra a seguridade**

Unha forma útil de clasificar os ataques contra a seguridade, utilizada tanto en X.800 como na RFC 4949, é por medio dos termos *ataque activo* e *ataque pasivo*.

Un ataque pasivo intenta coñecer ou facer uso da información do sistema pero sen afectar aos seus recursos. Os ataques pasivos son moi difíciles de detectar, xa que non provocan ningunha alteración dos datos. Non entanto, é posible evitar que teñan éxito mediante o cifrado da información e outros mecanismos que se han ver máis adiante.

Un ataque activo intenta cambiar os recursos do sistema ou alterar o seu modo de funcionamento. Os esforzos contra os ataques pasivos céntranse na prevención máis que na detección, namentres que fronte aos ataques activos o máis importante é recuperarse canto antes de calquera interrupción ou atraso causado.

## **1.2 Redes perimetrais**

O concepto de rede perimetral fai referencia ao conxunto de dispositivos e/ou mecanismos técnicos que son utilizados para separar a rede interna dunha organización (coñecida como *intranet* ou rede corporativa) do resto de redes externas (habitualmente a Internet).



### 1.2.1 Segmentación de redes

En organizacións cunha infraestrutura de rede moderada, recoméndase segmentar as redes en subredes co obxectivo de illar ao máximo os danos que se poidan producir pola intrusión dun atacante no sistema.

Se un atacante consegue acceso á rede da organización, e esta non está segmentada, terá acceso á totalidade dos seus equipos, aumentando a gravidade dos danos que poderá causar.

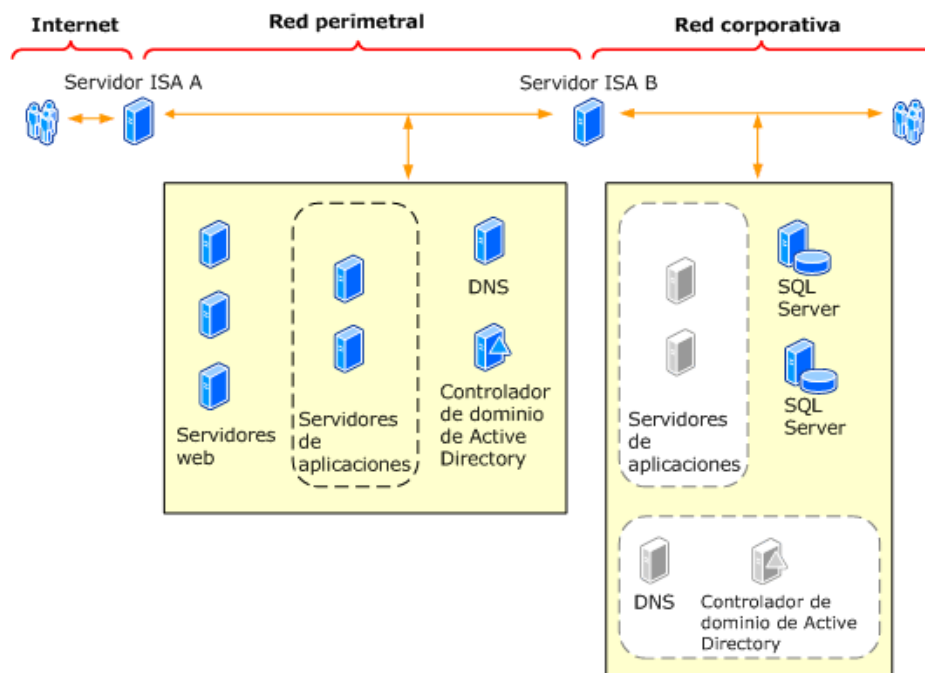
Se, pola contra, se dispón dunha rede segmentada en subredes, un atacante quedará nun segmento, podendo acceder a todos os recursos contidos nel, pero non ao resto dos recursos da organización, o que minimizará os danos.

Parece lóxico segmentar a rede en función do diferente nivel de control que require a rede interna en comparación coa comunicación co exterior, e da necesidade de establecer distintos niveis de control de acceso nunha e outras redes.

Porén, determinar onde remata exactamente a rede interna pode non ser una tarefa trivial. O auxe das comunicacións e a dispersión xeográfica de moitas organizacións complican a tarefa de determinar e protexer a rede propia.

A seguinte figura ilustra o concepto de rede perimetral e o seu contexto de rede interna e externa. Nela pódese observar que o concepto básico sobre o que xira a seguridade perimetral consiste en separar aqueles servidores que poden ser accedidos dende o exterior daqueles que son accesibles só dende o interior, de modo que os controis de seguridade de cada subrede poidan ser deseñados á medida das necesidades, habitualmente moi diferentes, de cada unha delas.





Fonte: <http://technet.microsoft.com>

O termo seguridade perimetral é moi amplo e tivo diversas atribucións ao longo do tempo. O perímetro como tal, está formado polas máquinas e os dispositivos que se sitúan na fronteira da nosa rede, onde esta interactúa co exterior, con outras redes.

Non obstante, a miúdo o perímetro xa non está circunscrito a unha sala e a seguridade física resulta insuficiente para protexer a nosa rede. É necesario incorporar certas medidas lóxicas para controlar os accesos.

O establecemento de VLAN (*Virtual Local Area Network*) permite crear redes lóxicamente independentes dentro dunha mesma rede física. Varias VLAN poden coexistir nun único equipo físico ou nunha única rede física. Son útiles para reducir o tamaño do dominio de difusión (*broadcast*) e axudan na administración da rede separando segmentos da rede interna mediante un criterio concreto (por exemplo por Departamentos). Deste xeito, pódese afirmar que os traballadores acceden unicamente aos recursos da súa rede, denegando o acceso aos recursos do resto.



Por outra parte na medida que aumenta o uso das arquitecturas orientadas ao servizo ou SaaS (do inglés *Software as a Service*) (ademais da virtualización e a computación na nube ou *cloud computing*) e na medida que se fai máis común o uso de ferramentas dirixidas á web 2.0 (redes sociais, blogs, wikis, etc.), as políticas baseadas en portos e protocolos, tradicionalmente utilizadas para o control do perímetro, son cada vez menos efectivas. Por iso son necesarias novas ferramentas para facer fronte á evolución do comportamento dos usuarios, ás novas formas en que os procesos de negocio utilizan as TIC e aos cambios nos mecanismos dos ataques que comprometen os sistemas de información.

### 1.2.2 O perímetro difuso

Na actualidade, o perímetro estendeuse e converteuse nunha fronteira dinámica (o que algúns denominan perímetro difuso), no que tamén se fai necesario protexer, e protexerse de, cada dispositivo que pode ocasionalmente formar parte del.

As medidas lóxicas de control de acceso e vixilancia das comunicacións son na actualidade máis complexas sendo ademais necesario autenticar os dispositivos que acceden á rede para monitorar o uso que fan dos recursos da rede corporativa ou doméstica.

Con este concepto actual de perímetro difuso podemos distinguir os seguintes elementos destinados a conseguir un nivel de seguridade perimetral axeitado:

- **Devasas** (*firewalls*): dispositivos (hardware o software) da rede que se sitúan normalmente entre 2 redes (ou subredes) para filtrar o tráfico en función de diferentes criterios.



- **IPS/IDS<sup>2</sup>**: os sistemas de prevención e detección de intrusións permiten detectar intrusións habitualmente en tempo real. Tamén poden detectar patróns de ataques e tomar accións contra eles.
- **Redes privadas virtuais (RPV)<sup>3</sup>**: unha rede privada virtual é unha tecnoloxía de rede que permite estender unha rede local sobre unha rede pública ou non controlada, como por exemplo a Internet, garantindo as condicións técnicas e de seguridade dunha rede local común.
- **Xestión e control de acceso e identidade**: trátase que permiten xestionar usuarios e os seus datos de identificación. A partir diso é posible asociar roles, perfís, políticas de seguridade e controlar o acceso aos recursos. No contexto da seguridade en rede podemos distinguir:
  - o Ferramentas de control de acceso á rede corporativa: en inglés NAC (*Network Access Control*), son ferramentas destinadas a proporcionar mecanismos para administrar e controlar o acceso de usuarios e doutras redes aos servizos da rede corporativa. Adoitan incluír unha función preventiva ante intrusións e usos indebidos e unha función de reforzo de políticas baseada na identidade, roles e permisos dos usuarios.
  - o Xestión de identidade e autenticación: son ferramentas centradas na xestión da identidade, que provén un repositorio centralizado de usuarios e permiten realizar unha autenticación e autorización centralizada aos sistemas e recursos dunha organización. Aplícanlles aos usuarios

---

<sup>2</sup> Siglas en inglés de *Intrusion Prevention Systems/ Intrusion Detection Systems*

<sup>3</sup> É frecuente a denominación en inglés VPN (*Virtual Private Network*)



perfís, privilexios, roles e políticas de uso dos recursos. Así mesmo tamén están incluídos os servidores de autenticación.

- o Ferramentas *Single Sign-On*: son ferramentas que permiten o acceso a distintos sistemas ou situacións cun mecanismo de identificación común. Isto realízase mediante a propagación dunha identidade única e a súa asociación aos diversos servizos e recursos dunha organización.

### **1.2.3 Zona desmilitarizada (DMZ)**

Unha DMZ é unha rede local que se sitúa entre a rede interna dunha organización e unha rede externa, xeralmente a Internet.

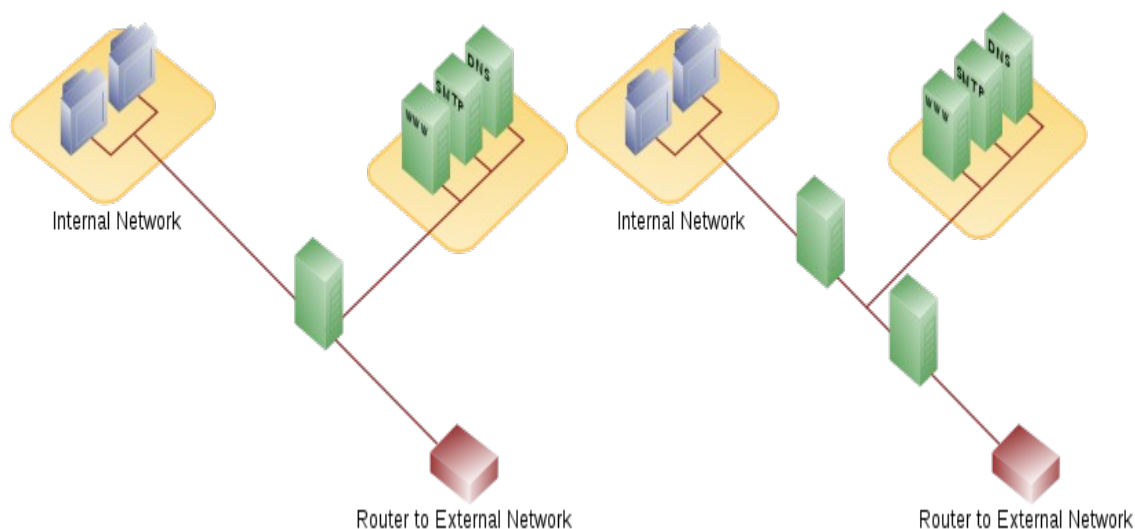
O obxectivo dunha DMZ é que as conexións dende a rede interna e a externa á DMZ estean permitidas, mentres que as conexións dende a DMZ só se permitan á rede externa, é dicir, os equipos na DMZ non poden conectar coa rede interna.

Isto permite que os equipos da DMZ poidan dar servizos á rede externa á vez que protexen a rede interna no caso de que intrusos comprometan a seguridade dos equipos (*host*) situados na zona desmilitarizada.

O resultado é que a DMZ se converte nunha liña de protección para a rede interna, xa que os buratos de seguridade explotables dende o exterior xamais poderán alcanzar a rede interna.

O concepto de DMZ é moi sinxelo. Trátase de incluír a DMZ entre a Internet e a rede interna, separándoa por unha ou dúas devasas que garanten as restricións de acceso descritas.





DMZ implantadas cunha ou dúas devasas. Fonte: Wikipedia

A DMZ úsase habitualmente para situar servidores que sexan necesariamente accesibles desde o exterior e que por iso sexan especialmente vulnerables a ataques. Trátase de servidores de correo electrónico, Web e DNS. Cada un dos equipos situados na DMZ adoita denominarse *Bastion Host*<sup>4</sup>, e considéranse como equipos especialmente sensibles a ataques, cuxa seguridade debe ser xestionada con especial atención.

As conexións que se realizan dende a rede externa cara á DMZ contrólanse xeralmente utilizando *Port Address Translation* (PAT). O PAT é parte do estándar *Network Address Translation* (NAT) que traduce conexións TCP e UDP feitas por un ordenador anfitrión e un porto nunha rede externa a outro enderezo e porto da rede interna. Permite que un único enderezo IP sexa utilizado por varias máquinas da intranet. Con PAT, un IP externo pode responder ata a 64000 enderezos internos.

Á inversa, cando un [ordenador](#) da intranet manda un paquete cara a fóra, queremos ocultar o seu enderezo real. O servizo NAT substitúe o IP interno co novo IP do propio servizo. Logo asígnalle á conexión un porto da lista de

---

<sup>4</sup> En analoxía cos bastións das fortalezas medievais, que eran puntos delas especialmente defendidos.



portos dispoñibles, insire o porto no campo apropiado do paquete de datos e envía o paquete.

O servizo NAT crea unha entrada na súa táboa de enderezos IP internos, portos internos e portos externos. A partir dese momento, todos os paquetes que proveñan deste anfitrión serán traducidos cos mesmos portos. Se hai algunha resposta para ese paquete, o mecanismo PAT será capaz agora de determinar cal é a máquina á que lle corresponde o porto de destino.

Unha DMZ créase a miúdo a través das opcións de configuración da devasa, onde cada rede se conecta a un porto distinto desta. Esta configuración chámase devasas en trípode (*three-legged firewall*). Unha formulación máis segura é usar dúas devasas, onde a DMZ se sitúa no medio e se conecta a ambas as dúas devasas, unha conectada á rede interna e a outra á rede externa. Esta configuración axuda a previr configuracións erróneas accidentais que permitan o acceso dende a rede externa á interna. Este tipo de configuración tamén se chama devasas de subrede monitorada (*screened-subnet firewall*).

## **2 TÉCNICAS DE SEGURIDADE PREVENTIVAS E REACTIVAS. DEVASAS. INTRUSIÓN. ACCESOS NON AUTORIZADOS**

A seguridade dunha rede non só consiste en previr os posibles ataques que esta poida sufrir, senón que se deben implementar técnicas que permitan reaccionar ante eles. Falamos así de técnicas preventivas e reactivas.

As técnicas preventivas permitirannos reducir o número de ataques con impacto na nosa rede namentres que as reactivas nos axudarán a reaccionar para conseguir evitar un ataque e a recuperar a operatividade



da rede en caso de que un ataque chegue a ter éxito e dane os nosos sistemas.

Neste contexto, os mecanismos de seguridade máis habituais son: devasas, sistemas de detección de intrusos e redes privadas virtuais.

## **2.1 Devasas**

O maior nivel de seguridade que se podería adoptar nun sistema informático consistiría en desenchufalo da rede. Con todo, esta non adoita ser unha opción aceptable xa que, a día de hoxe, a maior parte das empresas necesitan intercambiar información, ben sexa a través dunha rede local ou ben a través de internet.

Aínda que se poden ter todos os servidores e estacións de traballo protexidos con fortes medidas de seguridade, normalmente non abonda coa seguridade baseada en *host*. Unha medida complementaria —que é moi aceptada— consiste en complementar a seguridade do *host* cun **servizo de devasa**<sup>5</sup>. Unha devasa é un elemento que se sitúa entre a rede interna e os intrusos potenciais externos constituíndo unha barreira de protección. Normalmente establécese entre a rede local e internet, facendo así de muro de protección exterior da rede local. A devasa pode ser un software nun equipo ou pode ser un conxunto de varios sistemas específicos destinados a controlar o acceso á rede interna da organización.

### **2.1.1 Características dunha devasa**

Un servizo de devasa presenta unha serie de características entre as que se poden salientar as seguintes:

- Define un punto único de control que permite afastar os usuarios non autorizados da rede protexida e proporciona protección contra ataques de *spoofing* e encamiñamento. Isto simplifica a

---

<sup>5</sup> É moi común o uso do termo en inglés *firewall*.



xestión, porque as características de seguridade están centralizadas nun único sistema ou conxunto de sistemas.

- Proporciona unha localización para monitorar eventos relacionados coa seguridade. Nun sistema *devasa* pódense implementar auditorías de seguridade e alarmas.
- Proporciona unha plataforma para outras funcións de internet que non están relacionadas coa seguridade: tradución de enderezos, funcións de xestión para auditoría e rexistros de log para medir o uso de internet.
- Pode ser utilizado para implementar redes privadas virtuais (RPV).

### 2.1.2 Tipos de devasas

Atendendo á función desempeñada polo sistema devasa podemos establecer a seguinte clasificación:

- **Devasa de filtrado de paquetes:** aplica un conxunto de regras para cada paquete IP de entrada e saída e, a seguir, reenvía ou descarta o paquete. As regras de filtrado están baseadas na información contida en cada paquete: IP orixe, IP destino, protocolo, interface, etc.
- **Devasa de inspección de estados:** Ademais de facer o filtrado en función de paquetes, garda información das sesións e conexións abertas. Isto permite dispoñer de políticas de seguridade máis avanzadas e evita moitos ataques.
- **Proxy de aplicacións:** Actúa como intermediario no tráfico da capa de aplicación, permitindo acceder a certas características das aplicacións e reenviando a información.

### 2.1.3 Localizacións da Devasa e configuracións

Atendendo á localización da devasa, pódense obter distintas configuracións:



- **Redes DMZ:** empréganse para illar nunha ou en varias subredes (coñecidas como “zonas desmilitarizadas”) os principais servizos da organización aos que cómpre acceder desde o exterior (web, correo, DNS). A este segmento aplícanse unhas regras de filtrado para garantir unha conectividade controlada desde o exterior. Ao resto da rede interna péchase o acceso desde o exterior.
- **Redes privadas virtuais (RPV):** é unha solución que lles ofrece grandes vantaxes aos xestores de rede. Consiste en dar acceso mediante unha conexión segura a un equipo ou segmento dunha rede privada a través dunha rede considerada insegura, como pode ser internet.
- **Devasas distribuídas:** consiste en agrupar baixo un mesmo mecanismo de control centralizado a xestión de dispositivos *devasa* e a xestión de devasas baseadas en *host*. Con estas ferramentas, o administrador pode establecer políticas de seguridade e aplicarllelas a equipos *devasa*, servidores e estacións de traballo tanto locais como remotas. Ademais proporcionan capacidades de monitorización e alertas de seguridade.

#### 2.1.4 Netfilter/iptables

Os sistemas Linux inclúen no seu *kernel* unha aplicación que permite desenvolver a nosa propia devasa personalizada.

**Netfilter** é un framework dispoñible no núcleo Linux que permite interceptar e manipular paquetes de rede. Este *framework* permite realizar o manexo de paquetes en diferentes estados do procesamento. *Netfilter* é tamén o nome que recibe o proxecto que se encarga de ofrecer ferramentas libres para devasas baseadas en Linux.



O compoñente máis popular construído sobre *Netfilter* é *iptables* (anteriormente coñecido como *ipchains*), unha ferramenta de devasas que permite non só definir políticas de filtrado do tráfico que circula pola rede, senón tamén realizar tradución de enderezos de rede (NAT) para IPv4 ou manter rexistros de *log*.

Iptables configúrase mediante regras que determinan que facer con cada paquete analizado pola devasa. As regras agrúpanse en *cadeas*, de modo que cada cadea é unha lista ordenada de regras. E á súa vez, as cadeas agrúpanse en *táboas*, nas que cada unha está asociada a un tipo diferente de procesamento de paquetes. O contido das regras é sinxelo, xa que se basea en especificar que paquetes a cumpren (*match*) e cal é o destino do paquete se é que cumpre a regra.

As táboas que se inclúen por defecto (é posible incorporar máis) son:

- **Táboa *Filter*** (táboa de filtros): é a encargada de filtrar os paquetes permitindo ou non o seu paso. Contén tres cadeas e todo paquete pasará por unha delas: INPUT (entrada), OUTPUT (saída) e FORWARD (redirección).
- **Táboa *NAT*** (táboa de tradución de enderezos de rede): encárgase de configurar as regras de reescritura de enderezos ou de portos dos paquetes, xa que é habitual que a devasa estea nun punto da rede no que decida sobre o encamiñamento dos paquetes cara a redes internas (mediante NAT e/ou PAT). Conta tamén con tres cadeas: PREROUTING, POSTROUTING E OUTPUT.
- **Táboa *Mangle*** (táboa de esnaquizamento): esta táboa é a responsable de axustar as opcións dos paquetes, como por exemplo a [calidade de servizo](#). Todos os paquetes pasan por esta táboa. Contén todas as cadeas predefinidas nas anteriores.



En canto ao destino dos paquetes, *iptables* inclúe por defecto os seguintes destinos como consecuencia de que unha regra se cumpra:

- **ACCEPT:** o paquete acéptase, e polo tanto realizarase a acción determinada polo paquete e a cadea pola que entrou. Por exemplo, se fose INPUT permitirase que sexa recibido polo ordenador anfitrión de destino.
- **DROP:** o paquete non é aceptado e descártase sen ningún outro tipo de acción nin notificación á orixe nin ao destino.
- **QUEUE:** o paquete envíase a unha cola que pode ser manipulada por outra aplicación. Se ningunha aplicación le esta cola, a súa semántica é a mesma que a de DROP.
- **RETURN:** o paquete deixa de circular pola cadea na regra da cal se executou o destino RETURN. Se esa cadea é unha subcadea doutra, o paquete continuará pola cadea superior coma se nada pasara. Se pola contra a cadea é unha cadea principal (por exemplo a cadea INPUT), ao paquete aplicaráselle a política por defecto da cadea en cuestión (ACCEPT, DROP ou similar).

Un exemplo da sintaxe para engadir unha regra que descartaría todos os paquetes UDP recibidos sería a seguinte: `iptables -A INPUT -p udp -j DROP`.

## 2.2 Sistemas de detección de intrusións (IDS/IPS)

Tal e como apuntamos, unha devasa é un mecanismo de seguridade que permite pechar todos aqueles portos de servizos que non se estean a utilizar e así reducir a posibilidade de ataques por parte de intrusos. Pero aínda que se permita só o acceso aos servizos básicos e teoricamente seguros, estes ás veces teñen vulnerabilidades que poden ser aproveitadas por un atacante con fins maliciosos para saltarse esta medida de protección. Facendo un símil coa protección dunha casa, a devasa



corresponderíase coa porta de entrada, pero é necesario un sistema de alarma que se encargue de avisar en caso de que un intruso dea entrada. O elemento correspondente ao sistema de alarma no mundo da seguridade informática é o IDS (*Intrusion Detection System*). Pódese definir un IDS como un sistema que se encarga de vixiar a rede, absorbendo todo o tráfico e inspeccionándoo en busca de patróns de ataque. As características principais dos IDS son as seguintes:

- Engade un alto nivel de integridade ao resto da rede, xa que, en certa forma, sabemos que o resto de sistemas están ben porque o IDS non avisa do contrario.
- Pode monitorar a actividade dun atacante. Dependendo da infraestrutura de IDS, poderase monitorar esta actividade nun único segmento ou en varios.
- Alerta ante patróns de ataque comúns coñecidos.
- Automatiza a busca de novos patróns de ataque, xa que proporcionan ferramentas estatísticas de busca e monitorización de tráfico anómalo.
- Pode detectar ataques en tempo real.
- Pode detectar erros de configuración nos equipos.

Os sistemas IDS constan dun equipo cunha consola central de administración e unha ou varias "sondas" que se encargan de capturar o tráfico que se debe analizar. Dependendo da arquitectura de rede da organización pódense seguir diversos criterios para situar as sondas: na DMZ, atrás da devasa, nos accesos de usuario, entre a Extranet e a Internet, etc.

Unha variante dos sistemas de detección de intrusións son os IPS (*Intrusion Prevention System*). A diferenza con respecto aos IDS estriba en que os IPS



además de detectar un ataque e xerar a correspondente alerta son capaces de actuar e intentar neutralizar o ataque, polo que, a pesar do seu nome, é un mecanismo de defensa reactivo. Un exemplo claro sería cando o IPS detecta actividade maliciosa por parte dun usuario conectado a un servidor a través dunha conexión remota. Unha das accións drásticas que podería tomar o IPS é cortar a conexión.

Os sistemas IPS/IDS pódense instalar como un software nun servidor (ex. Snort) ou pode ser un equipo hardware co seu software completo e independente proporcionado por un fabricante de dispositivos de seguridade.

Un IDS consegue o seu obxectivo de detección usando usa algunha das dúas seguintes técnicas:

- **Heurística:** determina actividade normal de rede, como a orde de largo de banda usada, protocolos, portos e dispositivos que xeralmente se interconectan, e alerta a un administrador ou usuario cando este varía do considerado como normal, clasificándoo como anómalo.
- **Patrón:** analiza paquetes na rede, e compáraos con patróns de ataques coñecidos, e preconfigurados. Estes patróns denomínanse sinaturas. Esta técnica provoca que exista un período de tempo entre o descubrimento do ataque e o seu patrón, ata que este é finalmente configurado nun IDS. Durante este tempo, o IDS será incapaz de identificar o ataque.

### **2.3 Redes privadas virtuais (RPV)**

As redes privadas virtuais (RPV<sup>6</sup>) son un elemento cada vez máis habitual en moitas organizacións que contribúe ao concepto de perímetro difuso da rede que introducíamos ao principio deste tema.

---

<sup>6</sup> É frecuente a denominación en inglés VPN (*Virtual Private Network*)



Proporcionan unha forma segura de conectarse dende unha localización remota cunha rede de área local privada (LAN) a través da Internet ou calquera outra rede pública non segura. Unha RPV é unha conexión que ten a aparencia e moitas das vantaxes dunha ligazón dedicada pero traballando sobre unha rede pública.

Para isto utilízase unha técnica chamada *tunneling* que permite encamiñar os paquetes de datos pola rede pública nun túnel privado que simula unha conexión punto a punto. As RPV son utilizadas frecuentemente polos traballadores remotos ou empregados nas delegacións da organización, para compartir datos e recursos da rede privada.

### 2.3.1 Beneficios das redes privadas virtuais

Unha rede privada virtual pode proporcionar os seguintes beneficios para a organización:

1. **Rendemento predicible:** nas RPV con canles dedicadas pódese garantir o largo de banda entre os sitios e o rendemento da rede faise máis predicible.
2. **Independencia na elección das tecnoloxías de transporte para as redes de usuarios:** as posibilidades están limitadas pola elección dun provedor ou fabricante. Así a organización pode usar Ethernet, Frame Relay, IP e outras tecnoloxías de rede para conectar os seus sitios.
3. **Seguridade mellorada:** ao reducir o número de conexións co mundo exterior redúcese considerablemente a posibilidade dun ataque. Ademais tamén se reduce a posibilidade de interceptación do tráfico.
4. **Espazo de enderezos IP independente:** nas redes privadas é posible utilizar calquera enderezamento. Por exemplo, case todos os servizos de RPV permiten o uso de enderezos IP privados tales



como 10.0.0.1 ou 192.168.0.3, que non poden ser encamiñados a través das redes públicas.

Estas características serán de utilidade para algúns usuarios, pero de importancia relativa para outros. As vulnerabilidades e baixo rendemento das redes públicas poden facer que a "seguridade mellorada" e "rendemento predicible" sexan as características máis desexables dunha RPV.

Recentemente, a "independencia de elección de tecnoloxía" e o "espazo de enderezos independente" parece que se volveron menos importantes: o primeiro debido á dominación das tecnoloxías Ethernet en capa 2 e IP en capa 3. A segunda debido a que coa implantación de IPv6 se espera acabar co déficit de enderezos.

De todos os xeitos, ter un espazo de enderezos independente mellora a seguridade, utilizando rangos de enderezos para separar sitios dentro da organización e restrinxir accesos.

Á hora de crear unha RPV para proporcionar acceso remoto, hai que escoller a tecnoloxía que mellor se adapte ao escenario en cuestión. Esta elección de tecnoloxía implica técnicas de *tunneling*, autenticación, control de acceso e seguridade de datos.

### 2.3.2 Implantación dunha RPV

Xeralmente defínense tres arquitecturas principais de RPV:

- **RPV de acceso remoto:** neste tipo de RPV situaríanse os usuarios que dende un ordenador anfitrión remoto crean un túnel para conectarse á rede privada da organización. O dispositivo remoto pode ser un equipo persoal cun software cliente para crear RPV, e usar unha conexión conmutada ou unha conexión de banda larga permanente.



- **Extranet RPV:** este tipo de arquitecturas permiten que certos recursos da rede privada da organización sexan accedidos por redes doutras compañías, tales como clientes ou provedores. Neste escenario é fundamental o control de acceso.
- **Intranet RPV (LAN-to-LAN RPV):** neste escenario as redes remotas da organización son conectadas entre si utilizando a rede pública, converténdose deste xeito nunha única rede LAN corporativa global.

Hai unha ampla variedade de tecnoloxías que se poden utilizar para a implantación de RPV. Os criterios que deben cumprir estas tecnoloxías son:

- **Eficiencia:** : os tempos de resposta deben ser adecuados e comparables coas redes consideradas "non seguras".
- **Facilidade de administración:** os usuarios e administradores deste tipo de redes poden facer o seu traballo dun xeito rápido e efectivo.
- **Seguridade:** as conexións deben ser brindadas cumprindo cos requisitos de autenticación, autorización, privacidade, integridade e contabilidade.
- **Cumprimento de estándares e interoperabilidade:** hai moitas tecnoloxías que son estándares e que participan na creación de RPV: IPSec, MD5, SOCKSv5, IKE, ISAKMP, Diffie-Hellman, X.509, RADIUS, etc.

Para clasificar as tecnoloxías que se poden utilizar na implantación dunha arquitectura de RPV, pode tomar como referencia o modelo OSI e situalas en función do nivel no que son implantadas. A Figura 3 mostra os principais protocolos utilizados para o establecemento de conexións RPV e a súa situación no modelo de referencia OSI.



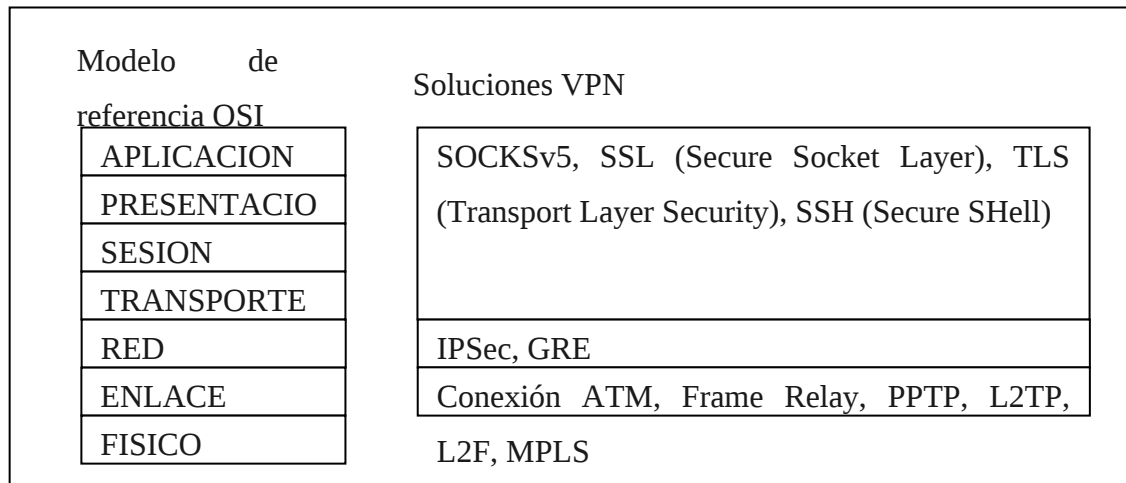


Figura 3: Situación das solucións RPV no modelo de referencia OSI

Á hora de implantar unha arquitectura de rede baseada en RPV hai dúas posibles opcións. Facer unha implantación por hardware ou por software.

As implantacións por hardware realizan o proceso de cifraxo e descifrado do tráfico a nivel físico entre os extremos da liña de comunicación. Os dispositivos utilizados normalmente son encamiñadores con capacidades de RPV incorporadas.

Como vantaxes desta solución pódese dicir que a instalación e configuración son relativamente sinxelas, non se necesita persoal especializado e o seu mantemento é mínimo. Pola contra presentan o inconveniente de que o sistema de cifraxo vén imposto polo fabricante e depéndese del para as actualizacións.

Por outro lado as implantacións baseadas no software estanse a impoñer cada día máis. A explicación radica en que a necesidade dos usuarios pequenos e medianos de implantar sistemas de seguridade no acceso ás súas máquinas vai en aumento.

As implantacións software son moito máis baratas que comprar hardware preparado para RPV e existe un gran número de RPV desenvolvidas por



software. Como inconvenientes desta aproximación pódese dicir que é necesaria unha máquina para dar soporte á solución, o sistema de claves e certificados reside en máquinas potencialmente inseguras e nos casos que se utilice software de libre distribución pode ser que teña portas traseiras ou outras deficiencias de seguridade.



### 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 editoriais para a documentación libre en español de Hispalinux.  
<http://es.tldp.org/>

**Autor: Juan Otero Pombo**

**Enxeñeiro en Informática no Concello de Ourense**

**Colexiado do CPEIG**

---

<sup>7</sup> Todas as ligazóns foron verificadas en novembro do 2011.



# **38. CERTIFICADOS DIXITAIS. TARXETAS CRIPTOGRÁFICAS. SINATURA DIXITAL. TÉCNICAS DE CIFRAXE. INFRAESTRUTURA DE CLAVE PÚBLICA (PKI).**





**Tema 38: Certificados dixitais. Tarxetas criptográficas. Sinatura dixital. Técnicas de cifrado. Infraestrutura de clave pública (PKI).**

**ÍNDICE**

<b>1. TÉCNICAS DE CIFRADO.....</b>	<b>2</b>
<b>2. SINATURA DIXITAL.....</b>	<b>18</b>
<b>3. INFRAESTRUTURA DE CLAVE PÚBLICA (PKI).....</b>	<b>20</b>
<b>4. REFERENCIAS.....</b>	<b>33</b>



## 1. TÉCNICAS DE CIFRADO

A palabra **criptografía** é unha palabra de orixe grega (*krypto* 'oculto' e *graphos* 'escribir') e defínese como a arte de escribir con clave secreta ou dun modo enigmático.

### 1.1. Criptografía e criptoanálise

A historia da criptografía remóntase a miles de anos atrás e ten unha longa tradición nas escrituras relixiosas que poderían ofender a cultura dominante ou as autoridades políticas. A finalidade desta técnica foi sempre enviar mensaxes confidenciais coa garantía de que só o seu destinatario puidese acceder á información contida na mensaxe.

O método consiste en aplicarlle á mensaxe unha transformación coñecida como **cifrado**, co obxectivo de que as persoas que descoñezan a transformación realizada sexan incapaces de acceder á información contida na mensaxe.

O estudo de técnicas destinadas a atoparlle o sentido a unha información cifrada, sen ter acceso á información secreta requirida, é a **criptoanálise**. A finalidade da criptoanálise é, xa que logo, descubrir a clave de cifrado. A vulnerabilidade dos algoritmos de cifrado depende da dificultade da tarefa de descubrir a clave. Unha ataque por forza bruta consiste en buscar a clave de cifrado probando un a un todos os posibles valores da clave de descifrado.

A **criptoloxía** é a disciplina que abarca a criptografía e a criptoanálise.

Outro concepto relacionado é a **esteganografía**. Igual que a criptografía, o que busca é ocultar unha mensaxe ante un posible atacante, pero a diferenza estriba en como ocultan a información ambas as técnicas: namentres que a criptografía pretende que a información non sexa



descifrada, a esteganografía pretende que a información pase desapercibida (por exemplo, un código secreto tatuado no coiro cabeludo e oculto polo pelo).

As técnicas criptográficas pódense clasificar en función de varios criterios. Seguindo un **criterio temporal**, pódense clasificar en clásicas ou extemporáneas e modernas ou contemporáneas.

## **1.2. Técnicas criptográficas clásicas**

As técnicas criptográficas clásicas realizan o cifrado a partir da substitución e transposición dos caracteres da mensaxe. O segredo está no algoritmo aplicado á mensaxe, polo que teñen o inconveniente de que se un atacante o descobre, será capaz de interpretar todas as mensaxes cifradas que capture. Como exemplos podemos citar:

- **Substitución monoalfabeto:** consiste na substitución de símbolos un a un. Como exemplo pódese citar o algoritmo de César.
- **Substitución polialfabeto:** consiste na substitución dun símbolo por un dun conxunto. Como exemplo pódese citar o cifrado de Vigenère.
- **Transposición:** consiste en cambiar a orde dos símbolos.
- **Combinación de substitución e transposición** (máquinas rotoras ).

## **1.3. Técnicas criptográficas modernas**

As técnicas criptográficas modernas, a diferenza das clásicas, empregan claves de cifrado para cifrar a información. Unha premisa fundamental da criptografía moderna é que a seguridade do método debe depender unicamente da clave de cifrado, e débense coñecer os algoritmos. Esta



premisas fai que estas técnicas resulten moito máis seguras e efectivas, xa que resulta máis sinxelo manter o segredo da clave, e, ademais, cambiar a clave de cifrado sempre será menos custoso que idear un novo algoritmo, sendo frecuente que a clave de cifrado se xere de forma automática.

Podemos clasificar os algoritmos de cifrado atendendo ás claves que utilizan ou ao modo en que procesan a información.

Se nos fixamos no tipo de claves que empregan, temos dous tipos de algoritmos:

- Algoritmos de **cifrado simétrico ou de clave privada**: empregan a mesma clave para o cifrado e o descifrado, polo que debe ser secreta e compartida polo emisor e o receptor. Exemplos de algoritmos deste tipo son DES, 3DES, AES, IDEA, RC5, etc.
- Sistemas de cifrado **asimétrico ou de clave pública**: empregan un par de claves xeradas polo emisor. Unha das claves é pública, é dicir, coñecida por todo o mundo, e a outra é privada ou secreta, de forma que o que se cifra cunha clave é descifrado pola outra e viceversa. Exemplos de algoritmos deste tipo son RSA, DSA, Diffie-Hellman, ElGamal, etc.

Se consideramos o modo de procesamento, temos 3 tipos de algoritmos:

- Técnicas criptográficas de cifrado en **modo fluxo** (*stream cipher*): estes algoritmos de cifrado baséanse na combinación dun texto en claro cun texto de cifrado obtido a partir dunha clave. A característica fundamental é que se vai cifrando un fluxo de datos bit a bit. Exemplos: RC4, SEAL.
- Técnicas criptográficas de cifrado en **modo bloque** (*block cipher*): caracterízanse porque o algoritmo de cifrado ou descifrado se aplica separadamente a bloques de lonxitude  $l$ , e para cada un deles o



resultado é un bloque da mesma lonxitude. Exemplos: DES, 3DES, AES.

- Técnicas criptográficas baseadas en funcións resumo (**hash functions**): a característica principal destes algoritmos é que permiten obter unha cadea de bits de lonxitude fixa a partir dunha mensaxe de lonxitude arbitraria. Exemplos: MD5, familia SHA.

#### **1.4. Criptografía de clave privada ou simétrica**

A criptografía de clave simétrica caracterízase porque a clave de descifrado **k** é idéntica á clave de cifrado ou pode obterse a partir desta, co que a fortaleza do algoritmo reside no segredo desta.

Se **M** é a mensaxe en claro que se quere protexer, ao cifrala cun algoritmo en función dunha clave privada **E<sub>k</sub>(M)** obtense outra mensaxe chamada texto cifrado **C**. Para que este cifrado sexa útil, existe outra función **D<sub>k</sub>(C)** que, a partir do texto cifrado polo emisor, permite obter de novo a mensaxe en claro **M**.

$$\mathbf{C} = \mathbf{E}_k(\mathbf{M})$$

$$\mathbf{M} = \mathbf{D}_k(\mathbf{C}) = \mathbf{D}_k(\mathbf{E}_k(\mathbf{M}))$$

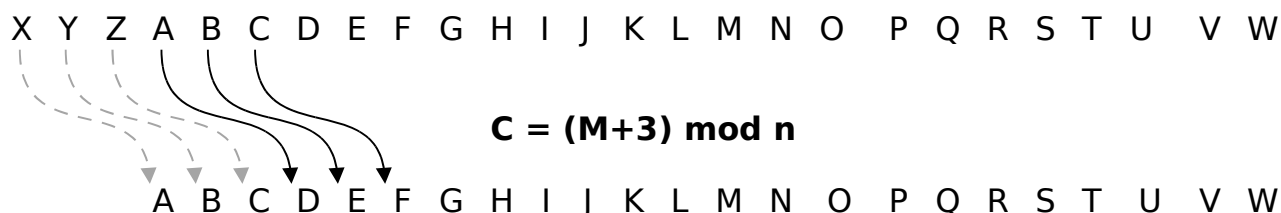
A seguridade do sistema reside daquela en manter en segredo a clave **k**. O inconveniente principal da criptografía simétrica é o **intercambio de claves**. Este problema soluciónase coa axuda da criptografía asimétrica.

##### **1.4.1. Substitución monoalfabeto**

Un exemplo de algoritmo de cifrado por substitución monoalfabeto é o cifrado César. É un tipo de cifrado por substitución en que cada letra no texto orixinal é substituída por outra letra que se atopa un número fixo de posicións máis adiante no alfabeto. Por exemplo, cun desprazamento de 3,



o A sería substituído polo D (situado 3 lugares á dereita do A), o B sería substituído polo E, etc. Este método débelle o seu nome a Xulio César, que o usaba para se comunicar cos seus xenerais.



### Exemplo:

<b>Mensaxe en</b>	<b>H O L A C E S A R</b>
<b>claro:</b>	
<b>Mensaxe</b>	<b>K R O D G H V D U</b>
<b>cifrada:</b>	

O descifrado dunha mensaxe consistiría en substituír cada letra do texto pola que hai tres posicións máis adiante no alfabeto.

A condición principal que debe cumprir a clave é que debe ser unha permutación do alfabeto, é dicir, non pode haber letras repetidas nin faltar ningunha. Se non, a transformación non sería invertible en xeral.

### 1.4.2. Substitución polialfabeto

O inconveniente dos algoritmos de substitución monoalfabeto é que o texto cifrado mantén a mesma distribución de frecuencia de caracteres que ten o texto claro orixinal, o que fai que sexan criptoanalizables por métodos estatísticos sinxelos. Unha posible mellora dos cifrados por substitución é intentar métodos que destrúan esa correspondencia de frecuencias entre a mensaxe en claro e o criptograma; por exemplo, empregando varios alfabetos á vez para o cifrado. Nos cifrados polialfabéticos, a substitución aplicada a cada carácter varía en función da posición que ocupe este



dentro do texto claro. En realidade corresponde a unha aplicación cíclica de  $n$  cifrados de substitución monoalfabeto. Un exemplo típico de cifrado polialfabético é o Cifrado **de Vigenère**.

### **1.4.3. Cifrado en bloque**

Un algoritmo de cifrado en bloque toma como entrada un bloque de lonxitude fixa e unha clave e xera un novo bloque cifrado da mesma lonxitude que o bloque de entrada.

A técnica consiste en dividir o texto que cómpre cifrar (con lonxitude  $L$ ) en bloques de tamaño  $b$  e deseguido cifrar cada un dos bloques. Se  $L$  non é múltiplo de  $b$ , agréganse bits adicionais para conseguir que todos os bloques estean completos. Para descifrar a mensaxe procédese de maneira análoga.

Moitos dos algoritmos de cifrado en bloque se basean na combinación de dúas operacións básicas: substitución e transposición.

- A **substitución** consiste en traducir cada bloque de bits que chegan como entrada a outro de saída seguindo unha permutación determinada. O cifrado César sería un exemplo simple de substitución onde cada grupo de bits correspondería a unha letra.
- A **transposición** consiste en reordenar a información do texto en claro segundo un patrón determinado. Un exemplo podería ser a formación de grupos de cinco letras, incluídos os espazos en branco, e reescribir cada grupo (1, 2, 3, 4, 5) na orde (3, 1, 5, 2, 4). Por exemplo:

Texto en claro: "HOLA MUNDO"

Texto cifrado: "LH OANMOUD"



A transposición non dificulta extraordinariamente a criptoanálise, mais pode combinarse con outras operacións para engadir-lles complexidade aos algoritmos de cifrado.

O **produto de cifras**, ou combinación en fervenza de distintas transformacións criptográficas é unha técnica moi efectiva para implementar algoritmos bastante seguros de forma sinxela. Por exemplo, moitos algoritmos de cifrado en bloque baséanse nunha serie de iteracións de produtos substitución-transposición.

Dúas propiedades desexables nun algoritmo criptográfico son a **confusión e a difusión**. A confusión consiste en agochar a relación entre a clave e as propiedades estatísticas do texto cifrado. A difusión propaga a redundancia do texto en claro ao longo do texto cifrado para que non sexa facilmente recoñecible.

A confusión consegue que, cambiando un só bit da clave, cambien moitos bits do texto cifrado; e a difusión implica que o cambio dun só bit do texto en claro afecte tamén a moitos bits do texto cifrado.

### ***Modos de operación do cifrado en bloque***

Un aspecto que cómpre ter en conta cando se utiliza o cifrado é que, inda que se pode conseguir que un atacante non descubra directamente os datos transmitidos, en ocasións é posible que se poida deducir información indirectamente. Por exemplo, nun protocolo que utilice mensaxes cunha cabeceira fixa, a aparición dos mesmos datos cifrados varias veces nunha transmisión pode indicar onde empezan as mensaxes. Para tentar contrarrestar isto, o cifrado en bloque opera en varios modos:

- O modo **ECB** (*Electronic Code Book*): consiste en dividir o texto en bloques e cifrar cada un deles de forma separada. O inconveniente



deste método é que bloques idénticos de mensaxe sen cifrar producirán idénticos textos cifrados.

- No modo **CBC** (*Cipher Block Chaining*), a cada bloque de texto aplícaselle, antes de ser cifrado, unha operación XOR bit a bit, co bloque previo xa cifrado. Deste xeito, cada bloque é dependente de todos os bloques de texto previos ata ese punto. Ademais, para facer que cada mensaxe sexa única, pódese usar un vector de inicialización. CBC é o modo máis usado. A súa principal desvantaxe é que é secuencial e non pode funcionar en paralelo.
- No modo **CFB** (*Cipher Feedback*), o algoritmo de cifrado non se lle aplica directamente ao texto en claro senón a un vector auxiliar (inicialmente igual ao IV). Do resultado do cifrado tómanse  $n$  bits que se suman a  $n$  bits do texto en claro para obter  $n$  bits de texto cifrado. Estes bits cifrados utilízanse tamén para actualizar o vector auxiliar. O número  $n$  de bits xerados en cada iteración pode ser menor ou igual que a lonxitude de bloque  $b$ . Tomando como exemplo  $n=8$ , temos un cifrado que xera un byte cada vez, sen que sexa necesario agardar a ter un bloque enteiro para podelo descifrar.
- O modo **OFB** (*Output Feedback*) opera como o CFB, pero, en lugar de actualizar o vector auxiliar co texto cifrado, actualízase co resultado obtido do algoritmo de cifrado. A propiedade que distingue este modo dos demais consiste en que un erro na recuperación dun bit cifrado afecta só ao descifrado deste bit.

### ***Exemplos de algoritmos de cifrado en bloque: DES***

DES<sup>1</sup> é un dos algoritmos de cifrado máis usados no mundo. Foi publicado en 1977 no documento **FIPS**<sup>2</sup> PUB 46 do Instituto Nacional de Estándares e Tecnoloxía (NIST).

---

<sup>1</sup> Siglas en inglés de *Data Encryption Standard*.

<sup>2</sup> Siglas en inglés de *Federal Information Processing Standard*.



O algoritmo foi controvertido ao principio, con algúns elementos de deseño clasificados, unha lonxitude de clave relativamente curta, e as continuas sospeitas sobre a existencia dalgunha porta traseira para a NSA<sup>3</sup>. Posteriormente DES foi sometido a unha intensa análise académica e motivou o concepto moderno do cifrado por bloques e a súa criptoanálise.

Hoxe en día, DES considérase inseguro para moitas aplicacións. Isto débese principalmente a que o tamaño de clave de 56 bits é curto. A finais do 2001 o algoritmo foi substituído polo novo AES<sup>4</sup>.

DES é o prototipo de algoritmo de cifrado por bloques: toma un texto en claro dunha lonxitude fixa de bits e transfórmalo mediante unha serie de operacións básicas noutro texto cifrado da mesma lonxitude, dividindo para iso a mensaxe en bloques de 64 bits. O algoritmo DES utiliza unha clave criptográfica para modificar a transformación, de modo que só poden realizar o descifrado aqueles que coñezan a clave concreta utilizada no cifrado. A lonxitude da clave é de 64 bits, inda que en realidade só 56 deles son empregados polo algoritmo. Os oito bits restantes utilízanse unicamente para comprobar a paridade, e despois son descartados.

A parte central do algoritmo consiste en dividir a mensaxe de entrada en grupos de bits, facer unha substitución distinta sobre cada grupo e, a seguir, unha transposición de todos os bits. Esta transformación repítese dezaseis veces: en cada iteración, a entrada é unha transposición distinta dos bits da clave sumada bit a bit (XOR) coa saída da iteración anterior. Este entrecruzamento coñécese como esquema *Feistel*.

A estrutura de *Feistel* garante que o cifrado e o descifrado sexan procesos moi semellantes. A única diferenza é que as subclaves se aplican en orde inversa cando desciframos. Isto simplifica enormemente a implementación, en especial sobre hardware, ao non seren necesarios algoritmos distintos para o cifrado e o descifrado.

---

<sup>3</sup> Siglas en inglés de *National Security Agency*.

<sup>4</sup> Siglas en inglés de *Advanced Encryption Standard*.



### ***Exemplos de algoritmos de cifrado en bloque: 3DES***

Inda que ao longo dos anos o algoritmo DES se mostrou moi resistente á criptoanálise, o seu principal problema actualmente é a vulnerabilidade aos ataques de forza bruta, por mor da lonxitude da clave, de só 56 bits. Nos anos setenta era moi custoso realizar unha busca entre as  $2^{56}$  combinacións posibles, pero a tecnoloxía actual permite romper o algoritmo nun tempo cada vez máis curto. Por este motivo, en 1999 o NIST cambiou o algoritmo DES polo “Triplo DES” como estándar oficial, en tanto non estivese dispoñible o novo estándar AES. O Triplo DES, como o seu nome indica, consiste en aplicar o DES tres veces consecutivas. Isto pódese realizar con tres claves ( $k_1, k_2, k_3$ ) ou ben só con dúas ( $k_1, k_2$ , e outra vez  $k_1$ ). A lonxitude total da clave coa segunda opción é de 112 bits (dúas claves de 56 bits). A primeira opción proporciona máis seguridade, pero á custa de empregar unha clave total de 168 bits (3 claves de 56 bits), que pode ser un pouco máis difícil de xestionar e intercambiar. Para conseguir que o sistema sexa adaptable ao estándar antigo, no Triplo DES aplícase unha secuencia cifrado-descifrado-cifrado (E-D-E) en lugar de tres cifrados:

$$\mathbf{C} = \mathbf{E}(k_3, \mathbf{D}(k_2, \mathbf{E}(k_1, \mathbf{M})))$$

$$\text{ou ben: } \mathbf{C} = \mathbf{E}(k_1, \mathbf{D}(k_2, \mathbf{E}(k_1, \mathbf{M})))$$

Deste xeito, para cifrar unha mensaxe  $M$ , primeiro cífrase con  $k_1$ , logo descífrase con  $k_2$  e finalmente vólvese cifrar con  $k_3$  (ou  $k_1$ ). Óllese que no caso de usar 2 claves, se facemos que  $k_1=k_2$ , temos un sistema equivalente ao DES simple.

### ***Exemplos de algoritmos de cifrado en bloque: AES***



A lonxitude da clave do algoritmo DES foise convertendo nun problema a medida que ían aumentando as capacidades de procesamento e o algoritmo se facía cada vez máis vulnerable a un ataque por forza bruta.

En 1977, en vista de que o Triplo DES non era excesivamente eficiente cando se implementa en software, o NIST convocou a comunidade criptográfica a que presentase propostas para un novo estándar que substituíse o DES. Dos quince algoritmos candidatos que se aceptaron, escolléronse cinco como finalistas (MARS, RC6, RIJNDAEL, SERPENT e TWOFISH), e en outubro do 2000 deuse a coñecer o gañador: o algoritmo Rijndael, proposto polos criptógrafos belgas Joan Daemen e Vincent Rijmen. En novembro do 2001 publicouse o documento FIPS 197, onde se adoptaba AES oficialmente.

O Rijndael pode traballar en bloques de 128, 192 ou 256 bits, e a lonxitude da clave tamén pode ser de 128, 192 ou 256 bits. Dependendo desta última lonxitude, o número de iteracións do algoritmo é 10, 12 ou 14, respectivamente. Cada iteración inclúe unha substitución fixa byte a byte, unha transposición, unha transformación consistente en desprazamentos de bits e XOR, e unha suma binaria (XOR) con bits obtidos a partir da clave.

#### **1.4.4. Cifrado en fluxo**

Para algunhas aplicacións, tales como o cifrado de conversas telefónicas, o cifrado en bloques é inapropiada, porque os fluxos de datos se producen en tempo real en pequenos fragmentos. As mostras de datos poden ser tan pequenas como 8 bits ou mesmo 1 bit, e sería un desperdicio reencher o resto dos 64 bits antes de cifrar e transmitilos.

O funcionamento dun cifrado en fluxo consiste na combinación dun texto claro ***M*** cun texto de cifrado ***S*** que se obtén a partir da clave simétrica ***k***, co que se obtén un texto cifrado ***C***. Para descifrar, só cómpre realizar a operación inversa co texto cifrado ***C*** e o mesmo texto de cifrado ***S***.



A operación de combinación que se emprega normalmente é a suma, e como operación inversa, a resta. Se o texto está formado por caracteres, o algoritmo sería como un cifrado César onde a clave vai cambiando dun carácter a outro. A clave que corresponde vén dada polo texto de cifrado **S** (chamado *keystream* en inglés).

Considerando o texto formado por bits, a suma e a resta son equivalentes. Cando se aplican bit a bit, ambas son idénticas á operación lóxica “ou exclusiva”, denotada co operador XOR (eXclusive OR). Daquela:

$$\mathbf{C} = \mathbf{M} \text{ XOR } \mathbf{S(k)}$$

$$\mathbf{M} = \mathbf{C} \text{ XOR } \mathbf{S(k)}$$

Nos esquemas de cifrado en fluxo, o texto claro **M** ten unha lonxitude variable e o texto de cifrado **S** debe ser como mínimo igual de longo. Non é necesario dispoñer da mensaxe enteira antes de empezar a cifrala ou descifrala, xa que se pode implementar o algoritmo para que traballe cun “fluxo de datos” que se vai xerando a partir da clave (o texto cifrado). De aí procede o nome deste tipo de algoritmos.

Existen varias formas de obter o texto cifrado **S** en función da clave **k**:

- Se se escolle unha secuencia **k** máis curta que a mensaxe **M**, unha posibilidade sería repetila ciclicamente tantas veces como sexa necesario para ir sumándolla ao texto en claro. O inconveniente deste método é que se pode romper facilmente, sobre todo canto máis curta sexa a clave.
- No outro extremo, poderíase tomar directamente **S(k) = k**. Isto quere dicir que a propia clave debe ser tan longa como a mensaxe que hai que cifrar. Este é o principio do coñecido **cifrado de Vernam**. Se **k** é unha secuencia totalmente aleatoria que non se repite ciclicamente, estamos ante un exemplo de cifrado incondicionalmente seguro. Este método de cifrado chámase en inglés *one-time-pad* (“caderno dun só uso”). Un exemplo de uso do cifrado de Vernam ocorre ás veces



entre os portaavións e os avións. Neste caso, aprovéitase que nun momento dado (antes de despegar) tanto o avión como o portaavións están no mesmo sitio, co cal intercambiarse un disco duro de 20GB cunha secuencia pseudoaleatoria non é ningún problema. Posteriormente, cando o avión despega, pódese establecer unha comunicación segura co portaavións utilizando un cifrado de Vernam coa clave aleatoria que ambos comparten.

- O que se utiliza na práctica son funcións que xeran **secuencias pseudoaleatorias** a partir dunha **semente** (un número que actúa como parámetro do xerador), e o que se intercambia como clave secreta  **$k$**  é só esta semente. En cada paso o algoritmo atópase nun determinado estado, que virá dado polas súas variables internas. Como as variables serán finitas, haberá un número máximo de posibles estados distintos. Isto significa que, ao cabo dun certo período, os datos xerados volveranse repetir. Para que o algoritmo sexa seguro, interesa que o período de repetición sexa canto máis longo mellor (con relación á mensaxe que hai que cifrar), co fin de dificultar a criptoanálise.

As características deste tipo de cifrado fano apropiado para contornos onde se precisa un rendemento alto e os recursos (capacidade de cálculo, consumo de enerxía) son limitados. Por iso se empregan normalmente en comunicacións móbiles: redes sen fíos, telefonía móbil, etc.

Un exemplo clásico de cifrado en fluxo é **RC4** (*Ron's Code 4*). Foi deseñado por Ronald Rivest en 1987 e publicado en internet por un remitente anónimo en 1994. É coñecido por ser o algoritmo de cifrado empregado no sistema de seguridade **WEP** (*Wired Equivalent Privacy*) recoñecido no estándar **IEEE 802.11**. RC4 utiliza claves de 64 bits (40 bits mais 24 bits do vector de iniciación IV) ou de 128 bits (104 bits mais 24 bits do IV).



#### **1.4.5. Cifrado a partir de funcións resumo**

Ás veces os algoritmos de cifrado non só se usan para cifrar datos, senón que se empregan para garantir a súa autenticidade. Como exemplo de algoritmo destas características pódense citar as chamadas **funcións hash**, tamén coñecidas como **funcións de resumo** de mensaxe<sup>5</sup>.

En xeral, podemos dicir que unha función resumo nos permite obter unha cadea de bits de lonxitude fixa, relativamente curta, a partir dunha mensaxe de lonxitude arbitraria:

$$H = h(M)$$

Para mensaxes **M** iguais, a función **h** debe dar resumos **H** iguais. Pero se dúas mensaxes dan o mesmo resumo **H** non deben ser necesariamente iguais. Isto é así porque só existe un conxunto limitado de posibles valores **H**, xa que a súa lonxitude é fixa.

Para que unha función **h** se poida aplicar en sistemas de autenticación, debe cumprir unha serie de condicións que permitan considerala unha **función resumo segura**. Entre elas destacan a **unidireccionalidade e a resistencia a colisións**.

Para dificultar os ataques contra as funcións de resumo, por unha banda os algoritmos teñen que definir unha relación complexa entre os bits de entrada e cada bit de saída. Por outra banda, os ataques por forza bruta contrárréstanse alongando abondo a lonxitude do resumo.

Ata hai pouco, o algoritmo de resumo máis usado era o MD5 (*Message Digest 5*). Pero como o resumo que obtén é de só 128 bits, e separadamente atopáronse outras formas de xerar colisións parciais no algoritmo, actualmente recoméndase utilizar algoritmos máis seguros, como o **SHA-1**<sup>6</sup>. O algoritmo **SHA-1**, publicado en 1995 nun estándar do

---

<sup>5</sup> *Message Digest*, en inglés.

<sup>6</sup> Siglas en inglés de *Secure Hash Algorithm-1*.



NIST (como revisión dun algoritmo anterior chamado simplemente SHA), obtén resumos de 160 bits. No ano 2002 o NIST publicou variantes deste algoritmo que xeran resumos de 256, 384 e 512 bits.

### 1.5. Criptografía de clave pública ou asimétrica

Os **sistemas de cifrado de clave pública** ou **sistemas de cifrado asimétricos** inventáronse co fin de evitar por completo o problema do intercambio de claves dos sistemas de cifrado simétricos.

Nun algoritmo criptográfico de clave pública empréganse claves distintas para o cifrado e o descifrado. Unha delas, a **clave pública**, pódese obter facilmente a partir da outra, a **clave privada**, pero o caso contrario é practicamente imposible. Os algoritmos de clave pública típicos permiten cifrar coa clave pública ( $k_{pub}$ ) e descifrar coa clave privada ( $k_{pr}$ ):

$$C = e(k_{pub}, M)$$

$$M = d(k_{pr}, C)$$

Inda que tamén pode haber algoritmos que permitan cifrar coa clave privada e descifrar coa pública:

$$C = e(k_{pr}, M)$$

$$M = d(k_{pub}, C)$$

Na práctica, os algoritmos utilizados permiten cifrar e descifrar facilmente, pero todos eles **son considerablemente máis lentos que os equivalentes con criptografía simétrica**. Por iso a criptografía de clave pública se usa normalmente só nos problemas que a criptografía simétrica non pode resolver: o intercambio de claves e a autenticación con non repudio (sinaturas dixitais).



Os mecanismos de **intercambio de claves** permiten que dúas partes se poñan de acordo nas claves simétricas que utilizan para se comunicar, sen que un terceiro que estea a escoitar o diálogo poida deducir cales son estas claves.

A **autenticación** baseada en clave pública pódese utilizar se o algoritmo permite utilizar as claves á inversa: a clave privada para cifrar e a clave pública para descifrar. Se A envía unha mensaxe cifrada coa súa clave privada, todo o mundo poderá descifrala coa clave pública de A, e ao mesmo tempo todo o mundo saberá que a mensaxe só a pode xerar quen coñeza a clave privada asociada (que debería ser A). Esta é a base das **sinaturas dixitais**.

#### **1.5.1. Exemplos de algoritmos de clave pública: Diffie-Hellman**

É un mecanismo que permite que dúas partes se poñan de acordo de forma segura sobre unha clave secreta utilizando unha canle insegura. O algoritmo baséase na dificultade de calcular logaritmos discretos e úsase normalmente como medio para acordar claves simétricas que serán empregadas para o cifrado dunha sesión.

#### **1.5.2. Exemplos de algoritmos de clave pública: RSA**

É o algoritmo máis utilizado na historia da criptografía de clave pública. O seu nome procede das iniciais dos que o deseñaron en 1977: Ronald Rivest, Adi Shamir e Leonard Adleman. A clave pública está formada por un número  $n$ , calculado como produto de dous factores primos moi grandes ( $n = p * q$ ) e un expoñente  $e$ . A clave privada é outro expoñente  $d$  calculado a partir de  $p$ ,  $q$  e  $e$ , de xeito que o cifrado e o descifrado se pode realizar da seguinte maneira:



Cifrado:  $C = M^e \bmod n$

Descifrado:  $M = C^d \bmod n$

Como se pode ver, a clave pública e a privada son intercambiáveis: se se usa calquera delas para cifrar, deberase utilizar a outra para descifrar. A fortaleza do algoritmo RSA baséase, por unha banda, na dificultade de obter  $M$  a partir de  $C$  sen coñecer  $d$  (problema do logaritmo discreto) e, por outra banda, na dificultade de obter  $p$  e  $q$  (e, xa que logo,  $d$ ) a partir de  $n$  (problema da factorización de números grandes, que é outro dos problemas considerados difíciles).

## 2. SINATURA DIXITAL

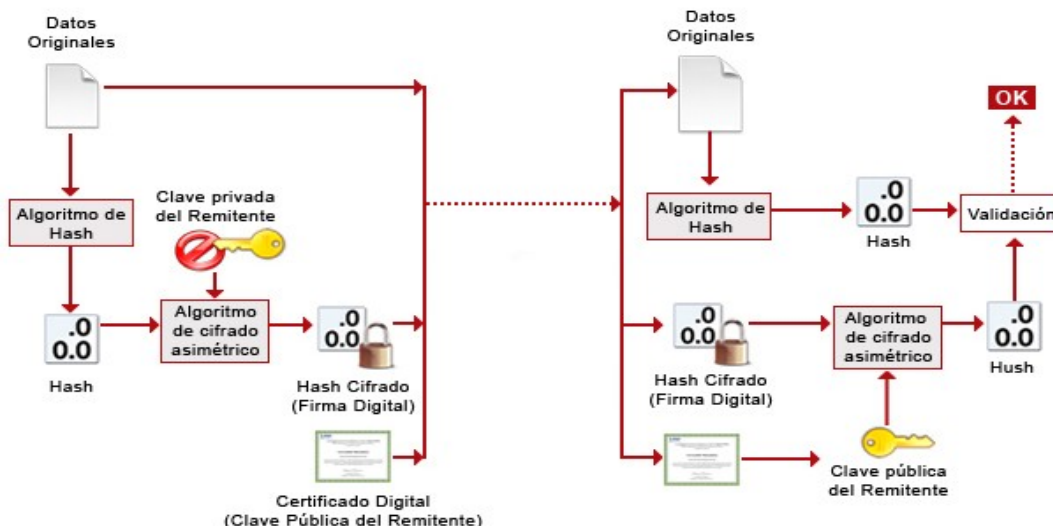
Unha sinatura dixital é, basicamente, unha mensaxe cifrada coa clave privada do asinante. Pero, por cuestións de eficiencia, o que se cifra non é directamente a mensaxe que se vai asinar, senón só o seu resumo calculado cunha función *resumo* segura.

A sinatura dixital está baseada en algoritmos criptográficos asimétricos en que son necesarias un par de claves para o intercambio da información: unha clave pública e unha clave privada. A clave privada está custodiada polo emisor e só a coñece el. A clave pública distribúese entre todos os posibles destinatarios das mensaxes ou documentos asinados. O proceso para realizar unha sinatura dixital resúmese a seguir:

- O emisor obtén un resumo da mensaxe a través dunha función resumo (*hash*). A propiedade máis importante dese resumo ou *hash* é que dous documentos diferentes sempre deben producir resumos diferentes.
- O resumo obtido cífrase coa clave privada do asinante e obtense a sinatura dixital do documento.



- O receptor da mensaxe asinada utiliza a clave pública para descifrar a sinatura, obtén o resumo do documento recibido e comproba que é igual que o resumo que lle chegou cifrado na sinatura dixital. Desta forma garántese que non se manipule o contido da mensaxe.



**Figura 1: Proceso de creación dunha sinatura dixital (fonte: INTECO)**

LEENDA: Datos Orixinais / Algoritmo de Hash / Clave privada do Remitente / Algoritmo de cifrado asimétrico / Hash Cifrado (Sinatura Dixital) / Certificado Dixital (Clave Pública do Remitente) / Validación.

A sinatura dixital, por si mesma, non lle proporciona confidencialidade á mensaxe, pero é habitual enviar as mensaxes asinadas de forma electrónica cifradas coa mesma clave privada utilizada para maior seguridade. A sinatura dixital proporciona:

- Identificación do asinante: a sinatura identifica o asinante de forma única igual que a súa sinatura manuscrita.
- Integridade do contido asinado: é posible verificar que os documentos asinados non sexan alterados por terceiras partes.



- Non repudio do asinante: un documento asinado de forma electrónica non pode ser repudiado polo seu asinante.

### 3. INFRAESTRUTURA DE CLAVE PÚBLICA (PKI)

Como vimos ata o de agora, a criptografía de clave pública permite resolver o problema do intercambio de claves utilizando as claves públicas dos participantes. Pero preséntase outro problema: se alguén afirma ser  $A$  e a súa clave pública é  $k_{pub}$ ,

*como podemos saber que realmente  $k_{pub}$  é a clave pública de  $A$ ?*

Porque é perfectamente posible que un atacante  $Z$  xere o seu par de claves  $(k'_{pr}, k'_{pub})$  e afirme “eu son  $A$ , e a miña clave pública é  $k'_{pub}$ ”.

Unha posible solución a este problema é que exista unha entidade de confianza que nos asegure que, efectivamente, as claves públicas pertencen aos seus supostos propietarios. Esta entidade pode asinar un documento que afirme “a clave pública de  $A$  é  $k_{pub}$ ”, e publicalo para que todos os usuarios o saiban. Este tipo de documento chámase **certificado de clave pública** ou **certificado dixital**, e é a base do que se coñece como **infraestrutura de clave pública** ou **PKI**.

Unha **PKI** está formada, entre outros, polos seguintes elementos:

- **Certificados dixitais:** Son documentos asinados de forma electrónica polas autoridades de certificación que certifican que unha clave pública pertence a un determinado usuario.
- **Autoridades de Certificación (AC, Certification Authority ou CA en inglés):** Son entidades de confianza que se encargan de emitir e revogar os certificados dixitais.



- **Autoridades de Rexistro (AR, *Registration Authority* ou RA en inglés):** Son entidades que rexistran as peticións que fagan os usuarios para obter un certificado, comprobando a veracidade e corrección dos datos que achegan os usuarios nas devanditas peticións e envíanas a unha AC para que sexan procesadas.
- **Autoridades de validación (AV, *Validation Authority* ou VA en inglés):** proporcionan información sobre a vixencia dos certificados electrónicos que, pola súa vez, fosen rexistrados por unha AR e certificados pola AC.
- **Autoridades de selado de tempos (*Time Stamping Authority* ou TSA en inglés):** proporcionan certeza sobre a preexistencia de determinados documentos electrónicos nun momento dado, cuxa indicación temporal xunto co *hash* do documento é asinada pola autoridade de selado de tempo.
- **Directorios de certificados:** proporcionan almacenamento e distribución de certificados e listas de revogación (*Certificate Revocation List* ou CRL en inglés).
- **Hardware criptográfico (*Hardware Security Modules* ou HSM en inglés):** dispositivos criptográficos baseados en hardware que xeran, almacenan e protexen claves criptográficas e adoitan proporcionar aceleración hardware para operacións criptográficas.
- **Tarxetas criptográficas (TI, *Tarxetas Intelixentes*):** son tarxetas que inclúen un *chip* cun microprocesador con módulos hardware específicos para realizar operacións criptográficas.



### 3.1. Certificados dixitais

Un certificado dixital é un documento emitido e asinado de forma electrónica por unha autoridade certificadora onde certifica a asociación entre unha clave pública e un participante.

O certificado garante que a clave pública pertence ao participante identificado e que o participante posúe a clave privada correspondente.

Os certificados dixitais só son útiles se existe unha *Autoridade de Certificación (AC)* de confianza para as dúas partes que os valide.

Os certificados dixitais proporcionan un **mecanismo criptográfico** para levar a cabo a **autenticación**. Tamén proporcionan un mecanismo seguro e escalable para **distribuír claves públicas** en comunidades con gran número de participantes.

O formato dos certificados X.509 é unha recomendación do *ITU*<sup>7</sup> publicada por vez primeira en 1988. A revisión actual do estándar publicouse en 1996 e coñécese co nome de X.509 v3. Os elementos que compoñen un certificado X.509 v3 son:

- **Versión.** É o número de versión do certificado codificado. Os valores aceptables son 1, 2 e 3.
- **Número de serie do certificado.** É un enteiro asignado pola autoridade certificadora. Cada certificado emitido por unha AC debe ter un número de serie único.
- **Identificador do algoritmo de sinatura.** Especifica o algoritmo empregado para asinar o certificado (ex.: *sha1withRSAEncryption*).
- **Nome do emisor.** identifica a AC que asinou e emitiu o certificado.
- **Período de validez.** É o período de tempo durante o cal o certificado é válido e a AC está obrigada a manter información sobre o seu estado.

---

<sup>7</sup> Siglas en inglés de *International Telecommunication Union*.



- **Nome do suxeito.** Identifica o suxeito cuxa clave pública está certificada no campo seguinte. O nome debe ser único para cada entidade certificada por unha AC dada, inda que pode emitir máis dun certificado co mesmo nome se é para a mesma entidade.
- **Información de clave pública do suxeito.** Almacena a clave pública, os seus parámetros e o identificador do algoritmo co que se emprega a clave.
- **Identificador único do emisor.** Este é un campo opcional que permite reutilizar nomes de emisor.
- **Identificador único do suxeito.** Este é un campo opcional que permite reutilizar nomes de suxeito.
- **Extensións.** As extensións do X.509 v3 proporcionan unha maneira de asociar información adicional a suxeitos, claves públicas, etc.
- **Sinatura da AC.** Neste campo almacénase a sinatura dixital do certificado por parte da AC.

Os certificados dixitais diferéncianse **segundo a finalidade** para a que se solicitan. Así, podemos ter certificados para **persoas físicas**, certificados de **servidor**, certificados para a **sinatura de código**, certificados de **entidade**, etc.

### **3.1.1. Autoridade de Certificación**

Unha autoridade de certificación (AC) é unha entidade de confianza encargada de emitir e revogar os certificados dixitais que garanten de forma unívoca e segura a identidade asociada a unha clave pública.

A autoridade de certificación, por si mesma ou por medio dunha autoridade de rexistro, verifica a identidade do solicitante dun certificado antes da súa expedición ou, en caso de certificados expedidos coa condición de revogados, elimina a revogación dos certificados ao comprobar a identidade.



Os certificados son documentos que recollen certos datos do seu titular e a súa clave pública e están asinados de forma electrónica pola autoridade de certificación utilizando a súa clave privada.

A autoridade de certificación é un tipo particular de prestador de servizos de certificación que lexitima ante os terceiros que confían nos seus certificados a relación entre a identidade dun usuario e a súa clave pública. A confianza dos usuarios na AC é importante para o funcionamento do servizo e xustifica a filosofía do seu emprego, pero non existe un procedemento normalizado para demostrar que unha AC merece esta confianza.

A autoridade de certificación encárgase de renovar os certificados, proporcionar servizos de *backup* e arquivo de claves de cifrado. Tamén crea a infraestrutura de seguridade para a confianza dos participantes, establece políticas de operación segura e xera información de auditoría.

O mecanismo habitual **de solicitude dun certificado** de servidor web a unha AC consiste en que a entidade solicitante, utilizando certas funcións do software de servidor web, completa certos datos identificativos (entre os que se inclúe o localizador URL do servidor) e xera unha parella de claves pública/privada. Con esa información, o software de servidor compón un ficheiro que contén unha petición CSR<sup>8</sup> en formato *PKCS#10* que contén a clave pública e que se lle fai chegar á AC elixida. Esta, tras verificar por si mesma ou mediante os servizos dunha autoridade de rexistro a información de identificación achegada e a realización do pagamento, envíalle o certificado asinado ao solicitante, que o instala no servidor web coa mesma ferramenta coa que xerou a petición CSR.

As AC dispoñen dos seus propios certificados públicos, cuxas claves privadas asociadas son empregadas polas AC para asinar os certificados que emiten. Un certificado de AC estará asinado por outra AC de rango superior establecéndose así unha xerarquía de certificación.

---

<sup>8</sup> *Certificate Signing Request.*



Existen **certificados de AC raíz** que están autoasinaados pola propia AC que os emite e que constitúen o elemento inicial da xerarquía de certificación.

Unha **xerarquía de certificación** consiste nunha estrutura xerárquica de autoridades de certificación en que se parte dunha AC autoasinaada, e, en cada nivel, existe unha ou máis autoridades de certificación que poden asinar certificados de entidade final (servidor web, persoa, aplicación de software) ou ben certificados doutras autoridades de certificación subordinadas plenamente identificadas e cuxa política de certificación sexa compatible coas autoridades de certificación de rango superior.

Unha das formas polas que se establece a confianza por parte dun usuario nunha AC consiste na “instalación” no ordenador do usuario (terceiro que confía) do certificado autoasinado da AC raíz da xerarquía en que se desexa confiar.

Cando o modelo de AC inclúe unha **xerarquía**, cómpre **establecer explicitamente a confianza nos certificados de todas as cadeas de certificación** en que se confíe. Para iso, pódense localizar os seus certificados mediante distintos medios de publicación en internet, mais tamén é posible que un certificado conteña toda a cadea de certificación necesaria para ser instalado con confianza.

Un **certificado revogado** é un certificado que non é válido malia que se empregue dentro do seu período de vixencia. Un certificado revogado ten a condición de suspendido se se pode restablecer a súa vixencia en determinadas condicións.

É necesario establecer un mecanismo que permita revogar un certificado antes de que este caduque para os casos de subtracción, erros, cambios de dereitos, ruptura da AC, etc.

Para comprobar se un certificado está revogado, normalmente utilízanse as **CRL (Certificate Revocation List)**. Deste xeito, cando se quere verificar



a sinatura dun documento, o usuario non só debe verificar o certificado e a súa validez, senón que tamén debe comprobar que o certificado non foi revogado, consultando para iso a versión máis recente da *CRL*.

Coas *CRL* opérase seguindo dous modelos:

- **Modelo *pull*:** o cliente que ten que facer a verificación obtén a *CRL* da AC cando o necesita.
- **Modelo *push*:** unha vez que a AC actualiza a *CRL*, a información é enviada aos clientes que necesitan verificar certificados.

Outro método alternativo de comprobación é o *protocolo de estado de certificado en liña OCSP*<sup>9</sup>. Este método permítelles aos clientes desprenderse da xestión do estado dos certificados e obter unha confirmación en liña do estado. Para iso a AC debe poñer a disposición de todos os usuarios potenciais un servizo seguro en liña de alta dispoñibilidade. Este protocolo está definido polo IETF no RFC 2560.

As mensaxes *OCSP* codifícanse en *ASN.1* e habitualmente transmítense sobre o protocolo *HTTP*. A natureza das peticións e respostas de *OCSP* fai que os servidores *OCSP* se coñezan como “***OCSP responders***”. As autoridades de certificación delegan a responsabilidade de proporcionar información de revogacións nos *responders*, creando así unha arquitectura distribuída. Os **clientes envíanlle unha petición de estado** a un *responder* e suspende a súa aceptación ata recibir a resposta. Este modo de funcionamento evita o uso de *CRL*, reducindo así o ancho de banda consumido e o uso de CPU e evítanse os problemas asociados á xestión de información sensible que conteñen as *CRL*.

### **3.1.2. Autoridade de Rexistro**

A autoridade de rexistro (AR) **xestiona o rexistro de usuarios e as súas peticións de certificación/revogación**, así como os certificados de

<sup>9</sup> Siglas en inglés de *Online Certificate Status Protocol*.



resposta a estas peticións. Indícalle á AC se debe emitir un certificado. A autoridade de rexistro é a que autoriza a asociación entre unha clave pública e o titular dun certificado. Durante o ciclo de vida dun certificado, a autoridade de rexistro é a que se encarga das seguintes operacións:

- Revogación.
- Expiración.
- Renovación (extensión do período de validez do certificado, respectando o plan de claves).
- Reemisión do par de claves do usuario.
- Actualización de datos do certificado.

### **3.1.3. Autoridade de Validación**

A autoridade de validación **proporciona información en liña acerca do estado dun certificado**. A autoridade de validación adoita proporcionar dous servizos de validación: o tradicional, permitindo a descarga de **CRL** para que o usuario as interprete el mesmo, ou a través do protocolo **OCSP** (*Online Certification Status Protocol*).

Os usuarios e as aplicacións que desexen obter o estado dun certificado só teñen que realizar unha petición **OCSP** contra a autoridade de validación para obter este estado. A AC actualiza a información da autoridade de validación cada vez que se modifica o estado dun certificado, co que, usando **OCSP**, se dispón de información en tempo real.

### **3.1.4. Autoridade de selado de tempos**

A autoridade de selado de tempos (*Time Stamping Authority* ou **TSA** en inglés) permite **asinar documentos con selos de tempo**, de maneira



que se pode obter unha proba de que un determinado dato existía nunha data concreta. O selo de tempo é un dos servizos máis importantes da sinatura electrónica. Co selo de tempo pódese demostrar que unha serie de datos existiron e non foron alterados desde un momento específico no tempo. Este protocolo descríbese no RFC 3161 e está no rexistro de estándares de internet. Unha autoridade de selado de tempo actúa como terceira parte de confianza, xa que testifica a existencia destes datos electrónicos nunha data e hora concretas. Os pasos que se seguen para xerar un selo de tempo son os seguintes:

- Un usuario quere obter un selo de tempo para un documento electrónico que el posúe.
- Xérase un resumo dixital (tecnicamente, un *hash*) para o documento no ordenador do usuario.
- Este resumo forma a solicitude que se lle envía á autoridade de selado de tempo (*TSA*).
- A *TSA* xera un selo de tempo con esta pegada, a data e hora obtida dunha fonte fiable e a sinatura electrónica da *TSA*.
- O selo de tempo envíaselle de volta ao usuario.
- A *TSA* mantén un rexistro dos selos emitidos para a súa futura verificación.

As aplicacións do selado de tempo son innumerables, xa que os certificados dixitais se emiten cun período de validez determinado e é fundamental, por exemplo, poder verificar que unha sinatura dun documento realizada hai *X* anos atrás efectivamente se fixo cun certificado que non estaba revogado nese momento. Exemplos de uso: factura electrónica, voto electrónico, protección da propiedade intelectual, etc.



### 3.1.5. Directorio de certificados

Os directorios proporcionan almacenamento e distribución de certificados e listas de revogación (*CRL*). Cando unha autoridade de certificación emite un certificado ou *CRL*, envíallo ao directorio e, ademais, garda o certificado ou *CRL* na súa base de datos local. Polo xeral utilízase *LDAP* (*Light-weight Directory Access Protocol*) para acceder aos directorios. O usuario pode obter certificados doutros usuarios e comprobar o estado destes.

### 3.2. Hardware criptográfico

Os **Módulos de Seguridade Hardware** (*Hardware Security Modules* ou *HSM* en inglés) son dispositivos especializados en realizar labores criptográficos. Proporcionan almacenamento seguro de claves e/ou realización de funcións criptográficas básicas como cifrado, sinatura, xeración de claves, etc. Para iso usan interfaces estándar como *PKCS#11* e *CryptoAPI*. Este tipo de dispositivos aumentan significativamente a seguridade en comparación cos certificados baseados en disco polo seguinte:

- A clave privada e as sinaturas dixitais xéranse dentro do *HSM*.
- A clave privada almacénase cifrada dentro do *HSM*.

Se se compara o hardware criptográfico coas tecnoloxías de cifrado baseado en software pódese dicir que o hardware criptográfico é moito máis rápido á hora de realizar o proceso. Dependendo do tipo de hardware *HSM*, os índices de traballo oscilan das 600 ás 4000 operacións de sinatura *RSA* por segundo. Ademais, proporcionan seguridade física ao non poder modificar os algoritmos de cifrado e limitar o acceso ao almacenamento seguro de claves. Isto permite que estas solucións poidan ser certificadas por un terceiro en xerarquías de certificación.



### 3.3. Tarxetas e *chip* criptográficos

Unha tarxeta intelixente (***smart card***), ou tarxeta con circuíto integrado (*TCI*), é calquera tarxeta de tipo peto con circuítos integrados que permiten a execución de certa lóxica programada. Inda que existe un rango diverso de aplicacións, hai varias categorías de *TCI*: **as tarxetas de memoria** conteñen só compoñentes de memoria non volátil e posiblemente algunha lóxica de seguridade. **As tarxetas microprocesadoras** conteñen memoria e teñen capacidade de procesamento limitada. **As tarxetas con *chip* criptográfico** son tarxetas microprocesadas avanzadas onde hai módulos hardware para a execución de algoritmos usados en cifrado e sinaturas dixitais.

Unha **tarxeta intelixente cun *chip* criptográfico** pódese definir como unha chave moi segura, non duplicable e inviolable, que contén as claves e certificados necesarios para a sinatura electrónica gravados na tarxeta e que, ademais, está protexida por un PIN secreto e/ou biometría. O ***chip* criptográfico** contén un microprocesador que realiza as operacións criptográficas coa clave privada, coa característica adicional de que non é posible acceder á clave desde o exterior. As características principais son as seguintes:

- Dobre seguridade: posesión da tarxeta e PIN de acceso (ou mecanismos biométricos).
- Pode ser multipropósito: tarxeta de identificación gráfica, tarxeta de control de acceso/horario mediante banda magnética ou *chip* de radiofrecuencia, tarxeta moedeiro, tarxeta xeradora de contrasinais dun só uso (*OTP*).
- Precísase un middleware (*CSP*) específico para utilizar a tarxeta, así como un lector (*USB*, integrado en teclado ou *PCMCIA*).



- O número de certificados que se poden cargar depende do perfil de certificado, da capacidade do *chip* e do espazo que se reserve para os certificados.

### **3.4. Marco legal e estándares**

#### **Lexislación española**

- **Lei 59/2003**, do 19 de decembro, de sinatura electrónica (BOE n.º 304, 20/12/2003).

#### **Directiva europea**

- **Directiva 1999/93/CE do Parlamento Europeo** e do Consello do 13 de decembro de 1999 pola que se establece un marco comunitario para a sinatura 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 os estándares de sinatura electrónica de acordo coa iniciativa de estandarización europea).

#### **Internet Engineering Task Force (IETF) - Request For Comment**

- RFC 3280: Certificate and Certificate Revocation List (CRL) Profile.
- RFC 3739: Qualified Certificate Profile.
- RFC 3647: Certificate Policy and Certification Practices Framework (Obsoletes RFC2527).



**PKCS (Public Key Cryptography Standards):** Familia de estándares para os sistemas de criptografía de clave pública definidos polos Laboratorios RSA:

- PKCS#1,#2,#4: RSA Cryptography Standard.
- PKCS#3: Diffie-Hellman Key Agreement Standard.
- PCKS#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**

- Universidade de Vigo. Materia de Seguridade en sistemas de información.  
(<http://ccia.ei.uvigo.es/docencia/SSI/>)
- Centro Criptolóxico Nacional (<https://www.ccn.es/>)
- Instituto Nacional de Tecnoloxías da Comunicación - DNI Electrónico.  
(<http://cert.inteco.es>)
- Universidade Politécnica de Madrid - Departamento de Matemática Aplicada da Facultade de Informática.  
(<http://www.dma.fi.upm.es/java/matematicadiscreta/aritmeticamodular/>)
- Universidade Pontificia Comillas (Madrid) - Materia de Seguridade Informática. (<http://www.iit.upcomillas.es/seguridad>)

**Autor: Juan Otero Pombo**

**Enxeñeiro en Informática no Concello de Ourense**

**Colexiado do CPEIG**

(Todas as ligazóns foron verificadas en novembro do 2011)



# **39. VIRUS E OUTRO SOFTWARE MALIGNO. TIPOS. MEDIOS PREVENTIVOS E REACTIVOS. SISTEMAS ANTIVIRUS E DE PROTECCIÓN.**



## **TEMA 39: VIRUS E OUTRO SOFTWARE MALIGNO. TIPOS. MEDIOS PREVENTIVOS E REACTIVOS. SISTEMAS ANTIVIRUS E DE PROTECCIÓN.**

### **ÍNDICE**

#### **1. VIRUS E OUTRO SOFTWARE MALIGNO. TIPOS**

ÍNDICE.....	1
2. MEDIOS PREVENTIVOS E REACTIVOS.....	22
3. SISTEMAS ANTIVIRUS E DE PROTECCIÓN.....	24
4. REFERENCIAS.....	30



## **1. VIRUS E OUTRO SOFTWARE MALIGNO. TIPOS.**

### **1.1. Introducción**

Os sistemas informáticos atópanse permanentemente expostos á ameaza dos virus informáticos cuxo nome provén da analoxía do seu comportamento cos virus biolóxicos.

Desde que apareceron os primeiros virus na década dos setenta, existiron sempre ameazas deste tipo que, nalgúns casos, chegaron a converterse en verdadeiras epidemias que infectaron millóns de ordenadores e deron lugar a perdas económicas importantes. Exemplos destas grandes epidemias foron as xeradas polo verme de *Morris*, *Melissa* ou *ILoveYou*.

En realidade os virus son un subtipo do software maligno en xeral que adoita denominarse como *malware*. Os troianos e vermes son tamén tipos de *malware* que, xunto aos virus, constitúen os tres tipos principais de software maligno que atacan os sistemas informáticos, e existen moitas outras variantes ou subtipos destes: *adware*, *spyware*, *scareware*, etc.

Na actualidade, o tipo de *malware* máis estendido son os troianos, destinados a se instalaren en sistemas de usuarios para roubar información confidencial —nomes de usuarios, números de tarxetas de crédito, etc.— e enviárllela aos atacantes. Os atacantes sérvense da rede internet para recibir e usar a información a milleiros de quilómetros, noutros países onde as leis non recollen delitos deste tipo e se encontran a salvo. Isto permítelles obter benéficos económicos, xa que a creación de programas maliciosos é un negocio lucrativo.



## **1.2. Malware**

O *malware*<sup>1</sup> é un tipo de software que ten como obxectivo infiltrarse ou danar unha computadora sen o consentimento do seu propietario. O termo *malware* é moi utilizado por profesionais da informática para se referiren a unha variedade de software hostil, intrusivo ou molesto.

Colateralmente, o *malware* adoita perseguir un lucro de modo directo ou indirecto por parte do atacante. O nivel de dano que recibe o usuario pode ir desde pequenas alarmas inofensivas a efectos desastrosos, como a perda masiva de datos. Internet resulta ser un medio moi apropiado para distribuír o *malware* de forma que se maximice o número de usuarios afectados.

Podemos establecer unha primeira clasificación de *malware* en tres tipos principais perfectamente diferenciados: virus, vermes e troianos. A partir de aí existen multitude de elementos perigosos que poderían ser catalogados nun ou noutro tipo (ou en varios á vez). Así, é común oír falar de:

- *Adware*: é un software que desprega publicidade de distintos produtos ou servizos. Estas aplicacións inclúen código adicional que mostra a publicidade en ventás emerxentes, ou a través dunha barra que aparece na pantalla simulando ofrecer distintos servizos útiles para o usuario. Normalmente, agregan iconas gráficas nas barras de ferramentas dos navegadores de internet ou nos clientes de correo que teñen palabras clave predefinidas para que o usuario chegue a sitios con publicidade, sexa o que sexa que estea a buscar.
- *Spyware*: ou software espía, é unha aplicación que recompila información sobre unha persoa ou organización sen o seu

---

<sup>1</sup> Das palabras inglesas *malicious software*.





coñecemento nin consentimento. O obxectivo máis común é distribuírllela a empresas publicitarias ou outras organizacións interesadas. Normalmente este software envíalles información aos seus servidores, en función dos hábitos de navegación do usuario. Tamén recolle datos acerca dos sitios web polos que se navega e a información que se solicita neses sitios, así como enderezos IP e URL que se visitan. Esta información é explotada con fins de comercialización, e moitas veces é a orixe doutra praga coma o SPAM, xa que pode dirixir publicidade personalizada cara ao usuario afectado. Con esta información, ademais, é posible crear perfís estatísticos dos hábitos dos internautas. Estes tipos de software adoitan “disfrazarse” de aplicacións útiles que cumpren unha función para o usuario, e moitos sitios recoñecidos ofrecen a súa descarga.

Cómpre salientar que o atacante non ten por que ser un delincuente. Por exemplo, o FBI desenvolveu a súa propia aplicación *spyware*, chamada MagicLantern, usada en investigacións criminais para obter información dos sospeitosos.

- *Crimeware*: é un tipo de programa de ordenador deseñado especificamente para cometer crimes de tipo financeiro ou semellantes e que tenta pasar desapercibido ante a vítima. Por extensión, tamén fai referencia a aplicacións web cos mesmos obxectivos.

Un *crimeware* pode roubar datos confidenciais, contrasinais, información bancaria, etc. e tamén pode servir para roubar a identidade ou espiar unha persoa ou empresa.

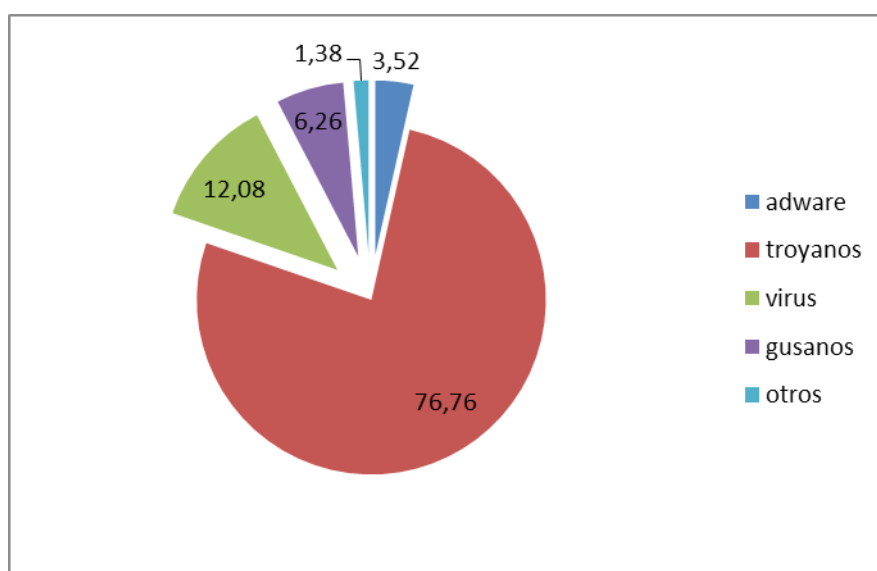
- *Scareware*: termo creado recentemente; é o que se coñece como “software de seguridade falso”. Normalmente este software benefíciase da intención dos usuarios de manter os seus equipos



protexidos, e especificamente emprega técnicas de enxeñería social que se basean sobre todo en crear “paranoia” neles, ofrecéndolles unha solución definitiva aos seus problemas de seguridade. Unha vez instalados, este tipo de programas dedícanse a roubar información como o faría calquera troiano bancario ou usurpador de identidade no equipo da vítima.

- Un novo tipo de *malware* comeza a aparecer con forza asociado ao crecemento das redes sociais. Trátase de aplicacións maliciosas que se moven dentro do contexto de redes sociais destinadas a roubar datos persoais, suplantar identidades, etc. Un exemplo é o verme *Koobface*, que ataca a varias redes sociais moi coñecidas, incluíndo Facebook, Twitter e Myspace.

O seguinte gráfico mostra a distribución de tipos de *malware* detectados en datas recentes<sup>2</sup>:



<sup>2</sup> A maioría de grandes empresas dedicadas ao desenvolvemento de antivirus ofrecen frecuentemente informes sobre os tipos de *malware* detectados, os virus máis activos, alertas especiais, etc.



## **Estatísticas de *malware* por tipos do terceiro trimestre do 2011 (fonte: Panda Security)**

LENDAS: adware / troianos / virus / vermes / outros.

### **1.3. Virus**

Os virus informáticos son porcións de código malicioso que se poden introducir nos ordenadores e sistemas informáticos de formas moi diversas, producindo efectos molestos, nocivos e mesmo destrutivos e irreparables. Ademais, o efecto inicial que o caracteriza é unha infección do equipo, seguida dunha propagación do virus a medida que se executan os arquivos onde reside. É dicir, o código do virus execútase só cando se executa o programa ou se abre o arquivo infectado. Isto é o que diferencia os virus dos vermes (que veremos máis adiante): se non se accede ao programa ou arquivo, o virus non se executa e, xa que logo, non se propaga.

#### **1.3.1. Funcionamento dos virus**

A propia denominación de virus informáticos provén da analoxía do funcionamento destes cos virus biolóxicos. Un virus biolóxico non pode sobrevivir de forma independente. En realidade, non é máis que unha cadea de ADN protexida por unha cobertura externa que se reproduce introducíndose nunha célula hóspede e usando o mecanismo reprodutor desa célula para reproducirse el mesmo.

Un virus informático tampouco é independente nin ten a capacidade de sobrevivir e reproducirse de xeito illado. Trátase dunha secuencia de código maligno que, para poder propagarse, necesita infectar un hóspede, que, neste caso, será un programa informático ou un documento.

Moitos virus agóchanse dentro de arquivos de programas aparentemente limpos. Estes virus coñécense como virus de arquivo e o seu código



execútase ao cargarse na memoria principal da máquina xunto co programa que o contén. Desde alí, o código do virus pode buscar outros programas no sistema que poidan ser infectados. Se atopa un, o virus introduce o seu código nese programa, que, unha vez infectado, tamén pode ser usado para infectar máis programas, iniciando así unha cadea infecciosa semellante a unha epidemia dun virus biolóxico.

A maioría dos virus non só se replican a eles mesmos, senón que tamén realizan outras operacións que habitualmente son daniñas para os seus hóspedes. Así, por exemplo, un virus pode eliminar certos arquivos vitais dun sistema, sobrescribir o sector de arranque dun disco duro deixando o disco inhabilitado, amosar mensaxes na pantalla, emitir ruídos, e moitas outras accións de maior ou menor poder destrutivo.

En xeral, os virus están deseñados para executar o seu código daniño no momento de seren executados. Con todo, hai outros que non atacan nese momento, senón que están deseñados para agardar a unha data concreta ou a un evento particular. Estes virus permanecen en estado latente no sistema ata que actúan.

### **1.3.2. Caracterización dos virus**

Existen moitos tipos de virus, pero podemos indicar algunhas características comúns a todos eles:

- **Latencia:** os virus teñen capacidade para permanecer inactivos, nun estado latente, ata que un evento os active. Este evento pode ser a execución dun programa, a lectura do sector de arranque, etc. Unha vez activado, o virus pode realizar as tarefas para as que estaba programado e replicarse.



- **Tipo de residencia:** os virus máis avanzados son capaces de camuflarse para evitar ser detectados e erradicados. Os virus máis básicos necesitan permanecer continuamente na memoria principal, namentres que estoutros virus máis avanzados son capaces de residir noutros lugares como a memoria secundaria. Poden chegar mesmo a modificar a táboa de asignación de arquivos para camuflar a súa presenza.
- **Forma de infección:** trátase dunha característica común de todos os virus. O medio polo que un virus infecta varía enormemente duns a outros: execución de programas, de macros, conexións mediante determinados protocolos, o simple arranque dunha máquina, etc.
- **Composición:** calquera virus informático conta con tres partes destinadas a cumprir cos seus obxectivos:
  - o **Sistema de reprodución:** encargado de infectar outros sistemas ou arquivos mediante o aproveitamento (ilícito) dos recursos da máquina hóspede.
  - o **Sistema de ataque:** en caso de existir, trátase dunha serie de rutinas destinadas a danar o sistema hóspede dun xeito ou doutro. O dano pode ir desde pequenos inconvenientes, como mensaxes molestas, ata ataques desastrosos, como a perda masiva de datos.
  - o **Sistema de defensa:** destinado a ocultar a presenza do virus mentres sexa posible e a dificultar a súa eliminación.

### 1.3.3. **Ciclo de vida dos virus**

Existen unha serie de fases polas que un virus pasa ao longo da súa vida:



- **Creación:** o virus é creado por programadores que escriben o seu código.
- **Contaxio:** o contaxio inicial ou os contaxios posteriores realízanse cando o programa contaminado está na memoria para ser executado. As vías polas que se pode producir a infección do sistema son disquetes, memorias flash, redes de ordenadores e calquera outro medio de transmisión de información. Estes disquetes contaminantes adoitan conter programas de fácil e libre circulación e carecen de toda garantía. É o caso dos programas de dominio público, as copias ilegais dos programas comerciais, xogos, etc.
- **O virus activo:** Cando se di que un virus se activa significa que o virus toma o control do sistema e, ao mesmo tempo que deixa que funcionen normalmente os programas que se executan, realiza actividades non desexadas que poden causar danos nos datos ou nos programas.

O primeiro que adoita facer o virus é cargarse na memoria do ordenador e modificar determinadas variables do sistema que lle permiten “facerse un oco” e impedir que outro programa o utilice. A esta acción chámase “facerse residente”. Así, o virus permanece á espera de que se dean certas condicións —que varían duns virus a outros— para replicarse ou atacar.

A replicación, que é o mecanismo máis característico e, para moitos expertos, definitorio da condición de virus, consiste basicamente na produción polo propio virus dunha copia de si mesmo, que se situará nun arquivo. O contaxio a outros programas adoita ser a actividade que máis veces realiza o virus, xa que, canto máis rápido e máis discretamente se copie, máis



posibilidades terá de danar un maior número de ordenadores antes de ser descuberto.

- **O ataque:** mentres se van copiando noutros programas, os virus comprobán se se cumpre determinada condición para atacar; por exemplo, se é cinco de xaneiro, no caso do coñecido virus Barrotes. É importante ter en conta que os virus son deseñados coa intención de non seren descubertos polo usuario e, normalmente, sen programas antivirus non son descubertos ata que se produce o dano, na terceira fase do ciclo de funcionamento do virus, coa conseguinte perda de información.
- **Descubrimento:** esta fase comeza cando o virus é detectado, identificado e documentado por vez primeira.
- **Asimilación:** as compañías fabricantes de antivirus modifican as súas solucións para conseguir detectar e eliminar as infeccións causadas polo virus.
- **Erradicación:** o uso exhaustivo de antivirus limita as infeccións do virus eliminando a súa ameaza.

Un virus pode ser catalogado en función do seu índice de perigo, que varía dependendo da fase do ciclo de vida en que se atope. O índice de perigo é unha medida do impacto que produce o seu código malicioso e da capacidade de propagación a outros sistemas. Existen varios niveis de perigo que van desde aqueles virus que causan danos leves e están pouco estendidos, ata aqueles outros que causan danos catastróficos e están moi estendidos.



O nivel de dano indica o prexuízo que un virus causa ao infectar un sistema informático. Un virus pode ser catalogado como daniño (mensaxes ou ruídos molestos, apertura de ventás involuntariamente, etc.) ou moi daniño (destrución ou modificación de arquivos, formato de discos duros, envío de información a terceiros, xeración de moito tráfico en servidores, degradación do rendemento dos sistemas, apertura de buracos de seguridade, etc.).

O grao de propagación indica o estendido que se atopa o virus. Evidentemente, canto máis estendido estea un virus, maiores son as probabilidades de estar afectado por el. A propagación dun virus determínase mediante o “cociente de infección”, que mide a porcentaxe de ordenadores infectados en relación co total de equipos explorados. O grao de propagación dun virus podería variar daquela desde “pouco estendido” ata “epidemia”.

#### **1.3.4. Tipos de virus**

En función de qué é exactamente o que un virus pode infectar, é habitual clasificalos en:

- Virus de sector de arranque mestre<sup>3</sup>: foi o primeiro tipo de virus. Agóchase no código executable do sector de arranque de discos de memoria secundaria ou de discos de arranque externos (disquetes, CD-ROM, etc.). Ata non hai moito tempo, iniciar o ordenador desde un disco de arranque era algo bastante usual, o que significaba que os virus se espallaban rapidamente.

---

<sup>3</sup> Coñecido habitualmente polo seu nome en inglés, *Master Boot Record* (MBR).



- Virus de arquivo: caracterízanse por unirse a arquivos, onde residen, e que poden usar para propagarse entre sistemas. Adoitan infectar arquivos executables, pero tamén existen virus capaces de unirse a arquivos de código fonte, librerías ou módulos de obxectos e mesmo a arquivos de datos. O virus *Jerusalem* (tamén coñecido como *Venres 13*), un dos máis coñecidos, pertence a esta categoría.
- Macrovirus ou virus de macro: os virus de macro fan uso da capacidade que teñen certas aplicacións, como Word e Excel, para executar internamente certos códigos programados chamados macros. O virus anéxase ás devanditas macros e transmítese dun documento a outro a través delas. O virus causante dunha das maiores epidemias da historia, *Melissa*, corresponde a este tipo.
- Virus mixtos, bimodais ou multiparte: trátase dunha combinación de virus de arquivo, de macro e de sector de arranque. Realizan infeccións empregando varias técnicas para se instalar en calquera das localizacións posibles. Considéranse moi perigosos pola gran capacidade de infección que teñen.
- Virus de BIOS: alóxanse na BIOS do ordenador e, cada vez que esta se arranca, infectan os arquivos do sistema. O virus *Chernobyl* realizaba unha infección deste tipo.

Dentro dos anteriores, seguindo un criterio máis amplo, outros tipos de virus identificables serían:

- Virus de compañía: estes virus non modifican os arquivos infectados, senón que crean unha copia do arquivo orixinal e modifican esta copia. Cando o arquivo orixinal se executa, o virus fai que pase o control ao arquivo infectado. Unha variante da infección consiste en



volver nomear o arquivo orixinal e substituílo por outro co nome orixinal que contén directamente o virus.

- **Retrovirus:** trátase de virus especialmente deseñados para evitar ser detectados e infectar os programas antivirus.
- **Virus de sobrescritura:** o código do virus escríbese por riba dun ficheiro executable, destruíndoo. Se o tamaño do virus é menor que o do executable infectado, o resultado final non aumenta de tamaño, dificultando a detección do virus.
- **Virus parasitos:** os arquivos hóspede son modificados só en parte, de modo que non son destruídos e poden mesmo simular que seguen funcionando. Nestes casos o virus alóxase en lugares do arquivo, normalmente ao comezo ou ao final, onde permite que o arquivo hóspede siga parcialmente activo.
- **Virus mutantes:** cando infectan un hóspede, modifican o seu propio código para evitar que a súa “pegada” sexa detectada por programas antivirus.
- **Virus sen punto de entrada:** coñecidos como virus EPO (*Entry Point Obscuring*). Destacan porque a instrución que fai pasar o control ao virus se insire nun lugar indeterminado do arquivo hóspede. Isto fai que o virus non se manifeste ata que se realiza unha acción concreta dentro do executable infectado, o que lles permite permanecer nun estado latente durante moito tempo.
- **Virus de enlace:** caracterízanse porque a infección consiste na inserción dun enlace a algún outro lugar (por exemplo, o último clúster dun disco duro ou un clúster marcado como erróneo) onde realmente se aloxa o código do virus.



- Virus OBJ, LIB e código fonte: en lugar de infectar directamente un executable, infectan librerías ou módulos usados por outros executables. Evidentemente, a súa capacidade para actuar e reproducirse só aparecerá cando a librería sexa utilizada por outro programa.
- Virus de script: trátase de virus que actúan sobre linguaxes de script habitualmente asociadas a páxinas web. Actúan incluíndose en páxinas web ou modificando os scripts que conteñen esas páxinas.
- Virus en estado salvaxe: son aqueles que se atopan en circulación nestes momentos e que están en condicións de infectar.
- Virus de zoológico: son virus que non se encontran en liberdade ou que perderon a súa capacidade para infectar; por exemplo, o coñecido como “virus da pelotíña”.
- Xeradores de virus: son virus que, ao mudaren, xeran novos virus.
- Virus que crean dependencia: cando un destes virus infecta unha máquina, instálase en lugares vitais, de modo que non é posible eliminalo sen facer que a máquina perda eses elementos vitais e deixe de funcionar.
- Bombas de tempo: ocúltanse nos sistemas ata que chega unha data concreta ou transcorre un determinado período de tempo. Nese momento pasan a un estado activo e realizan as súas accións de infección e replicación.

En canto ao modo en que o virus produce a infección do sistema, podemos enumerar os seguintes:



- Engadido ou empalme: é un dos modos máis básicos e clásicos. O código do virus engádese ao final dun arquivo hóspede (habitualmente un executable) e modifícase a estrutura de arranque deste, facendo que o virus se execute en primeiro lugar antes de pasar o control ao ficheiro orixinal. O resultado é un aumento do tamaño inicial do arquivo, permitindo así unha fácil detección.
- Inserción: é un modo máis avanzado de infección en que o virus se instala en zonas de código non utilizadas ou en segmentos de datos para que o tamaño do arquivo non varíe.
- Reordenación: introdúcese o código do virus en sectores do disco duro que quedan marcados como defectuosos e distribúense enlaces aos devanditos sectores no código doutros programas executables que quedan así infectados. A vantaxe deste método é que, ao atoparse o código do virus fóra do arquivo, este código pode ser de gran tamaño e, xa que logo, ter moita funcionalidade. En cambio, este tipo de virus é moi fácil de eliminar; abonda con sobrescribir os sectores defectuosos.
- Polimorfismo: é probablemente o método máis avanzado. Baséase en infectar un arquivo executable, mais realizando unha compactación do código do arquivo hóspede ou do código do propio virus para evitar así un aumento do tamaño que o delate. No momento de actuar, o virus descomprime en memoria o código necesario para se executar.
- Substitución: este método, moi pouco sutil, consiste en substituír directamente o código do programa hóspede por completo polo código do virus. Deste xeito o programa orixinal simplemente desaparece e o único que se executa é o virus.



#### **1.4. Troianos**

Hai miles de anos, ante a imposibilidade de traspasar as murallas da cidade de Troia, os gregos construían un gran cabalo de madeira no interior do cal se ocultaban unha selección dos seus mellores soldados. Colocaron o cabalo diante da cidade e os troianos, cativados pola maxestade do que aparentaba un agasallo dos deuses completamente inofensivo, introduciron o cabalo na cidade.

O tipo de *malware* denominado troiano débelle o seu nome a que, para a súa propagación, utiliza a mesma estratexia que idearon os gregos para entrar en Troia. O concepto básico dun troiano consiste en introducir código malicioso dentro dun sistema que resulte atractivo para a vítima e que, ademais, pareza seguro, de modo que o conxunto parece inofensivo. Este disfraze podería ser desde un xogo descargado de internet ata unha mensaxe de correo electrónico de aparencia inofensiva.

Os troianos son códigos maliciosos que tentan mostrarse como algo útil ou apetecible para que unha vítima os execute. Caracterízanse porque o seu obxectivo é introducirse no sistema e pasar desapercibidos. Namentres se atopan no sistema, pódense dedicar a enviar información (nese caso coñécense normalmente como *spyware*) ou a preparar o sistema para un ataque posterior mediante a instalación de *rootkits*<sup>4</sup> ou a creación de portas traseiras (*backdoors*). Ademais, a diferenza dos virus, non teñen a capacidade de se replicar e infectar outros sistemas por si mesmos.

Actualmente os troianos son moi utilizados co obxectivo de obter datos dunha vítima que os ten instalados no seu sistema sen o seu coñecemento.

---

<sup>4</sup> Conxunto de ferramentas destinadas a permitir que un atacante acceda aos privilexios de administrador do sistema de forma remota.



Isto pódeno facer de moi diversas maneiras, que van desde capturar as pulsacións de teclado (os chamados *keyloggers*) ata acceder e manipular as carpetas de documentos dun usuario.

#### **1.4.1. Tipos de troianos**

Os troianos poden ter características moi diversas, polo que resulta difícil catalogalos. Así e todo, en función do seu obxectivo, podemos establecer a seguinte clasificación:

- **Troianos de control remoto:** o seu obxectivo é proporcionarlle ao atacante o control da máquina onde reside o troiano. Habitualmente, o troiano tentará abrir conexións de rede clandestinas desde as que podería escoitar as ordes do atacante. Exemplos deste tipo de troianos son *Back\_orifice* ou *Netbus*.
- **Troianos que envían datos:** o seu obxectivo é enviarlle ao atacante datos confidenciais tomados da máquina atacada e da información que calquera usuario almacene nela. Polo xeral, o obxectivo é obter usuarios e claves de acceso, números de tarxetas de crédito, contas bancarias, etc. O envío de datos pódese facer de diferentes formas: mediante o envío dun correo electrónico a través dun servidor de correo público, directamente á páxina web do atacante mediante un formulario, etc. Un exemplo deste tipo é o *badtrans.b*, que é capaz de recoller as pulsacións do teclado e envialas vía correo electrónico.
- **Troianos destrutivos:** compórtanse en certa maneira como un virus, xa que o seu obxectivo é causar danos mediante a destrución de información. Poden levar a cabo esta tarefa inmediatamente trala



infección ou actuar como unha bomba lóxica que se activará cando se produza un determinado evento.

- **Troianos de ataque de denegación de servizo:** o obxectivo destes troianos é converter as vítimas en participantes involuntarios en ataques de denegación de servizo distribuídos (DDoS). A máquina infectada denomínase comunmente *botnet* ou máquina *zombi*. O atacante pode utilizar todas as máquinas infectadas para lanzar un ataque coordinado contra outro servidor ou mesmo contra unha conta de correo electrónico (cada máquina infectada podería enviar mensaxes con remitentes capturados). Un exemplo deste tipo é o *WinTrinoo*, unha ferramenta de DDoS moi estendida polo doado do seu uso.
- **Troianos Proxy:** trátase de troianos que permiten converter a máquina infectada nun Proxy a disposición do atacante. Este poderá utilizalo como punto intermedio para outros ataques, dificultando así que o ataque poida ser rastrexado ata a máquina orixinal. É común encadear varios saltos entre máquinas Proxy involuntarias para conseguir evitar que se rastrexo o ataque.
- **Troianos FTP:** caracterízanse por infectar o sistema a través do porto do protocolo FTP (o 21) e permitirlle ao atacante usar o devandito protocolo libremente contra a máquina infectada. FTP permite a transmisión bidireccional de arquivos entre equipos remotos. É dicir, o atacante terá capacidade para copiar e borrar arquivos como lle pete na máquina infectada.
- **Deshabilitadores de software de seguridade:** trátase de troianos avanzados que inclúen ferramentas para evitar ou mesmo eliminar software de protección como antivirus ou *firewalls*. Normalmente acompañan a virus e vermes e instálanse cando se produce a



infección; coma no caso do verme *Goner*, que incluía un troiano deste tipo.

Como dicíamos, existen troianos de difícil catalogación. A realidade é que non deixan de aparecer novos troianos con funcións e obxectivos cada vez máis extravagantes. Por exemplo, o *SMSlock.A*, que literalmente secuestra o equipo infectado impedindo o seu uso e pide un rescate económico a cambio da súa recuperación.<sup>5</sup>

#### **1.4.2. Modos de infección**

As dúas formas máis frecuentes polas que un troiano pode acceder e instalarse nun equipo hóspede son as seguintes:

- **Mediante adxuntos en correos electrónicos ou mensaxería instantánea:** o simple feito de abrir un arquivo adxunto a un correo electrónico ou enviado a través dunha ferramenta de mensaxería instantánea (como *Win32/SdBot*, que se instalaba a través de *MSN Messenger*) pode deixar que un troiano se instale no noso sistema. Incluso algúns xestores de correo que non estean configurados de forma adecuada poden permitir que os adxuntos se executen sen que o usuario o solicite.
- **Mediante a instalación voluntaria de software:** normalmente é software de procedencia dubidosa, *shareware*, *freeware*, versións de proba, etc. Ao descargar da rede este software e instalalo, o troiano instálase nun segundo plano sen que a vítima se decate de nada. Contra isto, cabe destacar o uso cada vez máis habitual de certificados dixitais que permiten identificar tanto o servidor ao que

---

<sup>5</sup> Este tipo de *malware* dedicado a secuestrar recursos e pedir rescate por eles é denominado tamén *ransomware*.



nos conectamos como o propio software que podemos descargar asinado. Deste xeito asegurámonos de que o estamos descargando realmente de onde pretendemos e que non foi modificado durante a comunicación.

En todo caso, a infección por troianos adoita ter un compoñente indispensable de enxeñería social. Polo xeral, os ataques sérvense de enganar para conseguir que a vítima abra os adxuntos ou chegue mesmo a executar o seu contido. Poden simular ser correos de amigos, peticións solidarias, etc. Moitas veces tamén aparentan ser programas útiles que se distribúen gratuitamente para conseguir que a vítima os descargue e instale. Incluso non é estraño atopar troianos detrás de programas antivirus gratuítos.

### **1.5. Vermes**

Un verme é un programa que, unha vez executado, se replica sen necesidade da intervención humana e é capaz de enviar copias de si mesmo a través de redes de comunicacións, sen que sexa preciso que un usuario envíe un correo electrónico infectado nin estableza comunicación explícita ningunha. Propágase de anfitrión en anfitrión facendo un uso indebido de servizos desprotexidos: correo electrónico, ferramentas de mensaxería instantánea, etc. Aínda que a propagación en si non ten por que ser daniña, sucede o mesmo que cos virus: é habitual que os vermes inclúan código malicioso destinado a danar os sistemas infectados. De feito, algunhas das infeccións máis nocivas e coñecidas foron provocadas por vermes: *ILoveYou*, *Kournikova*. Algúns vermes non inclúen código malicioso, pero o seu ataque consiste en reenviarse a si mesmos ata conseguir esgotar os recursos da máquina atacada.

No entanto, ao contrario que os virus, os vermes son programas completos. Non só no sentido de que son capaces de enviar copias de si mesmos a



través de internet, senón porque non necesitan ningún programa hóspede para facelo. Non precisan corromper outros programas e inserir o seu código alí. O seu funcionamento baséase en erros en sistemas operativos, aplicacións ou protocolos, que son aproveitados polos vermes para executarse.

### **1.5.1. Tipos de vermes**

Podemos clasificar os vermes atendendo ao medio que utilizan para a súa propagación:

- **Vermes de redes de área local:** propáganse a través dos recursos compartidos dunha rede local, chegando a bloqueala ou degradando as súas medidas de seguridade. Un exemplo é o verme *Lovgate*, que pode infectar a rede local e tamén enviarse por correo electrónico a outras máquinas.
- **Vermes de redes P2P<sup>6</sup>:** usan este tipo de redes e a súa gran popularidade para incluír nelas arquivos que, ao seren descargados, traen consigo o verme. O verme *Redisto.b* utiliza este tipo de mecanismos.
- **Vermes de correo electrónico:** é probablemente o método máis habitual de propagación e realízase utilizando certos programas clientes de correo. O verme accede aos enderezos de correo da axenda dun usuario e reenvíase usando a propia conta do usuario. Algúns virus máis avanzados poden contar incluso co seu propio servidor SMTP co que enviar as súas copias. *Sircam* ou *Nimda* son exemplos de vermes de correo electrónico.

---

<sup>6</sup> *Peer to Peer*.



- **Vermes de mensaxería instantánea** (IRC, MSN Messenger): outra fonte habitual de entrada para vermes son as aplicacións de mensaxería instantánea, que ademais permiten o envío de arquivos adxuntos.
- **Vermes que se propagan directamente por internet:** os vermes máis avanzados non dependen de ningunha aplicación en particular para se propagar. A súa estratexia de infección pódese basear en atopar portos abertos nas máquinas obxectivo e conseguir introducirse na máquina sen que os usuarios se decaten. Outros vermes infectan os servidores de información, facendo que coa simple petición dunha páxina web o verme poida pasar ao cliente, como é tamén o caso do verme *Nimda* noutro dos seus modos de contaxio.

## **2. MEDIOS PREVENTIVOS E REACTIVOS**

Os medios preventivos contra o *malware* en xeral baséanse en medidas de índole técnica combinadas cunha política de seguridade que promova boas prácticas por parte dos administradores e do persoal da organización. Entre todas estas medidas destinadas á prevención de infeccións e á reacción no caso de que se chegasen a producir, poderíamos citar:

- Utilización de programas antivirus perfectamente configurados e actualizados. É a ferramenta principal na loita contra infeccións. Permiten detectar e, en moitos casos, eliminar os virus. No seguinte apartado falaremos máis en profundidade sobre eles.
- Os sistemas operativos son elementos fundamentais nos sistemas informáticos. Moitos dos métodos de infección baséanse en debilidades ou vulnerabilidades dos sistemas operativos; xa que logo,



deben manterse sempre actualizados, especialmente coas actualizacións específicas de seguridade.

- Tamén hai que manter actualizado o resto do software instalado. Os administradores deben prestarlles atención ás noticias sobre novas vulnerabilidades do software e protexer o sistema ante ataques que se puidesen aproveitar delas.
- Controlar o software xa instalado nas máquinas e todo aquel que se vaia instalar, que, unha vez máis, debería ser soamente o necesario para as tarefas da organización. Por suposto, non se debe permitir a instalación de ningún software que non sexa orixinal; o software pirata é un dos principais puntos de propagación do *malware*.
- Os servizos activos no sistema deben manterse no mínimo número necesario; só deberían permanecer activos aqueles realmente necesarios para os obxectivos da organización. Hai que controlar este feito periodicamente para evitar a apertura de conexións ilegais.
- Manter copias de seguridade dos datos, dos programas e dos sistemas operativos.
- Xestionar adecuadamente as cotas de uso de disco e memoria de cada usuario. Isto evitará que se un usuario provoca unha infección se consuman todos os recursos da máquina afectada, senón só os asignados a este usuario.
- Tanto os administradores como os usuarios deben asumir boas prácticas de prevención: non abrir nunca arquivos adxuntos de procedencia dubidosa, desactivar as opcións de visualización de imaxes e vista previa de xestores de correo, non aceptar descargas nin instalacións de software non iniciadas voluntariamente, comprobar os certificados de procedencia do software, etc.



### **3. SISTEMAS ANTIVIRUS E DE PROTECCIÓN**

#### **3.1. Antivirus**

Os antivirus constitúen a pedra angular sobre a que descansa a maior parte da defensa contra virus, troianos, vermes e *malware* en xeral. O seu obxectivo é o de detectar, bloquear, eliminar e previr infeccións provocadas por virus informáticos. A maioría das solucións existentes hoxe en día son capaces tamén de detectar outros tipos de *malware*: troianos, vermes, *spyware*, *rootkits*, etc.

Cando un antivirus detecta unha infección, poderá actuar de dúas maneiras distintas. Se ten capacidade, podería eliminar a infección sen danar os recursos afectados. Se non é así, ha propoñer poñer en corentena os recursos afectados (habitualmente arquivos), que quedarán illados do resto do sistema, e impedirá que sexan executados. Isto último non elimina a infección, pero bloquéaa, impedindo que o virus execute o seu código malicioso e evitando que se poida propagar.

Os antivirus poden detectar as infeccións seguindo unha destas dúas estratexias:

- Detección de patróns: cada virus ten un patrón de identificación, que adoita ser unha secuencia de código que o identifica. Os antivirus posúen unha base de datos de patróns de virus e compáranas cos arquivos do sistema para ver se existe algunha infección. Así e todo, os virus actuais teñen sinaturas moi pequenas. Isto dificulta a tarefa dos antivirus e pode dar lugar a falsos positivos, é dicir, deteccións que aparentan ser virus e non o son.

Os antivirus baseados en patróns obteñen moi bos resultados. Detectan un gran número de virus, pero para iso cómpre que as súas bases de datos de



patróns estean actualizadas. Todo aquel virus que non figure na base de datos será simplemente indetectable. Ademais, para funcionar dunha forma eficiente, requiren o uso de algoritmos de busca optimizados, xa que a estratexia de detección se basea en escanear o contido de todos os recursos sospeitosos.

- **Heurísticas:** usan técnicas de intelixencia artificial para lograr recoñecer secuencias de accións ou comportamentos asociados a virus. Trátase de recoñecer accións que os virus realizan normalmente (borrar ficheiros, conectarse a internet, modificar arquivos executables, etc.) ou que non realizan (abrir ventás, emitir mensaxes visibles, etc.).

As técnicas heurísticas teñen unha vantaxe fundamental: permiten detectar virus novos sen necesidade de actualizacións. Pero, por outra banda, poden dar lugar a moitos falsos positivos sobre programas que en realidade non son virus.

Existe un variado abano de solucións antivirus, dispoñibles en versións propietarias e tamén en versións ofrecidas gratuitamente ou como software libre. Entre os primeiros podemos atopar solucións moi coñecidas desenvolvidas por grandes empresas: Norton, Symantec, PandaSoftware, McAfee, Kaspersky, etc. Entre os segundos podemos citar AVG, Avast!, ClamWin, etc.

Á hora de seleccionar o antivirus máis apropiado para os nosos sistemas, deberíamos ter en conta as seguintes características:

- **Frecuencia de actualización:** canto máis actualizado estea o noso antivirus, máis preparados estaremos para loitar contra as infeccións.
- **Protección en tempo real:** é conveniente que o axente antivirus resida na memoria principal e realice un escaneamento continuo en



busca de posibles infeccións. Por suposto, isto ten un custo en recursos para a máquina que cómpre ter en conta antes de optar por unha solución deste tipo.

- Capacidade de centralización: algúns antivirus permiten ser instalados en varias máquinas, pero ser controlados e xestionados desde unha única máquina. Isto facilita a tarefa dos administradores.
- Programación de tarefas: é moi interesante que os escaneamentos, que poden consumir bastante tempo e recursos da máquina, poidan ser programados para que se executen en horas de pouca actividade (durante a noite, por exemplo).
- Protección do correo: hoxe en día, un dos puntos de entrada máis comúns para os virus é a través dos correos electrónicos e os seus arquivos adxuntos. É conveniente que o noso antivirus sexa capaz de analizar automaticamente eses arquivos adxuntos.
- Xeración de informes: é moi útil para os responsables da seguridade que o antivirus sexa capaz de xerar informes cos seus resultados e as accións que levou a cabo.

Unha vez que seleccionemos un antivirus, é conveniente ter claro como e onde utilízalo. A norma base é “un equipo, un antivirus”. É dicir, todos os equipos do sistema deberían ter o seu propio antivirus instalado. Con todo, nalgúns equipos concretos o antivirus pódelle ofrecer unha protección máis eficiente ao resto de equipos; por exemplo, en equipos que actúan como Proxy de conexión con redes externas. Un antivirus ben configurado e escaneando en tempo real protéxese a si mesmo de virus, pero tamén evita que gran parte do *malware* pase aos equipos da rede interna e, xa que logo, aos usuarios finais.



No ámbito dos antivirus, o European Institute for Computer Antivirus Research (EICAR) desenvolveu unha proba para validar a súa operatividade. O seu obxectivo é comprobar que un antivirus funciona realmente sen poñer en perigo unha máquina con virus reais. A proba **EICAR**, que así se coñece, consiste nun inofensivo arquivo de texto que debe ser gardado con extensión de arquivo executable. Feito isto, todo antivirus debería detectalo como un virus que ten que eliminar.

### **3.2. Outras medidas de protección**

Para protexer os nosos sistemas contra infeccións sería conveniente dispoñer dalgúns outros elementos:

#### **3.2.1. Devasas (*firewalls*)**

Os vermes propáganse pola rede conectándose a servizos con buracos de seguridade, aloxados en diferentes sistemas ao longo da rede. Ademais de asegurarse de que estes servizos vulnerables non se estean executando, o seguinte paso que debe seguir o administrador é verificar que o *firewall* non permita conexións a estes servizos. Moitos *firewalls* modernos son capaces de filtrar no tráfico da rede aqueles paquetes nos que se detecten certas sinaturas asociadas a virus ou vermes.

#### **3.2.2. NIDS (Sistemas de detección de intrusos de rede)**

Os sistemas de detección de intrusiones de rede son semellantes aos antivirus pero aplicados ao tráfico da rede. Buscan no tráfico de rede sinaturas ou patróns de comportamento relacionados con virus ou vermes. Son capaces de alertar ao usuario atacado ou de deter o tráfico de rede que tenta distribuír o *malware*.



### **3.2.3. HIDS (Sistemas de detección de intrusos de *host*)**

Os sistemas de detección de intrusión de *host*, como por exemplo as ferramentas de software libre Tripwire e Aide, son capaces de detectar cambios realizados sobre arquivos aloxados nun servidor. Baséanse na hipótese de que un arquivo executable, unha vez compilado, non necesita ser modificado. Entón, mediante o control das súas características, tales como tamaño, data de creación e control de integridade, poden detectar decontado se ocorreu algo irregular que apunte a unha infección.

### **3.2.4. Sandboxes**

O concepto de *sandbox* baséase en que unha aplicación ou programa ten o seu propio contorno para executarse e non pode afectar ao resto do sistema. Isto quere dicir que os recursos e os privilexios que a aplicación ten mentres é executada son limitados. A vantaxe dos *sandboxes* é que restrinxen o dano que un *malware* lle pode ocasionar ao sistema infectado simplemente restrinxindo os accesos dos que dispón o *malware*.

Outra opción en auxe é a virtualización, que consiste en crear unha máquina virtual completa mediante produtos como VMWare. Isto illa a máquina virtual do sistema anfitrión “real”, limitando o acceso a este segundo o configurase o administrador.

### **3.2.5. Honeypot**

Denomínase *honeypot* un sistema especialmente preparado para ser ou parecer vulnerable, de maneira que sexa fácil de infectar por *malware*. Por suposto, este sistema non contén información nin programas valiosos (e moitas veces trátase dun sistema virtualizado). Aínda así, este sistema está estritamente monitorado, de modo que o administrador pode obter



información de antemán sobre as ameazas ás que se enfronta o sistema real. Isto permítelle ao administrador xestionar as medidas de seguridade do sistema real para protexerse contra novos virus ou ataques.



#### **4. REFERENCIAS**

- RFC 4949 Internet Security Glossary, Version 2

<http://tools.ietf.org/html/rfc4949>

- Observatorio de seguridade do Instituto Nacional de Tecnoloxías da Comunicación

<http://www.inteco.es/Seguridad/Observatorio/>

- Web e enciclopedia do *malware* de PandaSoftware

<http://www.pandasecurity.com>

- Lista de virus, noticias e 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**

**Enxeñeiro en Informática no Concello de Ourense**

**Colexiado do CPEIG**



**40. REDES LAN, MAN E WAN.  
ESTRUTURA DE REDES:  
TRONCAL, DISTRIBUCIÓN  
ACCESO. REDES PÚBLICAS DE  
TRANSMISIÓN DE DATOS.  
MODELO OSI. PROTOCOLOS DE  
REDE. TCP/IP. ELEMENTOS DE  
INTERCONEXIÓN DE REDES.  
CONCENTRADORES,  
CONMUTADORES,  
REPETIDORES, PONTES,  
ENCAMIÑADORES,  
PASARELAS. XESTIÓN DE  
REDES. CONFIGURACIÓN DE  
REDES.**



**Tema 40. Redes LAN, MAN e WAN. Estrutura de redes: troncal, distribución acceso. Redes públicas de transmisión de datos. Modelo OSI. Protocolos de rede. TCP/IP. Elementos de interconexión de rede: concentradores, conmutadores, pontes, encamiñadores, pasarelas. Xestión de redes. Configuración de redes**

## **INDICE**

### 40.1 Redes LAN, MAN e WAN

#### 40.1.1 PAN

#### 40.1.2 LAN

#### 40.1.3 MAN

#### 40.1.4 WAN

### 40.2 Estrutura de redes: troncal, distribución e 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 xerais

### 40.4 Modelo OSI

#### 40.4.1 Introducción

#### 40.4.2 Conceptos xerais

#### 40.4.3 Transmisión entre entidades do mesmo nivel (pares)

#### 40.4.4 Conexións

#### 40.4.5 Capas do modelo OSI

#### 40.4.6 Críticas ó modelo OSI

### 40.5 Protocolos de rede

#### 40.5.1 Redes de conmutación de circuítos

#### 40.5.2 Redes de conmutación de paquetes

### 40.6 TCP/IP

#### 40.6.1 Enderezamento e sistemas de nomes de dominio

#### 40.6.2 Protocolos IP, TCP



40.7 Elementos de interconexión de rede

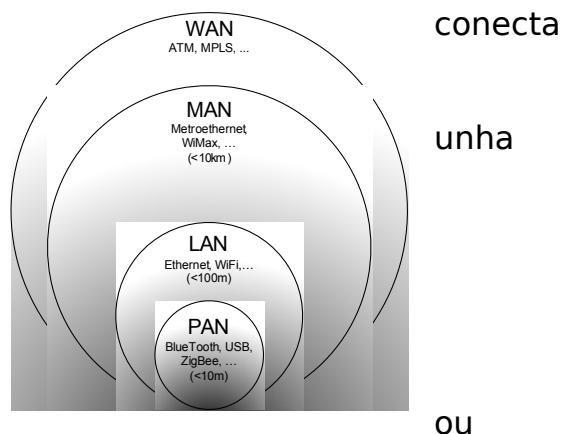
40.8 Xestión de redes. Configuración de redes

40.9 Bibliografía

## 40.1 REDES LAN, MAN E WAN

Unha das formas clásicas de clasificar as redes é pola súa extensión física (ou alcance) da que se obteñen os seguintes tipos:

- Personal Area Network (PAN): rede que conecta elementos próximos a unha persoa.
- Local Area Network (LAN): elementos en unha área xeográfica limitada, como son casa, un edificio, etc.
- Metropolitan Area Network (MAN): algunhas veces referida como Medium Area Network conecta elementos ao longo dunha cidade espazos de similar tamaño.
- Wide Area Network (WAN): son redes de gran extensión cubrindo cidades, unha provincia ou incluso varios países.



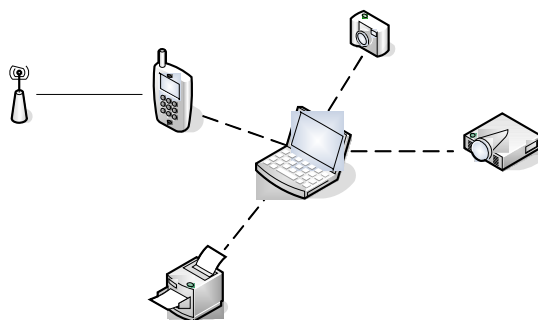
### 40.1.1 PAN

Una PAN é unha rede usada para a comunicación de ordenadores e diferentes dispositivos informáticos próximos a unha persoa. Algúns exemplos destes dispositivos usados en unha PAN son PCs, impresoras, teléfonos, PDAs ou consolas de videoxogos.

A necesidade destas redes é dobre, por un lado conectar dispositivos de uso persoal próximos coma o teléfono móbil cun mans libres ou con dispositivos de recollida de datos médicos. E polo outro permitila mobilidade das persoas aproveitando o uso e conectividade destes



dispositivos, seguindo o exemplo anterior, o móbil pode cambiar de conectarse a un mans libres no coche a conectarse a outro na casa. Unha PAN pode incluír dispositivos conectados por cable e dispositivos sen fíos alcanzando un máximo de 10 metros de distancia.

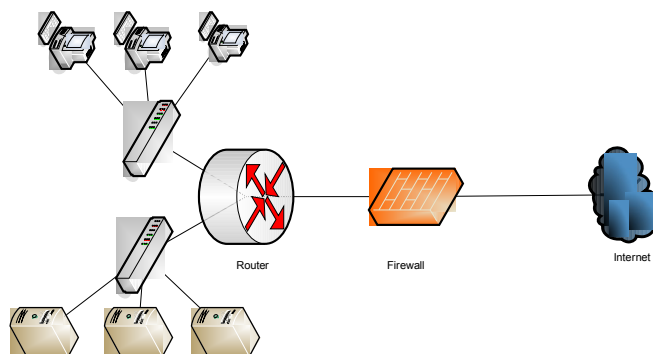


As típicas tecnoloxías nas que se basean as PAN son USB e FireWire para as cableadas e Bluetooth e ZigBee para as sen fíos.

#### **40.1.2 LAN**

Unha LAN conecta ordenadores e outros dispositivos nun espacio limitado como pode ser unha casa, un edificio, unha oficina ou un conxunto de edificios próximos entre si.

Tipicamente a distancia que abarca unha LAN non supera os 100 metros.



O exemplo máis común de LAN dáse no ámbito doméstico e das pequenas empresas onde varios equipos están conectados a un concentrador (switch), posiblemente varios servidores están conectados a outro e eses concentradores están conectados a un encamiñador (router) para a conexión a Internet.

As tecnoloxías dominantes usadas nas LAN son Ethernet (hoxe en día gigabit ethernet) e WiFi (habitualmente 802.11g) aínda que existen moitas



outras tecnoloxías que se empezan (ou continúan) a empregar, como poden ser as baseadas en PLC, por exemplo HomePlug.

#### **40.1.3 MAN**

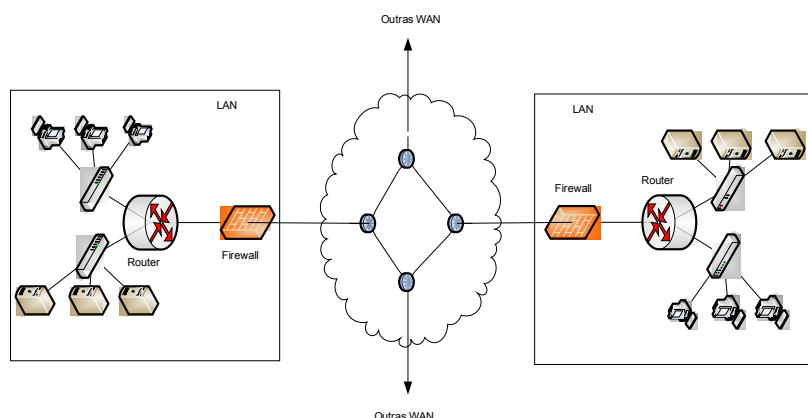
Unha MAN é unha rede optimizada para unha area xeográfica maior que unha LAN, que vai dende varios bloques de edificios ata una cidade. Unha MAN pode ser propiedade dunha organización pero normalmente é usada por moitos individuos e organizacións distintas. A súa utilidade típica e proporcionar conectividade entre varias LAN.

A distancia típica que cubre unha MAN é de 10 km.

Tipicamente estas redes están baseada sen tecnoloxías como MetroEthernet, en redes cableadas, ou Wimax, en redes sen fíos.

#### **40.1.4 WAN**

As WAN son redes de ordenadores que cobren grandes áreas e soen enlazar varias cidades, provincias ou países.



De forma análoga ás MAN a función típica das WAN é conectar varias redes de menor extensión como varias MAN ou varias LAN ademais de conectarse con outras redes WAN.

En contraste cos outros tipos de redes, as WAN non están limitadas a un tamaño máximo.

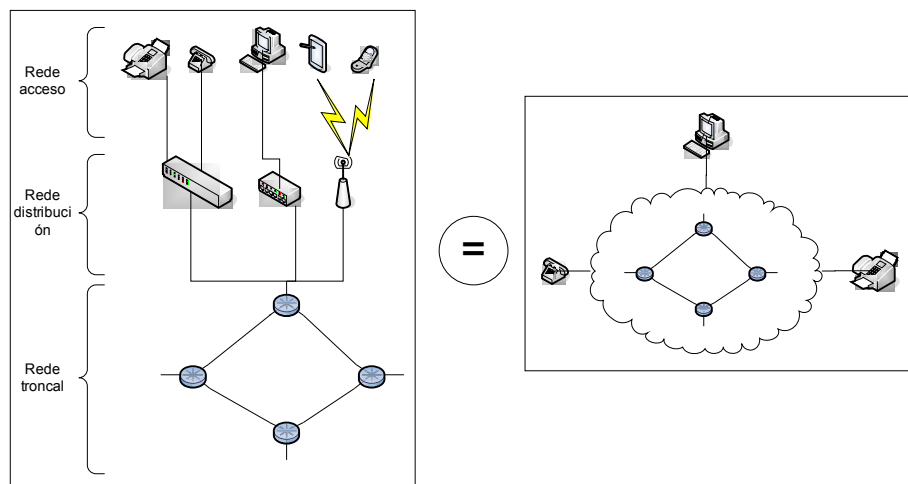
ATM ou MPLS son algunha das tecnoloxías usadas para despregar redes WAM.

### **40.2 ESTRUCTURA DE REDES: TRONCAL, DISTRIBUCIÓN E ACCESO**

As redes (xa sexa a rede dun operador ou dunha compañía) estrutúranse de forma xerárquica. Esta estrutura apórtalles modularidade, permite



aumentar os elementos dun nivel para expandir a rede sen afectar o resto da mesma (escalabilidade) e facilita a identificación e resolución de problemas.



Normalmente esta xerarquía divídese en 3 niveis (aínda que, dependendo da rede concreta, pode haber máis como, por exemplo, nas redes de cable):

- Troncal (en inglés *backbone* ou *core*): é a espiña dorsal da rede conectando os distintos elementos do nivel de distribución.
- Distribución: conecta varios elementos do nivel inferior co nivel superior e soe estar limitada a unha das zonas físicas (por exemplo unha cidade) nas que a rede está presente.
- Acceso: os elementos do nivel de acceso permite conectar os equipos finais.

#### **40.2.1 TRONCAL**

A rede troncal é o nivel xerárquico mais alto dentro da división en niveis e está formada por unha parte da infraestrutura da rede de ordenadores que conecta varias partes da mesma (subredes).

Normalmente a capacidade de transferencia de datos da rede troncal é maior que a das subredes que conecta e posúe camiños redundantes (os operadores soen usar varios aneis e nas redes corporativas soe haber varias conexións).



De tratarse dunha rede corporativa o acceso a Internet soe atoparse na rede troncal.

Un exemplo deste tipo pode ser a rede que conecta as distintas cidades onde da servizo un provedor de internet.

#### **40.2.2 DISTRIBUCIÓN**

No nivel de distribución agréganse os datos provintes dos elementos do nivel de acceso para seren enviados ao nivel superior.

Os elementos deste nivel tamén soen estar redundados pero en menor medida que no nivel superior.

De tratarse dunha rede corporativa é neste nivel onde se establecen as VLANs para cada departamento / división e onde se limitan os dominios de broadcast.

Seguindo o exemplo de antes estas poderían ser a rede que conecta os distintos nodos dentro dunha cidade.

#### **40.2.3 ACCESO**

O nivel de acceso é o nivel da rede onde se conectan os equipos finais (ordenadores, teléfonos, ...).

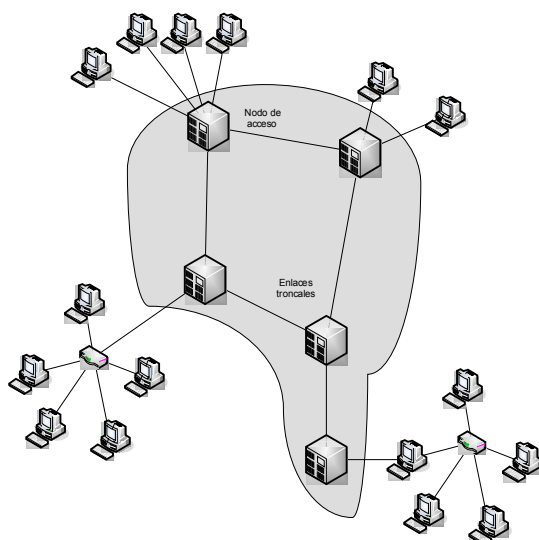
No exemplo anterior esta sería a rede que conecta os distintos usuarios a un nodo.

### **40.3 REDES PÚBLICAS DE TRANSMISIÓN DE DATOS**

Dende o momento que naceu a necesidade de conectar dúas localizacións pasando polo dominio público, naceu a necesidade das redes públicas.

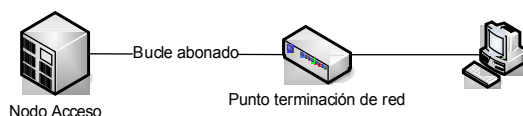
Unha rede pública é aquela a que calquera entidade pode conectarse (normalmente baixo pago dunha cuota) en orde a comunicarse con outra entidade conectada á mesma rede pero en distinta localización xeográfica. Esta dispoñibilidade de conectar calquera dúas entidades leva a que haxa varias entidades (individuos, compañías, gobernos, ...) conectadas a esta rede, como oposición a unha rede privada onde só hai unha entidades aproveitando os recursos da rede. Isto da como resultado que en moitas ocasións as entidades queiran usala rede pública como unha rede privada (xa sexa por simplicidade na configuración, seguridade ou outras razóns) dando lugar ás Redes Privadas Virtuais (VPN polas súas siglas en inglés).





### 40.3.1 CONCEPTOS XERAIS

O bucle de abonado (bucle local ou lazo local) é o cableado que se estende dende os nodos de acceso (centrais de teléfonos, ...) ata o domicilio ou local do usuario.

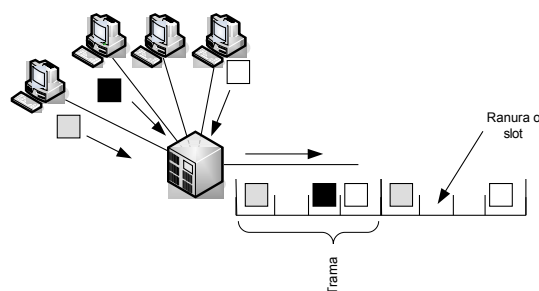


A conmutación é a conexión que realizan os diferentes nodos para lograr un camiño apropiado para conectar dous usuarios dunha rede de telecomunicacións . A conmutación permite a desconxestión entre os usuarios da rede reduciendo o tráfico e aumentando o ancho de banda (comparándoa cos sistemas baseados en bus, por exemplo).

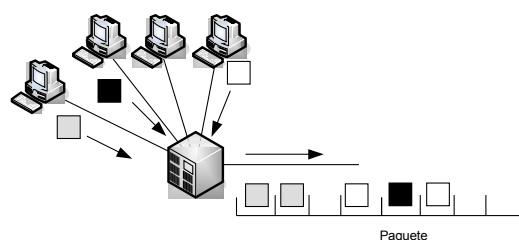
A multiplexación é a combinación de dous ou máis canles de información nun só medio de transmisión usando un dispositivo chamado multiplexor. O proceso coñécese como demultiplexación. Existen moitas estratexias de multiplexación según o protocolo de comunicación empregado pódense combinar para alcanzar o uso máis eficiente; as máis utilizadas son:

- A multiplexación por división de tempo o TDM (Time division multiplexing ). Dentro de esta estratexia podemos encontrar:
  - a. Multiplexación estática ou síncrona.





b. Multiplexación estatística ou asíncrona.



- A multiplexación por división de frecuencia o FDM (Frequency-division multiplexing) e o seu equivalente para medios ópticos, por división de lonxitude de onda ou WDM (de Wavelength).
- A multiplexación por división en código ou CDM (Code division multiplexing).

## 40.4 MODELO OSI

### 40.4.1 INTRODUCCIÓN

Inicialmente, os computadores eran elementos illados que almacenaban nos seus propios ficheiros e precisaban a conexión dos seus propios periféricos. A independencia era tal que, se se necesitaba imprimir un documento aloxado nunha máquina que non dispoñía de impresora, era necesario copiar o ficheiro nun disquete e levalo ata un equipo cunha impresora, conectala e imprimilo neste. Para evitar isto, a solución era instalar una impresora no computador inicial, coa conseguinte duplicación de recursos e dispositivos.

Con instalacións informáticas así, a configuración e xestión de tódolos ordenadores e periféricos a eles conectados supoñía un custo e unha tarefa moi grande, chegando a ser pouco práctica cando o número de computadores foi crescendo nas distintas empresas.



Por esta razón, apareceu a necesidade de conectar os diferentes ordenadores entre si e implantar métodos de comunicación e transferencia de datos entre eles. Nace o concepto de “redes de ordenadores” e “traballo en rede”.

A mediados dos 70 diversos fabricantes desenvolven os seus propios sistemas de redes locais. Sen embargo, a comunicación entre ordenadores pertencentes a redes distintas de distintos fabricantes era imposible, debido a que os sistemas de comunicación de cada rede eran propietarios. É dicir, estaban desenvoltos con hardware e software propios e usaban protocolos e arquitecturas diferentes ós doutros fabricantes.

As empresas déronse de conta da necesidade de abandonar os sistemas propietarios e definir unha arquitectura de rede cun modelo común que permitise conectar varias redes sen problemas.

En 1977, a Organización Internacional de Normas (ISO, International Standard Organization), integrada por industrias representativas do sector, creou un subcomité para o desenvolvemento de estándares de comunicación de datos que permitise a interoperabilidade entre produtos de diferentes fabricantes. Tras varias investigacións acerca dos modelos de rede, elaboraron o modelo de referencia OSI (Open Systems Interconnection), en 1984.

#### **40.4.2 CONCEPTOS XERAIS**

O modelo de referencia para a Interconexión de Sistemas Abertos caracterízase por:

- Ocupase da conexión de sistemas abertos, e dicir, sistemas que permiten a comunicación con outros sistemas.
- Consta de sete capas. Por capa entendese unha (ou varias) entidade(s) que realizan por sí mesma unha función específica. As entidades do mesmo nivel denomínanse entidades pares.
- Representa o primeiro paso á estandarización internacional dos protocolos que se usan nas diversas capas.



- Non é unha arquitectura de rede en sí, xa que non especifica os servizos e protocolos exactos que se teñen que usar en cada capa, se non que só define o que debe facer cada capa.

No modelo OSI existen tres conceptos fundamentais:

- **Servizo:** Capacidade de comportamento dunha capa. Cada capa presta algúns servizos ás entidades que se atopan sobre ela, que acceden ós mesmos a través dos puntos de acceso ó servizo (SAP), intercambiando primitivas de servizo.
- **Interface:** Indica aos procesos da capa superior cómo acceder a ela, especificando cales son os parámetros e qué resultados esperar. A interface entre dúas capas en unha máquina non ten porque ser igual á correspondente noutra máquina.
- **Protocolo:** Conxunto de regras que determinan o comportamento de comunicación horizontal entre entidades pares. Pódense cambiar os protocolos dunha capa sen afectar ás demais.

#### **40.4.3 TRANSMISIÓN ENTRE ENTIDADES DO MESMO NIVEL (PARES)**

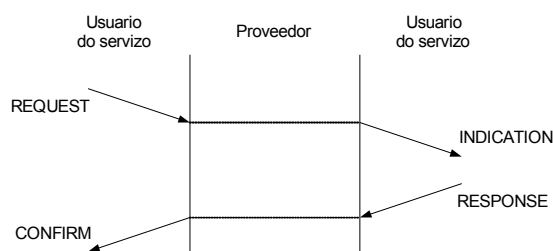
As entidades pares residentes no nivel  $N+1$  comunícanse entre sí a través do nivel  $N$ , mediante o uso de primitivas de servizo. Sen embargo, existe unha comunicación lóxica horizontal entre entidades pares. As regras que regulan esta comunicación veñen reflexadas no protocolo de pares. Polo tanto, na especificación de cada capa existen dous documentos:

- Especificación do servizo, que informa sobre as primitivas existentes. Na descrición das primitivas indícase cuántos parámetros pode ou debe haber e qué información conteñen, pero non se especifica cómo nin con qué formato teñen que ser “pasados”. Isto é un asunto local e definir isto equivale a definir a Interface. Existen catro tipos de primitivas.
  - De petición (REQUEST). Empregada para invocar un servizo e pasarlle os parámetros necesarios para a súa execución.
  - De indicación (INDICATION). Usada para indicar que un procedemento foi invocado polo usuario par do servizo na



conexión e pasalos parámetros asociados ou para indicar ó usuario do servizo o inicio dunha acción por parte do provedor.

- De resposta (RESPONSE). Empregada polo usuario do servizo para recoñecer ou completar algún procedemento previamente iniciado por unha indicación do provedor.
- De confirmación (CONFIRM). Usada polo provedor do servizo para recoñecer ou completar algún procedemento previamente iniciado por unha petición do usuario.



- Especificación do protocolo, que describe as PDUs (Protocol Data Units) e as regras que determinan o seu intercambio entre unidades pares. Existen dúas clases de PDUs:
    - De datos, que contén os datos do usuario final (no caso da capa de aplicación) ou a PDU do nivel inmediatamente superior.
    - De control, que serve para gobernar o comportamento completo do protocolo nas súas funcións de establecemento e ruptura da conexión, control de fluxo, control de erros, etc.
- Non conteñen información algunha proveniente do nivel N+1.

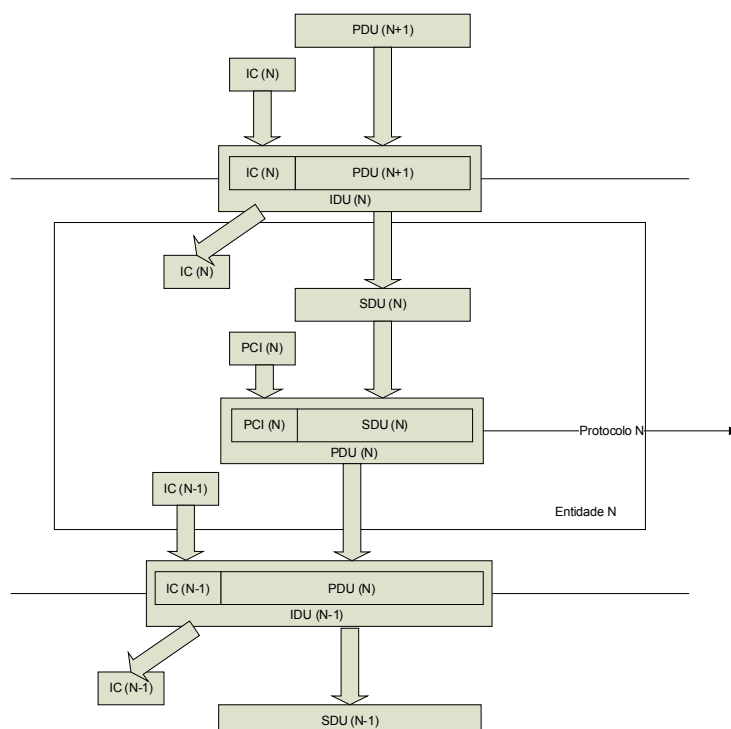
Na seguinte figura ilustrase a nomenclatura utilizada por ISO na pila que envía a información.

- PDU (N) -> Unidade de Datos do Protocolo do nivel N. Contén información de control do protocolo e, posiblemente, datos de usuario. Debe estar puntual e perfectamente definida (sintáctica e semanticamente).
- ICI (N) -> Información de Control da Interface do nivel N. É transferida entre unha entidade N+1 e una entidade N para coordinar o funcionamento local conxunto. A súa sintaxe e semántica son un



asunto local cando actúa como información complementaria na transferencia dunha PDU.

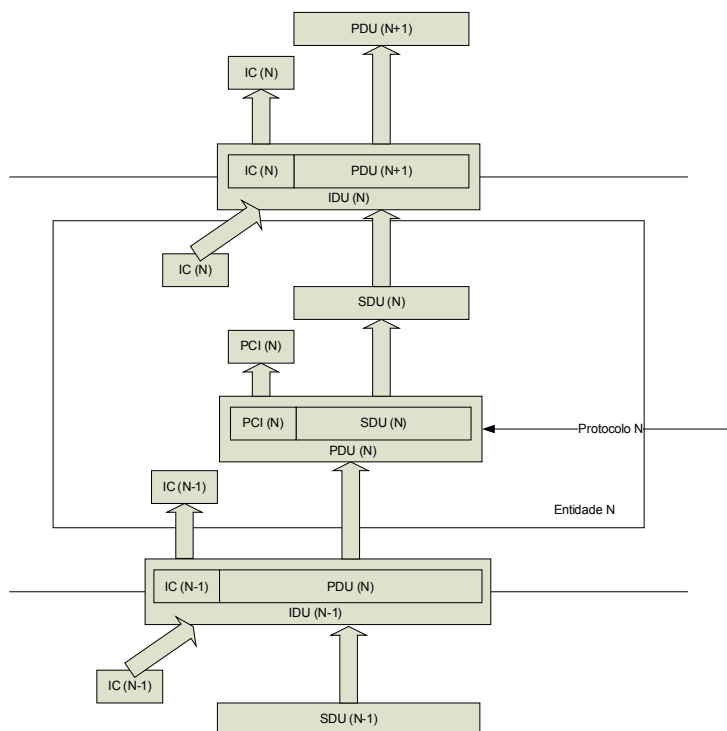
- IDU (N) -> Unidade de Datos da Interfaz do nivel N. É transferida a través do punto de acceso ó servizo N. Contén a ICI máis a totalidade (ou parte) da información da PDU (N+1). A estrutura das informacións da IDU é un asunto local.
- SDU (N) -> Unidade de Datos do Servizo do nivel N. Representa a información entregada polo nivel inmediatamente superior.
- PCI (N) -> Información de Control do Protocolo do nivel N. Información xerada pola entidade N para coordinar o funcionamento conxunto con outra ou outras entidades do nivel N coas que está intercambiando información “horizontal”.
- UD (N) -> Datos do Usuario. Datos transferidos entre entidades do nivel N en nome das entidades do nivel N+1.



Na pila que recibe a información, a relación existente entre as unidades de datos residentes nas entidades do nivel N-1, N e N+1 é similar á representada mais arriba, só que no lugar de engadir elementos o que se



producen son reducións e as frechas son ascendentes, como mostra a figura seguinte.



Cando as unidades de datos dos niveis limítrofes non teñen tamaños compatibles, recórrese a algunha das seguintes funcións:

- Segmentación de la SDU. Cando a SDU é demasiado grande, repartese en máis dunha PDU. A función simétrica no extremo receptor é o reensamblado, que consiste na identificación de varias PDUs cunha sola SDU. A PCI inclúe, neste caso, información adicional para posibilitar o reensamblado.
- Empaquetado da SDU. Cando o tamaño da SDU é máis pequeno que o da PDU, pode ser conveniente ou necesario agrupar varias SDUs nunha soa PDU. O empaquetado é o caso contrario á segmentación da SDU. A función inversa do empaquetado é o desempaquetado, que consiste en descompoñer unha PDU en varias SDUs. O caso de segmentación dáse con máis frecuencia que o de empaquetado.
- Segmentación da PDU. Se a PDU é moi grande, pode ser necesario repartila en máis dunha IDU do nivel inferior. Por iso, na definición de IDU dise que contén a ICI máis a totalidade (ou parte) da PDU.



Tamén neste caso deben existir información adicionais que posibiliten o reensamblado no extremo receptor.

- Concatenación de PDU. Se o tamaño da SDU do nivel inferior, e como resultado, da IDU do nivel inferior, é maior co da PDU, pode convir agrupar varias PDUs sobre unha soa SDU. A función inversa a esta, que se realiza no extremo receptor, é a separación. A concatenación-separación é o caso contrario da segmentación-reensamblado da PDU, sendo este último o máis frecuente.

#### **40.4.4 CONEXIÓNS**

O modelo de referencia OSI está orientado a conexión. Isto significa que, en tódolos niveis, é necesario que se estableza previamente unha conexión para que poida existir intercambio de datos. Sen embargo, existen protocolos que non requiren esta condición, son os non orientados a conexión (connectionless).

Nas comunicacións orientadas a conexión perdese tempo e recursos en establecer e liberala conexión entre dous nodos, pero se garante que o nodo remoto está a escoitar. Polo contrario, nas comunicacións non orientadas a conexión aforrase tempo e recursos, pero á costa de non saber se o outro extremo está a escoitar.

A nivel N-1 establecece unha asociación, una conexión N-1, para que dúas entidades do nivel N poidan comunicarse. A conexión N-1 é un servizo ofrecido polo nivel N-1, a través do cal circulan unidades de información do nivel N.

O Punto de Acceso ó Servizo(SAP) do nivel N identifica a dirección do nivel N á que se conectan as entidades do nivel N+1. A relación entre direccións de dous niveis consecutivos pode ser 1 a 1 (1 dirección do nivel N por cada dirección do N +1), N a 1 (Varias direccións do nivel N por cada dirección do N +1) (non confundir coa multiplexación, que se explica máis adiante) ou 1 a N (1 dirección do nivel N para varias direccións do N +1).

##### **40.4.4.1 ESTABLECEMENTO DE CONEXIÓNS**

Para que dúas entidades N establezan unha conexión, é necesario:



- Dispor dunha conexión N-1 por debaixo. É necesario establecer previamente a conexión N-1 antes de intentar establecer a conexión N, descendendo ata que se encontra unha dispoñible a nivel físico (o nivel mais baixo). Sen embargo, nos niveis altos, aprovéitase a circunstancia de establecemento da conexión N para establecer, o mesmo tempo, a conexión N+1.
- Estar ambas entidades conformes co establecemento.

Unha vez establecida a conexión, é como se a entidade dispuxese dun “tubo” a través do cal puidese enviar datos á súa entidade de comunicación correspondente.

#### **40.4.4.2 LIBERACIÓN DE CONEXIÓNS**

A liberación dunha conexión N é iniciada, normalmente, por unha das entidades N+1 que a está usando. Sen embargo, esta ruptura pode ser tamén iniciada por unha das entidades N que lle dan soporte.

Ó contrario que ocorre no establecemento, a liberación dunha conexión N-1 non implica a liberación da conexión N. Isto é así para permitir que, se a conexión N-1 rompe por dificultades da comunicación, poida intentarse reestablecela ou sustituíla por outra.

A liberación dunha conexión pode ser:

- Abrupta. Libérase de inmediato e os datos pendentes de envío pérdense.
- Suave ou diferida. Antes de rompela conexión, espérase a non ter datos pendentes.

#### **40.4.4.3 MULTIPLEXACIÓN E DIVISIÓN**

A multiplexación é a función que permite utilizar unha soa conexión N-1 para soportar varias conexións do nivel N. Tódolos “tubos” de conexión N viaxan por dentro do “tubo” de conexión N-1. Varias comunicacións entre entidades pares de nivel N realízanse apoiadas nunha soa conexión do nivel N-1. É dicir, as distintas



entidades usan un só punto de acceso ó servizo N-1. A función inversa realizada no receptor denomínase demultiplexación. Non debe confundirse o concepto de multiplexación co de concatenación, xa explicado.

A división é a función que permite a utilización de mais dunha conexión N-1 por unha soa conexión de nivel N. Con elo, o fluxo de datos que soporta pode ser maior. O fluxo de datos do “tubo” correspondente á conexión N repártese entre tódolos “tubos” de conexións N-1. No extremo receptor, a función inversa denomínase recombinação e debe ser capaz de recuperar a orde na que as PDUs foron xeradas polo extremo emisor. Non debe confundirse o concepto de división con de segmentación, xa explicado.

#### **40.4.4.4 TRANSMISIÓN DE DATOS**

Unha capa dunha máquina non pode transferir os datos de forma directa á súa capa par noutra máquina, se non que necesita dos servizos de tódalas capas que se encontran por debaixo dela na xerarquía de capas, pasándose a información cara abaixo ata chegar ó nivel físico, onde se transmiten á máquina receptora.

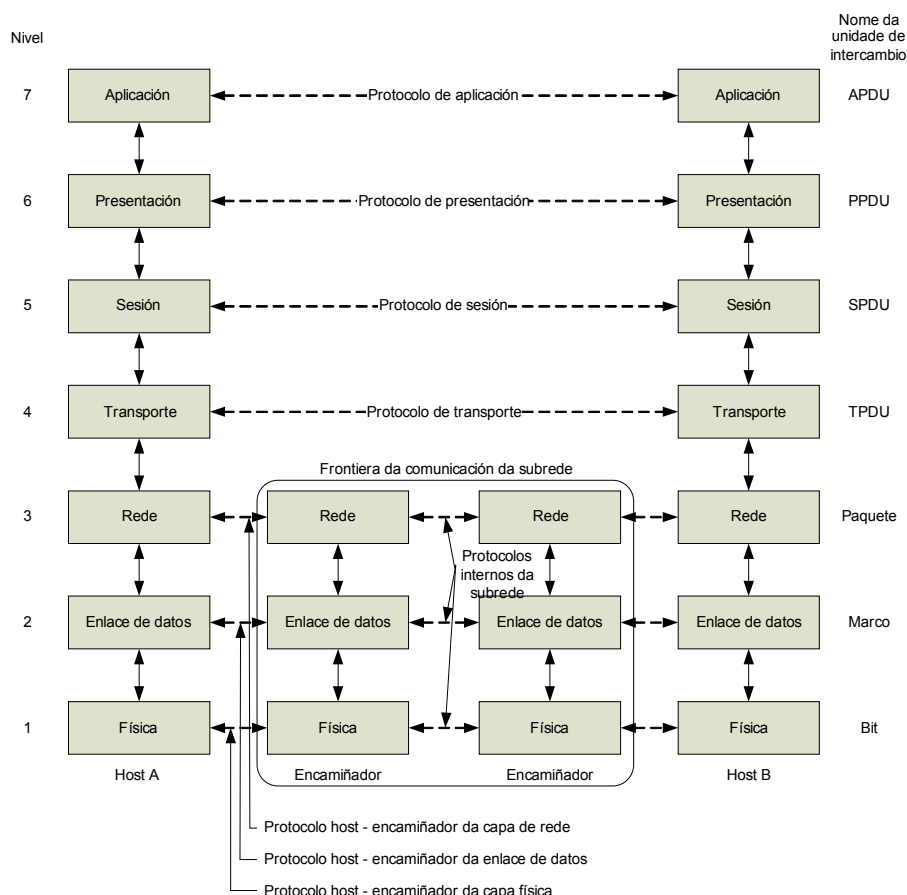
Cada capa utiliza o encapsulamento para colocar a PDU da capa superior no seu campo de datos e agregar calquera encabezado e información final da capa que necesite para realizar a súa función. Desta forma, a medida cos datos se desprazan cara abaixo a través das capas do modelo OSI, o tamaño da mensaxe vai crescendo. A nivel 3, a PDU chamase paquete e inclúe as direccións lóxicas orixe e destino. A nivel 2, a trama inclúe as direccións físicas. E, finalmente, a capa física codifica os datos da trama de enlace de datos nun patrón de uns e ceros para a súa transmisión a través do medio.

Na máquina receptora realízase o proceso inverso, retirando os distintos encabezados, un por un, conforme a mensaxe se propaga cara arriba polas capas.

#### **40.4.5 CAPAS DO MODELO OSI**



Como se dixo no punto anterior, o modelo de referencia OSI dividiuse en sete niveis ou capas, para poder simplificar a implementación da arquitectura necesaria.



Os principios que se aplicaron para chegar a estas sete capas son os seguintes:

- Debese de crear unha nova capa sempre que se precise un grado diferente de abstracción.
- A cada capa asignaselle unha función ben definida ou un conxunto de funcións relacionadas entre si, tratando de resolver en cada capa un problema distinto.
- A funcionalidade de cada capa débese elixir tendo en conta a posibilidade de definir protocolos normalizados a nivel internacional.
- A fronteira das capas será tal que se minimice o fluxo de información a través da interface existente entre ambas.



- O número de capas debe ser o suficientemente grande para non reunir en un mesmo nivel funcionalidades distintas e suficientemente pequeno para que a arquitectura resultante sexa manexable.

#### **40.4.5.1 CAPA FÍSICA (NIVEL 1)**

A capa física está relacionada coa transmisión de bits por un canle de comunicación, de forma que só recoñece bits individuais, sen estrutura algunha. É dicir, a PDU do nivel físico correspóndese con un bit ou, dito doutro modo, cada bit considérase unha unidade de datos.

As consideracións de deseño teñen que ver coas interfaces mecánica, eléctrica e de procedemento, así como co medio de transmisión que está baixo a capa física, asegurando que cando un lado envíe un bit co valor “1”, se reciba no outro extremo o valor “1”, non coma o valor “0”.

A capa física proporciona os seus servizos á capa de enlace de datos. As súas principais funcións son:

- Definición de características materiais (compoñentes e conectores mecánicos) e eléctricas (niveis de tensión, tipo de sinal) que se van a utilizar na transmisión dos datos polo medio físico.
- Definición das características funcionais da interface en canto ó establecemento, mantemento e liberación do enlace físico.
- Definición de regras de procedemento, e dicir, a secuencia de eventos para transmitir.
- Transmisión de fluxos de bits a través do medio.
- Manexo de voltaxes e pulsos eléctricos para representar 1's ou 0's.
- Especificación de cables, polos nun enchufe, compoñentes da interface co medio, etc.
- Especificación do medio físico de transmisión (coaxial, fibra óptica, par trenzado, etc.)
- Garantía conexión física, pero non a fiabilidade da mesma. É dicir, non se realiza ningún control de erros neste nivel. Eso corresponde ó nivel superior.

#### **40.4.5.2 CAPA DE ENLACE DE DATOS (NIVEL 2)**



Posto que a capa física só acepta e transmite una corrente de bits sen preocuparse polo seu significado ou estrutura, corresponde ó nivel de enlace tomalo medio de transmisión en bruto e transformalo nunha liña que pareza estar libre de erros aos ollos da capa de rede.

A capa de enlace de datos pode ofrecer á capa de rede varias clases de servizo con diferentes calidades.

Algunhas das funcións máis importantes da capa de enlace son:

- Establecemento dos medios necesarios para a comunicación fiable e eficiente entre dúas máquinas da rede.
- Estructuración dos datos nun formato predefinido, denominado trama, que soe ser duns centos de bytes, engadindo unha secuencia especial de bits ó principio e ó final da mesma.
- Sincronización no envío de tramas.
- Detección e control de erros provintes do medio físico mediante o uso de bits de paridade, CRC (Códigos de Redundancia Cíclica) e envío de acuses de recibo por parte do receptor que debe procesalo emisor.
- Utilización de números de secuencia nas tramas para evitar perdas e duplicidades.
- Utilización da técnica de “piggybacking”, consistente no envío de acuses de recibo dentro de tramas de datos.
- Resolución dos problemas provocados polas tramas danadas, perdas ou duplicadas.
- Control da conxestión da rede.
- Mecanismos de regulación de tráfico ou control de fluxo, para evitar que un transmisor veloz sature de datos a un receptor lento.
- Control do acceso ó canle compartido nas redes de difusión.

#### **40.4.5.3 CAPA DE REDE (NIVEL 3)**

A capa de rede é unha capa complexa que ofrece os seus servizos á capa de transporte. Responsable da conmutación e encamiñado da información, as súas funcións pódense resumir da seguinte forma:



- Coñecemento da topoloxía da rede, é dicir, da forma en que están interconectados os nodos, con obxecto de determinar a mellor ruta para a comunicación entre máquinas que poidan estar situadas en redes xeográficamente distintas.
- División das mensaxes da capa de transporte en unidades máis complexas, chamadas paquetes (NPDUs), e asignación de direccións lóxicas ós mesmos.
- Ensamblado de paquetes no host destino.
- Establecemento, mantemento e liberación das conexións de rede entre sistemas.
- Determinación do camiño dos paquetes dende a fonte ata o destino a través de dispositivos intermedios (routers):
  - As rutas poden basearse en táboas estáticas.
  - As rutas pódense determinar o inicio de cada conversa.
  - As rutas poden ser dinámicas, determinándose con cada paquete en función da carga da rede.
- Envío de paquetes de nodo a nodo usando un circuíto virtual (orientado a conexión) ou datagramas (non orientado a conexión).
- Control da conxestión.
- Control de fluxo.
- Control de erros.
- Reencamiñamento de paquetes en caso de caída dun enlace.
- Con frecuencia, funcións de contabilidade, para determinar cuántos paquetes, caracteres ou bits envía cada cliente e producir información de facturación.

Esta capa só é necesaria nas redes de conmutación ou redes interconectadas. En redes punto a punto ou de difusión existe un canle directo entre os dous equipos, polo que o nivel 2 proporciona directamente conexión fiable entre os dous equipos.

#### **40.4.5.4 CAPA DE TRANSPORTE (NIVEL 4)**



Tratase dunha verdadeira capa extremo a extremo, dende a orixe ata o destino. A comunicación nos niveis inferiores é entre máquinas adxacentes. A capa de transporte proporciona os seus servizos á capa de sesión, efectuando a transferencia de datos de maneira transparente entre dúas entidades de sesión.

O nivel 4 ten a interface mais sinxela de todo o modelo OSI, sendo a que ten menos primitivas. Non ten primitivas de confirmación, pois considerase a tódolos efectos que é un nivel fiable.

A súa función mais importante é a aceptación de datos da capa de sesión, división en unidades mais pequenas, se é preciso, denominadas segmentos, e envío desta información á capa de rede, asegurando que tódalas partes cheguen correctamente ó outro extremo de forma eficiente, onde son reensambladas.

Outras funcionalidades son:

- Establecemento, mantemento e terminación adecuados dos circuitos virtuais (conexións que se establecen dentro de una rede). Cando se inicia a conexión determinase unha ruta dende a fonte ata o destino, ruta que é usada para todo o tráfico de datos posterior.
- Determinación, no momento do establecemento da sesión, do tipo de clase de servizo de transporte que se proporcionará á capa de sesión:
  - Canle punto a punto libre de erros, que entrega os mensaxes ou bytes na orde en que se envían.
  - Mensaxes illados sen garantía respecto á orde de entrega.
  - Difusión de mensaxes a múltiples destinos.
- Control de fluxo, que desempeña un papel clave nesta capa. O control de fluxo entre nodos é distinto do control de fluxo entre encamiñadores, que ten lugar na capa de rede. Os datos poden ser normais ou urxentes. Estes últimos saltan os mecanismos de control de fluxo.
- Detección e recuperación de erros de transporte.
- Control da conxestión.



- Numeración dos segmentos para previr perdas e dobre procesamento de transmisións.
- Garantía de recepción de tódolos datos e na orde adecuada, sen perdas nin duplicados.
- Asignación dunha dirección única de transporte a cada usuario.
- Illamento das capas superiores dos cambios inevitables da tecnoloxía do hardware.
- Contabilidade a través da rede.

O normal é que a capa de nivel 4 cree unha conexión de rede distinta para cada conexión de transporte que require a capa de sesión. Sen embargo, é posible crear múltiples conexións de rede, dividindo os datos entre elas para aumentar o volume, se se require un volume de transmisión alto. De igual forma, se resulta custoso manter unha conexión de rede, o nivel 4 pode multiplexar varias conexións de transporte na mesma conexión de rede para reducir o custo. Na cabeceira que engade este nivel envíase a información que identifica a qué conexión pertence cada mensaxe. En calquera caso, a capa de transporte debe facer isto de forma transparente á capa de sesión.

#### **40.4.5.5 CAPA DE SESIÓN (NIVEL 5)**

Esta capa proporciona os seus servizos á capa de presentación, facilitando o medio necesario para que as entidades de presentación de dúas máquinas diferentes organicen e sincronicen o seu diálogo e procedan ó intercambio de datos, mediante o establecemento de sesións.

Por tanto, a función principal da capa de sesión é o establecemento, administración e finalización ordenada de sesións entre dúas máquinas. Unha sesión permite o transporte ordinario de datos, como efectuar un login nun sistema remoto ou transferir un ficheiro entre dous nodos, pero tamén proporciona servizos mellorados, útiles nalgúns aplicacións, como os que se detallan a continuación.

- Manexo do control do diálogo (quén fala, cando, canto tempo, half duplex ou full duplex). As sesións poden permitir que o tráfico vaia



nunha única dirección, comunicacións bidireccionais alternadas (half duplex), ou en ambas direccións ó mesmo tempo, comunicacións bidireccionais simultáneas (full duplex). Nas comunicacións half duplex, a capa de sesión axuda a levalo control dos turnos, mediante o manexo de fichas, tamén chamadas testigos ou tokens. Só o lado que posúa a ficha pode efectuala operación.

- Sincronización do diálogo, mediante a inserción de puntos de verificación na corrente de datos (APDU), de modo que si se produce unha interrupción só é necesario repetila transferencia dos datos despois do último punto de verificación. A decisión de ónde colocar os puntos de sincronización é competencia directa do nivel de aplicación. Os puntos de sincronización poden ser de dous tipos.
  - Maior. Necesita confirmación do outro extremo para seguir coa transferencia do seguinte bloque.
  - Menor. Intercálanse entre dous puntos de sincronización maiores. Non necesitan confirmación. Ó confirmarse un punto de sincronización maior, danse por confirmados os puntos menores intermedios.

O bloque entre o primeiro e o último punto de sincronización maior chámase actividade. Cando se establece unha conexión de sesión, automaticamente ábrese unha actividade, para poder traballar. Só un tipo de datos concreto pode enviarse fora dunha actividade, os datos de capacidades (CD), que son datos de control. As actividades divídense en unidades de diálogo, que é o contido entre dous puntos de sincronización maior consecutivos.

Nesta capa a referencia ós dispositivos é polo nome e non pola dirección. Ademais, é aquí onde se definen as API's (Application Program Interface). O protocolo de nivel de sesión é orientado á aplicación, xa que as súas funcionalidades adáptanse ás necesidades da aplicación.

As unidades de datos do nivel de sesión, SPDUs, que regulan o diálogo, flúen horizontalmente a través do nivel 5, pero son postas en circulación



por iniciativa dos correspondentes procesos de aplicación que residen no nivel 7. É dicir, a capa de sesión non é un nivel autónomo que teña capacidade para tomar decisións sobre quen fala e quen escoita. Estas decisións están reservadas ás entidades da capa de aplicación. O nivel 5 só proporciona os mecanismos para que as entidades de aplicación poidan regularo diálogo entre si.

No parágrafo anterior falase como se a capa de aplicación residise directamente encima da de sesión. Isto non é así. Como se indica na enumeración de capa, a capa de sesión ofrece os seus servizos á capa de presentación. O que ocorre é que o protocolo de nivel 6 non é un protocolo “normal”. De feito, a maior parte das primitivas que comunican a capa de presentación co nivel 7 son translación exacta das correspondentes primitivas entre o nivel 6 e a capa de sesión.

#### **40.4.5.6 CAPA DE PRESENTACIÓN (NIVEL 6)**

A diferenza das capas inferiores, as explicadas ata agora, que se ocupan só do movemento fiable de bits dun lado a outro, a capa de presentación encargase da sintaxe e a semántica da información que se transmite. Ademais, illa de ditas capas inferiores o formato dos datos das aplicacións específicas.

As estruturas de datos a intercambiar teñense que definir de forma abstracta, mediante a codificación destes datos dunha maneira estándar acordada, facendo posible así a comunicación entre computadoras con representacións locais diferentes. A capa de presentación manexa estas estruturas de datos abstractas e convérteas da representación da computadora á representación estándar da rede e viceversa.

Ademais desta funcionalidade, a capa de presentación ofrece á capa de aplicación os servizos de:

- Garantía de que a información que envía a capa de aplicación dun sistema poida ser entendida e utilizada pola capa de aplicación doutro sistema.



- Acordo e negociación da sintaxe de transferencia na fase de establecemento da conexión. A sintaxe escollida pode ser cambiada durante o tempo que dure a conexión.
- Definición do código a utilizar para representar unha cadea de caracteres (ASCII, EBCDIC, etc.)
- Interpretación dos formatos de números...
- Compresión dos datos, se é necesario.
- Aplicación de procesos criptográficos, se así se require. É o nivel clave para o sistema de seguridade do modelo OSI.
- Formateo da información para a súa visualización ou impresión.

#### **40.4.5.7 CAPA DE APLICACIÓN (NIVEL 7)**

É a capa do modelo OSI máis próxima ó usuario. Difire das demais capas en que non proporciona servizos a ningunha outra capa OSI, se non as aplicacións que se encontran fora do modelo. Tódalas capas anteriores serven de mera infraestrutura de telecomunicacións, é dicir, manteñen en bo estado o camiño para que flúan os datos. É a capa de aplicación a que fai posible que unha rede se poida usar, a pesar de estar abstraída de tódalas restantes funcións necesarias para o establecemento da comunicación.

As aplicacións mais importantes que fan uso desta capa, para que os procesos das aplicacións accedan ó entorno OSI son, entre outras:

- Correo electrónico. Primeira aplicación que se normalizou en OSI.
- Terminal virtual de rede abstracta, que diferentes editores e programas poidan manexar.
- Transferencia de arquivos.
- Carga remota de traballos.
- Servizos de directorio.
- Login remoto (rlogin, telnet).
- Acceso a bases de datos.
- Sistemas operativos de rede.



- Aplicacións Cliente/Servidor...

Por suposto, no nivel 7 tamén hai cabida para aplicacións “particulares” deseñadas por e para un núcleo reducido de usuarios, pero carecen de demasiado interese.

As PDUs da capa de aplicación, APDUs, son de formato moi flexible e variable. Entre dúas APDUs poden encontrarse diferencias substanciais en cuanto ó seu tamaño, número de campos presentes, etc, que dependen das necesidades de cada momento.

A cada unha das partes dunha aplicación que se encarga dunha tarefa específica denomínase Elemento de Servizo de Aplicación (ASE). O conxunto de tódolos ASEs que forman unha aplicación concreta e a relación entre eles forman o contexto de aplicación.

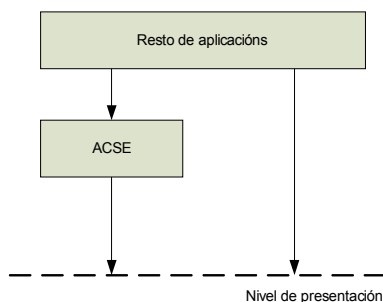
Hai ASEs válidos para varias aplicacións:

- ACSE (Association Control Service Element). Establecemento, manexo e liberación ordenada ou abrupta de conexións. Utilízanse tódalas aplicacións.
- RTSE (Reliable Transfer Service Element). Garante a fiabilidade na transferencia de datos, solucionando os problemas que se produciran do nivel 4 cara arriba. É responsable de manexar tódalas funcións de nivel 5. Utilízanse algunhas aplicacións, non todas.
- ROSE (Remote Operation Service Element). Facilita o traballo de petición de operacións remotas e devolución dos resultados.

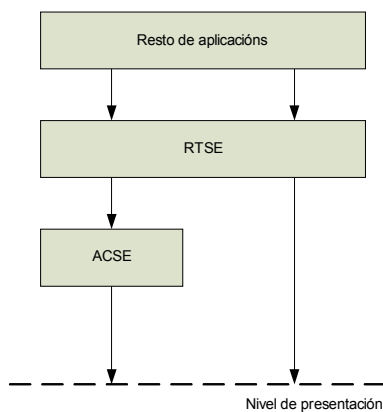
As aplicacións compóñense dunha mestura de elementos específicos e comúns. As seguintes figuras ilustran as relacións entre ASEs comúns.

- Para aplicacións que non manexan RTSE nin ROSE. Teñen que xestionar por si mesmas os puntos de sincronización, os token, etc.

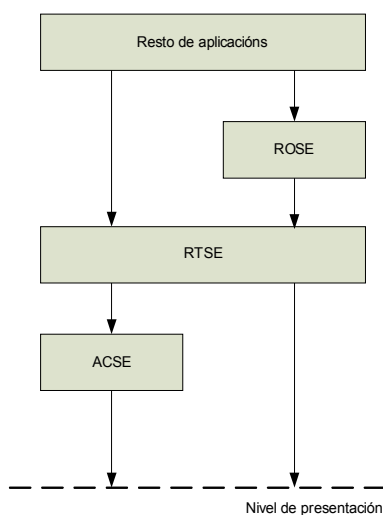




- RTSE é o que se encarga das RTSE actividades, os puntos de sincronización, etc. Neste caso, a aplicación non manexa directamente o nivel 5, se non que o fai a través de RTSE.

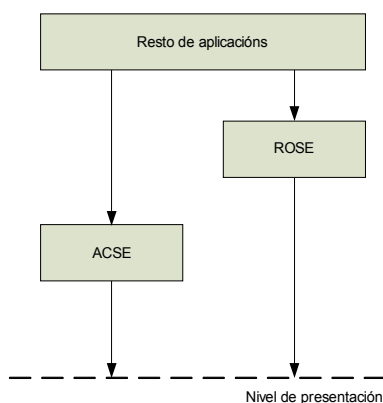


- Para as aplicacións que fan uso dos tres ASEs comúns.



- Neste caso, ROSE traballa directamente sobre o nivel 5.





#### 40.4.6 CRÍTICAS Ó MODELO OSI

A verdadeira razón de que o modelo OSI teña sete capas e que, no momento de deseño, IBM tiña un protocolo patentado de sete capas, chamado SNA (System Network Architecture, Arquitectura de Rede de Sistemas) e, nesa época, IBM dominaba a industria da computación. Por outro lado, o proceso de estandarización foi demasiado longo. Cando aínda se traballaba na definición de OSI, xa existían implementacións completas e gratuítas de TCP/IP e aplicacións como e-mail, telnet, ftp, etc. Algúns dos problemas ou fallos que se detectaron no modelo de referencia OSI son:

- Aínda que o modelo OSI, xunto coas súas definicións e protocolos de servizos, é moi completo; hai que recoñecer que os estándares son difíciles de implementar e ineficientes na súa operación. As implementacións iniciais foron enormes, inmanexables e lentas.
- OSI desenvolveuse antes de que se inventaran os protocolos. Así que os deseñadores non souberon ben qué funcionalidade por en cada capa.
- A capa de sesión ten pouco uso na maior parte das aplicacións.
- A capa de presentación está practicamente baleira.
- Polo contrario, as capas de rede e de enlace de datos están moi cheas, ata tal punto que chegaron a dividirse en múltiples subcapas, cada unha con funcións distintas.
- Algunhas funcións, como o direccionamento, o control de fluxo e o control de erros, reaparecen unha e outra vez en cada capa.



- Omisión da administración da rede no modelo.
- Aínda que no presente documento situouse na capa de presentación a función de cifrado e seguridade dos datos, inicialmente deixouse fora do modelo por falta de acordo sobre en qué capa colocalo.
- Na capa de rede ofrecese servizo orientado a conexión e non orientado a conexión. Sen embargo, na capa de transporte, onde o servizo é visible aos usuarios, só se ofrece comunicación orientada a conexión.
- O modelo está dominado por unha mentalidade de comunicacións. As computadoras son diferentes dos teléfonos. Moitas das decisións tomadas son inapropiadas para a forma de traballar das computadoras e o software. O modelo dun sistema controlado por interrupcións non se axusta conceptualmente cás ideas modernas da programación estruturada.

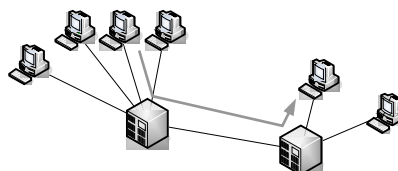
## **40.5 PROTOCOLOS DE REDE**

### **40.5.1 REDES DE CONMUTACIÓN DE CIRCUÍTOS**

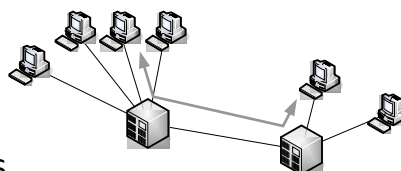
A conmutación de circuítos é un tipo de conexión que realizan os diferentes nodos dunha rede para lograr un camiño apropiado para conectar dous usuarios. Neste tipo de conmutación establececese un canal de comunicacións dedicado entre as dúas estacións. Resérvanse recursos de transmisión e de conmutación da rede para o seu uso exclusivo no circuítos durante a conexión.

Neste tipo de redes a comunicación ten tres fases:

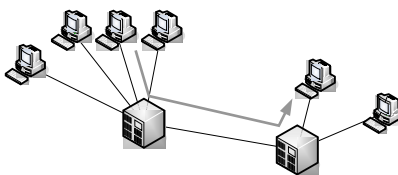
1. Establecemento do circuítos



2. Transmisión dos datos







### 3. Liberación do circuío

Este tipo de redes usa TDM, o que permite dispor dun retardo fixo e predicible. Dende o punto de vista do usuario o enlace é punto a punto. Un exemplo deste tipo de rede é a Rede Telefónica Conmutada (RTC).

#### **40.5.2 REDES DE CONMUTACIÓN DE PAQUETES**

A conmutación de paquetes é o sistema máis usado para o envío de datos nunha rede de ordenadores. Un paquete é un grupo de información que consta de dúas partes: os datos propiamente ditos, e unha información de control, que especifica a ruta a seguir ó longo da rede ata o destino do paquete. Existe un límite superior para o tamaño dos paquetes; en caso de superalo é necesario dividir o paquete en outros máis pequenos. Tamén pode existir un límite inferior para o tamaño do paquete dependendo da tecnoloxía de transmisión usada.

Existen dúas técnicas para a transmisión de paquetes nas redes de conmutación de paquetes:

- A baseada en circuítos virtuais: Moi similar á conmutación de circuítos, a diferenza radica en que cos circuítos virtuais a ruta non é dedicada, se non que un único enlace entre dous nodos pódese compartir dinamicamente no tempo por varios paquetes (TDM asíncrono). Require as mesmas 3 fases que a conmutación de circuítos (Establecemento do circuío, transmisión de datos e liberación do circuío).
- A baseada en datagramas: Non debemos establecer o circuío de forma previa á transferencia de información. Cada paquete debe levar a dirección de destino e tratase de forma individualizada, sen establecer ningún vínculo cos demais paquetes que levan datos de A a B, sexan ou non da mesma aplicación.

Os usuarios comparten os medios de transmisión por TDM estatístico. Os retardos son agora variables, dependentes da carga instantánea na rede.



Cando se establece o circuíto virtual ou cada vez que se transmite un datagrama conmutador debe seleccionar por que enlace encamiña os datos usando un algoritmo de encamiñamento. Esta decisión ten que ser tomada por cada nodo da rede implicado. Esta decisión debe tomarse minimizando o custo (tempo, recursos, ...) e, con este fin, cada nodo constrúe unha táboa (usando o mencionado algoritmo), chamada táboa de encamiñamento que indica por que enlace debe transmitir os datos para chegar o destino.

No caso dos circuítos virtuais (ademais da táboa de encamiñamento que se usará para seleccionar a ruta do circuíto virtual), o nodo debe construír una táboa cos circuítos virtuais, onde se asigna o identificador dun circuíto virtual dun enlace (entrada) a outro enlace (saída).

Un exemplo de rede que usa circuítos virtuais é X.25 e un de conmutación de paquetes é IP.

#### **40.6 TCP/IP**

O alumbramento do modelo TCP/IP remontase á rede ARPANET. Esta era unha rede de investigación controlada polo Departamento de Defensa de EE.UU. Pouco a pouco foron conectándose institucións, mediante o uso de liñas da rede telefónica. A necesidade de buscar unha arquitectura de referencia nova xurdiu cando empezaron a engadirse redes de satélite e radio cos conseguíntes problemas de interactuar cos protocolos existentes. Un dos principais obxectivos desta nova arquitectura foi a capacidade de conexión de múltiples redes entre si desembocando no que hoxe coñecemos como modelo TCP/IP.

TCP/IP ten unha maior aplicación que o modelo OSI, xa que se desenvolveu antes e implantouse TCP/IP mentres se esperaba ao protocolo OSI.

Ademais, como tódalas especificacións asociadas aos protocolos TCP/IP son de dominio público, e polo tanto non hai que pagar nada para usalos, foron utilizados extensivamente por entidades comerciais e públicas para crear entornos de redes abertos.

As vantaxes de TCP/IP son:



- Agrupa redes, creando unha rede maior chamada Internet.
- É independente do hardware dos nodos, do sistema operativo e da tecnoloxía do medio e do enlace.
- Ofrece capacidade de encamiñamento adaptativo, transparente ó usuario.
- É o software de rede mais dispoñible universalmente.

As diferenzas co modelo OSI son:

- Mentres que en OSI a distinción entre os conceptos de servizo, interface e protocolo é clara, en TCP/IP non existía esta distinción inicialmente. Posteriormente, intentouse axustar isto para acercarse máis a OSI.
- OSI desenvolveuse antes de que se definiran os protocolos, mentres que TCP/IP foi, en realidade, o resultado dos protocolos existentes.
- Unha diferenza clara é que OSI conta con sete capas ben definidas e TCP/IP só ten 4.
- O modelo OSI considera os dous tipos de comunicación, orientada e non orientada a conexión, na capa de rede. Sen embargo, na capa de transporte, ofrece unicamente orientada a conexión. Por outro lado, o modelo TCP/IP na capa de rede só soporta comunicacións non orientadas a conexión pero considera ambos modos na capa de transporte.

#### **40.6.1 ENDEREZAMENTO E SISTEMAS DE NOMES DE DOMINIO**

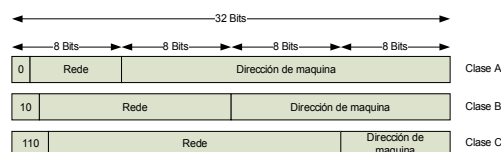
##### **40.6.1.1 ENDEREZAMENTO IP**

Cada nodo da rede ten unha dirección IP única, formada polo número da rede e o número do nodo. A asignación de direccións IP esta regulado pola ICANN (Internet Corporation for Assigned Names and Numbers) para evitar que unha mesma dirección sexa usada por varias máquinas.

Unha dirección IP na súa versión 4 consta de 32 bits de lonxitude e xeralmente escríbese como concatenación de 4 bytes en formato decimal separados por puntos.

Tradicionalmente as direccións agrúpanse en clases que podemos ver na seguinte imaxe.





Ademais historicamente existen as clases D (para multicast) e E (orixinalmente reservada para uso futuro).

Hai determinadas direccións reservadas, estas son:

- 0.0.0.0 Esta IP usase polas estacións cando aínda non teñen unha IP asignada.
- A dirección da rede representase por unha dirección IP onde a primeira parte é a parte de rede e o resto de bits están a 0.
- 127.X.X.X Rede se loopback (acceso á propia máquina dende a propia máquina) sendo a dirección predilecta para este fin 127.0.0.1 aínda que todas funcionan da mesma forma.
- 255.255.255.255 É a dirección de broadcast (normalmente non retransmitido polos encamiñadores). De forma equivalente a dirección de broadcast para a rede e da dirección da rede co resto de bits a 1.

Cando o número de rede vai todo a ceros asúmese que esa dirección se refire á rede actual.

Non tódalas direccións son únicas na rede (estas direccións son coñecidas como IPs públicas) se non que existe nunha serie de rangos reservados para o seu usos en rexistro:

- Para a clase A: de 10.0.0.0 a 10.255.255.255 (8 bits rede, 24 bits estación).
- Para a clase B: 172.16.0.0 a 172.31.255.255 (16 bits rede, 16 bits estación). 16 redes clase B contiguas, uso en universidades e grandes compañías.
- Para a clase C: 192.168.0.0 a 192.168.255.255 (24 bits rede, 8 bits estación). 256 redes clase C contiguas, uso de compañías medias e pequenas.

#### **40.6.1.1.1 SUBREDES**



Tódolos ordenadores dunha rede deben ter o mesmo número de rede. Esta característica pode chegar a ser un problema a medida que crecen as redes. A solución a este problema é a división dunha rede en varias partes ou subredes. Desde o punto de vista do mundo exterior as subredes non son visibles, se non que se ve a rede como un todo.

A parte da dirección que define o número de rede permanece igual unha vez dividida, pero o número de estación divídese en número de subrede e número de estación.

#### **40.6.1.1.2 MÁSCARAS DE REDE**

A ferramenta que permite obter a dirección de rede dunha dirección IP dada é a máscara de rede. É unha especie de dirección IP especial que, en binario, ten tódolos bits que definen a rede postos a 1 e os bits correspondentes ó host postos a 0. Así, as máscaras de rede dos diferentes tipos de redes principais son:

- Rede de clase A: Máscara de rede = 255.0.0.0
- Rede de clase B: Máscara de rede = 255.255.0.0
- Rede de clase C: Máscara de rede = 255.255.255.0

A máscara de rede posúe a propiedade de que cando se combina, mediante unha operación AND lóxica, coa dirección IP dun host obtense a dirección propia da rede na que se encontra o mesmo.

Nas redes onde existen definidas subredes aplicase o concepto de máscara de subrede, que é o resultado de por a 1 tódolos bits que representan rede ou subrede e a 0 tódolos bits que representan una estación.

#### **40.6.1.1.3 DIRECCIÓNS IPV6**

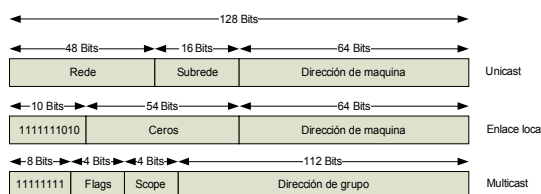
A diferenza de IPv4, que utiliza unha dirección IP de 32 bits, as direccións IPv6 están compostas de 128 bits, ampliando enormemente a capacidade de direccións do protocolo IP.

De similar forma que as direccións IPv4 as direccións IPv6 agrúpanse en 8 números de 4 díxitos hexadecimais separados entre si por dous puntos (:). Cando un destes números é todo ceros, pódese omitir na escritura da dirección.



Existen 3 tipos de direccións:

- Unicast que identifica un único interface de rede. O protocolo IP entrega os paquetes enviados a unha dirección unicast á interface específico.
- Anycast que é asignada a un grupo de interfaces, normalmente de nodos diferentes. Un paquete enviado a unha dirección anycast entregase unicamente a un dos membros, tipicamente o host con menos custe, según a definición de métrica do protocolo de encamiñamento. As direccións anycast non se identifican facilmente pois teñen o mesmo formato que as unicast, diferenciándose unicamente por estar presente en varios puntos da rede. Case calquera dirección unicast pode utilizarse como dirección anycast.
- Multicast que tamén é usada por múltiples hosts, que conseguen a dirección multicast participando do protocolo de multidifusión (multicast) entre os routers de rede. Un paquete enviado a unha dirección multicast é entregado a tódolos interfaces que se uniron ó grupo multicast correspondente.



#### 40.6.1.2 DNS

O sistema de nomes de dominio (DNS, polas súas siglas en inglés: Domain Name System) é un sistema de nomes xerárquico e distribuído para equipos, servizos ou calquera tipo de recurso conectado a Internet. A súa función máis importante é traducir os nomes comprensibles polos usuarios a direccións IP numéricas co propósito de localizalos na rede.

O espazo de nomes está organizado en árbore con cada nodo ou folla na arbore contendo cero ou máis rexistros de recursos (sendo estes os que almacenan a información correspondente ó nome de dominio).

Unha zoa DNS consiste nun ou varios dominios e subdominios dependendo da autoridade delegada nese xestor. Esta autoridade pode dividirse creando mais zoas (que normalmente ocuparanse de subdominios) cuia



autoridade será asumida por outra entidade, perdendo a orixinal a autoridade sobre estas novas zoas.

#### **40.6.2 PROTOCOLOS IP, TCP**

##### **40.6.2.1 CAPAS DO MODELO TCP/IP**

A arquitectura do modelo TCP/IP consta de 4 capas:

- Aplicación. Proporciona comunicación entre procesos ou aplicacións en ordenadores distintos. Contén tódolos protocolos de alto nivel.
- Transporte. O igual que a capa de transporte de OSI, permite que se poida establecer unha comunicación entre as entidades pares dos nodos orixe e destino. Proporciona, por tanto, transferencia de datos extremo a extremo, asegurando que os datos chegan no mesmo orde en que foron enviados e sen erros. Esta capa tamén pode incluír mecanismos de seguridade. Pódese resumir a funcionalidade da capa de transporte como calidade de servizo. Neste nivel defínense dous protocolos importantes, que se explican en detalle máis adiante:
  - TCP : Protocolo orientado a conexión que proporciona entrega fiable de mensaxes entre máquinas. Realiza control de fluxo para que un emisor rápido non poda saturar a un receptor lento.
  - UDP: Protocolo sen conexión e non fiable. Utilízase en aplicacións onde a entrega rápida é máis importante que a entrega precisa, como transmisión de voz e vídeo, e en aplicacións de consulta de petición e resposta.
- Rede. Esta capa é o eixo da arquitectura e permite que os nodos inxecten paquetes en calquera rede e os fagan viaxar de forma independente ao seu destino, sen importar se está na mesma rede, ou se hai outras redes entre elas. A súa misión principal, por tanto, é o encamiñamento dos paquetes, pero sen garantía de que cheguen ó extremo final nin de que o fagan no mesmo orde no que se enviaron. Se se desexa unha entrega ordenada, as capas superiores deben reordenar os paquetes. Neste nivel defínese un formato de paquete e o protocolo IP, que se detalla seguidamente.



- Capa do nodo á rede. O modelo TCP/IP non dí moito do que sucede baixo a capa de rede, existe un gran baleiro. Esta abstracción da topoloxía de rede pon en relieve a capacidade da capa de rede de soportar calquera tipo de rede por debaixo. O que está claro é que debe permitir que un nodo se conecte á rede para que poida enviar por ela paquetes IP. O protocolo que regula esta conexión pode variar de un nodo a outro.

#### **40.6.2.2 PROTOCOLO IP**

IP proporciona un servizo de distribución de paquetes caracterizado por:

- Transmisión de datos en datagramas (paquetes IP).
- Non é orientado a conexión, polo que os paquetes son tratados de forma independente e cada un pode seguir una traxectoria diferente na súa viaxe cara o host destino.
- Non é fiable, polo que non garante a entrega dos paquetes, nin a entrega en secuencia, nin a entrega única. Isto é responsabilidade do protocolo TCP da capa superior.
- Non implementa control de erros nin control de conxestión.
- Pode fragmentalos paquetes si é necesario.
- Direcciona os paquetes empregando direccións lóxicas IP de 32 bits (en IPv4).
- Verifica a integridade do paquete en sí, non dos datos que contén.

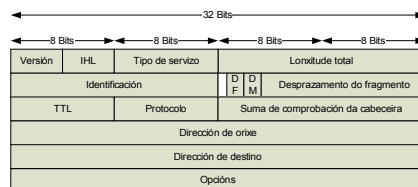
##### **40.6.2.2.1 DATAGRAMA IP**

Os datos proporcionados pola capa de transporte son divididos en datagramas e transmitidos a través da capa de rede. Ao longo do camiño poden ser fragmentados en unidades máis pequenas para atravesar unha rede ou subrede cuxa MTU (Unidade de Transferencia Máxima) sexa máis pequena que o paquete. Na máquina destino, estas unidades son reensambladas para volver a ter o datagrama orixinal que é entregado á capa de transporte.

Un datagrama IP está formado por un corpo e una cabeceira. O corpo corresponde co segmento TCP/UDP da capa de transporte. A cabeceira



ten unha parte fixa de 20 bytes e unha parte opcional de lonxitude variable. Na seguinte figura pode verse a estrutura de dita cabeceira.



#### 40.6.2.2 ICMP PROTOCOLO DE CONTROL DE MENSAXES DE INTERNET

Cando ocorre algún suceso, ICMP (Internet Control Message Protocol) é o protocolo encargado de informar do mesmo. Non toma ningunha decisión ó respecto, isto é tarefa das capas superiores. Os mensaxes de ICMP encapsulanse dentro do campo de datos dos paquetes IP.

A mensaxe ICMP ten tres campos fixos e a continuación, o corpo da mensaxe que varía en función do tipo. Os campos obrigatorios son:

- Tipo (8 bits). Utilízase para distinguilos tipos de mensaxes ICMP, descritos máis abaixo, e determinar o seu formato.
- Código (8 bits). Nalgunhas mensaxes ICMP utilízase este campo para distinguir distintos subtipos dentro dun tipo de mensaxe, é dicir, para ofrecer unha descrición concreta do error que se produciu.
- Checksum (16 bits). Código de protección contra erros de transmisión.

Existen diversos tipos de mensaxes ICMP. Por un lado, están as mensaxes informativas. Todos eles conteñen, ademais dos 3 campos fixos, un identificador de 16 bits e un número de secuencia, tamén de 16 bits.

O resto son mensaxes de erro. Todos eles conteñen, ademais dos 3 campos fixos, o encabezado e os 8 primeiros bytes do datagrama que ocasionou o erro.

#### 40.6.2.3 PROTOCOLO TCP

TCP (Transmisión Control Protocol) deseñouse especificamente para enviar unha secuencia de bytes fiable a través dunha rede non fiable. As súas características principais son:

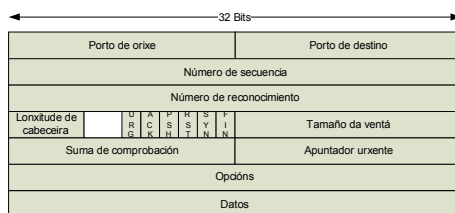


- É un protocolo orientado a conexión. TCP establece unha conexión entre un socket da máquina transmisora e un socket da máquina receptora. Un socket é un punto terminal ó que se lle asigna un número de socket formado pola dirección IP do host e un número de 16 bits local a ese host, chamado porto. Unha vez establecida a conexión pódense transferir datos entre a orixe e o destino. Aínda que cada paquete enviado desde o host orixe pode viaxar por un camiño ou ruta diferente ata chegar ó host destino, por medio do protocolo IP, TCP consegue que pareza que existe un único circuíto de comunicación entre ambos hosts.
- É un protocolo fiable.
- É un protocolo de fluxo non estruturado, con posibilidade de enviar información de control xunto aos datos.
- É un protocolo con transferencia de memoria intermedia. Co obxecto de minimizar o tráfico de rede e conseguir unha transferencia eficiente, vanse almacenando os datos do fluxo de transmisión ata completar un paquete o suficientemente largo como para ser enviado. No destino, almacénanse os datos recibidos ata completar unha secuencia completa e correcta para pasala ó proceso de aplicación destino.
- Usa conexións full-dúplex, é dicir, o tráfico pode ir en ambos sentidos ó mesmo tempo.

#### **40.6.2.3.1 SEGMENTO TCP**

Cada segmento comeza con unha cabeceira de formato fixo de 20 bytes. Esta pode ir seguida de opcións de cabeceira. Tras as opcións, se as hai, encóntranse os datos. Tamén pode haber segmentos sen datos, usados normalmente para acuses de recibo e mensaxes de control. O formato da cabeceira TCP é a seguinte:





#### 40.6.2.3.2 CONEXIÓNS TCP

Para establecer unha conexión un dos lados espera pasivamente unha conexión entrante e o outro executa unha primitiva de conexión, especificando a dirección e o porto IP co que se desexa conectar, o tamaño máximo de segmento TCP que está disposto a aceptar e, opcionalmente, algúns datos de usuario. Esta primitiva xera un segmento TCP co bit SYN a 1 e o bit ACK a 0. Ó chegar o segmento ao destino a entidade TCP revisa se hai algún proceso escoitando no porto indicado no campo de porto de destino. Se non o hai, envía una contestación co bit RST a 1 para rexeitala conexión. En caso contrario, o proceso recibe o segmento TCP entrante e pode aceptar ou rexeitala conexión. Se a acepta, envíase de volta un segmento de acuse de recibo.

Cando se cae un host, por seguridade, non pode reiniciarse durante o tempo máximo de paquete (120 seg.) para asegurar que no haxa paquetes de conexións previas vagando por Internet.

Para liberar unha conexión, calquera das partes pode enviar un segmento TCP co bit FIN activado, indicando que non ten máis datos que transmitir. Ó recoñecerse o FIN, ese sentido apágase. Sen embargo, o fluxo de datos no outro sentido pode continuar. Cando ambos sentidos se apagan libérase a conexión. Nalgunhas implementacións de TCP existe un temporizador de seguir con vida (keepalive timer). Cando unha conexión esta ociosa durante demasiado tempo este contador pode esgotarse. Se isto ocorre, un lado da conexión comproba se o outro aínda responde. Se non se recibe resposta se termina la conexión.

#### 40.6.2.3.3 DETECCIÓN DE ERROS



As técnicas máis efectivas e usadas son as seguintes:

- Detección de erros. Consiste en engadir un ou mais bits de información a cada segmento de forma que indiquen claramente se se alterou algún dos bits do mesmo no camiño dende emisor ó receptor (Pardidad, CheckSum, CRC, ...).
- Confirmacións positivas. O receptor devolve un acuse de recibo positivo por cada un dos segmentos recibidos correctamente. Usase para detectar e solicitar o reenvío de segmentos perdidos. Esta é a técnica que se utiliza no sistema de parada e espera.
- Expiración de intervalos de tempo. O emisor inicia un contador de tempo tras enviar un segmento (o temporizador de retransmisión). Se este contador se esgotase sen que se reciba un ACK positivo o emisor volve a transmitir o mesmo segmento.
- Confirmación negativa e transmisión. O receptor só confirma os segmentos recibidos erroneamente para que o emisor os volva a enviar. Por tanto, utilízase para solicitar o reenvío de segmentos danados.

#### **40.6.2.3.4 CONTROL DE FLUXO**

O control de fluxo máis simple é o que se leva a cabo mediante o sistema de parada e espera. O transmisor garda un rexistro de cada segmento que envía, esperando un ACK antes de enviar o seguinte. Tamén arranca un temporizador cando envía o segmento. Se o temporizador expira antes de recibilo acuse de recibo, retransmite o segmento e reinicia o temporizador. Este mecanismo é o máis barato e o máis usado cando se transmiten tramas moi grandes pero é ineficiente xa que está a canle de transmisión desaproveitado a maior parte do tempo.

O control de fluxo mediante ventá deslizante permite que o transmisor envíe varios segmentos sen esperar os ACK correspondentes. Neste sistema o emisor e o receptor póñense de acordo no número de segmentos sen procesar que pode gardar este último, dependendo do tamaño dos seus buffers. Tamén se poñen de acordo no número de bits a utilizar para



numerar cada segmento. Cando a ventá ten un tamaño cero o emisor non pode enviar máis segmentos, salvo en dous casos excepcionais: cando se trata de datos urxentes e cando o emisor envía un segmento de 1 byte para provocar que o receptor xere un novo acuse de recibo con un novo tamaño de ventá, evitando así un bloqueo indefinido da conexión.

Unha variedade mellorada do sistema de ventá deslizante é o sistema de control de fluxo con adiante-atrás-N, no que cando a estación destino encontra un segmento erróneo devolve un ACK negativo, rexeitando tódolos que lle cheguen ata que non reciba outra vez o segmento incorrecto en boas condicións. O emisor, ó recibir o ACK negativo, sabe que ten que volver a transmitir ese segmento e tódolos seguintes.

Por último, existe outro sistema denominado sistema de control con rexeitamento selectivo, que se basea en que os únicos segmentos que se volven a retransmitir son aqueles rexeitados polo receptor ou aqueles cuxo temporizador expira sen confirmación. Este método é máis eficiente que os anteriores pero precisa que o receptor dispoña dun buffer intermedio de gran capacidade no que gardar tódolos segmentos recibidos tras o rexeitamento dun dado ata recibir de novo o segmento.

#### **40.6.2.3.5 CONTROL DE CONXESTIÓN**

Cando a carga ofrecida á rede é maior que a que pode xestionar prodúcese conxestión. Tódolos algoritmos TCP supoñen que as terminacións de temporización son causadas por conxestións e as revisan en busca de problemas.

Cada transmisor mantén dúas ventás diferentes:

- Ventá negociada co receptor ó establecerse a conexión, cuxo tamaño está baseado no tamaño do buffer de memoria de destino. Isto permite que o transmisor non envíe máis datos dos que o receptor pode almacenar evitando así que o sature.
- Ventá de conxestión, determinada polo tamaño dos datos que se poden enviar sen que se produza timeout.



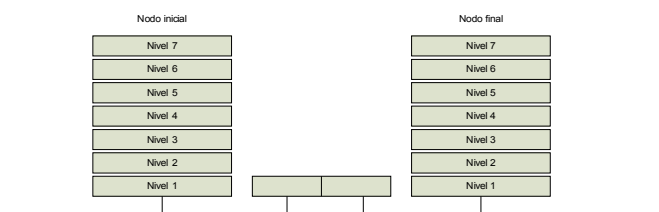
O transmisor só pode mandar un número de segmentos limitado polo tamaño da ventá máis pequena. Ó establecerse unha conexión, o transmisor asigna á ventá de conxestión o tamaño de segmento máximo usado pola conexión. Cada envío recoñecido con éxito duplica a ventá de conxestión. Este algoritmo chamase arranque lento (slow start) e permite que o tamaño da ventá de conxestión creza exponencialmente ata que se produza unha terminación de temporización (timeout) ou se alcance o tamaño da ventá receptora.

Este crecemento exponencial pode producir saturación. Para evitalo, introdúcese outro parámetro, denominado umbral, que toma como valor inicial 64 KBytes. Cando se produce o timeout cámbiase o valor do umbral á metade do tamaño da ventá de conxestión, establecece o valor da ventá ó do tamaño dun segmento máximo e inicialízase outra vez o proceso de arranque lento. Agora, cando o tamaño da ventá chega ó do umbral esta crece soamente en saltos dun segmento máximo, é dicir, con un progreso lineal ata que se produza unha nova terminación de temporización.

## **40.7 ELEMENTOS DE INTERCONEXIÓN DE REDE**

### **40.7.1 REPETIDORES**

O repetidor é un elemento que permite a conexión de dous tramos de rede e que ten como función principal rexenerar o sinal para permitir alcanzar distancias maiores. Tipicamente o repetidor recibe o sinal dende un dos segmentos, amplifícao e emíteo no outro segmento.



As súas características principais son:

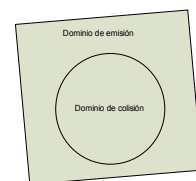
- É a forma mais simple e barata de conectar segmentos de rede.
- Utilízanse para superar limitacións de distancia.
- Só valen para conectar topoloxías de rede compatibles.



- Non illan tráfico nin segmentan a rede.

### 40.7.2 CONCENTRADORES

Un concentrador é un dispositivo que funciona como centro de cableado para unha rede con topoloxía en estrela. A súa función consiste en que o tráfico que chega a calquera dos portos propáguese a través dos demais portos. Isto un medio de rede compartido e reúne ás computadoras conectadas á rede nun único dominio de colisión e de difusión, da mesma forma que se estiveran conectadas a un único cable. Como implicación directa desto último temos que a velocidade de transmisión entre todos os nodos conectados a un concentrador é a mesma que entre dúas máquinas conectadas por un cable.



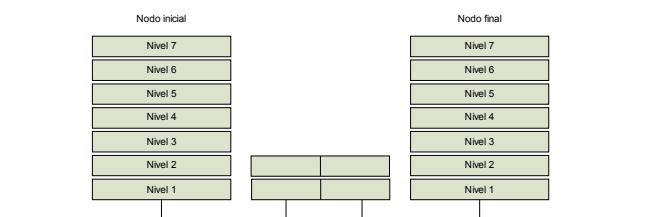
seus  
crea

Os concentradores pódense apilar ou interconectar entre eles, funcionando coma un único concentrador con máis portos, coas mesmas vantaxes e limitacións.

### 40.7.3 PONTES

Unha ponte é un dispositivo utilizado para conectar segmentos de redes. Opera no nivel de enlace de datos e é selectivo respecto aos paquetes que pasan a través del. Fronte ós repetidores que traballan so con sinais, as pontes traballan con tramas.

Unha ponte non transmite datos ós segmentos conectados ata que chega toda a trama. Por este motivo, dous sistemas que se encontran en segmentos separados por unha ponte poden transmitir á vez sen que se produza unha colisión. Unha ponte conecta segmentos de rede de tal forma que mantén no mesmo dominio de difusión pero en distintos dominios de colisión.



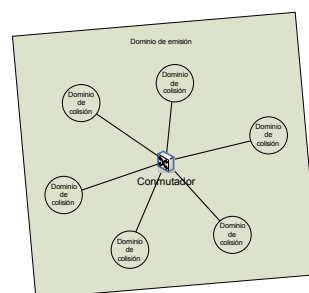
As súas características principais son:



- Poden illar o tráfico baseándose na dirección MAC.
- O igual que os repetidores, as pontes non son direccionables na rede (transparentes para niveis superiores).
- Só operan no nivel MAC de enlace (conectan segmentos da mesma rede).
- Estenden a topoloxía da rede (i.e. Anel-bus)
- Illan erros MAC (i.e. Tramas demasiado longas)
- Existen dous tipos:
  - Transparentes: Conectan topoloxías de rede compatibles (i.e. 10BaseT-10Base2) e non modifican ningunha parte de la trama.
  - De traslación: Conectan diferentes topoloxías de rede (i.e. 10BaseT-Token Ring) adaptando a trama ó protocolo MAC destino.

#### 40.7.4 CONMUTADORES

Un conmutador opera no nivel de enlace de e é en esencia unha ponte multiporto no que un dos portos é un segmento de rede independente. Un conmutador recibe tráfico seus portos e ó contrario que un concentrador, o cal reenvía o tráfico a través tódolos demais portos, só o reenvía polo porto necesario para alcanzar o seu destino.



datos  
cada  
polos  
de

Un sistema conectado a un conmutador posúe o equivalente a unha conexión dedicada con cada un dos sistemas restantes conectados o conmutador e con todo o ancho de banda.

As súas principais características son:

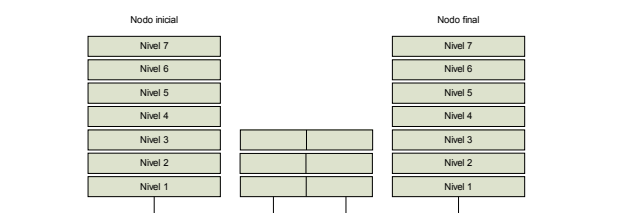
- Diseñados para solucionar problemas de rendemento de LAN (escaseo de ancho de banda, colos de botella na rede).
- Alto rendemento no envío de paquetes e baixa latencia.
- Segmentan un dominio de colisión en outros mais pequenos.



- Reduce ou case elimina a contenda polo acceso ó medio.

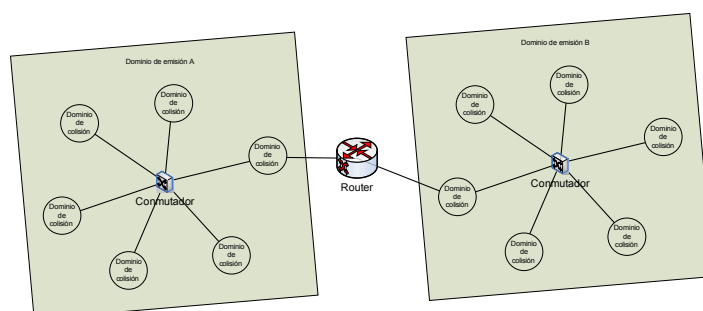
#### 40.7.5 ENCAMIÑADORES

A labor dun encamiñador é a de conectar dúas redes completamente independentes no nivel de rede. Os encamiñadores son máis selectivos que as pontes no tráfico que pasa entre as redes e son capaces de seleccionar de forma intelixente a ruta máis eficiente cara un destino específico.



As funcións básicas dun encamiñador son:

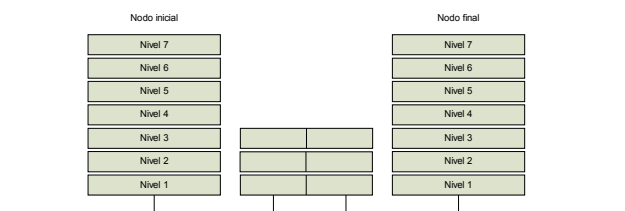
- Segmentala rede en dominios individuais de envío (redes illadas a nivel MAC)
- Proporcionan envío intelixente de paquetes: analizan o tráfico e para cada paquete seleccionan a rede que proporciona a mellor ruta cara o destino. Un paquete pode pasar por varios encamiñadores no seu camiño cara o destino, cada un deles coñecese como salto. O obxectivo soe ser que chegue co menor número de saltos. Para elo utilizan as chamadas táboas de encamiñamento.
- Proporcionan acceso a las WAN de maneira eficiente.
- Soportan camiños redundantes (tolerancia a fallos).
- Proporcionan seguridade / firewall: analizan todo paquete que chega de unha das redes á que está conectado. Se a dirección de orixe e de destino pertencen á mesma rede descártano, se non reenvíanolo ó seu destino a través de outra rede.





#### 40.7.6 PASARELAS

Unha pasarela conecta dúas redes distintas que usan protocolos e arquitecturas distintos a tódolos niveis. A súa función é traducir o protocolo dunha rede no protocolo da outra, pero tamén poden conectar redes que usen o mesmo protocolo. Neste último caso entran ás que traducen IP a IP, por exemplo facendo NAT (Network Address Translation, que converte unha dirección IP dunha rede –normalmente unha IP privada dunha LAN- noutra dirección IP –normalmente nunha IP pública- sendo capaz de inverter o proceso).



### 40.8 XESTIÓN DE REDES

#### 40.8.1 MODELO OSI DE XESTIÓN DE REDE

ISO, seguindo as directrices do grupo OSI, definiu o modelo de xestión de rede como a forma máis importante para entender a funcións principais dos sistemas de xestión de rede.

O modelo OSI de xestión de rede categoriza as funcións en 5 áreas que as veces se denominan modelo FCAPS (Fault, Configuration, Accounting, Performance e Security):

- **Falla (Fault):** Sendo o obxectivo desta área detectar, illar, corrixir e rexistrar as fallas que se produzan na rede.
- **Configuración (Configuration):** Os obxectivos desta área son recoller/fixar/facer seguimento da configuración dos dispositivos. A xestión da configuración ocupase de monitorizala información de configuración do sistema e calquera cambio que se produzan a mesma. A importancia deste área ven dada por que moitas incidencias na rede son o resultado de cambio sen arquivos de configuración, actualización



de versións, etc. Unha adecuada xestión da configuración obriga a rexistrar tódolos cambios na configuración software e hardware.

- **Contabilidade (Accounting):** Sendo o obxectivo principal recoller estatísticas. A xestión da contabilidade preocupase por manter a información referida á utilización da rede, de forma que se poda facturar a usuarios individuais, departamentos, etc.
- **Rendemento (Performance):** O obxectivo deste área é dobre: por un lado preparala rede par ao futuro e por outro medila eficiencia actual da mesma asegurándose que está dentro dos niveis aceptables. A xestión do rendemento preocupase de recoller regularmente a información de rendemento da rede como son os tempos de resposta, rátios de perda de paquetes, utilización de enlaces de datos, etc.
- **Seguridade (Security):** O obxectivo da xestión da seguridade é controlalo acceso ós recursos da rede. Este área non so se preocupa de que a rede sexa seguras e non de recadar e analizar a información referida á seguridade. Ás funcións típicas dentro deste área son a autenticación, autorización, auditoría, de forma que os usuarios (tanto internos como externos) teña no acceso adecuado ós recursos da rede.

#### **40.8.2 SNMP**

SNMP (Simple Network Management Protocol) é o protocolo definido polos comités técnicos de Internet para ser utilizado coma ferramenta de administración dos distintos dispositivos en calquera rede. O funcionamento de SNMP é sinxelo, como o seu propio nome indica, aínda que a súa implementación pode chegar a ser tremendamente complexa. SNMP utiliza a capa de transporte de TCP/IP mediante o envío de datagramas UDP (os axentes escoitan no porto 161 e as estacións xestoras no 162). Sen embargo, o feito de usar UDP fai que o protocolo non sexa fiable (en UDP non se garante a recepción dos paquetes enviados, como en TCP).

O protocolo SNMP está definido nun gran número de RFCs (Request For Comments), entre eles o RFC 1157, 1215 (que definen a versión 1), do

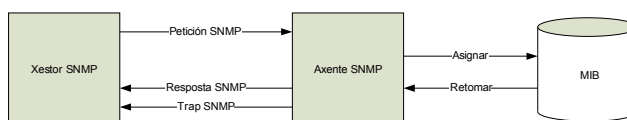


1441 ao 1452 (que definen a versión 2), do 2271 ao 2275 e do 2570 ao 2575 (para SNMP v3).

#### **40.8.2.1 FUNCIONAMENTO DE SNMP**

Cada axente (pódese ver a un axente coma unha máquina na que queremos monitorizar algún dos seus estados) ofrece unha determinada serie de variables, que poden ser lidas ou modificadas. Ademais, un axente pode enviar “alarmas” (Traps) a outros axentes para avisar de eventos que teñen lugar. O normal é que o axente encargado de recibir os eventos se denomine “xestor” (podemos ver a este como á máquina que monitoriza o estado de toda a rede). De forma moi resumida podemos ver as capacidades expostas:

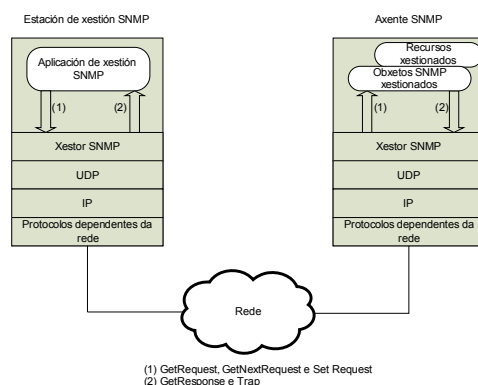
- GET : A estación xestora extrae (lee) o valor dun obxecto do axente
- SET : A estación xestora fixa (escribe) o valor dun obxecto do axente
- TRAP : Permite a un axente notificar á estación xestora eventos significativos



As variables ofrecidas para a consulta nos axentes SNMP defínense a través dunha MIB (Management Information Base, Base de Información de Xestión). A MIB é unha forma de determinala información que ofrece un dispositivo SNMP e a forma en que se representa. A versión da MIB actual é MIB-II e está definida no RFC 1213, aínda que hai múltiples extensións definidas noutros RFCs. A MIB está descrita en ASN.1 para facilitalo seu transporte transparente pola capa de rede.

Cada axente SNMP ofrece información dentro dunha MIB, tanto da estándar (definida nos distintos RFCs) como de aquelas extensións que desexe prover cada un dos fabricantes.





ASN.1 (Abstract Syntax Notation One) é un estándar de notación que describe a representación, a transmisión, a codificación e decodificación de estruturas de datos. Prové un conxunto de regras formais para describirla estrutura de obxectos que son independentes das técnicas de codificación dunha máquina, aportando unha notación formal que elimina ás ambigüidades.

#### **40.8.2.2 ESPECIFICACIÓNS TÉCNICAS SNMP MÍNIMAS REQUIRIDAS**

Existen diversas RFCs que definen SNMP. Por elo é importante establecer uns requisitos ou especificacións mínimas. Estas especificacións mínimas son:

- Versión do protocolo SNMPv2c (Community-based SNMPv2 - RFC 1901)  
 Utiliza o mesmo modelo que a primeira versión do protocolo SNMP, e como tal non inclúe mecanismos de seguridade. As únicas melloras introducidas nesta versión consisten nunha maior flexibilidade dos mecanismos de control de acceso, xa que se permite a definición de políticas de acceso consistentes en asociar un nome de comunidade con un perfil de comunidade formado por unha vista MIB e uns dereitos de acceso a dita vista (so lectura ou lectura e escritura).
- A MIB deberá ser compatible co formato ASN.1. As implementacións doutros estándares da MIB son opcionais. ASN.1 está deseñado para definir información estruturada (mensaxes) de tal forma que sexa independente da máquina utilizada. Para facer isto ASN.1 define tipos de datos básicos, como enteiros e cadeas de texto, e permite construír novos tipos de datos a partir dos xa definidos. Tamén utiliza palabras



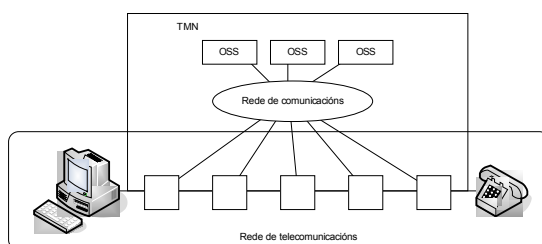
especiais (keywords) para definir os seus procedementos, definir novos tipos, asignar valores, definir macros e módulos.

- O acceso múltiple deberá ser permitido, existindo 3 niveis de acceso cos seus correspondentes “login” e “password”.
- O requirimento mínimo respecto á seguridade, é a xeración de “Traps” no caso dunha autenticación fallida. A información relevante do infractor que deberá ser enviada no TRAP, será a dirección IP.

### **40.8.3 TMN**

TMN (Telecommunications Management Network) define un marco de traballo para alcanzala interoperabilidade e comunicación entre redes de comunicacións e sistemas operativos heteroxéneos. TMN foi desenvolto por ITU coma unha infraestrutura para soportar a xestión e despregue de servicios dinámicos de telecomunicacións.

TMN provee un framework flexible, escalable, confiable, barato e fácil de mellorar. TMN permite crear redes mas capaces e eficientes definindo unha forma estándar de realizalas tarefas de xestión da rede e comunicacións. O procesamento en TMN pode ser distribuído para melloralas escalabilidade. Unha rede de telecomunicacións está composta de elementos de conmutación, circuitos, terminais, etc. En terminoloxía TMN todos estes elementos son Elementos da Rede (NE Network Elements). TMN permite a comunicación entre os NEs e OSS (Operations Support Systems).



TMN usa principios de orientación a obxectos e interfaces estándar para definir as comunicacións entre diferentes entidades na rede. O interface de xestión estándar chámase Q3. TMN basease en estándares OSI como CMIP (Common Management Information Protocol), GDMO (Guideline for definition of management objects), ASN.1 e o modelo de referencia OSI.



As funcións de xestión realízanse a través de operacións compostas de primitivas CMIS (Common Management Information Service).

A información de xestión da rede, así como as regras polas que a información se presenta e xestiona, chámanse MIB (Management Information Database). Os procesos que xestionan a informa chámanse entidades que poden ser de dous tipos: xestor ou axente.

TMN describe as redes de telecomunicacións dente distintos puntos de vista:

- Modelo funcional: representado por bloques que aportan unha visión xeral das funcións e características de TMN.
  - OS (Operation System): realiza funcións de operación do sistema incluíndo monitorización e control das funcións de xestión de telecomunicacións.
  - MD (Mediation Device): Realiza funcións de mediación entre os interfaces locais TMN e o modelo de información dos OS.
  - QA (Q-Adapters): Permite a TMN xestionar NEs que non teñen interfaces TMN.
  - NE (Network Entity): Contén información xestionable que é monitorizada e controlada polo OS.
  - WS (Workstations): Traducen a información entre formato TMN e un formato comprensible polo usuario.
  - DCN (Data Communication Network): Representa a rede de comunicacións cubrindo os niveis 1 a 3 de OSI.
- Conxunto de interfaces:
  - Q: Os interfaces Q existen entre dous bloques funcionais TMN que pertencen o mesmo dominio.
    - Qx: existe entre os NE e os MD, QA e MD e entre os MD e outro MD.
    - Q3: é a interface do OS e existe entre os NEs e OS, QA e OS, MD e OS e entre OS e outro OS.
  - F: As interfaces F existen entre os WS e OS e entre os WS e MD.



- X: Estas interfaces existen entre dous OS TMN de diferentes dominios ou entre un OS TMN e outro OS nunha rede non TMN.
- Modelo lóxico ou de negocio: Este modelo está baseado en capas de distintos niveis xerárquicos:
  - BML (Business Management Layer): Planificación de alto nivel, presupostos, BLAs (Busines Level Agreements), etc.
  - SML (Service Management Layer): Usa a información presentada pola capa NML para xestionar os servizos contratados por clientes actuais ou potenciais. Tamén é o punto clave de contacto con provedores de servizo e outras entidades administrativas.
  - NML (Network Management Layer): A NML ten visibilidade de toda a rede baseada na información dos OSs da capa EML. NML permite xestionar os NEs de forma individual ou coma un grupo.
  - EML (Element Management Layer): Xestiona cada elemento da rede contendo OSs, cada un dos cales xestiona certos NEs. Tamén contén a os MDs.
  - NEL (Network Element Layer): Representa a información xestionable por TMN nun NE. OS QA e os NE están localizados nesta 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. TECNOLOXÍAS DE ACCESO: REDES TELEFÓNICAS (RDSI, XDSL), REDES DE TELEFONÍA MÓBIL, CABLE, PLC, REDES RADIO (LMDS, WIMAX), SATÉLITE, LIÑAS PUNTO A PUNTO, METROETHERNET,**



**Tema 41. Tecnoloxías de acceso: redes telefónicas (RDSI, xDSL), redes de telefonía móbil, cable, PLC, redes radio (LMDS, Wimax), satélite, liñas punto a punto, MetroEthernet.**

**41.1 Redes telefónicas (RDSI, xDSL)**

**41.1.1 RDSI**

**41.1.1.1 Vantaxes de RDSI**

**41.1.1.2 RDSI de banda estreita**

**41.1.1.3 RDSI de banda ancha**

**41.1.1.3.1 Servizos 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 móbil**

**41.2.1 GSM**

**41.2.1.1 Arquitectura da rede 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 ou E-GPRS**

**41.2.2.2 HSCSD**

**41.2.3 Sistemas de terceira xeración: 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 Liñas punto a punto

##### 41.7.1 X.25

##### 41.7.2 FrameRelay

##### 41.7.3 MetroEthernet

#### 41.8 MetroEthernet

##### 41.8.1 MAN baseada en Ethernet

##### 41.8.2 MAN baseada en SDH

##### 41.8.3 MAN baseada en MPLS

#### 41.9 Bibliografía

### **41.1 REDES TELEFÓNICAS (RDSI, XDSL)**

Orixinalmente a única rede pública dispoñible era a Rede Telefónica Conmutada (RTC) que estaba composta por elementos analóxicos sendo o seu principal obxectivo o transporte da voz que transmitíase por liñas modulada como unha forma de onda analóxica.

Para poder transmitir datos sobre esta rede necesitábase convertela sinal dixital nunha sinal analóxica na orixe e volver a convertela en dixital no destino. Dado que as liñas de voz pensáronse só para transmitir voz usaban conmutación de circuítos. Ademais debido a que non foron deseñadas para garantir a transmisión sen perda, eran os protocolos de transmisión de datos os que debían garantir a corrección de erros, reconexión, etc.

Posteriormente, para solucionar o problema da perda de calidade do son nas chamadas a larga distancia apareceron as centrais dixitais, menos propensas a fallos, e permitiron controlar máis liñas de usuario e realizalas conexións moito máis rápido. Desta forma, unha comunicación por unha liña telefónica convencional realizase de forma analóxica no bucle de abonado, pero de forma dixital ata chegar á central onde está conectado o abonado destino. A RDSI (Rede Dixital de Servizos Integrados) supón o último avance: a comunicación dixital entre o abonado e a central telefónica.

#### **41.1.1 RDSI**



RDSI é unha rede desenvolta a partir da rede telefónica que proporciona unha conexión dixital extremo a extremo e que soporta unha gran variedade de servizos.

Denomínase “Dixital” porque basease en técnicas dixitais, garantindo a integridade da información e a transmisión da mesma libre de degradacións o perturbacións externas; e é “de Servizos Integrados” porque utiliza a mesma infraestrutura para moitos servizos que tradicionalmente requirían interfaces distintos (télex, voz, conmutación de circuítos, conmutación de paquetes, etc).

As características mais importantes son:

- A súa arquitectura está estratificada en niveis: físico, enlace e rede.
- Proporciona conexións de 64Kbps.
- A sinalización vai por un canal diferente á información propiamente dita. En certas ocasións utilízase a canle de sinalización para enviar información aínda que a unha velocidade máis baixa.
- Soporta unha gran variedade de aplicacións, independentemente de si están baseadas en conmutación de circuítos ou de paquetes.

#### **41.1.1.1 VANTAXES DE RDSI**

Entre as vantaxes que ofrece RDSI pódense destacar:

- Velocidade. Ofrece múltiples canles dixitais que poden operar simultaneamente a través da mesma conexión telefónica entre central e o usuario. Usando un protocolo de agregación de canles pódese alcanzar unha velocidade de datos sen comprimir duns 128 Kbps, no servizo de acceso básico. Este esquema permite unha transferencia de datos a una velocidade moito maior que a liña telefónica. Ademais, o tempo necesario para establecer una comunicación en RDSI é aproximadamente a metade do tempo empregado cunha liña analóxica.
- Conexión de múltiples dispositivos. É posible combinar diferentes fontes de datos dixitais e facer que a información chegue ó destino



correcto. Como a liña é dixital, é fácil controlalo ruído e as interferencias producidas ó combinar os sinais.

- Sinalización. Nunha conexión RDSI a chamada establece-se enviando un paquete de datos especial a través dun canle independente dos canles para datos. Permite establecer a chamada nun par de segundos.
- Servizos. A RDSI non se limita a ofrecer comunicacións de voz. Ofrece outros moitos servizos como transmisión de datos informáticos (servizos portadores), télex, facsímile, videoconferencia (usando, por exemplo, H.320), conexión a Internet e opcións como chamada en espera, identidade da orixe, etc.

En función do ancho de banda distínguense entre RDSI de “banda estreita” que permite velocidades de 64Kbps ou agrupacións desta velocidade ata 1984Kbps e RDSI de banda ancha onde a velocidade mínima á que se traballa é 2Mbps, podendo chegar ata os 100Mbps.

#### **41.1.1.2 RDSI DE BANDA ESTREITA**

A RDSI dispón de tres tipos de canles:

- Canle B. Os canles tipo B transmiten información en modo circuíto ou en modo paquete a 64Kbps e empréganse para transportar calquera tipo de información de usuario, ben sexa voz ou datos.
- Canle D. Utilízase principalmente para enviar información de control, como é o caso dos datos necesarios para establecer unha chamada ou para liberala. Estes canles traballan a 16Kbps ou 64kbps según o tipo de servizo contratado.
- Canles H. Combinando varios canles B obtéñense canles tipo H, que tamén son canles para transportar só información de usuario pero a velocidades moito maiores. Hai varios tipos de canles H:
  - Canles H0, que traballan a 384Kbps (6 canles B).
  - Canles H10, que traballan a 1472Kbps (23 canles B).
  - Canles H11, que traballan a 1536Kbps (24 canles B).
  - Canles H12, que traballan a 1920Kbps (30 canles B).



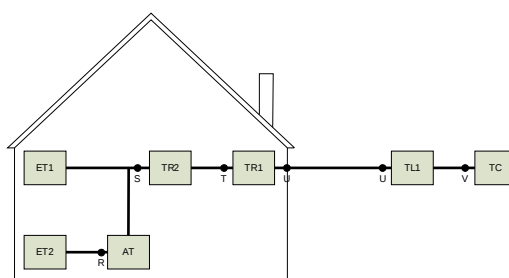
Un usuario pode contactar dous tipos de servizo diferentes co provedor telefónico según as súas necesidades:

- Acceso básico ou BRI (Basic Rate Interface). Proporciona dous canles B e un canle D.
- Acceso primario ou PRI (Primary Rate Interface). En Europa o PRI consta de 30 canles B e un canal D. Neste caso, os canles B tamén poden estar agrupados como 5 canles H0 ou un canle H12.

A RDSI ofrece a capacidade de agregar canles para realizar conexións a maior velocidade.

Nun acceso básico unha chamada a 128Kbps son en realidade dúas chamadas diferentes a 64Kbps cada una, existindo un protocolo por encima que permite ver esa chamada como unha soa. Moitos fabricantes de hardware para RDSI permiten a agregación de canles utilizando protocolos propios. Para garantir a compatibilidade entre equipos de diversos fabricantes é conveniente que o hardware soporte o protocolo MPPP (Multilink Point to Point Protocol).

A configuración de referencia defínese por agrupacións funcionais e puntos de referencia ou interfaces, como se mostra na figura:



As agrupacións funcionais son:

- TC (Terminación de Central). Situada na central de conmutación, encargase do mantemento do Acceso do Usuario.
- TL (Terminación de Liña). Situada na central, encargase dos aspectos de transmisión.
- TR1 (Terminación de Rede nº 1). Dispositivo fronteira que separa as instalacións de usuario das da rede e converte os dous fíos da interface U nos catro fíos empregados nunha interface T ou S/T.



Sempre o proporciona o provedor do servizo. En xeral, realiza funcións do nivel físico.

- TR2 (Terminación de Rede nº 2). Converte a interface T nunha interface S. Fai referencia a unha centraliña ou PABX (Private Automatic Branch Exchange). No acceso básico o TR2 non existe, co que o punto de referencia S e o T coinciden, pasándose a chamar este punto S/T.
- ET1 (Equipo Terminal nº 1). É un terminal específico para RDSI, preparado para a sinalización en modo paquete e xestión de canles de información.
- AT (Adaptador de Terminal). Trátase dun equipo RDSI que ten a capacidade de adaptar interfaces. Converte os sinais doutros equipos non RDSI a sinais adecuadas á interface correspondente, S e/ou T.
- ET2 (Equipo Terminal nº 2). Equipos non RDSI que poden conectarse mediante un AT ó bus RDSI.

Os Puntos de Referencia ou interfaces son:

- V. Representa a separación entre as funcións de conmutación e transmisión na central.
- U. Nun acceso básico está formado pola liña típica de un par trenzado de fíos procedente da rede telefónica. Nun acceso primario está formado por una liña de cable coaxial ou fibra óptica que se soe conectar directamente a unha central local de distribución ou PABX que actúa como TR2.
- T. Representa a separación entre a transmisión de liña e a transmisión no domicilio do cliente. Consta de catro fíos, dous para recibir e dous para enviar datos, permitindo tamén unha conexión full dúplex.
- S. Representa a interface de conexión física dos equipos terminais RDSI e define a estrutura da trama, a xestión do Canle D, a sincronización e as características de transmisión.
- R. Representa unha interface non normalizada en RDSI.



A RDSI estruturase en tres capas: física, de enlace e rede:

- Física. As funcións máis destacadas deste nivel son a codificación de datos dixitais para a transmisión a través da interface correspondente, transmisión full dúplex, formación da trama, activación e desactivación do circuíto físico, etc.
- Enlace. Este nivel emprega principalmente o protocolo LAP-D (Link Access Protocol ou protocolo de acceso ó enlace). Proporciona ó nivel superior un servizo orientado a conexión con transferencia de información confirmada, servizo sen conexión con transferencia de información non confirmada e servizos de administración, que permiten identificar os equipos específicos dentro do bus S/T asociado a unha conexión RDSI.
- Rede. Nesta capa, a Recomendación Q.931 especifica os procedementos para establecer, manter e liberar as conexións no interface usuario-rede. Manéxanse distintos tipos de mensaxes: de establecemento de chamada, durante a fase de transmisión de información, de liberación da chamada e outros.

#### **41.1.1.3 RDSI DE BANDA ANCHA**

A RDSI de banda ancha representa un termo medio entre a conmutación de circuítos pura e a conmutación de paquetes pura. O servizo ofrecido está orientado a conexión, pero internamente implementase con conmutación de paquetes. A RDSI-BA está baseada na tecnoloxía ATM. As razóns que levaron a elixir esta tecnoloxía foron, entre outras, que permite manexar tanto tráfico de velocidade constante como variable e que, a velocidades altas, a conmutación dixital de celdas é mais fácil que as técnicas tradicionais de multiplexación. As velocidades acadables por de RDSI-BA dependen das tecnoloxías concretas usadas na rede e van dende un mínimo de 2 Mbps ata os 155 ou 622Mbps.

En ATM, o fluxo de información organizase en bloques de tamaño fixo e pequeno (53 bytes), chamados celdas. Non se garante a entrega de tódalas



celdas, pero as que chegan fano en orde. O modelo ATM tamén se divide en capas:

- Capa física. Ten que ver co medio físico. Divídese en dúas subcapas: PDM (Physical Medium Dependent) e TC (Transmisión Convergence).
- Capa ATM. Ten que ver coas celdas e o seu transporte.
- Capa de adaptación de ATM (AAL, ATM Adaptation Layer). Permite os usuarios enviar paquetes maiores que unha celda. Divídese en dúas subcapas: SAR (Segmentation And Reassembly) e CS (Convergence Sublayer).

ATM tratarase noutro tema máis en detalle.

#### **41.1.1.3.1 SERVIZOS RDSI-BA**

Na RDSI de Banda Ancha pódense incorporar distintos tipos de servizos que podemos clasificar en servizos interactivos e servizos de distribución.

Dentro dos servizos interactivos podemos encontrar:

- Servizos conversacionais, coma poden ser:
  - o Videoconferencia
  - o Videovigilancia
  - o Fax de alta velocidade
  - o Transferenza de documentos
- Servizos de mesaxería:
  - o Video-mail
  - o Correo con contido multimedia
- Servizos de consulta:
  - o Videotex
  - o Recuperación de datos, documentos, etc.

Exemplos de servizos de distribución poderían ser:

- Servizos sen control de presentación:
  - o Televisión
  - o Televisión a la carta
  - o Distribución de documentos



- o Vídeo baixo demanda
- o ...
- Servizos con control de presentación:
  - o Vídeo

#### **41.1.2 XDSL**

Unha Digital Subscriber Line (DSL) é o nome que identifica a tódolos estándares dixitais sobre bucle de abonado, en exemplo é a RDSI. Pola súa parte xDSL identifica un conxunto de estándares para bucle de abonado sobre fío de cobre como son (entre outras):

- 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**

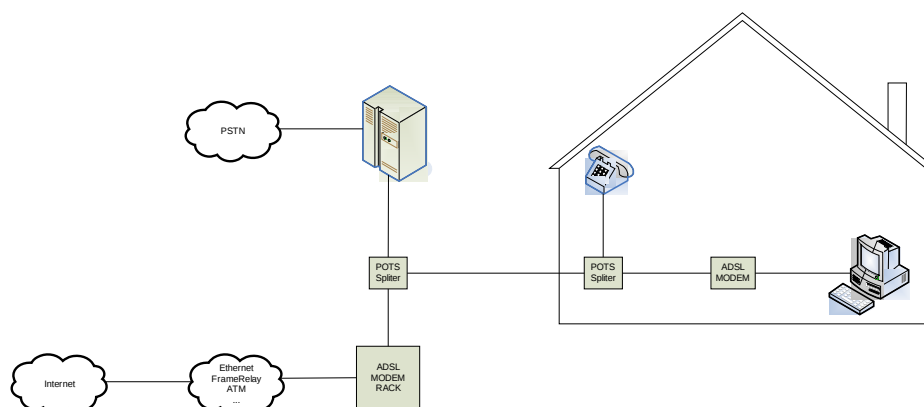
Proporciona servizos dixitais de alta velocidade sobre redes de pares de cobre existentes. Permitindo traballar sen interferir cos tradicionais servizos de voz analóxica (POTS Plain Old Telephone Service).

Utiliza técnicas eficientes de codificación de liña como QAM. E Soporta novos servizos sobre un par trenzado simple, como o acceso a Internet de alta velocidade.

O seu ancho de banda asimétrico (64-640 kbit/s upstream, 500 kbit/s - 8 Mbit/s downstream) faina atractiva para a maioría das aplicacións cliente/servidor como o acceso a Web, acceso a LAN remotas, onde tipicamente o cliente recibe moita mais información do servidor da que xenera.

##### **41.1.2.1.1 ARQUITECTURA ADSL**

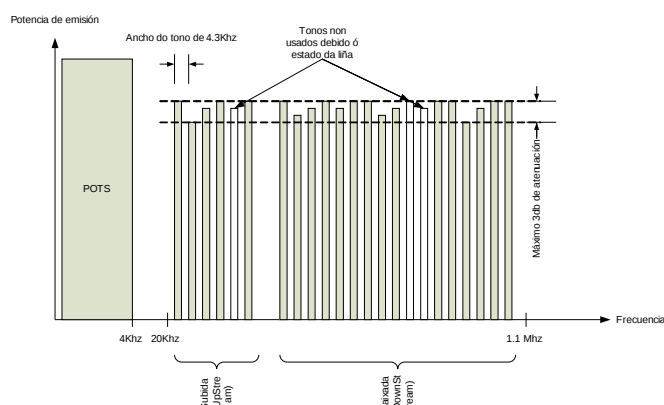




A arquitectura de ADSL fai uso de filtros tanto no domicilio do abonado coma na central para separar o sinal do teléfono e da conexión de datos, estes filtros chámanse splitters. No domicilio do abonado o teléfono conectárase directamente a saída correspondente do splitter mentres que os equipos de transmisión de datos necesitarán un MODEM ADSL. Na central a saída do splitter correspondente conéctase á PSTN (Public Switching Telephone Network ou RTC) mentres que a saída de datos conéctase a un dos modems da central que está conéctado á rede de distribución que a súa vez está conéctada a Internet.

#### 41.1.2.1.2 NIVEL FÍSICO

A canle divídese en tres bandas diferentes, utilízase DMT (Discrete Multi-Tone) que permite utilizar diferentes portadoras (codificadas en QAM) en distintas frecuencias:



DMT utiliza a codificación QAM para conseguir codificar máis bits en frecuencias



onde o sinal presenta menos interferencias. Desta forma modúlanse un número variable de bits en cada unha de esas portadoras, dependendo este número das características do cable de pares, do espectro de frecuencias e das interferencias na sinal. Deste modo, os rateos de velocidade poden ser optimizados facendo posible o uso do mesmo modem sobre bucles locais con diferentes características.

A velocidade de baixada dependen dun bo número de factores, entre eles:

- Lonxitude da liña de cobre.
- Sección do cable.
- Presencia de bobinas de carga, por atenuación en liñas analóxicas.
- Interferencias por paradiafonía.

O alcance depende da velocidade e vai dende 2,7 Km (á máxima velocidade no peor caso) ata 5,5km (á baixa velocidade no mellor caso). Actualmente este estándar evolucionou existindo ADSL2 e ADSL2+ que proporcionan maior velocidade (12 Mb no caso de ADSL2 e 24Mb no caso de ADSL2+) simplemente usando mais espectro de frecuencias.

#### **41.1.2.2 HDSL**

HDSL é simplemente unha forma mellor de transmitir circuítos T1 ou E1 (32 canles de

64 Kbs) sobre liñas de pares de cobre. Necesita un menor ancho de banda para transmitir estas liñas e non necesita utilizar repetidores.

Utilizando avanzadas técnicas de modulación, HDSL transmite 1,544 Mbps ou 2,048 Mbps utilizando rangos de frecuencia entre 80 kHz e 240 kHz, bastante menos cos 1,5 MHz necesarios para as E1/T1 tradicionais.

Sobre un cable con un calibre 24 AWG (0,5 mm) a distancia que se pode alcanzar é de aproximadamente 3,7 Km, aínda que pode chegar os 4,5 Km, sempre sobre dous pares de cobre.

Este tipo de tecnoloxía utilízase para conexións entre PBX, conexións entre estacións de antenas celulares, circuítos dixitais, servidores de Internet e



Redes de Datos Privadas. Pero non está falta de problemas, sendo os mais destacables:

- A existencia de gran cantidade de implementacións propietarias de HDSL pois o estándar só contempla as características básicas.
- Os beneficios técnicos son transparentes ós usuarios, pois eles seguen percibindo unha T1/E1 aínda se poderían alcanzar mellores rendementos en termos de velocidade e de custes.
- Para distancias maiores de 3,7 Km necesítanse repetidores.
- A necesidade de utilizar múltiples pares de fíos, reduce a dispoñibilidade do servizo T1/E1 nun área determinada entre un 50 y 66 por cento. Ademais, moitas veces existen problemas para encontrar 2 ou 3 pares libres dentro da mesma ruta.

Ó igual que con ADSL, HDSL evolucionou tratando de eliminar os problemas da primeira xeración a HDSL2 que simplemente usa menos fíos.

#### **41.1.2.3 SDSL**

SDSL (Symmetrical Digital Subscriber Line) pode referirse a:

- Nun sentido amplo a calquera tecnoloxía DSL simétrica
- Ou a un estándar concreto que proporciona servizos T1/E1 (o caso que nos ocupa)

SDSL permite a transmisión de sinais T1 o E1 sobre un único par de cobre. Isto supón unha gran vantaxe sobre HDSL posto que se poden usar liñas individuais estendéndose o servizo a domicilios particulares.

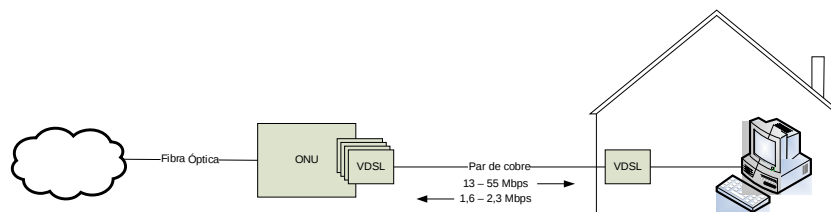
A distancia máxima de SDSL é de 3 Km, distancia á que un ADSL podería operar a 6 Mbps.

#### **41.1.2.4 VDSL**

É unha das máis recentes tecnoloxías da familia xDSL, confía en que o tendido de cobre será curto xa que as operadoras están instalando cada vez máis tramos de F.O. (Fibra Óptica). Xa que as operadoras queren ofrecer novos servizos que requiran a combinación de voz, vídeo e audio, o cal, só é posible con transportes como o ofrecido por ATM. VDSL está preparado para actuar como capa física que dé soporte a redes ATM. Para



logralo, VDSL inclúe unha unidade de rede óptica (ONU) que se encarga de converter e concentrar sinais VDSL sobre unha rede de fibra.



Problemas de VDSL:

- É moi sensible ás interferencias de radio. Ás frecuencias ás que traballa, un bucle de abonado comportase como unha antena receptora para este tipo de sinais.
- VDSL foi deseñada para traballar sobre redes ATM.
- ADSL ten un custe moito menor, pois os filtros poden instalarse na propia central local. Con VDSL necesítanse ONU residenciais que deben ser instalados e mantidos pola operadora.

O igual que coas tecnoloxías xDSL anteriores VDSL ten a súa evolución en VDSL2 capaz de proporcionar 250Mbps de baixada.

## **41.2 REDES DE TELEFONÍA MÓBIL**

### **41.2.1 GSM**

A tecnoloxía GSM pertence ós sistemas de segunda xeración, que ó igual que outros como CDMA, TDMA, NADC ou PDC caracterízanse porque son dixitais. GSM implantouse en Europa e en outros países do resto do mundo, mentres que TDMA e CDMA implantouse en EEUU, e PDC en Xapón.

GSM utiliza multiplexación por división do tempo (TDM), o que posibilita que en cada frecuencia se podan transmitir varias conversas; o tempo de transmisión divídese en pequenos intervalos de tempo, cada un dos cales pode ser utilizado por unha transmisión distinta. Ademais, unha mesma conversa levase a cabo en intervalos de distintas frecuencias, co que non se pode asociar unha chamada a unha frecuencia. Isto ten a vantaxe de que se unha das frecuencias vese afectada por unha interferencia, unha conversa que utilice esta frecuencia só observará problemas nos intervalos pertencentes a dita frecuencia. Isto denomínase TDMA.



O sistema GSM ten asignadas dúas bandas de frecuencias, 900 Mhz (chamado GSM-900) e 1800 Mhz (chamado DCS-1800). En cada unha das bandas, as frecuencias mais baixas úsanse para o enlace ascendente e as mais altas para o descendente. Así, na banda dos 900 Mhz, as frecuencias comprendidas entre los 890 e 915 Mhz comprenden 125 portadoras, cada unha delas con un ancho de banda de 200 Khz para transmisións mobil-estación base. Da mesma maneira, a banda comprendida entre os 935 e os 960 Mhz subdivídese en outras 125 portadoras de 200 Khz para transmisións estación base-mobil.

Ademais, existe tamén unha terceira banda de frecuencia nos 1900 Mhz, nos que GSM opera nalgúns países como EEUU.

Cada portadora subdivide en 8 canles lóxicos ou slots, cada un deles ten o seu uso particular: sinalización, control da comunicación ou tramas de información.

En GSM, a sinalización realízase pola canle común, segundo o protocolo SS7.

GSM é un sistema baseado en conmutación de circuítos e por tanto é un servizo orientado a conexión, é dicir, en toda comunicación haberá 3 etapas diferenciadas: establecemento, comunicación e liberación. En GSM, o que se tarifica precisamente é o establecemento dun canle.

A taxa de transmisión que se alcanza con GSM é de 9'6 Kbps.

#### **41.2.1.1 ARQUITECTURA DA REDE GSM**

A arquitectura GSM basease en tres subsistemas diferenciados:

- Subsistema estación base (BSS): Agrupa as máquinas específicas aos aspectos de radio e celulares do GSM. O BSS está en contacto directo cás estacións móbiles a través do interface radio. O BSS inclúe dous tipos de elementos: a Estación de Base (BTS, Base Transceiver Station) e o Controlador de Estaciones de Base (BSC, Base Station Controller).
- Subsistema conmutación (NSS): Inclúe as funcións básicas de conmutación do GSM, así como as bases de datos necesarias para os



datos de usuario e a xestión da mobilidade. A función principal do NSS é xestionar as comunicacións entre os usuarios GSM e os usuarios doutras redes de telecomunicación. Dentro do NSS, a función básica de conmutación realizase na MSC (Mobile services Switching Centre), cuxa misión principal é coordinar o establecemento de chamadas dende e ata usuarios GSM.

- Subsistema operación e mantemento (OMS): Controla e monitoriza o estado xeral da rede. Componse dos seguintes elementos:
  - o HLR: Home Location Registry. Existe un so HLR por compañía. Nesta base de datos gárdase a información estática do abonado (servizos contratados, etc) e dinámica (ónde se localiza o móbil nun determinado momento)
  - o EIR: base de datos utilizada para comprobar a pertenza do dispositivo móbil á rede do operador, mediante o chequeo do número IMEI, que é un código que identifica univocamente o móbil, unha especie de número de serie do aparato.
  - o AUC: centro de autenticación de usuarios
  - o OMC: Centro de xestión da rede (mantemento)

#### **41.2.2 GPRS, HSCSD**

GPRS y HSCSD pertencen aos denominados sistemas da xeración 2'5, que farán de ponte entre os de segunda e terceira xeración (UMTS, que se verá mais adiante).

##### **41.2.2.1 GPRS**

GPRS significa "General Packet Radio Service". Como o seu nome indica, tratase dun servizo portador baseado na conmutación de paquetes, que se realiza utilizando a rede GSM actual, aínda que necesita terminais que a soporten.

As principais características de GPRS son as seguintes:

- Velocidade: a velocidade máxima teórica é de 171'2 Kbps, aínda que se fala dunha velocidade de conexión máxima na práctica de 115 Kbps.



- Inmediatez: GPRS facilita as conexións instantáneas tan pronto como se necesita enviar ou recibir información; pódese dicir que con GPRS estamos sempre conectados.
- Novas e mellores aplicacións: grazas á maior velocidade de GPRS.
- Tarificación: GPRS tarifícase por volume de datos intercambiado, calidade de servizo e tipo de servizo; en GSM polo contrario tarifícase por duración da chamada.

#### **41.2.2.1.1 ARQUITECTURA DE GPRS**

Coma dixemos anteriormente, o sistema GPRS despregase sobre a rede GSM existente, aínda que require dalgúns elementos novos coma:

- O nodo GGSN (Gateway GPRS Support Node): este nodo é a interface coas redes de datos externas, como X.25 e as redes IP
- O nodo SGSN (Serving GPRS Support Node): nodo de conmutación de paquetes, ó mesmo nivel que as centrais convencionais de GSM (as MSC)
- Estructura principal ou rede troncal GPRS (backbone)

#### **41.2.2.1.2 EDGE OU E-GPRS**

EDGE responde ás siglas de Evolved (ou Enhanced) Data rates for GSM Evolution. Non é en sí una nova arquitectura, se non unha mellora da modulación do canle, é dicir, é unha versión mellorada de GPRS.

Con estas técnicas conséguense ata 384 Kbps, combinando ata 8 slots.

#### **41.2.2.2 HSCSD**

HSCSD (High Speed Circuit Switched Data) é unha especificación homologada polo ETSI (European Telecommunication Standard Institute), que supón unha evolución de GSM, xa que aproveita dita infraestrutura.

Trátase dun servizo multi-slot para a transmisión de datos a alta velocidade mediante circuitos conmutados.

O HSCSD aporta un esquema de codificación mellorado que permite 14'4 Kbps, fronte ós 9'6 Kbps de GSM, o cal posibilita velocidades de transmisión de datos de ata 57'6 Kbps combinando ata 4 canles GSM.



HSCSD foi desenvolvido en paralelo con GPRS pero son servizos de alta velocidade totalmente diferentes. Como o seu propio nome indica HSCSD utiliza a conmutación de circuitos a diferenza de GPRS que utiliza a conmutación de paquetes.

A vantaxe de HSCDS sobre GPRS é a calidade de servizo garantida, proporcionada pola canle de comunicación dedicado. Isto, sen embargo, faina menos eficiente para a transmisión de datos pois a conexión ten que manterse incluso nos momentos nos que non existe transmisión de datos. GPRS fai un uso máis eficiente do ancho de banda e permite facer a tarificación en función da cantidade de contido que se recibe e non só en función do tempo de conexión.

#### **41.2.3 SISTEMAS DE TERCEIRA XERACIÓN: UMTS**

UMTS é o terceiro escalón na historia da telefonía móbil, despois da analóxica e a dixital. UMTS son as siglas de Universal Mobile Telecommunication System ou Sistema Universal de Comunicaciones Móviles. UMTS é un membro da familia global IMT-2000 do sistema de comunicacións móbiles de “terceira xeración” do UIT (Unión Internacional de Telecomunicacións). Este sistema é revolucionario xa que por primeira vez trátase dun estándar universal.

O funcionamento tamén é novedoso, xa que o usuario paga segundo a cantidade de información que se descargue da rede, e non xa polo tempo de uso do servizo. Desta forma poderemos estar constantemente conectados á rede, o que permite, por exemplo, acceder ó correo electrónico de forma instantánea.

Esta tecnoloxía permite que os teléfonos transmitan e reciban datos con una velocidade 200 veces superior á de GSM. A máxima velocidade do UMTS é 2 Mbits. Este tope pode alcanzarse soamente se a rede está ó máximo nivel, o usuario está parado e sen móbiles o seu arredor.

UMTS tamén plantexa importantes innovacións con respecto á arquitectura de rede. UMTS R'99 definiu unha arquitectura que dá cabida a redes de acceso GSM e a rede de acceso UMTS (UTRAN), e propón unha rede central



(CN, Core Network) deseñada como unha evolución da rede GSM/GPRS para facilitar a migración de redes GSM/GPRS a UMTS.

UMTS ofrece un novo interface radio denominado UTRA (UMTS Terrestrial Radio Access). Dito interface está baseado en tecnoloxía CDMA (Code Division Multiple Access) permitindo aumentar considerablemente a velocidade de transferencia de datos, e soporta dous modos de operación el FDD (Frequency Division Duplex) e o TDD (Time Division Duplex). FDD está baseada nun esquema de Secuencia Directa CDMA e soporta unha velocidade de ata 384 Kbit/s. O TDD está baseado na multiplexación en tempo e en código, deseñouse e optimizouse para ser usado en zonas con alta densidade de tráfico, e soporta unha velocidade de ata 2 Mbit/s.

Entre as cousas que nos ofrece UMTS, destacamos a súa facilidade de uso e baixos custes, novos e mellores servizos, acceso rápido, transmisión de paquetes de datos e velocidade de transferencia de datos a pedido, entorno de servizos amigable e consistente, mobilidade e cobertura e servizos UMTS dispoñibles globalmente por satélite.

A súa velocidade sumada ó soporte inherente do Protocolo de Internet (IP), combínanse para prestar servizos multimedia interactivos e novas aplicacións de banda ancha, tales como servizos de vídeo telefonía e videoconferencia.

#### **41.2.3.1 HSPA**

High-Speed Downlink Packet Access (HSDPA) é un protocolo mellorado da terceira xeración que pertence á familia High-Speed Packet Access (HSPA) tamén coñecida como 3.5G, 3G+ ou turbo 3G. HSDPA permite ás redes UMTS ter un ancho de banda de descarga maior que actualmente pode ser de 1,8, 3,6, 7,2 e 14,4 Mbps. Xa está dispoñible tamén HSPA+ que entrega ata 84Mbps gracias ó uso de varias antenas (MIMO Multiple Input Multiple Output).

High-Speed Uplink Packet Access (HSUPA) é outro dos protocolos HSPA que permite unha velocidade de subida de 5,76Mbps. O nome HSUPA foia cuñado por Nokia, o nome oficial é Enhanced Uplink (EUL).

#### **41.3 CABLE**



As redes de cable actuais presenta as seguintes características: servizos integrados, alta capacidade, e redundancia. Os servizos que ofrece unha rede de cable moderna inclúen os seguintes: amplía oferta de canles de TV (terrestres, vía satélite, e de produción propia), vídeo a la carta, PPV, datos e Internet (mediante módem cable), telefonía (básica e RDSI, con opción de acceso a Internet), aluguer de liñas e fibras.

A transmisión do sinal ata o abonado levase a cabo mediante o canle denominado descendente ou directo (de 86 a 862 MHz), mentres que as que parten do abonado realízanse a través do canle ascendente ou de retorno (de 5 a 65 MHz).

A topoloxía dunha rede de cable baseada en tecnoloxía HFC (Hybrid fibre-coaxial):

- Rede troncal primaria: a nivel físico aneis redundantes de fibra óptica, a nivel lóxico topoloxía de estrela. Estes aneis comunican a cabeceira cos nodos primarios. O respaldo é activo.
- Rede secundaria ou de distribución: conecta un nodo primario con varios nodos secundarios a través de aneis con arquitectura en estrela formando lóbulos que abarcan 12.000 fogares. Cada lóbulo interconecta 5 ou 6 nodos secundarios. Hai redundancia en ruta e equipos. O servizo de telefonía a veces non se proporciona mediante a rede HFC (é dicir, no presenta telefonía integrada), se non que fai uso dunha rede paralela de tipo SDH (Synchronous Digital Hierarchy), falándose entón de telefonía superposta.
- Rede terciaria ou de dispersión: conecta cada nodo secundario con cada un dos catro nodos ópticos terminais que dependen del. Cada nodo óptico terminal cubre un área de 500 fogares. A rede de dispersión presenta una disposición en estrela sen redundancia en ruta. No nodo secundario realízase a interconexión das fibras provintes do nodo primario coas fibras que van ata os nodos terminais.



- Rede de distribución de coaxial: distribúe as sinais desde o nodo óptico terminal ata cada punto de derivación nos edificios aos que da servizo. A distribución realízase con estrutura en árbore, de forma que cada nodo óptico terminal da lugar a 4 ramas duns 125 fogares aproximadamente. Os nodos ópticos terminais ubícanse fisicamente en armarios de exterior. O nodo óptico terminal realiza a conversión óptico-eléctrica dos sinais transportadas en sentido descendente. De el vai aos amplificadores que atacan as catro ramas de coaxial que parten do nodo óptico. Cada rama de coaxial alimenta (se é necesario, mediante amplificadores) a unha rede de derivadores, cuxas saídas están conectadas ás acometidas individuais de abonado. Para o camiño de retorno utilízase a mesma infraestrutura de rede, equipando adecuadamente aos amplificadores.
- Rede de acometida de abonado: conecta a rede de distribución de coaxial co punto de terminación de rede. Existen dúas arquitecturas:
  - o Estrela: un mesmo derivador da servizo a tódalas vivendas das diferentes plantas dun edificio.
  - o Árbore: colócase un derivador en cada planta, do que parten os coaxiais que dan servizo aos abonados desa planta.A rede de acometida de abonado pódese dividir en dúas partes: cableado de edificio ou verticais, e cableado de vivenda.

As redes HFC teñen os seguintes puntos singulares:

- Cabeceira: está equipada para a prestación do servizo de difusión de televisión. Pódese descompoñer en catro bloques:
  - o Sistemas de recepción e transmisión analóxica: composto por antenas de recepción, equipos de recepción, equipos para banda base, etapa de codificación, e etapa de modulación e saída.
  - o Sistemas de recepción e transmisión analóxica de reserva: antenas de recepción, equipos de recepción, e etapa de modulación.
  - o Sistemas de monitorización.



- o Sistemas de transmisión óptica.
- Nodo primario: recibe o sinal da rede troncal primaria provinte da cabeceira de rede. O nodo primario presenta dous módulos independentes:
  - o O módulo do camiño descendente.
  - o O módulo do camiño ascendente.
- Nodo secundario: encamiñan os sinais procedentes do nodo primario (mediante a rede troncal secundaria) ata os nodos ópticos terminais (a través da rede terciaria). Ubicanse fisicamente nunha arqueta, habitualmente xunto a un dos seus nodos ópticos terminais.
- Nodo óptico terminal: dan servizo a áreas de aproximadamente 500 fogares. Ubicanse en armarios de exterior. Pódense descompor en dous grandes bloques:
  - o Canle descendente.
  - o Canle ascendente.
- Terminal direccionable de abonado: permite ó cliente acceder aos servizos TV da rede e é instalado no propio domicilio do abonado. Descodifica os canles correspondentes ó servizo contratado polo abonado e permite o cliente interactuar co sistema.
- Subredes de telefonía e datos:
  - o Subred de datos:
    - Servizos ofrecidos: portadores (aluguer de circuítos dixitais), de transmisión de datos (baséanse en conmutación de circuítos e conmutación de paquetes ou celdas), de acceso a redes (acceso a Internet e a outros provedores de contido multimedia ), e de valor engadido.
    - Estructura: os equipos que conectan a rede de datos con Internet están situados na cabeceira. O nodo primario que reside xunto a esta é o encargado do funcionamento dos módems cable, a través dos cales o abonado ten acceso á rede. Os seus elementos son:



- Router: encamiña o tráfico IP entre a rede de datos e Internet.
  - Servidor Proxy: actúa a modo de caché.
  - Firewall (cortafuegos): protexe a rede de datos de ataques externos.
  - Servidores: encárganse de dar diversos servizos: WWW, FTP, IRC, e-mail, DNS, etc.
  - Conmutador ATM multiservizo: permite a interconexión de equipos de diferentes tecnoloxías.
  - Conmutador LAN: conecta os servidores co conmutador ATM multiservizo.
  - Conmutador ATM de acceso: como o conmutador ATM multiservizo , pero de menor capacidade.
  - Cabeceira de modems cable: a cabeceira de modems e os modems cable, compoñen a rede de acceso a datos integrada en HFC.
  - Modems cable: sitúase no domicilio do abonado, e permite acceso á rede de datos mediante HFC.
- o Rede de telefonía:
- Servizos ofrecidos: telefonía analóxica tradicional, acceso dixital RDSI básico, acceso dixital RDSI primario.
  - Estructura: A rede soporta tanto telefonía integrada como superposta.
    - Centro de conmutación: canaliza todo o tráfico de chamadas.
    - Rede de acceso mediante telefonía integrada: aproveita a rede HFC de distribución de TV e datos para chegar ata o cliente. Para realizar o interface coa rede HFC, son necesarios dous equipos específicos (HDT y MDU).
    - Rede de acceso mediante telefonía superposta: non usa a rede HFC. Dende a cabeceira distribúese o sinal, mediante



fibra óptica ata os nodos primarios, e dende eles, ata os lóbulos de 12.000 fogares da rede SDH.

- Cableado de vivendas.

#### **41.4 PLC**

A tecnoloxía Power Line Communications, "PLC", posibilita a transmisión de voz e datos a través dos cables eléctricos, convertendo calquera enchufe da casa en conexión potencial a tódolos servizos de telecomunicacións. O cliente só necesitará conectar un pequeno módem para acceder a Internet, telefonía e datos ó mesmo tempo e a alta velocidade (banda ancha).

A rede eléctrica transporta electricidade a unha frecuencia de 50 Hz. En PLC engádense frecuencias na banda que vai dende 1,6MHz ata 30MHz para o transporte dos datos. Uns filtros instalados no transformador de baixa tensión separan as frecuencias altas de datos, da frecuencia de 50Hz da electricidade.

Power Line Communications emprega unha rede coñecida como High Frequency Conditioned Power Network (HFPCN) para transmitir simultaneamente enerxía e información. Unha serie de unidades acondicionadoras son as que se encargan do filtrado e separación de ambos sinais.

Na actualidade non existen estándares tecnolóxicos para o PLC de acceso. Este é un dos principais problemas desta tecnoloxía, ó non permitila interoperabilidade entre os equipos suministrados polos distintos fabricantes. Tampouco existía unha regulación en canto á utilización de frecuencias, ata o 2005. Garantindo agora a coexistencia de sistemas domésticos (como HomePlug) e as tecnoloxías de acceso.

É posible que o prezo da tecnoloxía PLC sexa bastante inferior ó dos actuais ADSL e Cable no mesmo rango de velocidades o que a converte nunha tecnoloxía interesante.

Os servizos típicos de telecomunicacións que poderían ser proporcionados son:

- Telefonía



- Acceso rápido a Internet
- Vídeo baixo demanda

#### Vantaxes de PLC:

- Como a PLC posicionouse coma un servizo de tipo IP utilizará routers de paquetes en vez dos de conmutación de circuítos típicos, dos subministradores de telecomunicacións tradicionais, mantendo así os custes dos equipos de IT baixos.
- As compañías eléctricas poderían pois comercializar un servizo básico de conexión a Internet con unha suscripción mensual de tarifa plana.
- Esta tecnoloxía ponse virtualmente ó alcance de calquera,
- Xa existen varias tecnoloxías que transforman os cables eléctricos existentes nun cableado LAN (Local Area Network)
- PLC podería facilitar ás compañías eléctricas a oportunidade de ofrecer servizos de valor engadido.

#### Inconvenientes de PLC:

- O número máximo de fogares por transformador. Como os sinais de datos de Power Line non poden sobrevivir o seu paso por un transformador, só se utilizan na última milla. O modelo europeo de rede eléctrica soe colocar un transformador cada 150 fogares aproximadamente.
- Polo que é necesario que tódolos transformadores veñan dotados de servidores de estación base PowerLine.
- Calquera liña conductora é, por definición, unha antena polo que a instalación eléctrica dunha casa actúa como tal, e é moi sensible ás interferencias que se produzan nas frecuencias de transmisión de datos, o redor dos 30 MHz.

### **41.5 REDES RADIO (LMDS, WIMAX),**

#### **41.5.1 LMDS**

LDMS nace no contexto das emerxentes tecnoloxías sen fíos e a crecente interese en IP como unha alternativa para proporcionar servizos multimedia



ó usuario final, xunto co aumento da demanda de novos servizos de telecomunicación orientados a voz y datos (acceso rápido a Internet, etc) LDMS (Local Multipoint Distribution Service) é unha tecnoloxía de acceso sen fíos de banda ancha ou bucle de abonado sen cable. Os sistemas LDMS utilizan ondas radioeléctricas de alta frecuencia para ofrecer servizos multimedia e de difusión a usuarios finais en distancias similares ás alcanzadas coas tecnoloxías de cable.

Entre as vantaxes que ofrece LDMS podemos sinalar:

- Rápido despregue, comparado coas tecnoloxías de cable: LDMS permite instalar redes rapidamente xa que, por exemplo, o emprazamento das antenas é moi sinxelo dado o pequeno tamaño destas.
- Posibilidade de integrar distintos tipos de tráfico (voz, vídeo, datos,...)
- Alta velocidade de acceso a Internet
- Flexibilidade y modularidade

Como outras características de LDMS, podemos sinalar que require LoS (Line of Sight), é dicir, visión directa entre os dous puntos que se comunican. As velocidades de acceso que se alcanzan encóntranse no entorno dos 512 Kbps - 2 Mbps.

O servizo LDMS prestase en dúas bandas:

- Banda S: esta banda traballa nos 3,5 GHz. É a que se utiliza para o despliegue do bucle de abonado. Ten un alcance de o redor de 15 Km e posúe un ancho de banda de 20 Mhz.
- Banda K: traballa nos 26 GHz. É a banda utilizada para o acceso de banda ancha. Dispón dun alcance menor que a banda S (o redor de los 3 Km), pero un maior ancho de banda (uns 56 Mhz).

A comunicación en LDMS establececese mediante radiodifusión punto-multipunto: os sinais viaxan dende ou ata unha estación central, ata ou dende os diferentes puntos de recepción distribuídos na zoa de cobertura. LDMS utiliza modulación QPSK (Cuadratura Phase Shift Keying), que permite reducirlas interferencias e aumentala reutilización do espectro, alcanzando un ancho de banda cercano a 1 Gbps.



#### **41.5.2 WIMAX**

WiMAX (Worldwide Interoperability for Microwave Access) é un protocolo de telecomunicacións que provee acceso a Internet en puntos fixos ou móbiles.

A actual revisión de WiMAX permite ata 40Mbps, coa nova versión (IEEE 802.16m) se esperan velocidades de ata 1 Gbps.

O nome WiMAX foi creado polo WiMAX Forum que foi fundado en xuño de 2001 para fomentar a interoperabilidade do estándar. Este foro describe WiMAX como unha tecnoloxía baseada en estándares permitindo o acceso de banda ancha de última milla como alternativa ó cable e ás xDSL.

O estándar 802.16 Broadband Wireless Access (BWA) define:

- **Capa 1 - Capa física:** A versión orixinal de IEEE 802.16 especifica unha capa física que operan o rango entre os 10 e os 66Ghz. 802.16a (actualización do 2004) engadiu a posibilidade de operar entre 2 e 11 Ghz. En 2005 a versión 802.16e-2005 viu a luz usando SOFDMA (Scalable Orthogonal Frequency-Division Multiple Access) en vez da orixinal OFDM (Orthogonal Frequency División Multiplexing). 802.16e tamén define o uso de varias antenas con MIMO.
- **Capa 2 - Capa de acceso o medio (MAC):** Usa un algoritmo de planificación para o que a estación do abonado necesita competir só unha vez: para conectarse á rede. A ventá de tempo pode alargarse ou contraerse, pero permanece asigna ó abonado. O algoritmo é estable en situacións de sobrecarga e gran número de abonados permitindo á estación base o control da calidade do servizo (QoS).
- **Mobilidade**
- **Características opcionais e obrigatorias do enlace de radio**

#### **41.6 SATÉLITE**

O acceso a Internet por satélite pode obterse en calquera parte do mundo usando satélites LEO (Low Earth Orbit) aportando unha relativamente baixa latencia pero baixa velocidade ou satélites geoestacionarios aportando maior velocidade pero tamén maior latencia e non podendo chegar a certas



partes dos polos. As desvantaxes deste sistema non se quedan ahí (alta latencia) se non que tamén inclúen problemas de cobertura cando chove, ademais require liña directa de visión ó satélite (nun terreo escarpado esto pode ser un problema).

O equipo do cliente para unha comunicación de satélite require a instalación dunha antena parabólica dun tamaño dependente da tecnoloxía concreta, do satélite e do modo en que se use (comunicación uni ou bidireccional, ...) ademais dun MODEM específico.

Existen distintas técnicas usadas para compartir cada portadora (TDMA, SCPC, ...) aportando velocidades de ata 40Mbps de descarga.

Os típicos modos de comunicación son:

- Comunicación bidireccional por satélite. Neste caso tanto a subida como a baixada prodúcese usando ó satélite, requirindo unha moi precisa orientación da antena.
- Comunicación unidirección mais conexión terrestre. Neste caso a antena só recibe o sinal do satélite, e o envío de datos prodúcese por unha liña terrestre (RTC, GSM, GPRS, ...). Ten a vantaxe sobre o modelo anterior de que a antena non necesita estar tan precisamente orientada e que, polo tanto, é mellor para instalación móbiles.
- Comunicación unidireccional / multicast, sen retorno. Dentro desta modalidade temos os servizos de multicast por IP que non requiren retorno (ademais da transmisión de audio e vídeo).

#### **41.7 LIÑAS PUNTO A PUNTO**

Unha liña punto a punto ou liña privada (coma contraposición a unha VPN) é unha liña física entre dúas ubicacións de forma transparente, de forma que unha liña punto a punto dende unha ubicación remota funciona da mesma forma que se estivese conectada na ubicación de destino.

Tradicionalmente cando se contrataba unha liña punto a punto especificábase as características e velocidade da mesma usando múltiplos do DS0, dende un DS0 (64Kbps) ata un T1/E1 (1,5Mbps) ou DS-3 (672



canles de 64Kbps). Os operadores usaban conmutación de circuítos ou circuítos virtuais con tecnoloxías coma X.25.

Na actualidade estas liñas sóense crear coma unha VPN dentro das redes do operador, por exemplo usando MetroEthernet e MPLS.

#### **41.7.1 X.25**

X.25 é un estándar para o acceso a redes públicas de conmutación de paquetes. Non especifica cómo está implementada a rede interiormente aínda que o protocolo interno soe ser parecido a X.25. Implementa un servizo de circuito virtual externo.

O servizo que ofrece é orientado a conexión, fiable, no sentido de que non duplica, nin perde nin desordena, e ofrece multiplexación, isto é, a través dun único interfaz mantéñense abertas distintas comunicacións.

Os elementos usados por X.25 denomínanse:

- DTE (Data Terminal Equipment): É o equipo final de usuario (PC con placa X.25 por exemplo).
- DCE (Data Circuit Terminating Equipment): Podemos interpretalo coma un nodo local. A nivel de enlace (LAPB) as conexións establécense DTE-DCE. Co nivel de rede, ampliamos as comunicacións mais ala do DCE, que fai de interconexión.

X.25 define 3 niveis:

- Nivel Físico:

Existen dúas posibilidades:

- X.21: Utilízase para o acceso a redes de conmutación dixital. (Similares ás de telefonía dixital.)
- X.21bis: Empregase para o acceso a través dun enlace punto a punto. (Similar a RS-232 en modo síncrono.)

En canto as características mecánicas, úsanse conectores Canon de 15 pines ou de 25 pines.

As velocidades van entre os 64kbps e os 2Mbps, velocidades que poden parecer baixas e, de feito, así son. X.25 presenta un problema



de baixa eficiencia pola esaxerada protección contra erros que implementa e que coas redes actuais non ten sentido.

- Nivel de Enlace (LAP-B):

En X.25, este nivel queda implementado co protocolo LAP-B (Link Access Procedure - B) que é un protocolo de enlace con rexeitamento simple e no cal as tramas de información poden ser utilizadas como tramas de control.

- Nivel de Paquete (PLP):

Este nivel está especificado polo PLP (Packet Layer Protocol) que é un protocolo de acceso a nivel de rede e que proporciona servizos ó nivel superior.

Permite establecer circuitos virtuais (CV): Que poderíamos definir como a asociación lóxica entre usuarios para comunicarse entre eles. Existen dous tipos de CV:

- Conmutados (CVC) : Hai que realizar un diálogo previo á transmisión co nodo local para establecelos.
- Permanentes (CVP): Están establecidos de antemán (por contrato), así que non fai falla fase de establecemento. Son moi útiles se se transmite moito e con moita frecuencia cara un mesmo destino.

#### **41.7.2 FRAMERELAY**

É posible usar FrameRelay como tecnoloxía de soporte para conseguir redes punto a punto. Esta tecnoloxía explícase noutro tema.

#### **41.7.3 METROETHERNET**

É posible usar MetroEthernet como tecnoloxía de soporte para conseguir redes punto a punto. Esta tecnoloxía explícase máis adiante neste mesmo tema.

#### **41.8 METROETHERNET**

MetroEthernet é unha rede tipo MAN (está deseñada para cubrir unha área metropolitana) baseada no moi coñecido estándar Ethernet. O uso habitual de MetroEthernet é servir de rede de acceso para conectarse a Internet ou servir de rede de interconexión de varias oficinas dunha compañía.



Tipicamente o proveedor dunha conexión MetroEthernet proporcionará a mesma por fibra óptica terminando nun equipo que habitualmente ten capacidades non só de nivel 2 se non de nivel 3 (rede).

Esta tecnoloxía pode despregarse de varias formas atendendo á base usada para MAN que a soporta:

- Ethernet pura, toda a rede está baseada en Ethernet, sen ningunha outra tecnoloxía de soporte. Esta opción aporta unha gran simplicidade e baixo custe, pero ten graves problemas de fiabilidade e de escalabilidade polo que está limitada a pequenos despregues.
- MetroEthernet baseadas en redes SDH xa existentes, coas restricións que impón SDH no manexo do ancho de banda.
- E, por último, MetroEthernet baseadas en MPLS. Esta é a opción mais cara pero a mais escalable e fiable sendo a típica que a despregar por operadores (de non ter unha rede SDH existente).

#### **41.8.1 MAN BASEADA EN ETHERNET**

Un despregue baseado unicamente en Ethernet fai so uso de switches de nivel 2. Isto permite un deseño e configuración moi simples a un baixo custo.

Este tipo de despregue só foi posible tras a incorporación das VLAN (Virtual LAN) aportando a posibilidade de “punto a punto” e “multipunto a multipunto” combinadas con VLAN Stacking (tamén coñecida como VLAN Tunneling) e VLAN Translation xa que previamente non era posible illar o tráfico de cada usuario (dada a natureza de Ethernet) para formar circuítos. VLAN Stacking permite o uso de varias LANs virtuais sobre o mesmo circuito da rede troncal gracias ó uso de dous identificadores: un para rede troncal e outro para a rede Ethernet (existindo 4096 identificadores distintos segun o estándar 802.1Q, que non 4096 VLAN distintas) . VLAN Translation permite converter un identificador de VLAN noutro de forma que o identificador usado nunha parte da rede sexa distinto ó usado noutra, evitando desta forma posibles conflitos entre identificadores de distintos usuarios.



#### **41.8.2 MAN BASEADA EN SDH**

Unha Ethernet MAN baseada en SDH (Synchronous Digital Hierarchy) é un paso intermedio entre redes tradicionais (baseadas en división de tempos) e redes máis modernas (como Ethernet). Neste modelo a infraestrutura SDH existente é usada para transportar conexións Ethernet de alta velocidade aportando unha gran fiabilidade grazas aos mecanismos intrínsecos das redes SDH (cun tempo de recuperación inferior a 50 ms). Este tipo de implantacións limítanse aos casos onde xa existe unha rede SDH debido ó alto custo dos equipos SDH e as limitacións de SDH para xestionar o tráfico (velocidade, ruta, ...) levando moitas veces á instalación de switches Ethernet na fronteira SDH para aliviar parte destas limitacións

#### **41.8.3 MAN BASEADA EN MPLS**

Neste caso as tramas Ethernet enviadas polo usuario son empacquetadas en MPLS que transmite os seus datos sobre (habitualmente) Ethernet, creando unha pila Ethernet sobre MPLS sobre Ethernet (aínda que podería haber outro protocolo por debaixo).

Este despregue usa LDP (Label Distribution Protocol) punto a punto para a etiqueta interna (etiqueta do VC) e RSVP-TE (Resource reSerVation Protocol-Traffic Engineering) ou LDP para a etiqueta externa usada na rede. Un dos mecanismos de restrición de MetroEthernet baseadas en MPLS é Fast ReRoute (FRR) que permite un tempo de restoración inferior ós 50ms. Isto é unha das cousas que máis hai que ter en conta á hora de decidirmos por MetroEthernet baseadas en MPLS fronte aquelas baseadas en Ethernet, xa que se temos un tempo de restoración equivalente usando unha solución Ethernet pura non merece a pena introducir unha baseada en MPLS.

#### **41.9 BIBLIOGRAFÍA**

José Manuel Huidrobo. Todo sobre comunicacións. 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. TECNOLOXÍAS DE TRANSPORTE: FRAME RELAY, ATM, DWDM, MPLS. REDES DE FIBRA ÓPTICA. REDES DE NOVA XERACIÓN (NGN).**



## **Tema 42. Tecnoloxías de transporte: Frame Relay, ATM, DWDM, MPLS. Redes de fibra óptica. Redes de nova xeración (NGN).**

### **42.1 Tecnoloxías 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 da conxestión**

###### **42.1.1.1.4 Tipos de tráfico transportado**

###### **42.1.1.1.5 Vantaxes**

#### **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 do tráfico**

###### **42.1.2.2.2.2 Clases de servizo**

###### **42.1.2.2.2.3 Asignación de ancho de banda e control da conxestión**

###### **42.1.2.2.3 Capa de adaptación (AAL)**

###### **42.1.2.2.3.1 Estructura da 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 Funcions de MPLS**

##### **42.1.4.2 LSRS e 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 Xerarquía Dixital Plesiócrona (PDH)

#### 42.2.2 Xerarquía Dixital Síncrona (SDH) / SONET

##### 42.2.2.1 Frame SDH / SONET

##### 42.2.2.1.1 Framing

##### 42.2.2.1.2 Estructura do frame STM-1

### 42.3 Redes de nova xeración (NGN)

#### 42.3.1 Tecnoloxías de soporte

### 42.4 Bibliografía

## **42.1 TECNOLOXÍAS DE TRANSPORTE**

### **42.1.1 FRAME RELAY**

Frame relay é un protocolo de transmisión de paquetes de datos en ráfagas de alta velocidade a través dunha rede dixital fragmentados en unidades de transmisión chamadas Frames. Require unha conexión exclusiva durante o período de transmisión.

Frame relay é unha tecnoloxía de paquete-rápido xa que o chequeo de erros non ocorre en ningún nodo da transmisión. Son os extremos os responsables deste chequeo de erros. (Sen embargo debido a que os erros en redes dixitais son extremadamente menos frecuentes en comparación coas redes analóxicas, isto non supón un verdadeiro inconveniente).

A diferenza dos paquetes que son de tamaño fixo, Frame Relay transmite Frames que son de tamaño variable (mil ou mais bytes ).

O estándar de Frame Relay (ITU-T I.122) é unha extensión do estándar ISDN. Implementa varios interfaces físicos como V.35 para velocidades menores de 2 Mb e G.703 para 2 Mb.

Unha conexión Frame Relay é coñecida como unha conexión virtual. Unha conexión virtual permanente é exclusiva ó par orixen-destino e pode transmitir por encima de 1,544 Mbps, dependendo das capacidades do par

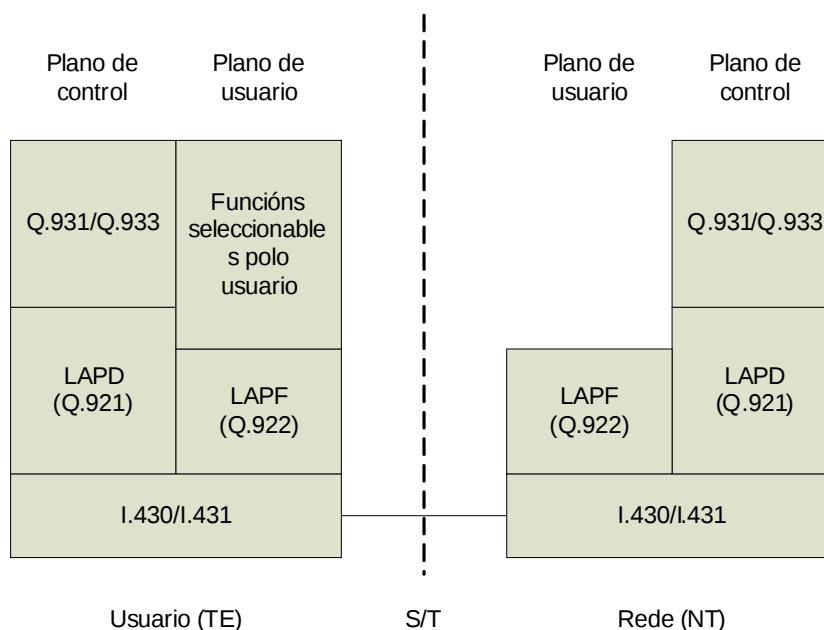


orixen-destino. É posible tamén unha conexión virtual conmutada usando a rede pública e pode proporcionar elevados anchos de banda.

#### 42.1.1.1 ARQUITECTURA DE PROTOCOLOS

En Frame Relay considéranse dous planos de operación:

- Plano de control (C): Involucrado no establecemento e liberación de conexións lóxicas. Os protocolos deste plano implementanse entre o usuario e a rede. É similar á sinalización en canles de servizos de conmutación de circuítos, xa que usa un canal lóxico separado para a información de control. Na capa de enlace usase o protocolo LAPD (Q.921) para proporcionar un servizo de control de datos fiable, mediante un control de fluxo e de erros entre a rede e o usuario. Este servizo de enlace de datos usa para o intercambio de mensaxes de control o protocolo Q.931.
- Plano de usuario (U): Responsable da transferencia de datos extremo a extremo mediante o protocolo LAPF (Procedemento de Acceso ó Enlace para Servizos en Modo Trama) definido no estándar Q.922.



##### 42.1.1.1.1 PROTOCOLO LAPF

Permite a retransmisión de tramas como un servizo orientado a conexión da capa de enlace cás seguintes propiedades:



- Envío secuencial das tramas a partir da información de direccionamento existente na cabeceira de control de cada trama.
- Existe unha probabilidade pequena de perda de tramas.
- Reduce ó mínimo o traballo da rede.

El formato de trama de LAPF de funcionamento mínimo (coñecido como protocolo central LAPF) é similar ó LAPD e LAPB con unha salvedade: non existe campo de control, o que ten as seguintes implicacións:

- Existe un único tipo de trama, usada para transportar datos de usuario. Non existen tramas de control.
- Non é posible o uso de sinalización en banda; unha conexión lóxica só pode transmitir datos de usuario.
- Non é posible levar a cabo control de erros dado que non existen números de secuencia.
- Os campos indicador e secuencia de verificación de trama (FCS), actúan como en LAPD y LAPB. O campo de información contén datos de capas superiores. Si o usuario decide implementar funcións adicionais de control de enlace de datos extremo a extremo, debe incluírse neste campo unha trama de enlace de datos.

#### **42.1.1.1.2 DIRECCIONAMIENTO**

O campo de dirección ten implícitamente unha lonxitude de 2 octetos, e pode ampliarse a 3 ou 4 octetos (indicase mediante os bits de ampliación do campo de dirección (EA)). Este campo contén un identificador de conexión de enlace de datos (DLCI) de 10, 17 ou 24 bits. O DLCI proporciona a mesma función que o número de circuíto virtual en X.25: permite a multiplexación de varias conexións lóxicas de retransmisión de tramas a través dun único enlace físico. Como en X.25, o identificador de conexión só ten significado local.

A función realizada por calquera rede que soporte a técnica de retransmisión de tramas, consiste no encamiñamento das tramas de acordo co seus valores DLCI. Para esta función existe un xestor de tramas



encargado de tomar as decisións de encamiñamento. Esta operación pode involucrar a múltiples xestores de tramas interconectados.

#### **42.1.1.1.3 CONTROL DA CONXESTIÓN**

Utilízanse dous tipos de sinalización para o control da conxestión:

- Sinalización implícita: prodúcese cando a rede descarta tramas, feito que é detectado polos protocolos de nivel superior no interface de usuario.
- Sinalización explícita: é de carácter opcional e constitúe unha sinalización da rede ó interface de usuario mediante dous bits. Ambos bits permiten que os dispositivos finais regulen a velocidade de transmisión de información á rede ata que a conxestión desapareza. Os bits son o BECN (conxestión no sentido oposto á trama) e o FECN (conxestión no sentido da trama).

Adicionalmente, existe outro mecanismo de control da conxestión (CLLM, “Consolidated Link Layer Management”), consistente en mensaxes que a rede envía aos dispositivos de acceso con códigos indicadores das causas da conxestión, así como unha lista de tódolos DLCI’s que deben reducir o seu tráfico para diminuír o nivel de conxestión.

En Frame Relay defínense dúas clases de tráfico por Circuito Virtual Permanente:

- Clase de Caudal ou CIR (Committed Information Rate): defínese como o caudal de información que a Rede comprométese a transmitir expresado en bit/sg. Nun dimensionamiento correcto, debe corresponder a un tráfico “normal” ou promedio. Recomendase que non supere o 75 % da velocidade da liña, aínda cando hai pouca simultaneidade entre os Circuitos Virtuais, pódese superar coa sobrecontratación que se recomenda non exceda do 200%.
- Exceso de Tráfico ou EIR (Excess Information Rate): sen contratación, e que vai dirixido a permitila transmisión de ráfagas de gran intensidade de tráfico, sen custe adicional: defínese como a cantidade de información en exceso do CIR contratado, que a Rede é



capaz de xestionar durante un período de tempo definido. Esta clase de tráfico, expresado en bit/sg, é sinalada pola Rede como de menor prioridade ( $DE=1$ ), e en condicións normais será retransmitida e en condicións de conxestión pode ser descartada.

Para os Circuitos Virtuais Commutados defínese un CIR e un EIR Total Agregado que engloba a tódolos CVC establecidos.

#### **42.1.1.1.4 TIPOS DE TRÁFICO TRANSPORTADO**

A necesidade de novas facilidades para as comunicacións de área extendida, en particular para as comunicacións de datos entre redes locais, impulsou enormemente o desenvolvemento e utilización de Frame Relay, debido a que satisfai as dúas características predominantes deste tipo de tráfico:

- Tráfico a ráfaga e impulsivo, que esixiría dimensionar en exceso o enlace para atender os picos de demanda. Frame Relay soluciona este problema, xa que require unha pequena cantidade de ancho de banda permanentemente reservada vía un circuítio virtual permanente, mentres que dinamicamente, e sempre que exista ancho de banda dispoñible, é posible asignar maior velocidade á conexión para atender picos de demanda.
- A necesidade de interconexión remota de LANs incrementa a necesidade de mallado das redes WAN resultantes. Frame Relay evita a necesidade de numerosos enlaces físicos, permitindo a definición de múltiples circuítos virtuais permanentes a diferentes destinos, a través dun mesmo porto dedicado a un enlace físico.

#### **42.1.1.1.5 VANTAXES**

Aforro nos custes de telecomunicacións: Co servizo Frame Relay os usuarios poderán transportar simultaneamente, compartindo os mesmos recursos de rede, o tráfico pertencente a múltiples comunicacións e aplicacións, e cara diferentes destinos.

Tecnoloxía punta e altas prestacións: Frame Relay proporciona alta capacidade de transmisión de datos pola utilización de nodos de rede de



alta tecnoloxía e baixos retardos, como consecuencia da construción de rede (backbone) sobre enlaces a 34 Mbps. e dos criterios de encamiñamento da Rede de Datos, orientados a minimizar o número de nodos de tránsito.

Flexibilidade do servizo : Frame Relay é unha solución adaptable ás necesidades cambiantes, xa que se basea en circuítos virtuais permanentes (CVP), que é o concepto de Rede Pública de Datos, equivalente ó circuíto punto a punto nunha rede privada. Sobre unha interface de acceso á rede pódense establecer simultaneamente múltiples circuítos virtuais permanentes distintos, o que permite unha fácil incorporación de novas sedes á Rede de Cliente.

Servizo normalizado: Frame Relay é un servizo normalizado según os estándares e recomendacións de UIT -T, ANSI e Frame Relay Forum, co que queda garantida a interoperatividade con calquer outro produto Frame Relay asimesmo normalizado.

#### **42.1.2 ATM**

Asynchronous Transfer Mode (ATM) é unha técnica de conmutación que leva a cabo a transmisión de datos por medio de paquetes (celdas), permite a multiplexación de varias conexións lóxicas sobre unha única interface física e tratase dunha técnica de conmutación de paquetes orientada a conexión.

O protocolo de ATM ten unha mínima capacidade de control de erros, de fluxo e un tamaño de paquete fixo (celda) co que facilita o uso de nodos de conmutación a velocidades elevadas.

As súas principais características son:

- Capacidade de integración de diverso tipo de tráfico.
- Asignación dinámica e flexible do ancho de banda.
- Optimización do compromiso entre caudal e latencia.
- Ganancia estatística: capacidade de optimizar a relación entre a suma das velocidades de pico das fontes e a velocidade do enlace.

##### **42.1.2.1 PRINCIPIOS DE OPERACIÓN**



As conexións lóxicas en ATM están relacionadas coas conexións de canais virtuais (VCC, “Virtual Channel Connection”). Unha VCC é a unidade básica de conmutación en un rede ATM. Unha VCC establecece entre dous usuarios finais a través da rede, intercambiándose celdas de tamaño fixo a través da conexión de un fluxo full-duplex e de velocidade variable. As VCC utilízanse tamén para sinalización de control e xestión de rede e encamiñamento.

Introducíuse unha segunda capa de procesamento en ATM para xestionar o concepto de camiño virtual. Unha conexión de camiño virtual (VPC, “Virtual Path Connection”) é un conxunto de VCC cos mesmos extremos, de maneira que tódalas celdas fluíndo a través das VCC dunha mesma VPC conmútanse conxuntamente.

A técnica de camiño virtual axuda a conter o custo de control agrupando nunha soa unidade conexións que comparten camiños comúns a través da rede. As accións da xestión de rede poden ser aplicadas a un pequeno número de grupos de conexións, en lugar de a un gran número de conexións individuais.

#### **42.1.2.2 CAPAS DE ATM**

As normalizacións ITU-U para ATM baséanse nunha arquitectura onde contéplanse os seguintes niveis:

- Capa física: Especifica o medio de transmisión e un esquema de codificación do sinal. Dividíndose en dúas capas:
  - Subcapa dependente do medio físico (PMD): que leva a cabo funcións de transmisión e temporización de bits.
  - Subcapa de Converxencia de Transmisión (TC): responsable das funcións relacionadas coa transmisión de células como control de HEC, delimitación de celdas, etc..
- Capa ATM: Define a transmisión de datos en celdas de tamaño fixo, ó tempo que establece o uso de conexións lóxicas. Realiza funcións de multiplexación de celdas e control de fluxo.



- Capa de adaptación ATM (AAL): Capa de adaptación para admitir compatibilidade con protocolos de transferencia de información non baseados en ATM. Divídese en dous:
  - Subcapa de Converxencia (CS).
  - Subcapa de Segmentación e reensambrado (SAR).

#### **42.1.2.2.1 CAPA FÍSICA**

A función básica do nivel físico é a codificación/decodificación da información en formato eléctrico ou óptico para a transmisión/recepción sobre o medio físico de comunicación utilizado. Outras funcións proporcionadas por este nivel son a desalineación de celdas e xeneración e proceso do checksum para o control de erros na cabeceira.

As recomendacións ITU-T detallan a velocidade de transmisión e as técnicas de sincronización para a transmisión de celdas ATM. As principais propostas de capas físicas en redes ATM son as seguintes:

- ATM sobre SDH: STM-1 (155,52) e STM-4 (622,08)
- ATM sobre PDH: E1 (2,048), DS1 (1,548), DS2 (6,312), E3 (34,368), E4 (139,264) e DS3 (44,736).
- ATM a 100 Mbps sobre FDDI.
- ATM a 25,6 Mbps (proposta por IBM).

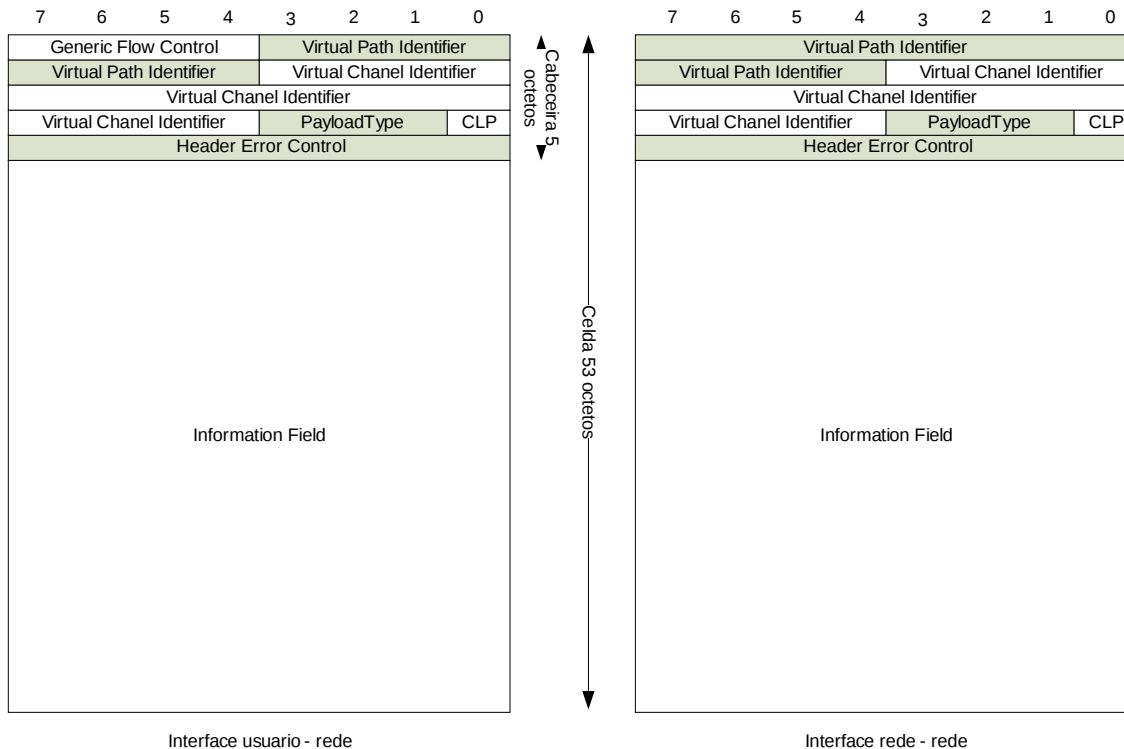
#### **42.1.2.2.2 CAPA ATM**

O modo de transferencia asíncrono utiliza celdas de tamaño fixo, que constan de 5 octetos de cabeceira e un campo de información de 48 octetos. A capa ATM é a encargada de incorporala cabeceira de 5 octetos ó campo de información. A cabeceira ten o seguinte formato:

- Control de Fluxo Xenérico (GFC, "Generic Flow Control"). Utilízase unicamente no interface UNI (User-Network Interface).
- Identificador de Camiño 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 se se trata de datos de usuario, información de xestión, información OAM, etc...



- Prioridade de Perda de Celda (CLP, “Cell Loss Priority”). As celdas marcadas son as primeiras en ser descartadas en caso de conxestión.
- Control de Erros de Cabeceira (“Checksum”).



#### 42.1.2.2.2.1 PARÁMETROS DE TRÁFICO

Cando se establece unha conexión ATM constitúese un “Contrato de tráfico” no que se especifican os parámetros de tráfico e os de QoS, entre os máis significativos están:

- PCR (“Peak Cell Rate”): Taxa máxima de celdas permitida sobre o circuíto.
- MCR (“Minimun Cell Rate”): Mínima taxa de celdas sobre o circuíto garantida polo proveedor do servizo.
- CDVT (“Cell Delay Variation Tolerance”): Nivel de tolerancia para a diferenza entre la celda co mínimo retardo e a celda co máximo retardo.
- SCR (“Sustained Cell Rate”): Taxa media de transmisión de celdas que se manterá durante a duración dunha transmisión.



- BT (“Burst Tolerance”): Límite que a transmisión pode alcanzar no seu nivel máis alto (PCR).

Garántese unha calidade de servizo con respecto a un mínimo ancho de banda dispoñible, a cantidade de retardo que afectará á transmisión e a máxima perda de celdas que se producirá nesta.

#### **42.1.2.2.2 CLASES DE SERVIZO**

A partir destas especificacións iniciais do ITU, definíronse cinco clases de servizo que a rede ATM debería proporcionar (non é obrigatorio):

- CBR (“Constant Bit Rate”): É un servizo determinístico, deseñado para soportar emulación de circuítos, tráfico de voz e vídeo (por exemplo, MPEG/JPEG) en taxa constante de bits. Proporciona ancho de banda reservado, garantido mínima perda de celdas e mínimas variacións en retardos. O servizo CBR proporciona ancho de banda reservado ata o PCR especificado para o circuíto. Con CBR o usuario debe declarar o PCR e o CDTV no momento de establecela conexión.
- VBR (“Variable Bit Rate”): Definíronse dous tipos:
  - VBR-RT (“Real Time”), proporciona un estreito control dos retardos para a transmisión de información como vídeo e voz sen silencios.
  - VBR-NRT (“Non Real Time”) ten menos esixencias que o anterior, respecto ás variacións en retardos e desenvolveuse para a transmisión de datos transaccionais.

O servizo VBR tamén require a especificación do PCR, se ben con un comportamento diferente: o usuario pode utilizar a canle por enriba do SCR (ata o PCR) só durante curtos períodos de tempo determinados polo parámetro BT, pero debe manter o SCR como unha taxa media. Se o usuario excede o SCR durante un período de tempo, a isto seguiralle un período similar por debaixo do SCR. Se o usuario excede o PCR, esto seguirase por un período de inactividade antes de que o usuario poda exceder o SCR de novo. Con VBR o usuario debe declarar PCR, CDVT, SCR y BT.



- UBR (“Unspecified Bit Rate”): Diseñouse para permitir o uso de ancho de banda excedente non utilizado para os servicios CBR e VBR. Non ofrece garantías en canto á perda de celdas ou variacións en retardos; é dicir, non inclúe ningún mecanismo de control no caso de conxestión na rede. Con UBR non hai descritores de tráfico, nin garantías de calidade de servizo nin mecanismos de realimentación en caso de conxestión na rede. A estación pode enviar tráfico cando o necesite e a rede o aceptará; iso si, en caso de conxestión, a estación non é notificada de ningún modo e o conmutador eliminará celdas cando os seus buffers estén cheos. Isto significa que a taxa potencial de perda de celdas con UBR pode ser inaceptablemente alta.
- ABR (“Available Bit Rate”): Ó igual que UBR diseñouse para aproveitalo ancho de banda excedente pero, ó contrario que aquel, implementa mecanismos de control e control en caso de conxestión na rede. ABR concebiuse para transportar o tráfico a ráfagas sen as limitacións de calidade de servizo de UBR, basicamente aplicacións que non funcionen en tempo real e polo tanto pouco sensibles a retardos. Utiliza basicamente os descritores PCR y MCR; o usuario comprométese a non enviar información mais rápido que o PCR e a rede a proporcionar como mínimo o MCR requirido. O usuario non está obrigado a especificar PCR e MCR; en ausencia de tal especificación, os valores por defecto serían o PCR a velocidade de acceso e o MCR a cero. Se se cumpren estes parámetros descritores de tráfico, garántese a calidade de servizo en canto a mínimo nivel de perda de celdas e mínimo ancho de banda asegurado; o retardo de celdas será minimizado, pero non existe garantía absoluta respecto ó retardo para o servizo ABR. Hai outro tipo de servizo en estudio polo ATM Forum (VBR+) que, como o ABR, contempla un sistema de realimentación para control da conxestión na rede pero,



adicionalmente, proporciona ademais garantías en canto a os retardos.

#### **42.1.2.2.3 ASIGNACIÓN DE ANCHO DE BANDA E CONTROL DE CONXESTIÓN**

Unha rede ATM debe garantir uns determinados parámetros de QoS, ademáis de proporcionar ganancia estatística. Para conseguilo utiliza métodos preventivos, denominados Control de Admisión de Conexión e de monitorización posterior, mediante una función de policía (UPC) que emprega diversos algoritmos para este fin.

Tamén existen métodos reactivos como o CLP (Cell Loss Priority) ou o GFC (Generic Flow Control) que controla o tráfico do usuario á rede.

Para o control da conxestión existen dúas opcións:

- Control baseado en créditos: o extremo receptor emite créditos, que indican o número de celdas que pode enviar o emisor, útil en entornos de área local.
- Control baseado na velocidade: baseado en mensaxes EFCI (Explicit Forward Congestion Indication) , as estaciones e conmutadores axustan a velocidade dinamicamente. Esta é a técnica máis amplamente utilizada.

#### **42.1.2.2.3 CAPA DE ADAPTACIÓN (AAL)**

A función básica do nivel de adaptación ATM (AAL) é proporcionar o enlace entre os servizos requiridos polos niveis superiores de rede e o nivel ATM.

Ata o momento, o ITU-T definiu para a AAL catro clases de servizo, respondendo esta

clasificación a tres parámetros básicos: a relación de tempo entre a fonte e o destino, taxa de bits constante ou variable e modo de conexión. As clases definidas son as seguintes:

<b>Clase</b>	<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>
<b>Relación orixe / destino</b>	Si	Si	No	No
<b>Velocidade</b>	Constante	Variable	Variable	Variable



<b>Orientado a conexión</b>	Si	Si	Si	No
---------------------------------	----	----	----	----

#### **42.1.2.2.3.1 ESTRUCTURA DA CAPA AAL**

A capa AAL está organizada en dúas subcapas:

- Subcapa de segmentación e reensambrado (SAR): Segmenta a información das capas superiores para construíla carga útil das celdas ATM e reciprocamente, reensambla os campos de información das celdas en unidades de información para as capas superiores.
- Subcapa de converxencia (CS): ten como misión realizar funcións específicas para cada servizo, como o tratamento da variación do retardo de celda, sincronización extremo a extremo, tratamento de celdas mal insertadas ou perdidas. Existen, por tanto, diferentes CS sobre a subcapa SAR. Debido á gran cantidade de servizos propostos sobre ATM, foi necesario distinguir entre unha parte común CS (CPCS) e unha parte específica de servizo (SSCS).

Inicialmente, el ITU-T recomendou catro tipos de protocolos AAL para soportar as catro clases de servizo definidas, os protocolos AAL de tipo 1, 2, 3, e 4. Así, o tráfico de clase 1 utilizará o protocolo AAL-1, o de clase 2 o AAL-2, e os de clase 3 e 4 o protocolo AAL-3/4. Sendo os protocolos das clases 3 e 4 un protocolo único.

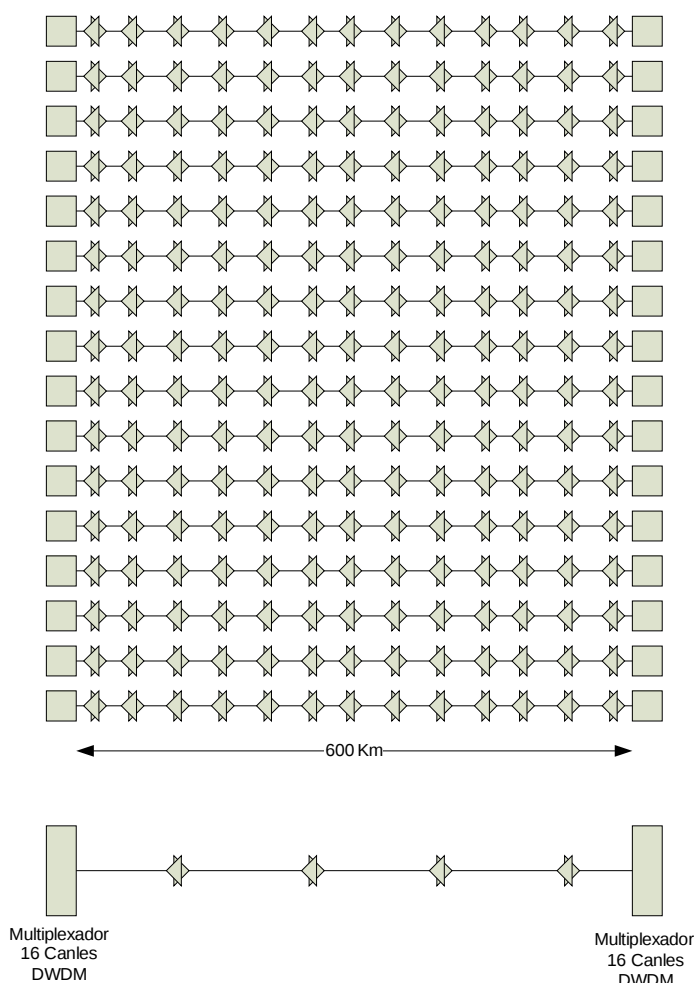
Debido á complexidade do protocolo AAL-3/4 propúxose como alternativa o AAL-5, a veces denominado SEAL ("Simple and Efficient Adaptation Layer"). En consecuencia, as clases de tráfico 3 e 4 poden utilizar o protocolo AAL-3/4 ou o AAL-5.

#### **42.1.3 DWDM**

A tecnoloxía DWDM usa unha composición óptica do sinal de distintos fluxos de datos cada un dos cales na súa propia lonxitude de onda óptica. A pesar de que a división e multiplexado usando o espectro óptico é unha tecnoloxía que se coñece dende hai tempo, as súas primeiras aplicacións restrinxían o seu uso a prover dous lonxitudes de onda moi grosos e moi separados ou a construción de compoñentes capaces de ata 4 canles. So



recentemente a tecnoloxía evolucionou ata o punto de poder empacar e integrar nun sistema de transmisión unha alta densidade de canais paralelos, simultáneos, a unha frecuencia extremadamente alta (192 - 200 Terahertz). Conforme ó plan de canles do ITU, este sistema asegura a interoperabilidade con outros equipos e permite os operadores posicionarse ben para despregar solucións ópticas na súa rede. O sistema de 16 canles proporciona basicamente un cable virtual con 16 fibras.



Para transmitir 40Gb/s a 600 Kms usando un sistema tradicional requirimos 16 pares de fibra óptica con rexeneradores cada 35 kms cun total de 272 rexeneradores. Un sistema DWDM de 16 canles usa só un par de fibra óptica e 4 amplificadores posicionados cada 120Km.

A forma mais común de DWDM usa un par de fibra (unha para transmitir e outra para recibir). Aínda que existen sistemas que só usan unha fibra para transmitir e recibir, estes sistemas deben sacrificar un pouco da



capacidade da fibra óptica para unha banda de garda e evitar así a mestura de canles reducindo tamén o rendemento dos amplificadores.

Adicionalmente, existe un gran risco de que os reflexos producidos durante o mantemento ou reparación podan danar os amplificadores.

A dispoñibilidade de tecnoloxías maduras de soporte como os multiplexadores precisos e os EDFA (Erbium Doped Fiber Amplifiers) permitiu a dispoñibilidade comercial de sistemas DWDM con oito, dezaseis, ou incluso un maior número de canles.

#### **42.1.3.1 DEMULTIPLEXADORES**

Con sinais tan precisos e densos como os usados en DWDM, ten que existir unha forma que aporte unha precisa separación dos sinais, ou filtrado, no receptor óptico. Esta solución tamén ten que ser fácil de implementar e non requirir mantemento. Os sistemas primitivos de filtrado eran ou demasiado imprecisos para DWDM ou demasiado sensibles a variacións de temperatura e polarización, demasiado vulnerables a cruces de comunicacións en canles adxacentes ou demasiado caros. Isto restrinxiu a evolución de DWDM. Para conseguir satisfacer os requisitos de alto rendemento, desenvolveuse unha nova tecnoloxía de filtrado que fixo DWDM posible a un custo aceptable: a fibra con rexilla de Bragg.

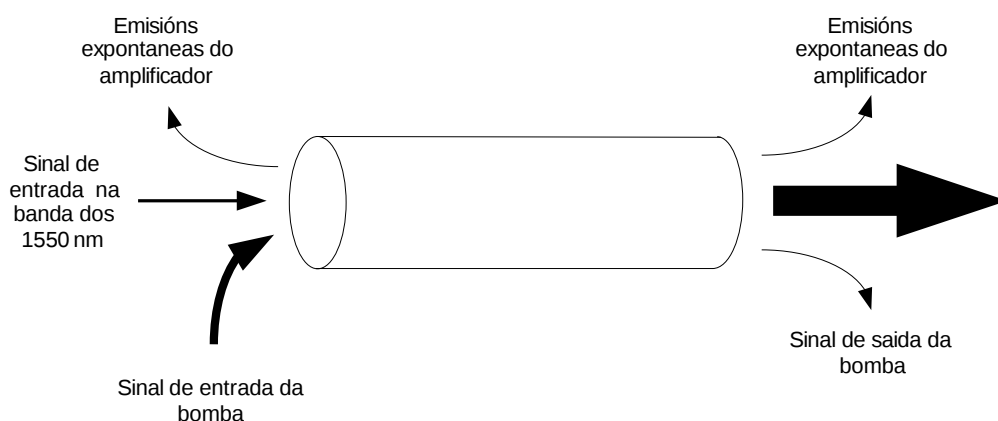
O novo compoñente de filtro (fibra con rexilla) consiste nunha determinada lonxitude de fibra óptica onde o índice de refracción do núcleo foi permanentemente modificado en puntos equidistantes, xeralmente por exposición a un patrón de interferencia ultravioleta. O resultado é un compoñente que reflecta a luz dependendo da lonxitude de onda e é útil para separar lonxitudes de onda. Noutras palabras, a rexilla crea un filtro altamente selectivo para unha lonxitude de onda moi estreita que funciona de forma similar a un espello e aporta maior selectividade de lonxitude de onda que calquera outra tecnoloxía. Como este é un dispositivo pasivo, fabricado en fibra de vidro, é robusto e durable.

#### **42.1.3.2 ERBIUM DOPED FIBER AMPLIFIER**

A chegada do EDFA permitiu o desenvolvemento de sistemas DWDM comerciais provendo unha forma de amplificar toda as lonxitudes de onda



ó mesmo tempo. Esta amplificación óptica faise incorporando ións de Erbium no núcleo dunha fibra especial nun proceso coñecido como dopado. Úsanse bombas ópticas láser para transferir altos niveis de enerxía á fibra especial, enerxizando os ións de Erbium que logo aumentan os sinais ópticos que pasan pola fibra. A estrutura atómica do Erbium proporciona amplificación ós grandes rangos do espectro que se requiren para DWDM.



En vez de múltiples rexeneradores electrónicos, que requiren que o sinal óptico sexa convertido a sinal electrónico e viceversa, o EDFA amplifica directamente os sinais ópticos. Desta forma o sinal pode ser enviado ata 600 Km sen rexeneración e ata 120Km entre amplificadores nun sistema DWDM dispoñible comercialmente.

#### **42.1.4 MPLS**

MPLS (MultiProtocol Label Switching) é unha solución versátil para resolver os problemas que afrontan as redes actuais de velocidade, escalabilidade, xestión da calidade de servizo (QoS) e enxeñería do tráfico. MPLS emerxeu coma unha solución elegante para aportar a xestión de ancho de banda e requirimentos de servizo para a nova xeración de redes troncais baseadas en IP. MPLS resolve problemas relacionados coa escalabilidade e encamiñamento (baseados en QoS e métricas da calidade de servizo) e pode existir sobre redes ATM e Frame Relay xa existentes.

##### **42.1.4.1 FUNCIÓNS DE MPLS**

MPLS realiza as seguintes funcións:



- Especifica mecanismos para xestionar os fluxos de tráfico de varias granularidades, coma os fluxos entre diferente hardware, maquinas ou incluso entre diferentes aplicacións.
- Mantense independente dos protocolos das capas 2 e 3.
- Aporta un método para mapear direccións IP a simples etiquetas de tamaño fixo, usadas por diferentes tecnoloxías.
- Intégrase con protocolos existentes coma RSVP (Resource Reservation Protocol) e OSPF (Open Shortest Path First).
- Da soporte a IP, ATM e Frame Relay.

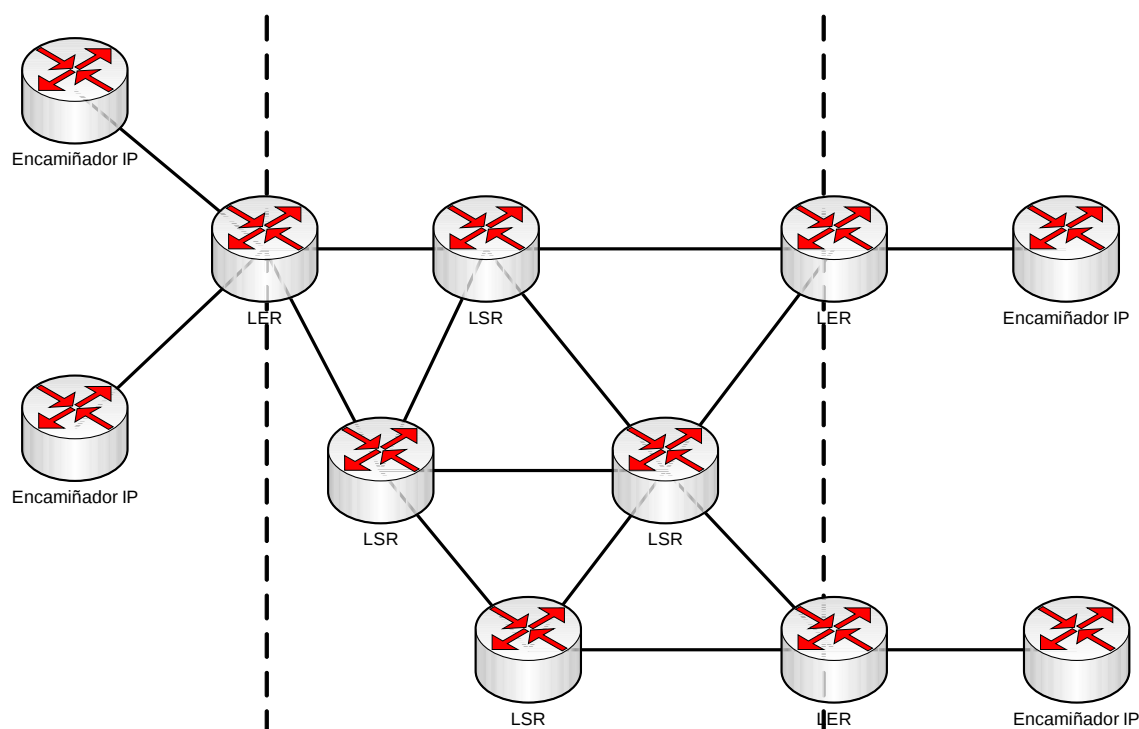
En MPLS a transmisión de datos prodúcese en LSPs (Label Switching Paths). Un LSP é unha secuencia de etiquetas para cada un dos nodos ó longo do camiño, desde a orixe ata o destino. Os LSPs establécense antes da transmisión dos datos (establecemento por control) ou ante a detección dun determinado fluxo de datos (establecemento por datos). As etiquetas (que son identificadores específicos dos protocolos subxacentes) distribúense usando LDP (Label Distribution Protocol) ou RSVP ou acompañando os datos de protocolos de encamiñamento como o BGP (Border Gateway Protocol) e OSPF. Cada paquete de datos encapsula e transporta as etiquetas durante o seu traxecto dende a orixe ata o destino. A alta velocidade de conmutación é posible pola lonxitude fixa das etiquetas e por que estas son insertadas ao principio do paquete ou celda e isto permite que hardware específico as use para conmutar paquetes a alta velocidade entre distintos enlaces.

#### **42.1.4.2 LSRS E LERS**

Os dispositivos que participan nos mecanismos do protocolo MPLS poden clasificarse en LERs (Label Edge Routers) e LSRs (Label Switching Routers). Un LSR é un encamiñador de alta velocidade no núcleo dunha rede MPLS que participa no establecemento dos LSPs usando o protocolo de sinais adecuado e proporciona conmutación de alta velocidade dos datos baseándose nos camiños establecidos.



Un LER é un dispositivo que opera no límite entre a rede de acceso e da rede MPLS. LERs soportan múltiples portos conectados a diferentes redes (Frame Relay, ATM, Ethernet, ...) e envía o tráfico á rede MPLS despois de establecer o seu LSP, tamén distribúe o tráfico á rede de acceso no proceso inverso. O LER xoga un rol moi importante na asignación e renovación de etiquetas, cando o tráfico entra ou sae dunha rede MPLS.



#### **42.1.4.3 FEC**

A FEC (Forward Equivalent Class) é a representación dun grupo de paquetes que comparten os mesmos requisitos de transporte. Todos os paquetes deste grupo reciben o mesmo tratamento na súa ruta cara o destino. Ó contrario que IP, en MPLS, a asignación dun determinado paquete a un FEC realízase só unha vez, cando o paquete entra na rede. Os FECs están baseados nos requisitos do servizo para un conxunto de paquetes ou simplemente nun determinado prefixo de rede. Cada LSR constrúe unha táboa para saber como un paquete debe ser enviado. Esta táboa, chamada LIB (Label Information Base) componse de relacións FEC – etiqueta.

#### **42.1.4.4 ETIQUETAS**



Unha etiqueta na súa forma mais simple, identifica o camiño que debe recorrer un paquete. As etiquetas encapsulanse nunha cabeceira de nivel 2 xunto co paquete. O encamiñador que o recibe, examina o paquete para obtela súa etiqueta para determinalo seguinte salto. Unha vez que o paquete esta etiquetado, o resto do camiño pola rede troncal basease en conmutación por etiquetas. Os valores das etiquetas só teñen valor local, esto quere dicir que so pertencen a saltos entre LSRs concretos.

Cando un paquete se clasifica como un FEC novo ou existente, asignáselle unha etiqueta. Os valores da etiqueta derívanse da capa de enlace inferior (DLCI para Frame Relay, VPI/ VCI para ATM, ...).

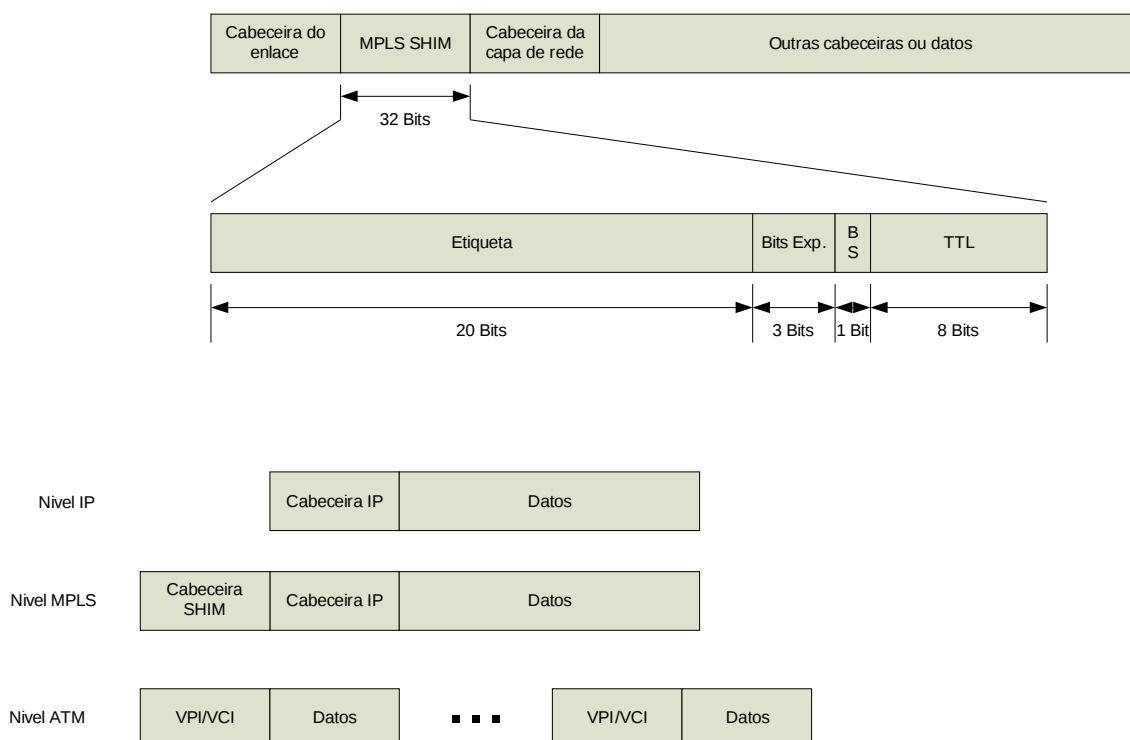
As etiquetas enlázanse cos FEC como resultado dun evento ou política que indica a necesidade dese enlace. Estes eventos poden ter como orixe as sinais de control ou as propias transmisión de datos sendo esta última a mellor opción polas súas propiedades de escalado.

A asignación de etiquetas basease en criterios de encamiñado coma:

- Destino Unicast
- Enxeñaría de tráfico
- Multicast
- VPN (Virtual Private Network)
- QoS

Na seguinte imaxe podemos ver a estrutura dunha etiqueta MPLS e un exemplo do proceso dende IP ata a capa de enlace (neste caso sobre ATM).





#### 42.1.4.5 DISTRIBUCIÓN DE ETIQUETAS

MPLS non impón un só método de distribución de etiquetas. Existen protocolos de encamiñado, como BGP e RSVP que foron mellorados para poder incorporar (usando o método *poggyback*) a información das etiquetas xunto coa información do protocolo. O IETF (Internet Engineering Task Force) tamén definiu un protocolo chamado LDP (Label Distribution Protocol) para xestionar as etiquetas. Extensións de LDP permiten definir rutas explícitas baseadas en requisitos de Qos. Estas extensións están recollidas na definición do protocolo CR-LSP (Constrint-based Routing-LDP). Un resumo dos varios esquemas para o intercambio de etiquetas é o seguinte:

- LDP: mapea direccións de unicast IP a etiquetas.
- RSVP, CR-LDP: úsanse para enxeñaría do tráfico e reserva de recursos.
- PIM (Protocol Independent Multicast): usase para mapear multicasts a etiquetas.
- BGP: etiquetas externas (VPN)

#### 42.1.4.6 LSP



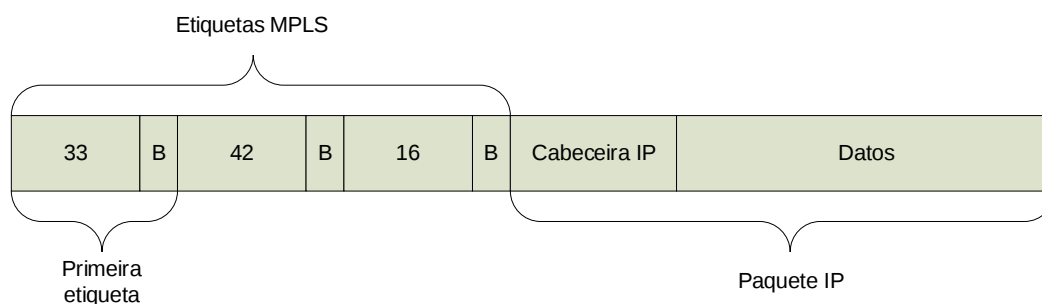
Unha colección de dispositivos MPLS defínese coma un dominio MPLS. Dentro dun dominio MPLS, fíxase un camiño para un paquete baseándose no seu FEC. O LSP fíxase antes da transmisión dos datos usando unha destas dúas opcións:

- Encamiñado salto a salto (hop-by-hop): cada LSR selecciona independentemente o seguinte salto para un FEC determinado. Esta é unha metodoloxía moi similar á usada en redes IP. O LSR usa os protocolos de encamiñamento dispoñibles coma OSPF, etc.
- Encamiñado explícito (ER-LSP): O LSR de entrada (o LSR onde o fluxo de datos comeza na rede) especifica unha lista de nodos que o ER-LSP atravesará. O camiño podería ser non óptimo. Os recursos necesarios poderían ser reservados para garantir QoS. Isto facilita a enxeñaría de tráfico na rede, e permite que servizos diferentes sexan dados usando fluxos baseados en métodos de políticas ou xestión de rede.

Un LSP para un determinado FEC é unidireccional en natureza, o tráfico de retorno terá que usar outro LSP.

#### **42.1.4.7 PILA DE ETIQUETAS**

O apilado de etiquetas (Label Stack) é un mecanismo que permite a operación xerárquica dentro dun dominio MPLS. Basicamente permite a MPLS ser usado simultaneamente para o encamiñado a nivel fino (entre encamiñadores individuais dentro dun ISP) e nun nivel groso (dominio a dominio). Cada nivel, nunha pila de etiquetas, pertence a un nivel xerárquico. Isto facilita o uso de túneles en MPLS.



## **42.2 REDES DE FIBRA ÓPTICA**

### **42.2.1 XERARQUÍA DIXITAL PLESIÓCRONA (PDH)**



A PDH (Plesiochronous Digital Hierarchy) é unha tecnoloxía usada en redes de telecomunicacións para transportar unha gran cantidade de datos sobre redes de fibra óptica ou radioenlaces. O termo alude a que as diferentes partes da rede funcionan de forma case sincronizada, pero non totalmente. A maior parte dos operadores están a actualizar PDH a SDH ou SONET (capaces de transmitir a maior velocidade) se non a reemplazan directamente por unha tecnoloxía completamente distinta.

PDH permite a transmisión de datos a unha velocidade similar entre as transmisións pero permitindo variacións sobre a velocidade nominal, xa que non hai unha sincronización exacta dentro da rede.

A velocidade de transferencia básica é de 2 Mb/s, para a transferencia de voz esta é dividida en 30 canles de 64Kb/s e dous canles para sinalización e sincronización; ademais todo o enlace (2Mb/s) pódese usar para transmitir datos.

A velocidade de transmisión está controlada por un reloxo no equipo que envía os datos, esta velocidade pode variar 50 ppm dende os 2 Mb/s isto quere dicir que as distintas transmisións poden estar sucedendo a diferentes velocidades (e probablemente o están).

Para transmitir varios fluxos de datos dende unha orixe, estes se multiplexan en grupos de 4 de forma que primeiro se transmite o bit 1 do fluxo 1, logo o bit 1 do fluxo 2, logo o bit 1 do fluxo 3, logo o bit 1 do fluxo 4 e así sucesivamente. O transmisor tamén engade información para poder reconstruír o fluxo cando se recibe. Como os fluxos de datos transmitidos poden non estar a ser recibidos á mesma velocidade no multiplexador de orixe, este supón que os está a recibir á máxima velocidade posible. Esta suposición provoca que as veces non se haxa recibido o bit correspondente dun fluxo, esta situación debe ser notificada ó multiplexador de destino para que este poda reconstruír os fluxos á velocidade correcta.

A velocidade resultante da multiplexación descrita é de 8.448 kbit/s, técnicas similares permiten combinar 4 streams de 8 Mb/s máis bit de recheo (proporcionando 34 Mb/s), 4 streams de 34Mb/s (proporcionando 140Mb/s) e 4 streams de 140 Mb/s (proporcionando 565Mb/s). 565 Mbit/s é



a velocidade típica para transmitir datos sobre fibra óptica para longas distancias.

#### **42.2.2 XERARQUÍA DIXITAL SÍNCRONA (SDH) / SONET**

Synchronous optical networking (SONET) e Synchronous Digital Hierarchy (SDH) son protocolos de multiplexación que transmiten varios streams sobre unha rede de fibra óptica, aínda que tamén se poden usar sobre interfaces eléctricos (a menor velocidade). Esta técnica desenvolveuse para reemplazar PDH na tarefa de transportar un largo número de chamadas de teléfono e tráfico de datos sobre a mesma fibra en problemas de sinalización. A maior diferenza con PDH é que SDH / SONET está sincronizada usando reloxos atómicos reducindo a necesidade de buffers na rede e aproveitándoa mellor.

SONET e SDH, son basicamente idénticos, e, aínda que SONET (ANSI T1.105) é anterior, dado que só é usado en Canada e EEUU mentres que SDH (ITU G.707, G.783, G.784 e G.803) é usado no resto do mundo, SONET considerase unha variación de SDH. Dada a gran similitude entre eles é extremadamente fácil a interconexión entre os dous a calquera velocidade.

##### **42.2.2.1 FRAME SDH / SONET**

Á unidade básica para a transmisión en SDH é a Synchronous Transport Module, level 1 (STM-1), que é transmitida a 155,52Mb/s. SONET cambia o nome desta estrutura a Synchronous Transport Signal 3 concatenated (STS-3c) ou OC-3c dependendo se o sinal se transporta electricamente (STS) ou opticamente (OC) pero a súa funcionalidade, velocidade e tamaño son iguais a STM-1.

SONET proporciona outra unidade STS-1 ou OC-1 que opera a 51,84Mb/s (un tercio de STM-1) esto busca poder transmitir unha canle DS-3 estándar (672 canles de 64Kb/s para voz) .

##### **42.2.2.1.1 FRAMING**

En tecnoloxías como Ethernet a trama consiste nunha cabeceira e un conxunto de datos, a cabeceira é transmitida primeiro, seguida dos datos e posiblemente dunha cola contendo o CRC ou equivalente. En SDH esto



modifícase lixeiramente, a cabeceira denomínase overhead e en vez de ser transmitida antes que o resto dos datos é entrelazada con eles durante a transmisión: transmítese parte da cabeceira, logo parte dos datos, ... ata que se transmite todo o frame.

No caso de STS-1, cada frame está composto de 810 octetos, mentres que no caso de STM-1/STS-3 cada frame está composto de 2.430 octetos. STS-1 transmite 3 octetos de overhead seguidos de 87 de datos durante nove veces ata transmitir os 810 octetos en 125 microsegundos. No caso de STM-1 (que opera a tres veces a velocidade de STS-1) transmítense 9 octetos de overhead e 261 de datos, tamén 9 veces ata que se transmite nos 2.430 octetos levando tamén 125 microsegundos. Isto soe representarse graficamente dibuxando o frame como un bloque de 90 columnas e 9 filas para STS-1 e 270 columnas e 9 filas para STM-1 desta forma a representación alínea toda a overhead e todos os datos da payload.

A estrutura interna da overhead e dos datos transmitidos dentro do frame varían lixeiramente para SONET e SDH, usando tamén diferentes nomes para describir estruturas.

#### **42.2.2.1.2 ESTRUCTURA DO FRAME STM-1**

A Overhead Section e os Administrative Unit Pointers ocupan as 9 primeiras columnas do frame. Estes punteiros (os bytes H1, H2 e H3) identifican AUs (Administrative Units) dentro da carga útil. Cada unha destas unidades administrativas pode albergar un ou varios Virtual Containers (VC) que a súa vez conteñen unha descrición do camiño (Path OverHead, POH) e datos. A primeira columna é para o POH e o resto son para datos que poden ser á súa vez outros VC. As unidades administrativas poden ter calquera tipo de alineación e é esta alineación a que é indicada polo punteiro da fila 4.







- o Punteiro AU: Apunta á localización do byte J1 nos datos (o primeiro byte do VC).
- Datos Path: os datos transmitidos de extremo a extremo chámanse datos path e están compostos de 2 elementos:
  - o POH (Path OverHead): nove octetos para sinalización extremo a extremo.
  - o Datos de usuario: 2340 octetos de datos

### **42.3 REDES DE NOVA XERACIÓN (NGN)**

Segundo o ITU-T, unha NGN é unha rede baseada en paquetes que pode prover servizos, incluíndo servizos de telecomunicacións, e capaz de usar varias tecnoloxías de banda ancha que incorporen capacidades de calidade de servizo. Nestas redes as funcións relacionadas co servizo deben ser independentes da tecnoloxía de transporte usada. Ofrecen un acceso sen restriccións ó usuario a diferentes provedores de servizo. Soporta a mobilidade o que permitirá una provisión de servizo consistente e ubicua aos usuarios.

Dende un punto de vista práctico, NGN implica tres cambios na arquitectura:

- Na rede troncal, NGN implica a consolidación de diversas redes de transporte, cada un das cales construída historicamente para un servizo diferente nunha soa rede troncal de transporte (normalmente baseada en IP e Ethernet). Implica entre outras, a migración da voz dende unha arquitectura de conmutación de circuítos (RTC) a VoIP, e tamén a migración de servizos como X.25 e Frame Relay (xa sexa unha migración comercial a nivel de usuario con servizos como VPN sobre IP, ou a migración técnica emulando estes servizos pero sobre a NGN).
- Na rede de acceso por par de cobre, NGN implica a migración dende o sistema dual cá voz independente do xDSL na central á que chega o bucle de aboado a unha configuración onde os DSLAMs (Digital Subscriber Line Access Multiplexer, o punto onde se concentran todas



ás conexións DSL da central) integran portos de voz ou VoIP, posibilitando a eliminación da infraestrutura de conmutación de voz.

- Na rede de acceso de cable, a converxencia NGN implica a migración da voz de servizos CBR a estándares de VoIP e SIP.

Nunha NGN existe unha maior distancia entre a parte da rede que proporciona o transporte (conectividade) e os servizos que se executan sobre este. Isto ten como consecuencia que cada vez que un operador quere dar un novo servizo só ten que definir o nivel da capa de servizos sen preocuparse polo transporte. Cada vez mais os servizos (incluíndo os de voz) tenden a ser independentes da rede de acceso e residen mais nos equipos de usuario final (PC, Set-Top Box, ...).

#### **42.3.1 TECNOLOXÍAS DE SOPORTE**

As redes de nova xeración están baseadas en tecnoloxías coma IP e MPLS. A nivel de aplicación SIP (Session Initiation Protocol) está a reempazar a H.323. Aínda que orixinalmente H.323 era o protocolo de VoIP / videoconferencia / ... sobre redes IP máis popular, ás súas limitacións para atravesar NAT (Network address translation) e firewalls reducen a súa implantación a nivel do bucle de aboado. Son estas algunhas das razóns (xunto coa complexidade de H.323) que están levando á implantación de SIP, sobre todo no bucle de aboado. Sen embargo, nas redes de operador (onde todo está baixo o seu control) moitos de eles usan H.323 para ás súas redes troncais.

Cos novos cambios introducido sen H.323 é posible que dispositivos H.323 atravesen NAT e firewalls facilmente, esta circunstancia pode propiciar que H.323 poda volver a ser usado en entornos onde isto sexa necesario.

Como contrapartida moitos operadores están á investigar e dar soporte a IMS (IP Multimedia Subsystem, unha arquitectura estandarizada para servizos multimedia en Internet definida polo ETSI e a 3GPP) que daría a SIP a oportunidade de ser protocolo máis usado.

Para aplicacións de voz o dispositivo mais importante da NGN é o Softswitch (este nome e ás súas funcións aínda son moi dependentes do



fabricante), un dispositivo que controla as chamadas VoIP permitindo a correcta integración de diferentes protocolos dentro da NGN. A súa función principal é crear a interface ás redes telefónicas existentes (RTC).

Un dos termos máis comunmente usados é o de GateKeeper, que orixinalmente referíase a un dispositivo que transformaba voz e datos dende os seus formatos analóxicos a IP. Cando este dispositivo comezou a usar Media Gateway Control Protocol pasou a chamarse Media Gateway Controller (MGC).

Un Axente (Call Agent, SIP Agent, ...) é un nome xeral para dispositivos capaces de controlar chamadas.

#### **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





## **43. TECNOLOXÍAS SEN FÍOS: BLUETOOTH, WIBREE, WIRELESS USB, WI-FI. RFID. TECNOLOXÍAS MÓBILES.**



## **Tema 43. Tecnoloxías sen fíos: Bluetooth, WiBree, Wireless USB, Wi-Fi. RFID. Tecnoloxías móbiles**

### 43.1 Tecnoloxías sen fíos

#### 43.1.1 Bluetooth

##### 43.1.1.1 Funcionamento de Bluetooth

##### 43.1.1.1.1 Especificacións e características

###### 43.1.1.1.1.1 Versión 1.0 e 1.0B

###### 43.1.1.1.1.2 Versión 1.1

###### 43.1.1.1.1.3 Versión 1.2

###### 43.1.1.1.1.4 Versión 2.0 + EDR

###### 43.1.1.1.1.5 Versión 2.1 + EDR

###### 43.1.1.1.1.6 Versión 3.0 + HS

###### 43.1.1.1.1.7 Versión 4.0

##### 43.1.1.1.2 Pila de protocolos

###### 43.1.1.1.2.1 Capa de banda base e interface de rádio

###### 43.1.1.1.2.2 Capa do protocolo de xestión de enlace (LMP)

###### 43.1.1.1.2.3 Interface de controlador Host (HCI)

###### 43.1.1.1.2.4 Capa do protocolo de adaptación e control do enlace lóxico (L2CAP)

###### 43.1.1.1.2.5 Capa do protocolo de descubrimento de servizos (SDP)

###### 43.1.1.1.2.6 Capa RFCOMM

###### 43.1.1.1.2.7 Comandos AT

#### 43.1.2 WiBree

#### 43.1.3 Wireless USB

##### 43.1.3.1 Descrición do sistema

###### 43.1.3.1.1 Topoloxía

###### 43.1.3.1.1.1 USB host

###### 43.1.3.1.1.2 Dispositivos Wireless USB

###### 43.1.3.1.1.3 Interface física

###### 43.1.3.1.1.3.1 Velocidades da capa física



- 43.1.3.1.1.3.2 Soporte de canles
          - 43.1.3.1.1.3.3 Selección de canle
          - 43.1.3.1.1.4 Xestión de enerxía
          - 43.1.3.1.1.5 Protocolo de bus
          - 43.1.3.1.1.6 Robustez
            - 43.1.3.1.1.6.1 Xestión dos erros
          - 43.1.3.1.1.7 Seguridade
          - 43.1.3.1.1.8 Configuración
            - 43.1.3.1.1.8.1 Conexión de dispositivos Wireless USB
            - 43.1.3.1.1.8.2 Desconexión
            - 43.1.3.1.1.8.3 Enumeración
          - 43.1.3.1.1.9 Tipos de fluxos de datos
          - 43.1.3.1.1.10 Dispositivos Wireless USB
            - 43.1.3.1.1.10.1 Características dos dispositivos
            - 43.1.3.1.1.10.2 Dispositivos e capa MAC
          - 43.1.3.1.1.11 Hardware e software do host
- 43.1.4 Wi-Fi
  - 43.1.4.1 Descrición do sistema
    - 43.1.4.1.1 Capa física(PHY)
      - 43.1.4.1.1.1 Infravermellos
      - 43.1.4.1.1.2 FHSS
      - 43.1.4.1.1.3 DSSS
      - 43.1.4.1.1.4 Tramas da capa física
    - 43.1.4.1.2 Capa de acceso ó medio (MAC)
      - 43.1.4.1.2.1 Tramas do nivel MAC
- 43.2 RFID
  - 43.2.1 Principios de RFID
  - 43.2.2 Compoñentes e operación
- 43.3 Tecnoloxías móbiles
  - 43.3.1 Android



43.3.2 Meego

43.3.3 Symbian

43.3.4 Windows Phone 7

43.4 Bibliografía

## **43.1 TECNOLOXÍAS SEN FÍOS**

### **43.1.1 BLUETOOTH**

É unha tecnoloxía sen fíos de curto alcance (PAN) deseñada para substituír os cables entre dispositivos que se converteu na solución sen fíos ideal para conectar teléfonos móbiles con portátiles para súa conexión a Internet, ou para que outros organizadores de mano, como PDAs poidan conectarse ó PC para coordinar os seus contactos, e incluso para poder imprimir desde un ordenador sen necesidade de cables.

As características intrínsecas das tecnoloxías con bluetooth permiten establecer conexións seguras, con capacidade de encriptación da canle, autenticación da rede e outros parámetros de seguridade como a localización e dispositivo do usuario.

#### **43.1.1.1 FUNCIONAMENTO DE BLUETOOTH**

Bluetooth traballa no rango de frecuencias de 2.4 a 2.48 Ghz, con espectro ensanchado (widespread) e saltos de frecuencia (frequency hopping), con posibilidade de transmitir en full-duplex con un máximo de 1600 saltos/seg. Os saltos de frecuencia realízanse entre un total de 79 frecuencias con intervalos de 1Mhz, o cal permite brindar seguridade e robustez. A frecuencia na cal traballa permítelle atravesar paredes, polo cal é ideal tanto para o móbil, como en oficinas.

A potencia de saída para transmitir a unha distancia máxima de 10m é 1-2,5 mW, mentres que a versión de largo alcance, ata 100m, transmite a 100 mW.

Para lograr alcanzar o obxectivo de baixo consumo e baixo custe, deseñouse unha solución integrada nun só chip. Desta maneira, logrouse crear unha solución de 9x9mm e que consume aproximadamente 97% menos enerxía que un teléfono celular común.



Cada unha das catro canles de voz na especificación Bluetooth pode soportar unha taxa de transferencia de 64 Kb/s en cada sentido, o cal é suficientemente adecuada para a transmisión de voz. Unha canle de datos asíncrono pode transmitir 721 Kb/s nunha dirección e 56 Kb/s na dirección oposta, sen embargo, para unha conexión asíncrona é posible soportar 432,6 Kb/s en ambas direccións se o enlace é simétrico.

Para relacionarse e intercambiar información os dispositivos Bluetooth ofrecen distintos servicios, chamados tecnicamente Perfiles, entre estes perfiles encóntranse o Acceso a Redes locais (LAN), Acceso Telefónico, Fax, Transferencia de Arquivos, Sincronización, Intercomunicador, ou Telefonía sen fíos, entre outros. Desta

forma cando dous dispositivos se comunican por primeira vez intercambian esta información para coñecer as súas posibilidades de intercomunicación. As compañías mais destacadas no desenvolvemento desta tecnoloxía foron Ericsson e Nokia. A primeira versión do estándar Bluetooth lanzouse en maio de 1998. Esta tecnoloxía sen fíos tiña unha velocidade de transferencia de datos de 1Mbps e conta con un alcance máximo de 100 metros. Sen embargo a versión mais utilizada é a de alcance 10 metros, xa que o consumo eléctrico aumenta rapidamente con unha maior potencia de transmisión.

#### **43.1.1.1.1 ESPECIFICACIÓN E CARACTERÍSTICAS**

Tras o deseño da primeira especificación en 1994 a especificación foi ratificada polo SIG (Bluetooth Special Interest Group) en 1998 e dende entón evolucionou en varias versións ata a actualidade. Todas as versións inclúen a compatibilidade cos dispositivos das versións anteriores.

##### **43.1.1.1.1.1 VERSIÓN 1.0 E 1.0B**

Estas versións tiveron moitos problemas sobre todo no que a interoperabilidade entre fabricantes se refire.

Tamén obrigaba a incluír a dirección física na transmisión durante o proceso de conexión o cal era contraproducente para algúns dos servizos planificados.

##### **43.1.1.1.1.2 VERSIÓN 1.1**



Foi establecida como estándar 802.15.1-2002 polo IEEE corrixindo moitos erros da versión 1.0B e engadindo soporte para canle son encristados e RSSI (Received Signal Strength Indicator).

#### **43.1.1.1.1.3 VERSIÓN 1.2**

As melloras desta versión inclúen:

- Maior velocidade de conexión e busca (Discovery).
- Uso de AFH (Adaptive Frequency-Hopping) que mellora a resistencia a interferencias evitando o uso de frecuencias moi ocupadas na secuencia de saltos.
- Maior velocidade de transmisión, na práctica ata 721 kbit/s.
- Uso de eSCO (Extended Synchronous Connections) que mellora a calidade dos enlaces de voz permitindo a retransmisión de paquetes corruptos e, opcionalmente, podendo incrementala latencia para mellorar o soporte para a transferencia de datos concorrente.
- Soporte do HCI (Host Controller Interface) para UARTs de 3 fíos.
- Estandarizado coma IEEE 802.15.1-2005.
- Introducción de modos de control de fluxo e retransmisión para L2CAP.

#### **43.1.1.1.1.4 VERSIÓN 2.0 + EDR**

A principal diferenza coa versión anterior é a posibilidade de uso de EDR (Enhanced Data Rate) como modo de transferencia rápido sendo a velocidade nominal de 3 Mbit/s aínda que a velocidade práctica real sexa de 2.1 Mb/s. EDR é opcional na especificación polo que poden existir dispositivos da versión 2.0 que non soporten EDR.

#### **43.1.1.1.1.5 VERSIÓN 2.1 + EDR**

Esta versión foi adoptada polo SIG no 2007 sendo a súa maior aportación ao estándar o SSP (Secure Simple Pairing) que mellor a experiencia de emparellamento dos dispositivos Bluetooth mentres mellora a seguridade.

#### **43.1.1.1.1.6 VERSIÓN 3.0 + HS**

Esta versión é do 2009 soportando en teoría velocidades de ata 24Mb/s, pero non sobre o enlace Bluetooth se non que Bluetooth usase para



negociar o establecemento dun enlace 802.11 sobre o que se transfiren os datos.

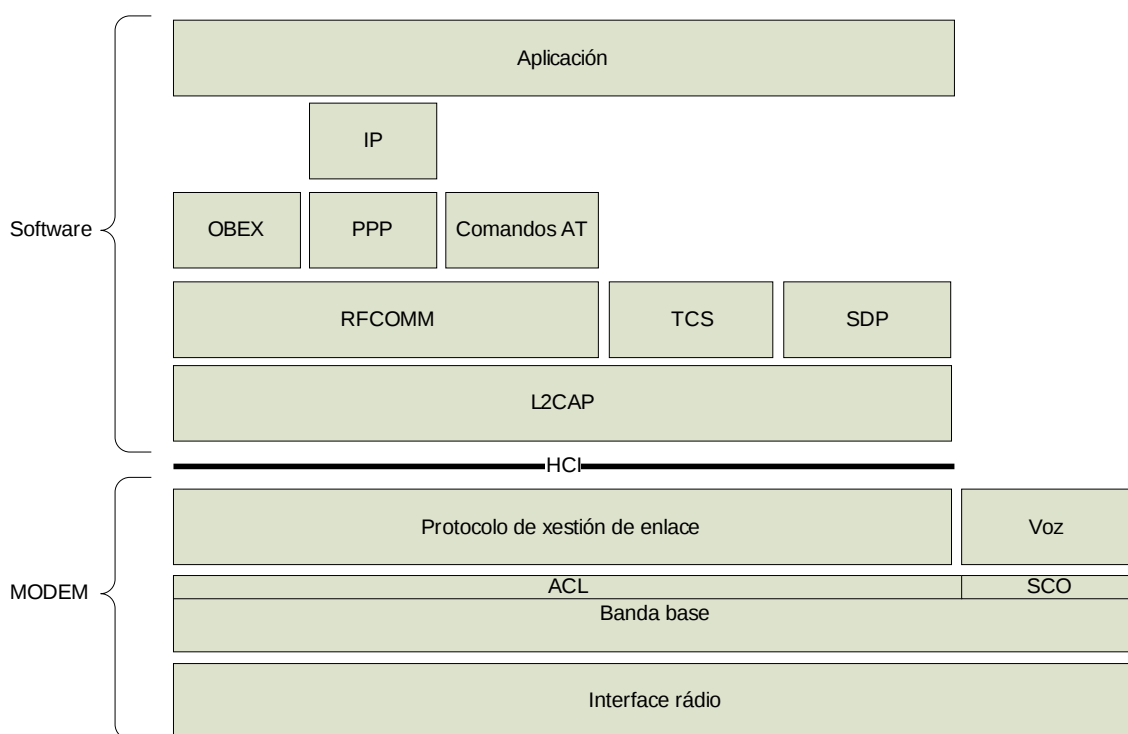
#### **43.1.1.1.1.7 VERSIÓN 4.0**

Esta versión data de xuño de 2010 e inclúe tres sabores do protocolo: Classic Bluetooth, Bluetooth high speed e Bluetooth low energy. Bluetooth high speed está baseado en WiFi e Classic Bluetooth está composto polos protocolos de versións anteriores.

Pola súa parte Bluetooth low energy define unha nova pila de protocolo para a creación rápida de enlaces simple e foi deseñado para executarse en dispositivos de moi baixo consumo.

#### **43.1.1.1.2 PILA DE PROTOCOLOS**

Dunha forma básica podemos ver a pila de protocolos Bluetooth na imaxe seguinte.



##### **43.1.1.1.2.1 CAPA DE BANDA BASE E INTERFACE DE RADIO**

Na base da pila de protocolos Bluetooth encóntranse a capa de banda base e o interface de radio. A súa función principal é permitir o enlace físico por radiofrecuencia (RF) entre unidades Bluetooth realizando tarefas de



modulación e demodulación dos datos en sinais RF que se transmiten polo aire.

O nivel de banda base proporciona dous tipos de enlace físico:

- ACL (Asynchronous ConnectionLess) para enlaces asíncronos sen conexión.
- SCO (Synchronous Connection-Oriented) para enlaces síncronos orientados a conexión.

#### **43.1.1.1.2.2 CAPA DO PROTOCOLO DE XESTIÓN DE ENLACE (LMP)**

A capa LMP é a responsable da configuración e control do enlace entre dispositivos Bluetooth, incluíndo o control e negociación do tamaño dos paquetes.

#### **43.1.1.1.2.3 INTERFACE DE CONTROLADOR HOST (HCI)**

A capa HCI (Host Controller Interface) actúa como fronteira entre as capas de protocolo relativas o hardware (módulo Bluetooth) e as relativas ó software (host Bluetooth). Proporciona unha interface de comandos para a comunicación entre o host e o firmware do módulo Bluetooth e permite dispoñer dunha capa de acceso homoxénea para tódolos módulos Bluetooth, aínda que sexan de distintos fabricantes.

#### **43.1.1.1.2.4 CAPA DO PROTOCOLO DE ADAPTACIÓN E CONTROL DO ENLACE LÓXICO (L2CAP)**

A especificación Bluetooth inclúe o protocolo L2CAP (Logical Link Control and Adaptation Protocol), que se encarga da multiplexación de protocolos, xa que o protocolo de banda base non soporta un campo tipo para identificar o protocolo de nivel superior o que quere transmitir a información, por exemplo SDP, RFCOMM e TCS.

Outra función que se realiza no nivel L2CAP é a segmentación e recomposición de paquetes, necesaria para permitirla utilización de protocolos que utilicen paquetes de maior tamaño que os soportados pola capa de banda base.

#### **43.1.1.1.2.5 CAPA DO PROTOCOLO DE DESCUBRIMENTO DE SERVICIOS (SDP)**



O descubrimento de servizos fai referencia á capacidade de buscar e encontrar servizos dispoñibles en dispositivos Bluetooth. A través dos servizos, dos dispositivos poden executar aplicacións comúns e intercambiar datos.

#### **43.1.1.1.2.6 CAPA RFCOMM**

O protocolo RFCOMM (Radio Frequency Communication) é un protocolo de emulación de liña serie baseado no estándar ETSI TS 07.10. Proporciona unha emulación dos portos serie RS-232 sobre o protocolo L2CAP.

#### **43.1.1.1.2.7 COMANDOS AT**

Os comandos AT son instrucións codificadas que conforman unha linguaxe de comunicación entre o home e un terminal módem. Os comandos AT denomínanse así por la abreviatura de attention.

#### **43.1.2 WIBREE**

En 2001, investigadores Nokia determinan que existen varios escenarios non cubertos polos sistemas sen fíos da época. Para solucionar este problema Nokia Research Center comezou a desenvolver unha tecnoloxías en fíos adaptada dende o estándar Bluetooth que proporcionase un dispositivo de mais baixo consumo e prezo pero sen ser moi distinto a un Bluetooth. O resultado foi publicado en 2004 usando o nome Bluetooth Low End Extension, e tras mais desenvolvemento a tecnoloxía foi publicada co nome de WiBree en outubro de 2006. Tras unha negociación co SIG de Bluetooth, en xuño de 2007 acordouse que WiBree sería incluído nunha futura especificación de Bluetooth coma Bluetooth ultra-low-power que hoxe se coñece coma Bluetooth low energy (presenta na especificación Bluetooth 4.0).

BLE (Bluetooth Low Energy) opera no mesmo rango de frecuencias que Classic Bluetooth (2402-2480 MHz) pero usa un conxunto de canles distinto, en vez de usar os 79 canles de 1Mhz. BLE usa 40 canles de 2 Mhz. BLE foi deseñado para permitir dúas alternativas: modo simple e modo dual. Os dispositivos sinxelos, coma sensores, reloxos, etc. están baseados no modo simple permitindo só BLE, mentres que os dispositivos de modo dual combinan BLE con Classic Bluetooth na mesma circuitería.



A pesar de que Classic Bluetooth e BLE poden coexistir non son compatibles entre sí. Podemos velas principais diferencias na seguinte taboa.

	Classic Bluetooth	Bluetooth Low Energy
Alcance	100 m	200 m
Velocidade da transmisión	1-3 Mb/s	1 Mb/s
Velocidade aproveitable	0.7-2.1 Mb/s	0.26 Mb/s
Latencia típica	100 ms	6 ms
Capacidade para voz	Si	Non
Topoloxía	Malla	Etrella - Bus
Consumo	1mW	0,01 a 0,5mW
Pico de consumo de corrente	<30mA	<20mA

### **43.1.3 WIRELESS USB**

Wireless USB é un protocolo de comunicación sen fíos de alta velocidade e reducido alcance baseado na plataforma común de radio UWB (Ultra-WideBand) e sendo capaz de transmitir a 480Mb/s ata unha distancia de 3 metros e a 11Mb/s ata unha distancia de 10 metros. Foi deseñado para operar no rango de frecuencias entre 3.1 e 10.6 Ghz aínda que as regulacións de cada país poden limitar este rango.

Sen embargo, a pesar de que existe unha gran excitación sobre Wireless USB e soporte dos grandes fabricantes, non acaba de despegar. A pesares de que tecnicamente Wireless USB ten moitas vantaxes sobre BlueTooth e WiFi (os seus principais competidores) estas outras tecnoloxías xa tiñan a súa posición no mercado e non había espacio para unha nova.

#### **43.1.3.1 DESCRICIÓN DO SISTEMA**

Un sistema USB esta composto por un host e un número indeterminado de dispositivos funcionando na mesma conexión lóxica e pode ser descrito por 3 áreas:

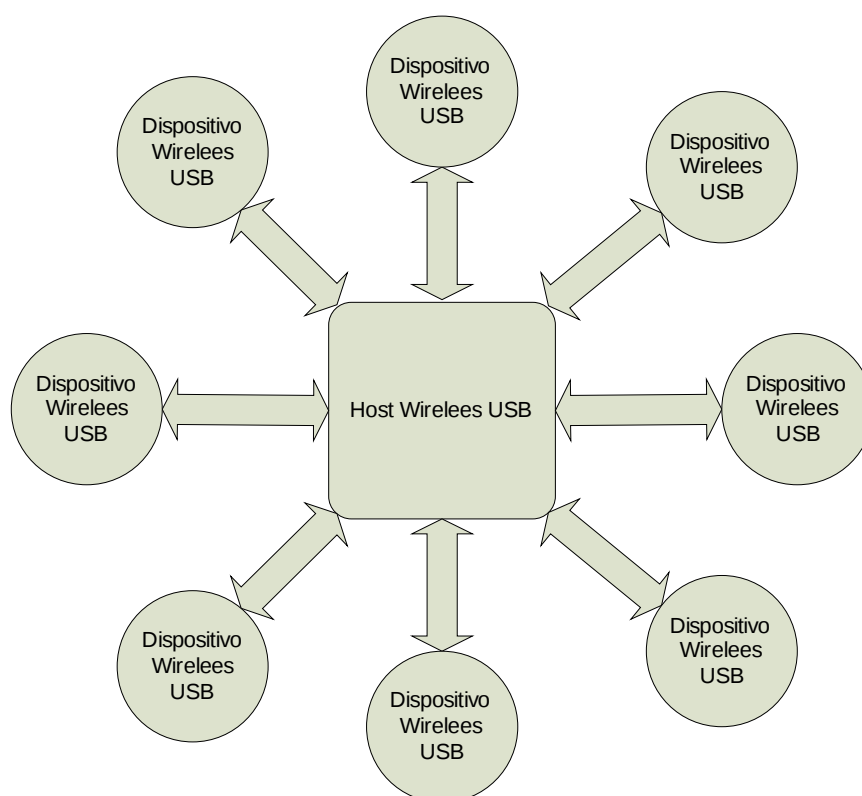
- Interconexión USB
- Dispositivos USB
- Host USB



Neste documento centrarémonos na descrición da Interconexión USB.

#### **43.1.3.1.1 TOPOLOXÍA**

Wireless USB conecta dispositivos usando un modelo de “conectarse ó concentrador e falar” (“hub and spoke”). O host é o concentrador no centro da topoloxía e cada dispositivo encontrase ó final dunha conexión punto a punto co host. Un host pode soportar ata 127 dispositivos debido a que Wireless USB non ten un interfaz físico para cada porto (conector como o de o USB) non hai necesidade de instalar ningún dispositivo para expandilos portos.



##### **43.1.3.1.1.1 USB HOST**

Solo hai un host en calquera sistema USB. A interface co ordenador host chamase Host Controller e tipicamente están conectados aos PCs usando buses internos como o PCI. O Host Controller pode estar implementado coma unha combinación de hardware, firmware e software.

Os adaptadores que usan cables e se conectan directamene ó PC usando USB coñécense como Host Wire Adapter, estes dispositivos proporcionan capacidade de Host Wireless USB ó PC.



Os adaptadores que proporcionan conexións USB pero se conectan a o host usando Wireless USB chamanse Device Wire Adapters e típicamente usan conectores USB tipo A.

Hai que ter en conta que cada un destes adaptadores crea un novo sistema USB con un host (o adaptador) e un ou varios dispositivos USB.

#### **43.1.3.1.1.2 DISPOSITIVOS WIRELESS USB**

Os dispositivos USB poden adoptar unha das seguintes formas:

- Funcións: provén capacidades ao sistema coma impresoras, cámaras dixitais, ...
- Device Wire Adapter: xa descrito mais arriba

Os dispositivos Wireless USB proporcionan unha interface estándar en termos de:

- A súa comprensión do protocolo Wireless USB
- A súa resposta a operacións USB estándar como confirmación ou reinicialización
- A súa capacidade de proporcionar información descriptiva

#### **43.1.3.1.1.3 INTERFACE FÍSICA**

##### **43.1.3.1.1.3.1 VELOCIDADES DA CAPA FÍSICA**

A capa física de Wireless USB esta descrita na especificación UWB PHY da WiMedia Alliance e soporta velocidades de transmisión de 53,3, 80, 106,7, 200, 320, 400, e 480Mb/s con múltiples canles.

PHY aporta tamén esquemas de detección e corrección de erros para prover unha canle de comunicación tan robusta como sexa posible.

As velocidades de 53,3, 106,7 e 200 son obrigatorias para os dispositivos Wireless USB o resto de velocidades son opcionais.

Os host Wireless USB están obrigados a implementar tódalas velocidades descritas.

##### **43.1.3.1.1.3.2 SOPORTE DE CANLES**

Todas as implementacións Wireless USB deben soportar as canles PHY da 9 á 15 (Band Group 1, Códigos TF 1-7) se está permitido polas regulacións



nacionais. Na versión 1.1 débense soportar Band Group 3 ou Band Group 6 (tódolos códigos TF) e no caso dos hosts tamén ás usadas na versión 1.0.

#### **43.1.3.1.1.3.3 SELECCIÓN DE CANLE**

As implementacións de Wireless USB deben soportar un canle inicial para encontrar outros dispositivos e, despois desta busca, poden moverse a outra canle.

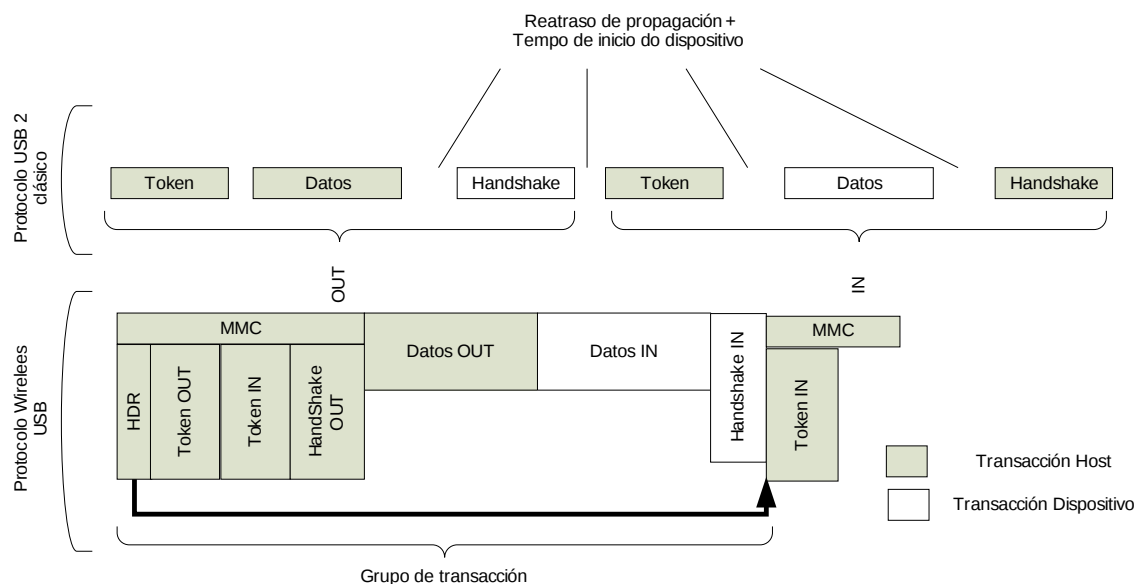
#### **43.1.3.1.1.4 XESTIÓN DA ENERXÍA**

Un host Wireless USB pode ter un sistema de xestión da enerxía independente do USB. O software do sistema USB interactuará co xestor de enerxía do host para procesar os eventos tales como a suspensión. Ademais os dispositivos USB implementarán características adicionais para a xestión da enerxía que poden ser usadas polo software do sistema.

#### **43.1.3.1.1.5 PROTOCOLO DO BUS**

Loxicamente Wireless USB é un protocolo baseado en TDMA similar ao de USB. O Host Controller inicia todas as transmisión de datos. Da mesma forma que no USB de cable, cada transferencia esta composta de 3 “paquetes”: token, datos e handshake. Sen embargo, para melloral a eficiencia da capa física eliminando transmisións constantes entre o emisor e o receptor, os hosts combinan información de múltiples tokens nun so paquete. Nese paquete o host indica o tempo apropiado para que os dispositivos escoiten un paquete OUT ou para que transmitan un paquete IN ou handshake.





Da mesma forma que en USB, o modelo de transferencia de datos entre unha orixe e un destino chamase pipe.

Wireless USB define un novo máximo tamaño de paquete para algunhas transmisións co obxectivo de mellorar o rendemento e o consumo.

#### **43.1.3.1.1.6 ROBUSTEZ**

Hai varios atributos de Wireless USB que contribúen á súa robustez:

- A capa física está deseñada para unha comunicación fiable e robusta con detección e corrección de erros.
- Detección de conexións e desconexións e configuración de recursos no sistema
- Autorecuperación no propio protocolo usando timeouts para paquetes perdidos e corruptos
- Control de fluxo metendo nun buffer e reintentando

##### **43.1.3.1.1.6.1 XESTIÓN DOS ERROS**

O protocolo permite a xestión de erros tanto en hardware como en software.

A xestión de erros en hardware incue o reporte e envío de transferencias fallidas. Un host reintentará unha transmisión na que se encontren erros ata un número limitado de veces antes de informalo software do fallo.



O software pode recuperarse dese fallo nunha forma que é dependente da implementación.

#### **43.1.3.1.1.7 SEGURIDADE**

Tódolos hosts e tódolos dispositivos Wireless USB deben soportar o nivel de seguridade definido na especificación. O mecanismo de seguridade asegúrase de que ambos, host e dispositivo, son capaces de autenticarse un ao outro (impedindo un ataque de man-in-the-middle) e de que as comunicacións son privadas.

Á seguridade básease en encriptación AES-128/CCM, as comunicacións entre o host e o dispositivo usan claves que só o host e o dispositivo posúen unha vez que están autenticados un no outro.

#### **43.1.3.1.1.8 CONFIGURACIÓN**

De igual forma que co USB, Wireless USB soporta dispositivos conectándose e desconectándose do host polo que o software do sistema debe soportar esta circunstancia.

##### **43.1.3.1.1.8.1 CONEXIÓN DE DISPOSITIVOS WIRELESS USB**

A diferenza de USB, un dispositivo Wireless Usb conectase a un host enviando unha mensaxe nun momento determinado. O host e o dispositivo auténtícanse entre eles usando os seus identificadores únicos e as claves de seguridade apropiadas.

Despois de que o dispositivo e o host se haxan autenticado e autorizado, o host asígnalle unha dirección USB única ó dispositivo e notifica ó software do sistema da conexión do dispositivo.

##### **43.1.3.1.1.8.2 DESCONEXIÓN**

A desconexión dos dispositivos pode ser realizada de forma explícita polo host ou o dispositivo usando os mecanismos definidos no protocolo. A desconexión tamén se produce se o host non pode comunicarse co dispositivo por un longo período de tempo.

##### **43.1.3.1.1.8.3 ENUMERACIÓN**

Esta actividade permite identificar e asignar direccións únicas a os dispositivos conectados ó bus lóxico. Dado que Wireless USB permite aos dispositivos conectarse ou desconectarse do bus lóxico en calquera



momento, a enumeración é unha tarefa continua do software do sistema USB. Adicionalmente a enumeración de Wireless USB permite tamén a detección e procesado das desconexións.

#### **43.1.3.1.1.9 TIPOS DE FLUXOS DE DATOS**

Wireless USB soporta os mesmos tipos de transferencia e “pipes” que USB. Debido ó seu maior nivel de erro (por razóns do medio de transmisión), o protocolo de Wireless USB define diferentes mecanismos para realizalas transmisións, estes mecanismos inclúen handshakes durante a recepción de datos e o uso de buffers para permitir unha certa confianza no pipe. A asignación de ancho de banda realízase de forma similar a USB.

#### **43.1.3.1.1.10 DISPOSITIVOS WIRELESS USB**

Da mesma forma que os dispositivos USB, os dispositivos Wireless USB están divididos en clases de dispositivos tales coma interface humana, impresoras ou dispositivos de almacenamento. Os dispositivos Wireless USB teñen que almacenar a información necesaria para a súa identificación e configuración. Tamén se require deles que mostren un comportamento consistente cos estados USB definidos.

Unha cousa importante é que os concentradores, non son dispositivos Wireless USB xa que o propio host soporta 127 dispositivos.

##### **43.1.3.1.1.10.1 CARACTERÍSTICAS DOS DISPOSITIVOS**

Da mesma forma que en USB, todos os dispositivos Wireless USB son accesibles usando unha dirección USB que é asignada cando o dispositivo se conecta e sofre o proceso de enumeración. Cada dispositivo USB soporta adicionalmente un ou varios “pipes” polos que o host pode comunicarse con el. Todos os dispositivos deben soportar un “pipe” especial ao que se conectara o “pipe” de control USB do dispositivo. Tódolos dispositivos soportan un mecanismo de acceso á información a través deste “pipe” de control. Asociado a este “pipe” está a información requirida para describir por completo o dispositivo USB.

##### **43.1.3.1.1.10.2 DISPOSITIVOS E CAPA MAC**

Os dispositivos deben implementar unha capa MAC que se comporte adecuadamente.



#### **43.1.3.1.1.11 HARDWARE E SOFTWARE DO HOST**

O host posúe unhas responsabilidade maiores que no caso de USB. O host Wireless USB debe comportarse de forma responsable con respecto á capa MAC e pode que teñan que compartir a UWB con outras aplicacións do host, por exemplo nun PC ó acceso radio pode ser compartido entre Wireless USB e a conexión de rede.

Os hosts son tamén responsables de conectarse a outros dispositivos UWB (incluíndo outros hosts Wireless USB) de forma ordenada para reducil a interferencia e mellorar o uso de ancho de banda.

Á especificación de hosts Wireless USB cubre hosts implementados coma parte de Pcs, no caso doutros dispositivos que non se conforman a este estándar (dispositivos portátiles, embebidos, ...) poden elixir implementar un subconxunto dos requisitos.

#### **43.1.4 WI-FI**

Hoxe en día existen varias tecnoloxías e estándares para as comunicacións de redes de área local sen fíos.

Estes estándares, definen unha rede formada por un medio inalámbrico compartido e transmisión encriptada da información.

IEEE 802.11 é un estándar para redes inalámbricas definido pola organización Institute of Electrical and Electronics Engineers (IEEE), instituto de investigación e desenvolvemento, de gran recoñecemento e prestixio, cuos membros pertencen a decenas de países entre profesores e profesionais das novas tecnoloxías.

O estándar IEEE 802.11 é un estándar en continua evolución, debido a que existen cantidade de grupos de investigación, traballando en paralelo para melloralo estándar, a partir das especificacións orixinais.

Na actualidade coexisten principalmente os seguintes estándares:

- 802.11b: Este estándar especifica transmisións na banda de frecuencias dos 2.4GHz, con velocidades de ata 11 Mbps.
- 802.11a: Este estándar, posterior ó 802.11b, especifica transmisións na banda dos 5 GHz (unha banda con menos ruído que a dos 2.4 GHz)



) e con unha velocidade de ata 54 Mbps. Posúe unha menor cobertura que 802.11b.

- 802.11g: Especifica transmisións de ata 54 Mbps na banda dos 2.4GHz e asegura a compatibilidade cos dispositivos 802.11b.
- 802.11n: Aprobado no 2009 especifica velocidades de transmisión de ata 600Mb/s.

A alianza WI-FI (Wireless Fidelity) é unha organización sen ánimo de lucro formada en 1999 para certificar a interoperabilidade dos produtos 802.11 e para promocionalos con un estándar global de WLAN en tódolos segmentos de mercado. Hoxe en día, existen mais de 500 produtos certificados, principalmente en 802.11b/g.

Trátase dunha especificación en continua evolución con posibilidade de adaptarse a novos requirimentos e demandas de usuario no futuro.

#### **43.1.4.1 DESCRICIÓN DO SISTEMA**

O estándar permite o uso de varios medios e técnicas para establecer conexións. Incluso o estándar orixinal permite usar infravermellos e espectro ensanchado, tanto en salto de frecuencias como secuencia directa, coa vantaxe de usar unha capa de acceso ó medio (MAC) común. Isto proporciona moita flexibilidade aos desenvolvedores e investigadores, que poden esquecerse de certos aspectos xa que non existe dependencia directa entre eles.

Os estándares de IEEE 802.11 son de libre distribución e calquera persoa pode ir á páxina Web do IEEE e descargarlos. Estes estándares só definen especificacións para as capas físicas e de acceso ao medio e para nada tratan modos ou tecnoloxías a usar para a implementación final.

Esto debe permitir e facilitar a interoperabilidade entre fabricantes de dispositivos IEEE 802.11 e para asegurarse diso creouse unha alianza denominada WECA para crear e definir procedementos para conseguir certificados de interoperabilidade e de cumprilas especificación, todo dentro dun estándar chamado WiFi (Wireless Fidelity). O nome ademais é un indicativo do enfoque doméstico e moi enfocado cara o usuario final.



O bloque constructivo básico dunha rede inalámbrica 802.11 é o denominado conxunto de servizo básico (BSS, Basic Service Set), que é un área xeográfica na que as estacións sen fíos se poden comunicar.

O tipo mais sinxelo de BSS consiste en dous ou mais equipos que entran dentro das áreas de transmisión respectivas. Este proceso polo que os dispositivos entran nun BSS denomínase asociación.

#### **43.1.4.1.1 CAPA FÍSICA (PHY)**

A capa física en calquera rede define a modulación e características de sinalización para a transmisión de datos nese medio. Para poder transmitir en redes sen fíos en bandas sen licenza necesítanse usar técnicas de espectro ensanchado, definidas nos requirimentos de case tódolos países. No estándar IEEE 802.11 defínense tres medios de nivel físico. Un usa sinais de infravermellos e os outros dous utilizan sinais de radio frecuencia (RF).

Os medios de RF 802.11 funcionan na banda de 2.4Ghz, con un ancho de banda de 83Mhz entre 2.400 e 2.483 GHz, aínda que en España tan só temos 23Mhz, como Francia e Xapón. Ademais hai definicións de potencia máxima de transmisión definidas polos distintos organismos de regulación. En EEUU defínese unha potencia máxima de 1W, para Europa 10mW cada 1 Mhz e para Xapón 10mW.

As definicións para a transmisión por radiofrecuencia nos estándares son de espectro ensanchado por salto en frecuencias (FHSS) e espectro ensanchado por secuencia directa (DSSS). Ambos están definidos para traballar na banda de 2.4Ghz, e DSSS ademais ten unha variante na banda dos 5Ghz, que consegue maiores velocidades de transmisión.

##### **43.1.4.1.1.1 INFRAVERMELLOS**

As comunicación por infravermellos utilizan frecuencias entre 850 e 950 nanómetros, xusto por debaixo do espectro da luz visible. A implementación IEEE 802.11 de infravermellos, a diferenza da maioría dos medios infravermellos, non require comunicación de visión directa, pode funcionar mediante sinais reflexadas.



Sen embargo, debido o seu limitado alcance comparado cos medios de RF e a que só pode funcionar adecuadamente nun ambiente interior cando as superficies proporcionan unha boa reflexión dos sinais, é raro que se implemente nas redes sen fíos. Ademais impón mais restriccións na ubicación física do dispositivo que FHSS ou DSSS.

#### **43.1.4.1.1.2 FHSS**

FHSS (Frequency-Hopping Spread Spectrum) refírese a un sistema que periodicamente cambia as frecuencias nas que transmite. Utilízase a banda enteira o que contribúe a aumentala seguridade fronte a escoitas á vez que axuda a suprimir o ruído ou as interferencias.

FHSS ten 22 patróns de saltos predefinidos usando 79 canles de 1Mhz a un mínimo de 2.5 saltos por segundo, e para resolver os problemas de sincronización, para que tanto transmisor como receptor salten á vez, defínense paquetes de sincronización.

A velocidade dos cambios de frecuencia é independente da velocidade de bit de transmisión de datos. Se a velocidade do salto de frecuencia é menor que a velocidade de bit do sinal, a tecnoloxía denomínase sistema de salto lento, e se é maior denomínase sistema de salto rápido.

Para a modulación FHSS usa FSK gaussiano de 2 ou 4 niveis. As velocidades típicas conseguidas son de 1 e 2 Mbps para FHSS.

#### **43.1.4.1.1.3 DSSS**

DSSS (Direct Sequence Spread Spectrum) traballa nun canle fixo e preconfigurado, o que lle permite obter maiores taxas de transferencia, pero coa desvantaxe de ser mais sensible a interferencia e a sinais procedentes de outros dispositivos usando a mesma frecuencia. É posible ter tres puntos de acceso con tres canles diferentes, sen solapar en un mesmo emprazamento e sen ter en conta ningún tipo de planificación. Aínda para mais de tres puntos de acceso é necesaria certa planificación, para poder mantelas velocidades, posto que o solape de celdas e frecuencias terá un deterioro sobre o rendemento.

As modulacións usadas para DSSS son BPSK e DQPSK para o estándar orixinal. Para 11b, que permite conseguir 11Mbps, utilízase CCK.



Ademais definiuse una variante de IEEE 802.11, que permite conseguir 54Mbps na banda de 5Ghz, con un ancho de banda de ata 300MHz e usando una modulación OFDM.

Esta mesma modulación é a usada por 802.11g e 802.11n.

#### **43.1.4.1.1.4 TRAMAS DA CAPA FÍSICA**

En lugar de ter un esquema de sinalización relativamente sinxelo coma en Ethernet e Token Ring que utilizan Manchester e Manchester diferencial respectivamente, os medios que funcionan en 802.11 teñen o seu propio formato de tramas, que encapsulan as tramas xeradas no nivel de enlace de datos.

A trama de FHSS contén os seguintes campos:

- Preámbulo (10 bytes): contén 80 bits de 1 e 0 alternos utilizados polo receptor para detectalo sinal e sincronizalos tempos.
- Delimitador de comezo de trama (SFD) (2 bytes): indica o comezo da trama.
- Lonxitude (12 bits): indica o tamaño do campo de datos.
- Sinalización (4 bits): contén un bit para indicar se se está utilizando a velocidade de 1 ou 2 Mbps. Os outros 3 bits resérvanse para uso futuro. Só o campo de datos se pode transmitir a 2 Mbps.
- CRC (2 bytes): contén un valor de comprobación de redundancia cíclica.
- Datos (de 0 a 4.095 bytes): contén a trama do nivel de enlace de datos que se transmite.

A trama DSSS contén os seguintes campos:

- Preámbulo (16 bytes): contén 128 bits que o sistema receptor utiliza para axustarse á sinal entrante.
- Delimitador de comezo de trama (SFD) (2 bytes): indica o comezo da trama.
- Sinal (1 byte): especifica a velocidade de transmisión.
- Servizo (1 byte): contén o valor hexadecimal 00 que indica que o sistema cumpre co estándar 802.11



- Lonxitude (2 bytes): indica o tamaño do campo de datos.
- CRC (2 bytes): contén un valor de comprobación de redundancia cíclica.
- Datos (variable): contén a trama do nivel de enlace de datos que se transmite.

A trama de infravermellos contén os seguintes campos:

- Delimitador de comezo de trama (SFD) (2 ranuras): indica o comezo da trama.
- Velocidade de datos (3 ranuras): especifica a velocidade de transmisión.
- Axuste do nivel de DC (DCLA) (32 ranuras): utilizado polo receptor para estabilizar o nivel DC despois de transmitir os campos precedentes.
- Lonxitude (12 bits): indica o tamaño do campo de datos.
- CRC (2 bytes): contén un valor de comprobación de redundancia cíclica.
- Sincronización (SYNC) (entre 57 y 73 ranuras): utilizadas polo sistema receptor para sincronizalo tempo e opcionalmente para estimar a relación sinal/ruído.
- Datos (de 0 a 2.500 bytes): contén a trama do nivel de enlace de datos que se transmite.

#### **43.1.4.1.2 CAPA DE ACCESO Ó MEDIO (MAC)**

A especificación da capa MAC do IEEE 802.11 ten moitas similitudes co estándar de Ethernet cableado (IEEE 802.3). O protocolo do 802.11 é un esquema de protocolo coñecido como detección de portadora, acceso múltiple, evitando colisións (CSMA/CA). Este protocolo evita as colisións, en vez de detectalas como o algoritmo de 802.3 (CSMA/CD). É extremadamente difícil detectar colisións nunha rede de transmisión de radiofrecuencias e de ahí que se trate de evitalas colisións.

A capa MAC opera xunto coa capa física muestreando a enerxía do medio de transmisión de datos. O protocolo CSMA/CA permite opcións para que se



poda minimizalas colisións usando tramas de transmisión RTS/CTS (Request-to-send/Clear-to-send), datos e recoñecementos dunha maneira secuencial. Nestas tramas sóense incorporar datos de duración dos envíos co obxectivo de asegurar que eses envíos non van a ser interrompidos: os demais nodos saben que deben estar calados durante ese intervalo de tempo. Todo elo ademais se asegura e confirma con tramas de recoñecemento (ACK).

Pero un problema común a calquera WLAN é o problema dos nodos ocultos. Isto pode chegar a reducir as prestacións nun 40% nunha WLAN con alta carga. Producecse cando un nodo non pode escoitar transmisións dun nodo e trata de transmitir a un nodo que si pode escoitalos, alí se pode xerar moitas colisións.

Algunhas melloras incluíronse para evitalo problema co uso de RTS/CTS dunha maneira intelixente.

Ademais utilízanse tempos entre tramas para evitar colisións isto, a parte de evitar colisións, permite ademais certo uso de clases de calidade ou polo menos de preferencia dun tráfico sobre outro, utilizando funcións de coordinación puntual e de permitilo acceso ó medio de tráfico prioritario antes que aos demais.

#### **43.1.4.1.2.1 TRAMAS DO NIVEL MAC**

O estándar 802.11 define tres tipos básicos de tramas neste nivel:

- Tramas de datos: úsanse para transmitir datos dos niveis superiores entre estacións.
- Tramas de administración: úsanse para o intercambio de información para realizar funcións de rede como a autenticación e a asociación.
- Tramas de control: úsanse para regular o acceso ó medio e para recoñecemento das tramas de datos transmitidas.

Unha trama MAC xenérica contén os seguintes campos:

- Control da trama (2 bytes): contén 11 subcampos que habilitan as distintas funcións do protocolo:



- Versión do protocolo (2 bits): especifica a versión do estándar que se está a usar.
- Tipo (2 bits): indica se a trama é de administración (00), control (01) ou datos (00).
- Subtipo (4 bits): identifica a función específica da trama.
- A DS (1 bit): un valor de 1 indica que a trama transmítese ó sistema de distribución a través dun punto de acceso.
- Dende DS (1 bit): un valor de 1 indica que a trama recibíuse dun sistema de distribución.
- Mais fragmentos (1 bit): un valor de 1 indica que o paquete contén un fragmento dunha trama e que hai mais fragmentos para a súa transmisión.
- Reintento (1 bit): un valor de 1 indica que a trama se está retransmitindo debido a unha falta de recepción dun ack.
- Administración de enerxía (1 bit): un valor de 0 indica que a estación está funcionando en modo activo; un valor de 1 en modo aforro de enerxía.
- Mais datos (1 bit): se vale 1 indica que o AP ten mais paquetes almacenados para a estación e en espera de transmisión.
- WEP (1 bit): se vale 1 indica que o corpo da trama cifrouse utilizando WEP.
- Orde (1 bit): se vale 1 indica que a trama de datos se está transmitindo utilizando a clase de servizo estritamente ordenado.
- Duración/AID (2 bytes): nas tramas de control de sondeo de enerxía contén a identidade de asociación (AID) da estación transmisora. No resto de tramas contén o tempo (en microsegundos) necesario para transmitir unha trama máis o intervalo entre tramas.
- Dirección 1 (6 bytes): contén unha dirección que identifica ó receptor da trama, dependendo dos valores dos subcampos A DS e Dende DS.
- Dirección 2 (6 bytes): contén unha dirección, dependendo dos valores dos subcampos A DS e Dende DS.



- Dirección 3 (6 bytes): contén unha dirección, dependendo dos valores dos subcampos A DS e Dende DS.
- Control de secuencia (2 bytes): contén dous subcampos:
  - Número de fragmento (4 bits): contén un valor que identifica un fragmento particular nunha secuencia.
  - Número de secuencia (12 bits): contén un valor que identifica os fragmentos da secuencia que compoñen o conxunto de datos.
- Dirección 4 (6 bytes): contén unha dirección, dependendo dos valores dos subcampos A DS e Dende DS.
- Corpo da trama (0 a 2.312 bytes): contén a información que se está transmitindo á estación receptora.
- Secuencia de verificación de trama (4 bytes): contén un valor CRC.

Os cinco tipos de dirección do subnivel MAC son:

- Dirección do emisor (TA): unha dirección MAC individual que identifica ó sistema que transmitiu a información que vai no corpo da trama no medio sen fíos actual (un AP).
- Dirección do receptor (RA): unha dirección MAC individual ou de grupo que identifica ó receptor inmediato da información no corpo da trama no medio inalámbrico actual (un AP).
- Dirección destino (DA): unha dirección MAC individual ou de grupo que identifica ó receptor final dunha unidade de datos de servizo.
- Dirección orixe (SA): unha dirección MAC individual que identifica ó sistema que xenerou a información que vai no corpo da trama.
- ID do conxunto de servizo básico (BSSID): nunha rede ad hoc o BSSID é un valor xenerado aleatoriamente durante a creación do BSS; nunha rede con infraestrutura é a dirección MAC da estación que funciona como AP do BSS.

## **43.2 RFID**

RFID (Radio-frequency identification) é unha tecnoloxía que usa a comunicación mediante ondas de radio para transferir datos entre un lector



e unha etiqueta electrónica adherida a un obxecto co propósito de identificación ou seguimento.

Unha etiqueta pasiva de RFID (unha sen alimentación propia) pódese ler cun lector RFID se se aproxima suficientemente.

#### **43.2.1 PRINCIPIOS DE RFID**

Existen moitos tipos de RFID, pero no maior nivel de abstracción podemos dividilas en dous clases: activo e pasivo.

As etiquetas activas requiren enerxía de unha fonte, están ou conectadas á rede eléctrica ou a unhas baterías. No caso de usar baterías a vida da etiqueta está limitada pola duración destas contra o número de lecturas que sufrirá o dispositivo. Un exemplo de etiqueta activa é o transpondedor dun avión que identifica á súa nación de procedencia.

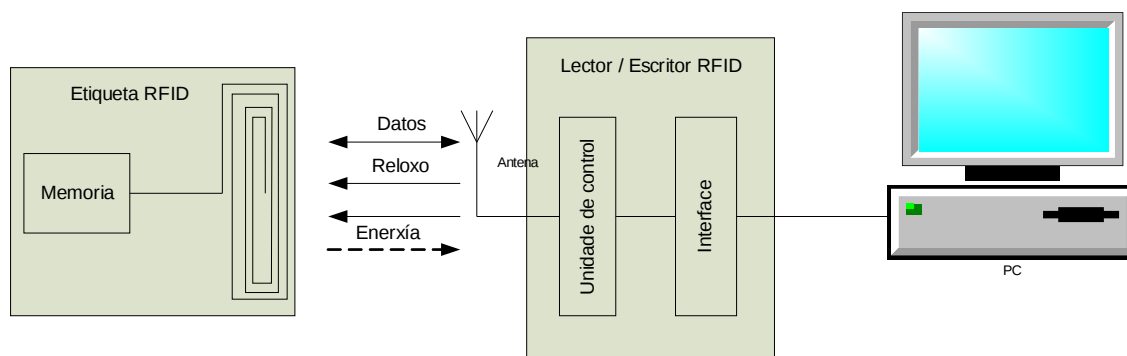
Sen embargo as baterías fan que o custo, tamaño e duración das etiquetas activas sexa pouco práctica. As etiquetas RFID pasivas son interesantes porque non necesitan mantemento, teñen un período de vida indeterminado e son suficientemente pequenas como para caber nunha etiqueta adhesiva.

Unha etiqueta pasiva esta composta por 3 elementos: unha antena, un chip conectado á antena e algún tipo de encapsulación. O lector de etiquetas é o responsable de aportala enerxía e comunicarse coa etiqueta para ler o seu ID (o chip da etiqueta coordina este proceso). A encapsulación mantén integridade física da etiqueta e protexe a antena e o chip das condicións ambientais. Esta encapsulación pode ser un cristal ou unha lamina de plástico con adhesivo por unha das caras para permitir a súa adherencia a una superficie.

Existen dúas aproximacións fundamentais distintas para transmitir enerxía á etiqueta dende o lector: inducción magnética e captura de onda electromagnética (EM). Estes dous deseños toman coma vantaxe as propiedades EM asociadas cunha antena cunha potencia típica de desde 10 microvatios ata 1 milivatio (podemos comparalo co consumo dun procesador Intel XScale que é de 500 milivatios ou co consumo dun procesador Intel Pentium 4 de 50 vatios).



### 43.2.2 COMPOÑENTES E OPERACIÓN



Un sistema RFID ten os seguintes compoñentes:

- Unha etiqueta RFID que almacena certa información (ás mais típicas son de 2KB pero hainas de moitos tamaños)
- Un lector RFID que emite nunha determinada frecuencia e é capaz de detectar a resposta da etiqueta
- Un equipo capaz de interpretar a información da etiqueta

A operación do sistema é moi sinxela:

- O lector RFID emite sinais de radio a unha determinada frecuencia co obxectivo de activar a etiqueta RFID e ler ou escribir nela
- Cando unha etiqueta RFID pasa polo rango de acción do lector RFID, este detecta o sinal da etiqueta
- O lector comunícase coa etiqueta (co propósito concreto para o que se crease ese sistema, identificación do produto, pago sen fíos, ...)

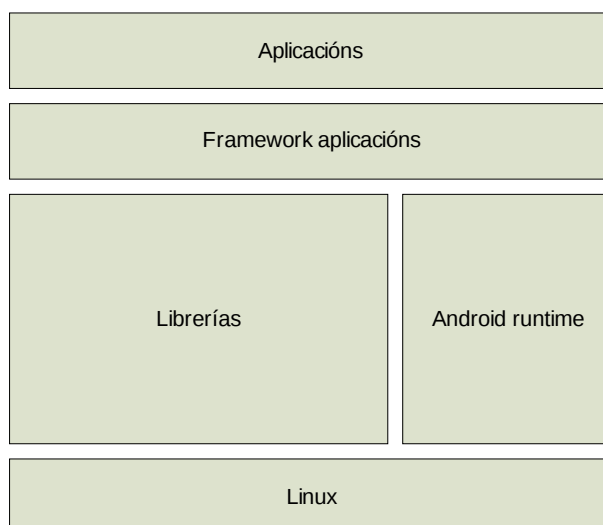
## 43.3 TECNOLOXÍAS MÓBILES

### 43.3.1 ANDROID

Android é unha pila de software para dispositivos móbiles que inclúe o sistema operativo baseado en Linux, un middleware e aplicacións clave sendo a plataforma para móbiles mais vendida.

Google comprou a compañía que comezou o desenvolvemento do produto en 2005 e xunto con outros membros da Open Handset Alliance colaborou no seu desenvolvemento e publicación e na actualidade AOSP (Android Open Source Project) leva o seu mantemento e desenvolvemento futuro. A grandes trazos a estrutura de Android pode verse na figura seguinte.





- Aplicacións: Android permite desenvolver aplicacións e interfaces de usuario específicas para produtos específicos
- Framework aplicacións: Permite estender a funcionalidade por enriba da de un dispositivo de man dando soporte ás aplicacións e permitindo definir clases específicas para unha industria ou produto.
- Librerías: Optimizadas para un hardware determinado.
- Android runtime: Optimización da VM Dalvik de Java para distintas CPU's e SoC.
- Linux: Sistema operativo base preparado para distintas CPU's e chipsets.

### **43.3.2 MEEGO**

MeeGo é un sistema operativo baseado en Linux e orientado a dispositivos móbiles. Aínda que está principalmente orientado a dispositivos móbiles e appliances no mercado de electrónica de consumo MeeGo esta deseñado para actuar como sistema operativo para outras plataformas hardware coma netbooks, tablets, televisions, ... Na actualidade MeeGo está amparada pola Linux Foundation.

### **43.3.3 SYMBIAN**

Symbian é outro sistema operativo e plataforma computacional para smartphones mantido por Nokia. A plataforma Symbian é a sucesora de Symbian OS do Nokia Series 60, a diferenza de Symbian OS, que requiría



unha interface de usuario adicional Symbian inclúe compoñentes de interface de usuario baseados na 5ª edición de S60. A última versión de Symbian (Symbian ^3) lanzouse a finais do 2010 co Nokia N8. Symbian so está preparado para executarse en procesadores ARM aínda que existe unha versión para procesadores x86 que non foi lanzada. Os dispositivos Symbian inclúen un 29,2% dos smartphones.

#### **43.3.4 WINDOWS PHONE 7**

Windows Phone 7 (anteriormente Windows Phone 7 Series) é un sistema operativo mobil desenvolto por Microsoft e é o sucesor da súa Windows Mobile platform. A diferenza do seu predecesor esta orientado ó mercado de consumo en vez de ó mercado empresarial.

Con Windows Phone 7 Microsoft ofrece un novo interface de usuario coa súa nova linguaxe Metro, integra o sistema operativo con servizos de terceiras partes e da propia Microsoft e controla o hardware no que se executa.

#### **43.4 BIBLIOGRAFÍA**

- Wireless Universal Serial Bus Specification 1.1 (2010)
- GAST, Matthew S. 802.11 Wireless Networks: The Definitive Guide. O'Reilly & Associates; 1st edition (2002)
- GEIER, James T. y Geier, Jim. Wireless LANs (2nd Edition). Sams; 2nd edition (2001)
- FLICKENGER, Rob. Building Wireless Community Networks. O'Reilly & Associates; 1st edition (2001)
- Roy Want. An Introduction to RFID Technology

**Autor:** Matías Villanueva Sampayo

Director de Informática Asociación Provincial de Pensionistas y Jubilados de A Coruña

Colegiado del CPEIG





**XUNTA  
DE GALICIA**



*ESCOLA GALEGA  
DE ADMINISTRACIÓN  
PÚBLICA*

# **INFRAESTRUTURA DE SISTEMAS.**





# **44. CONCEPTO, EVOLUCIÓN E TENDENCIAS DOS SISTEMAS OPERATIVOS. SISTEMA OPERATIVO UNIX-LINUX. SISTEMA OPERATIVO WINDOWS.**



## Tema 44. Concepto, evolución e tendencias dos sistemas operativos. Sistema operativo UNIX-LINUX. Sistema operativo Windows

---

### ÍNDICE

<b>44.1.- Sistema Operativo.....</b>	<b>2</b>
44.1.1 <i>Concepto.....</i>	2
44.1.2 <i>Evolución dos Sistemas Operativos.....</i>	4
44.1.3 <i>Clasificacións dos Sistemas Operativos.....</i>	7
44.1.4 <i>Estrutura dos Sistemas Operativos.....</i>	9
44.1.5 <i>Funcións dun Sistema Operativo.....</i>	10
44.1.6 <i>Tendencias.....</i>	11
<b>44.2.- Sistema Operativo UNIX-LINUX.....</b>	<b>13</b>
44.2.1 <i>Características.....</i>	14
44.2.2 <i>Arquitectura de Unix-Linux.....</i>	16
44.2.3 <i>Xestión de procesos.....</i>	17
44.2.4 <i>Xestión de memoria.....</i>	18
44.2.5 <i>Xestión de E/S.....</i>	20
44.2.6 <i>Xestión de arquivos.....</i>	22
<b>44.3.- Sistema Operativo Windows.....</b>	<b>24</b>
44.3.1 <i>Características.....</i>	24
44.3.2 <i>Arquitectura de Windows.....</i>	26
44.3.3 <i>Xestión de procesos.....</i>	27
44.3.4 <i>Xestión de memoria.....</i>	29
44.3.5 <i>Xestión de E/S.....</i>	30
44.3.6 <i>Xestión de arquivos.....</i>	31
44.3.7 <i>Seguridade.....</i>	35
<b>44.4.- Bibliografía.....</b>	<b>36</b>



## **44.1.- SISTEMA OPERATIVO**

### **44.1.1 Concepto**

Pódese definir un sistema operativo (SO) como un conxunto de programas que controlan directamente os recursos hardware (HW) ou físicos dun ordenador (CPU, memoria principal e periféricos) proporcionando unha máquina virtual que oculta os detalles físicos da máquina e lle ofrece á persoa un contorno máis amigable. O sistema operativo é a capa de software máis baixa dun ordenador. Cada capa ocúltalles ás capas superiores certos detalles das capas inferiores. Desta forma constrúese o software, baseándonos no que xa existe.

O sistema operativo ten unha serie de funcións, que se poden agrupar en 3:

1. Inicializar a máquina. Prepara a máquina para o funcionamento. Hai 2 formas de inicialización:

- a) TOTAL: Inicialización de todas as funcións e servizos que a máquina pode ofrecer. Por exemplo, MS-DOS ten inicialización total.
- b) PARCIALMENTE: Vaise ser selectivo cos tipos de servizos que se inician. Por exemplo, Linux e Windows.

A principal vantaxe/utilidade da inicialización parcial é a recuperación da máquina ante fallos: consiste en que se falla un servizo da máquina non fai falta apagala toda; só hai que lanzar de novo o servizo.

2. Servir de máquina virtual. Ocúltanse detalles de hardware proporcionando un contorno máis amigable. Isto ten 2 obxectivos:

- a) A SEGURIDADE: En lugar de que o usuario acceda directamente a un recurso HW, faino o SO para que non se produzan operacións non





desexadas, tamén chamadas operacións perigosas: operacións de entrada/saída (E/S), operacións de acceso á memoria.

O HW ten dúas formas de actuar: modo supervisor e modo usuario.

Todos os programas actuarán en modo usuario ata o momento en que haxa que acceder ao HW; será daquela cando cambie a modo supervisor para evitar as operacións que poidan causar problemas. Ao realizarse unha destas operacións susceptibles de fallo xérase unha interrupción. Esa interrupción cóllea o sistema operativo e actúa en consecuencia. O sistema operativo tomará o control do hardware e realizará a operación que se lle indica.

Unha interrupción é un sinal físico que xeran os dispositivos do sistema e que é tratada polo sistema operativo.

Ao conxunto de interrupcións chámase **interface interna** do sistema operativo.

- b) ABSTRACCIÓN: Abstráense as características físicas e reais da máquina ofrecendo unha serie de servizos incluso maiores dos que pode ofrecer a propia máquina. Por exemplo, para traballar con ficheiros utilízanse nomes, pero o ordenador non utiliza eses nomes para se referir a eles; emprega un enderezo.

A E/S: cando tecleamos un dato vémosto en pantalla tal como o imaxinamos aínda que realmente para a máquina son uns e ceros.

Isto constitúe a **interface externa** do SO, a linguaxe coa que nos imos comunicar con el. Denomínase SHELL.

Xa que logo, temos 2 tipos de interface:

- A externa: forma de comunicación entre nós e o ordenador a través de comandos/ordes (abstracción).



- A interna: forma de comunicación do sistema operativo co hardware (modo supervisor).

3. Administrar os recursos para o seu funcionamento. Esta administración ten que cumprir 3 características:

- ten que ser CORRECTA: se hai 2 procesos que queren acceder a un recurso, hai que darlle acceso primeiro a un e logo a outro, pero non mesturalos.
- ten que ser XUSTA: se temos 2 procesos hai que darlles saída aos dous; un proceso non pode monopolizar.
- ten que ser EFICIENTE: para mellorar o rendemento do sistema.

Para rematar, recordemos que un SO ten que ter estas 2 características:

- DETERMINISMO: se se repite a mesma operación cos mesmos datos de entrada, debemos obter os mesmos resultados.
- INDETERMINISMO: no sentido de que ten que responder oportunamente ás interrupcións; é dicir, non sabe que interrupción vai chegar primeiro, nin sabe en que orde, pero debe saber tratalas.

#### **44.1.2 Evolución dos Sistemas Operativos**

Os sistemas operativos evolucionaron en paralelo ao desenvolvemento do HW. Conforme o HW ía incorporando novas capacidades, os SO debían adaptarse para poder xestionar eficientemente esas novas capacidades.

A evolución dos SO pódese organizar en xeracións con algunhas características comúns:

##### **Primeira Xeración**

Abrangue desde 1945 a 1955. Caracterízase porque non existía un sistema operativo. Eran os propios deseñadores das máquinas os que as programaban a través de cableado e os que as manexaban.

##### **Segunda Xeración**



Desde 1955 a 1965. No campo do hardware aparece o transistor. Empézanse a utilizar as tarxetas perforadas. Distínguense 2 tipos de sistema operativo:

- Monitor Residente: o SO limitábase a cargar os programas na memoria, léndoos de tarxetas perforadas, e a executalos. O problema era atopar unha forma de optimizar o tempo entre a retirada dun traballo e a montaxe do seguinte.
- Traballo por lotes: utilízanse as fitas magnéticas. Como solución para optimizar o tempo de montaxe xurdiu a idea de agrupar os traballos en lotes, nunha mesma fita ou conxunto de tarxetas, de forma que se executasen un a continuación doutro sen perder apenas tempo na transición. Na memoria do ordenador existen dous ítems: a) o monitor de lotes: indicando qué traballo se está a executar e b) o traballo actual.

Estes dous tipos de sistema operativo caracterízanse por:

- Non existe ningún planificador (o que decide qué traballo se vai realizar): a razón da súa inexistencia é que non é necesario, xa que só hai un traballo na memoria principal.
- Tampouco existe un reloxo (mide o tempo que un traballo está ocupando a CPU): non é necesario porque só hai un proceso en memoria executándose.

### **Terceira Xeración**

Abarca desde 1965 a 1980. No campo do hardware temos os circuítos integrados con tecnoloxía LSI e VLSI. Antes de ver os distintos SO, paga a pena deixar claros dous conceptos: Programa: código (algo estático) e Proceso: programa en execución (algo dinámico).

- SO Multiprogramación. Na memoria principal vai haber máis dun programa. A CPU executa instrucións dun programa; cando o que se





atopa en execución realiza unha operación de E/S, en lugar de esperar a que remate a operación de E/S, pásase a executar outro programa. Desta forma é posible, tendo almacenado un conxunto adecuado de tarefas en cada momento, utilizar de maneira óptima os recursos dispoñibles.

Vaise definir o grao de multiprogramación como o número de programas que hai actualmente en memoria.

OverHead é un parámetro que mide a diferenza de tempo que hai entre o tempo que o sistema operativo está dedicado a facer as súas tarefas e o tempo dedicado ao cambio de contexto entre procesos.

- SO Multiproceso. Ten varios procesos en memoria principal. Hai que distinguir que:
  - o *Multiprogramación IMPLICA Multiproceso.* Multiprogramación: varios programas na memoria principal. Ao estaren na memoria principal estanse a executar. Programas en execución = procesos.
  - o *Multiproceso NON IMPLICA Multiprogramación.* Agora ben, podemos ter unicamente un programa en memoria e que este programa lance varios procesos. Un programa pode querer imprimir, ler algo por teclado... lanzar procesos. Varios procesos = multiproceso.
- SO Multiprocesador. Utilízanse onde hai 2 CPU ou máis.
- SO Interactivos. Sistemas que dalgunha forma manteñen un diálogo co usuario mediante o SHELL (linguaxe de comandos).

É moi importante o tempo de resposta: tempo que transcorre dende que o usuario lle manda facer algo ao sistema ata que obtén a resposta. Sempre se tenta minimizar o tempo de resposta.
- SO Multiusuario. Aqueles sistemas operativos nos cales varios usuarios poden acceder ao mesmo ordenador simultaneamente.





Por exemplo: calquera sistema UNIX ou LINUX. Pódese ter unha máquina e xerar usuarios accedendo a esa máquina (a través de rede local, internet...).

- SO de Tempo Compartido. Pretenden dotar a cada persoa dunha parte da CPU. O usuario ve o ordenador como seu propio, aínda que non o é.
- SO de Tempo real. Estes sistemas úsanse en contornos onde se deben aceptar e procesar en tempos moi definidos un gran número de sucesos, na súa maioría externos ao ordenador. Se o sistema non respecta as restricións de tempo no que as operacións deben entregar o seu resultado dise que fallou.

### **Cuarta Xeración**

Abarca desde 1980 ata os nosos días; está marcada polos ordenadores persoais.

Sistema Operativo de Rede. O usuario é consciente de que existen outras máquinas. O usuario xa non quere traballar só; quere traballar con outros usuarios. Ten que acceder de forma explícita a esas máquinas.

Sistema Operativo Distribuído. O parámetro clave é a transparencia: o usuario non é consciente de que existen outras máquinas; non sabe en que máquina está.

#### **44.1.3 Clasificacións dos Sistemas Operativos**

- Segundo o número de usuarios:
  - o SO monousuarios: só aceptan un usuario nun momento determinado.
  - o SO multiusuarios: aceptan simultaneamente máis dun usuario.
- Segundo o hardware:
  - o Segundo o número de CPU:
    - SO monoprocesador: só controla unha CPU.





- SO multiprocesador: varios procesadores (máis complicado de deseñar).
- o Segundo a organización da memoria principal:
  - SO centralizados: unha memoria principal; os procesadores van estar intentando acceder a esta memoria principal. Os procesos comunícanse a través da memoria.
  - SO distribuídos: cada procesador ten a súa propia memoria principal. Os procesos teñen outros mecanismos de comunicación.
- Segundo o modo de traballo cos sistemas operativos:
  - o Interactivo (on-line): o usuario dialoga coa máquina.
  - o Batch (off-line): non hai comunicación coa máquina cando está realizando o traballo.
- Segundo o obxectivo para o que foron deseñados:
  - o SO de propósito xeral: capaces de realizar calquera tarefa.
  - o SO de propósito específico: só poden realizar unha tarefa específica; instálanse en microprocesadores que controlan o funcionamento de electrodomésticos, vehículos, equipos de electrónica consumo, etc.
  - o SO de Tempo Real: ofrecen unha resposta nun intervalo de tempo ben definido.
  - o SO virtuais: operan sobre o HW dun ordenador ofrecéndolles aos niveis superiores copias exactas da máquina real, de forma que en cada copia se pode executar un sistema operativo distinto.
  - o SO de dispositivos móbiles: deben adaptarse ás limitacións que estes dispositivos presentan: procesadores lentos, memoria limitada, pantallas pequenas e consumo de enerxía limitado. Exemplos típicos para estes dispositivos son iOS de Apple, Windows Mobile, Android, etc.



#### **44.1.4 Estrutura dos Sistemas Operativos**

##### Sistemas operativos monolíticos

Estes SO non teñen unha estrutura definida. O SO escríbese como unha colección de procedementos entrelazados de tal forma que cada un pode chamar a calquera outro. As características deste tipo de estrutura son:

- Construción do SO baseado en procedementos compilados separadamente que se unen nun só ficheiro obxecto a través do enlazador (linker).
- Boa definición de parámetros de enlace entre os distintos procedementos existentes, o que xera acoplamento.
- Carece de protección ao entrar a procedementos que xestionan diferentes aspectos dos recursos do ordenador, como almacenamento, E/S, etc.
- Son feitos á medida, o que ten como vantaxe que son eficientes e rápidos, e como desvantaxe que carecen de flexibilidade para crecer.

##### Sistemas operativos con capas

O SO organízase nunha xerarquía de estratos, estando construído cada un deles sobre o outro que ten menor xerarquía ca el. Exemplos: THE, MULTICS.

##### Sistemas operativos Cliente-Servidor

Minimizar o kernel (núcleo) do SO, desprazando o código de todos os seus servizos a estratos o máis superiores posibles. Para iso, a maioría das súas funcións impleméntanse como procesos de usuario, denominados procesos servidores, de xeito que cando un proceso de usuario, chamado proceso cliente, necesita un servizo do SO, o que fai é enviarlle unha mensaxe ao proceso servidor correspondente, que realiza o traballo e devolve a resposta.



#### **44.1.5 Funcións dun Sistema Operativo**

As principais funcións que teñen os SO son a xestión de procesos, a xestión da memoria principal, a xestión do almacenamento secundario e a xestión dos dispositivos de entrada/saída.

##### **Xestión de procesos**

A CPU é o recurso principal do ordenador, de modo que é necesaria unha xestión eficiente da mesma para garantir o seu aproveitamento.

O SO ten que cargar os distintos procesos, inicialos, supervisar a súa execución levando a cabo os cambios de contexto necesarios e detectar a súa terminación normal ou anormal. Nos contornos multiusuario é fundamental a activación de mecanismos de protección que limiten as posibilidades de acceso de cada proceso a unha serie de recursos para os que conte coa debida autorización.

##### **Xestión de memoria principal**

Nun sistema multiproceso os procesos teñen que compartir a CPU tendo que atoparse na memoria principal para poder pasar a executarse inmediatamente; así, varios procesos teñen que compartir a memoria principal sen que uns poidan acceder aos recursos doutros.

Para iso hai que dividir a memoria en bloques, e estes asígnanselles a distintos procesos. Para facer a división utilízase a segmentación, a paxinación ou a segmentación paxinada.

##### **Xestión dos sistemas de arquivos**

Nun sistema de arquivos, o SO tense que facer cargo de: a xestión do espazo libre/ocupado, dos cachés de lectura e escritura, do vínculo entre nomes e arquivos, das asociacións entre os bloques físicos dos dispositivos e os bloques lóxicos, dos permisos para o acceso e modificación dos distintos elementos.

##### **Xestión de entrada/saída (E/S)**



A velocidade con que se comunican o procesador e a memoria principal contrasta coa velocidade cando os programas deben interactuar con algún tipo de dispositivo de E/S; durante este proceso a execución do programa vese interrompida, xa que a comunicación cos devanditos dispositivos é significativamente máis lenta que coa memoria.

Conxuntamente coa multiprogramación xorden dous conceptos: o acceso directo á memoria (DMA) e as interrupcións. O procesador cédelle o control da E/S a un módulo que se encarga de executar este tipo de operacións (o controlador de DMA), de agardar ata que estas se completen e, cando isto sucede, de avisar o procesador (que se atopa namentres executando outras instrucións —ben sexa do mesmo proceso ou dalgún outro—) que pode continuar coas operacións subseguintes que quedaron pendentes cando se realizou a petición de E/S mediante unha interrupción.

#### **44.1.6 Tendencias**

Actualmente, a mobilidade é o primordial na nosa sociedade e asociada a ela está a seguridade. Empézase a falar de sistemas operativos na nube (*cloud computing*) e consolídanse os sistemas operativos dos dispositivos móbiles.

Outro aspecto que cómpre destacar, derivado do momento económico, é o aforro de custos. Froito del pódese ver unha tendencia máis que clara cara a virtualización.

#### **SO en dispositivos móbiles**

Se existe unha carreira hoxe en día no desenvolvemento de SO, encóntrase nos SO para dispositivos móbiles. Os novos sistemas operativos converten o teléfono nun completo aparello multimedia. Ata hai moi pouco tempo, a elección dun móbil viña determinada polas súas características físicas: recepción do sinal, cámara... pero coa chegada dos smartphone, a elección do SO converteuse en algo moi importante.



A diferenza do mundo do ordenador persoal, e debido quizais á súa xuventude no mercado, non existe un dominador claro de SO móbiles. Hai fabricantes hardware que son os fabricantes dos seus propios sistemas operativos, como por exemplo, Apple, que só distribúe o SO iOS para iPhones e iPad; o mesmo que RIM, que distribúe o seu SO BlackBerry OS en dispositivos BlackBerry, etc. No outro extremo están os fabricantes que utilizan SO doutras compañías, como Android, Symbian, Windows Mobile...

As compañías que fabrican e distribúen o seu propio SO teñen a favor que as actualizacións dos dispositivos son moi controladas.

Outra tendencia ha vir da man da difusión dos dispositivos *tablets* (tabletas), se temos en conta os anuncios de lanzamentos de tabletas: Apple (iPad 2), RIM (PlayBook), Samsung (Galaxy Tab 2), etc. Isto halle dar aínda máis auxe aos SO móbiles.

### **SO na nube ou en rede**

Estes SO xorden do concepto de Computación na Nube (*Cloud Computing*), que é un novo paradigma que, basicamente, permite ter servizos computacionais a través de internet.

Unha das grandes vantaxes que se lle poden atopar a este novo paradigma é o baixo investimento que hai que realizar en HW, xa que toda a infraestrutura da computación na nube se encontra nos grandes provedores de servizos de internet. Abondaría cun hardware mínimo, un navegador e unha boa conexión a internet.

Outra das vantaxes sería que as aplicacións non se instalan no pc; son aplicacións Web, o que fai que sexan compatibles coa maioría dos formatos coñecidos.

Permite ter unha única copia dun ficheiro dispoñible en calquera lugar e momento.

Os seus puntos débiles son a seguridade e a necesidade dunha conexión a internet.



Estes SO son unha boa opción para os notebook que teñen pouco hardware, e mesmo poderían conseguir que os fabricantes apostasen por modelos máis baratos, o que permitiría difundir moito máis a informática.

Entre os SO máis importantes destacan: eyeOS, ChromeOS de Google, oOS, iCloud, etc.

#### **44.2.- SISTEMA OPERATIVO UNIX-LINUX**

UNIX é un sistema operativo creado en 1969 por un grupo de investigadores dos laboratorios Bell de AT&T —entre eles Ken Thompson, Dennis Ritchie e Douglas McIlroy— como unha versión reducida do proxecto MULTICS; primeiro foi escrito en ensamblador, pero isto impedía a portabilidade a diferentes ordenadores. Despois de que en 1973 Dennis Ritchie crease a linguaxe C, reescríbese UNIX totalmente nesta linguaxe de alto nivel, facendo así o código case totalmente independente do tipo de máquina e permitindo a instalación de UNIX en diferentes plataformas.

Inicialmente, os laboratorios AT&T Bell, consideraron que UNIX era máis ben un proxecto de investigación e chegarono a distribuír de forma gratuíta entre os departamentos informáticos das universidades, os cales podían modificalo e adaptar ás súas necesidades. Pero a gran demanda do sistema operativo fai que os laboratorios Bell inicien a súa venda a través de distribucións oficiais, concedéndolles aos usuarios que o requiren licenzas de uso.

Debido ás múltiples versións no mercado de UNIX, o IEEE especificou unha familia de estándares para definir unha interface de programación de aplicacións (API) para que todas as versións fosen 'compatibles'. Esta familia coñécese como POSIX (*Portable Operating System Interface*; o X vén de UNIX, como marca de identidade da API).

Linux creouse en 1991 por Linus Torvalds, que se baseou noutros dous sistemas operativos:

- O sistema aberto UNIX.



- O sistema educativo Minix, creado en 1987 por Andrew S. Tanenbaum.

Torvalds crea só o kernel, o núcleo do sistema, sen a capa de servizos, xestores, aplicacións gráficas, etc., que serán creados posteriormente por outros autores. O código do núcleo podémolo atopar no enderezo ([www.kernel.org](http://www.kernel.org)).

Na comunidade de programadores créase o proxecto GNU (*Gnu's Not Unix*), proxecto para xerar software libre, onde se xeran editores, compiladores, etc. baixo a licenza pública xeral GPL (*General Public License*): usar, copiar, distribuír e modificar sempre que se conserve a sinatura do autor, podendo cobrar por iso.

Linux créase con esta filosofía de libre distribución, e o sistema operativo completo que se constrúe con este núcleo tamén. A todo o sistema dáselle o nome de GNU/Linux (distribución completa do sistema operativo con Linux), que contén o núcleo máis as outras capas do sistema operativo e utilidades. Aínda que moitas veces se denomina a todo o sistema simplemente LINUX.

#### **44.2.1 Características**

As características máis relevantes do sistema UNIX son:

- UNIX foi deseñado como un sistema multiusuario en tempo compartido, ofrecendo protección dos datos privados sobre ficheiros e protección do contorno de execución.
- Portabilidade: UNIX foi escrito na linguaxe C, unha linguaxe de alto nivel, o cal fai que sexa relativamente doado de ler, entender, modificar e transportar a outras máquinas cunha arquitectura física diferente.
- Código e funcionamento escrito baixo a familia de estándares POSIX (*Portable Operating System Interface*).





- Interface de usuario simple e interactiva: o intérprete de ordes (*shell*) é un programa independente que o usuario pode substituír. A sintaxe de utilización é idéntica para todas as ordes.
- Modularidade: Proporciona primitivas que permiten construír grandes programas a partir doutros máis sinxelos, así como librerías para *linkage*.
- Posúe bibliotecas compartidas para facilitar o enlace dinámico.
- Protección de memoria.
- Soporta diferentes sistemas de arquivos, incluídos os de Microsoft Windows.
- Sistema de arquivos con estrutura de árbore invertida (de múltiples niveis, que permite un fácil mantemento) e xerárquica (permite a unión de diversos sistemas de ficheiros co sistema principal, e unha separación de directorios).
- Todos os arquivos de usuario son simples secuencias de bytes (8 bits ); non teñen ningún formato predeterminado.
- Independencia de dispositivos: Os discos e os dispositivos de entrada e saída (E/S) trátanse todos do mesmo xeito: como meros arquivos. As peculiaridades dos dispositivos mantéñense no núcleo (kernel).
- A arquitectura da máquina é completamente transparente para o usuario, o que permite que os programas sexan fáciles de escribir e transportar a outras máquinas con hardware diferente.
- UNIX non incorpora deseños sofisticados; de feito, a maioría dos algoritmos foron seleccionados polo súa sinxeleza e non pola súa rapidez ou complexidade.
- Incorpora todos os servizos de rede, TCP/IP, DNS, *sendmail*, etc.



- Proporciona un completo contorno de programación: os filtros son utilidades simples que se concentran en realizar ben unha soa función. Pódense combinar de forma moi flexible utilizando as *pipes* (tubaxes) e as redireccións de E/S segundo as necesidades e preferencias de cada usuario.
- Mantemento fácil: consecuencia directa da modularidade. O sistema segue evolucionando e perfecciónase e enriquecese con novas funcionalidades.
- Carácter aberto: permite ampliar facilmente a funcionalidade con novos compoñentes sen ter que depender dun único fabricante.

#### **44.2.2 Arquitectura de Unix-Linux**

A arquitectura está baseada en capas ou niveis, de forma que cada capa unicamente pode comunicarse coas capas que se atopan nos niveis inmediatamente inferior e superior.

Na capa inferior temos toda a parte do Hardware que o sistema operativo debe xestionar. Por riba deste sitúase o kernel de Unix, que é o encargado da administración de procesos, xestión do sistema de arquivos, entradas/saídas, etc. Aos procesos que traballan a ese nivel chámaselles procesos en modo kernel.

A biblioteca estándar sitúase por riba do kernel; encárgase, por exemplo, das operacións de apertura, peche, etc. A este nivel trabállase en modo usuario. A interface entre as dúas capas, ou o acceso da capa de biblioteca estándar á do kernel, realízase a través da interface de chamadas ao sistema.

A un nivel superior temos os programas e utilidades como o *shell*, compiladores, etc., que lles serven de axuda a desenvolvedores e usuarios que interactúan co sistema operativo. A interface entre esta capa e a inmediatamente inferior é a través da interface de biblioteca.



Por último, situaríanse os usuarios que, por medio da interface de usuario, se comunican co *shell*, ou outras utilidades do sistema Unix.

O núcleo de UNIX (kernel) é de tipo monolítico, diferenciándose dúas partes principais: o núcleo dependente da máquina e o núcleo independente. O núcleo dependente encárgase das interrupcións, os dispositivos de baixo nivel e parte da administración da memoria. O núcleo independente é igual en todas as plataformas e inclúe a xestión de chamadas do sistema, a planificación de procesos, a paxinación e intercambio, a xestión de discos e o sistema de arquivos.

#### **44.2.3 Xestión de procesos**

A xestión de procesos en UNIX é por prioridade e *round robin*. Nalgunhas versións xestiónase tamén un axuste dinámico da prioridade de acordo ao tempo que os procesos esperaron e ao tempo que xa usaron a CPU. O sistema dispón de facilidades para contabilizar o uso de CPU por proceso e unha pila común para todos os procesos cando necesitan estarse executando en modo privilexiado (cando fixeron unha chamada ao sistema ).

Os procesos traballan en modo usuario e en modo kernel. O paso de modo usuario a kernel ou viceversa realízase a través de *traps* que crean unha interrupción para acceder á interface de chamadas ao sistema e ao resto dos compoñentes de nivel kernel. O paso do modo kernel a usuario é un retorno tras a realización da petición que motivou o paso ao modo kernel.

UNIX permite que un proceso faga unha copia de si mesmo por medio da chamada «fork», o cal é moi útil cando se realizan traballos paralelos ou concorrentes; tamén se provén facilidades para o envío de mensaxes entre procesos (*pipes, signals*).

Os procesos non interactivos denomínanse *daemons* ou procesos background. Cando se inicia un proceso, asígnaselle un identificador PID, gárdase o proceso que o lanzou PPID, o propietario que o lanzou UID e o grupo de pertenza GID, o que definirá o perfil de permisos de acceso aos



que terá dereito. Existe a posibilidade de alterar o usuario ou grupo efectivo de permisos durante a execución do proceso mediante a chamada a **setuid** ou **setgid**, sempre que se dispoña dos permisos apropiados. Tamén existe unha bandeira de permisos **setuid** asociada ao arquivo do programa que permite executar este, cos permisos do propietario do arquivo en lugar dos do usuario que o executa.

LINUX combina multiprogramación e tempo compartido.

O xestor de procesos no kernel do sistema UNIX encárgase da asignación de CPU, a programación de procesos e as solicitudes dos procesos. Para realizar estas tarefas, o kernel mantén varias táboas importantes para coordinar a execución destes procesos e a asignación dos dispositivos.

Utilizando unha política predefinida, o programador de procesos selecciona un proceso da cola de procesos listos e comeza a súa execución durante un intervalo de tempo xa dado.

O algoritmo de programación de procesos selecciona o proceso con maior prioridade para ser executado primeiro. Se varios procesos teñen a mesma prioridade, aplícase o algoritmo *round-robin*.

#### **44.2.4 Xestión de memoria**

Os sistemas UNIX utilizan o manexo de memoria virtual sendo o esquema máis usado a paxinación por demanda e combinación de segmentos paxinados, en ambos os casos con páxinas de tamaño fixo.

En todos os sistemas UNIX úsase unha partición de disco duro para a área de intercambio (*swap*). Esa área resérvase durante a instalación do sistema operativo.

Unha regra moi difundida entre administradores de sistemas é asignar unha partición de disco duro que sexa polo menos o dobre da cantidade de memoria real do ordenador. Con esta regra permítese que se poidan intercambiar flexiblemente todos os procesos que estean na memoria RAM nun momento dado por outros que estean no disco.



Se non caben todos os programas na memoria principal faise uso da partición de intercambio (*swapping*).

- Swap out. Cando non caben en memoria procesos activos, “expúlsase” un proceso de memoria principal, copiando a súa imaxe a swap, aínda que non é necesario copiar todo o mapa. Existen diversos criterios de selección do proceso que se intercambia: dependendo da prioridade do proceso; preferencia aos procesos bloqueados; non intercambiar se está activo DMA sobre mapa do proceso.
- Swap in. Cando haxa espazo na memoria principal, intercámbiase o proceso a memoria copiando a imaxe desde swap.

Todos os procesos que forman parte do kernel non poden ser intercambiados a disco.

Cada proceso dispón do seu propio espazo de enderezos, organizado en segmentos segundo:

- Text Segment: que almacena o código.
- Data Segment: que almacena os datos ou variables que utilizan os procesos; este segmento ten dúas partes: *Initialized Data* (datos inicializados) e *Uninitialized Data* (datos non inicializados).
- Stack Segment: que almacena a información referente a chamadas a outras funcións.

É posible compartir código entre procesos mediante o emprego de *Shared Text Segments*. Dous procesos nunca comparten os segmentos de datos e de pila (salvo os *thread*); a forma de compartir información lévase a cabo mediante o emprego de segmentos especiais de memoria compartida, *Shared Segments*.

Linux comparte moitas características dos esquemas de xestión de memoria doutras implementacións UNIX, pero ten características de seu.



No que respecta á memoria virtual, o direccionamento de memoria virtual de Linux fai uso dunha estrutura de táboa de páxinas con tres niveis, formada polos seguintes tipos de táboas (cada táboa individual é do tamaño dunha páxina): Directorio de páxinas: un proceso activo ten un só directorio de páxinas, que é do tamaño dunha páxina. Cada entrada no directorio de páxinas apunta a unha páxina do directorio intermedio de páxinas. Para un proceso activo, o directorio de páxinas ten que estar na memoria principal. Directorio intermedio de páxinas: este directorio pode ocupar varias páxinas e cada entrada deste directorio apunta a unha páxina da táboa de páxinas. Táboa de páxinas: esta táboa de páxinas tamén pode ocupar varias páxinas, e cada entrada da táboa de páxina fai referencia a unha táboa virtual do proceso.

Para utilizar esta estrutura da táboa de páxinas a tres niveis, un enderezo virtual en Linux vese como un conxunto de catro campos. O campo máis á esquerda (máis significativo) utilízase como índice no directorio de páxinas. O seguinte campo serve como índice no directorio intermedio de páxinas. O terceiro campo serve como índice na táboa de páxinas. E o cuarto e último campo indica o desprazamento dentro da páxina seleccionada da memoria.

#### **44.2.5 Xestión de E/S**

Os dispositivos de entrada e saída son considerados ficheiros especiais. Toda entrada/saída está baseada no principio de que todos os dispositivos se poden tratar como ficheiros simples aos que se accede mediante descritores de arquivos cuxos nomes se atopan polo xeral no directorio «/dev».

Cada proceso en UNIX mantén unha táboa de arquivos abertos (onde o arquivo pode ser calquera dispositivo de entrada/saída). Esa táboa ten entradas que corresponden aos descritores, os cales son números enteiros obtidos por medio da chamada do sistema.

As chamadas ao xestor de entrada/saída fanse de dúas formas: síncrona e asíncrona. O modo síncrono é o modo normal de traballo e consiste en



facen peticións de lectura ou escritura, e o proceso espera a que o sistema lle responda.

O xestor de entrada/saída utiliza como elementos principais o buffer de cache; o código xeral de xestión de dispositivos, e drivers de dispositivos de hardware. Existen dous tipos de dispositivos:

Dispositivos de bloques:

- Usan secuencias de bytes (bloques).
- Utilizan buffer-cache.
- Están estruturados en bloques de tamaño fixo (512 bytes).
- Permiten optimizar o rendemento.

Dispositivos de carácter:

- Son dispositivos sen estrutura (terminais, impresoras, etc).
- Non usan buffer.
- As operacións realízanse carácter a carácter.

## **Interrupcións e excepcións**

UNIX permite interromper a CPU asincronamente. Ao recibir a interrupción, o kernel almacena o contexto actual, determina a causa e responde á interrupción. Tras responder a esta, devolve o contexto interrompido e segue executando. O HW asígnalles as prioridades aos dispositivos de acordo coa orde de actuación nas interrupcións.

Así como as interrupcións están causadas por factores externos a un proceso, as excepcións son sucesos inesperados producidos por procesos —tales coma a execución de instrucións reservadas— de forma que o sistema, ao se atopar con unha, tende a reiniciar a instrución, en lugar de pasar á seguinte.



Non obstante, o kernel debe ter a posibilidade de impedir a aparición de interrupcións en momentos críticos para evitar a degradación dos datos. O sistema que se utiliza é o de dispoñer dun conxunto de instrucións restrinxidas que colocan o nivel de execución do procesador no estado de palabra (*status word*). Ao asignar un nivel de execución do procesado, todas as interrupcións dese nivel e inferiores quedan suprimidas, permitíndose só as superiores.

#### **44.2.6 Xestión de arquivos**

Un sistema de arquivos permite realizar unha abstracción dos dispositivos físicos de almacenamento da información para que sexan tratados a nivel lóxico, como unha estrutura de máis alto nivel e máis sinxela que a estrutura da súa arquitectura hardware particular.

O sistema de arquivos UNIX caracterízase porque posúe unha estrutura xerárquica, realiza un tratamento consistente dos datos dos arquivos, protexe os datos dos arquivos e trata os dispositivos e periféricos (terminais, unidades de disco, fita, etc.) coma se fosen arquivos.

O sistema de arquivos está organizado, a nivel lóxico, en forma de árbore invertida, cun nodo principal coñecido como nodo raíz ("/"). Cada nodo dentro da árbore é un directorio e pode conter pola súa vez outros nodos (subdirectorios), arquivos normais ou arquivos de dispositivo.

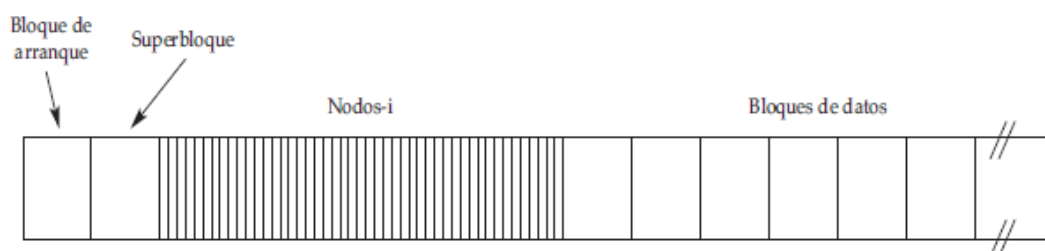
Os nomes dos arquivos (*pathname*) especifícanse mediante a ruta (*path*), que describe como localizar un arquivo dentro da xerarquía do sistema. A ruta dun arquivo pode ser absoluta (referida ao nodo raíz) ou relativa (referida ao directorio de traballo actual).

Todos os sistemas UNIX poden manexar múltiples particións de disco, cada unha cun sistema de arquivos distinto.

Unha partición de disco clásica en UNIX contén unha estrutura coma a da figura.



- Bloque de arranque: contén o código para arrincar o ordenador con esa partición.
- Superbloque: Contén información crucial acerca da organización do sistema de arquivos, incluído o número de nodos-i (*i-nodes*), o número de bloques de disco e o principio da lista de bloques de disco libres. A destrución do superbloque fai que o sistema de arquivos xa non se poida ler.
- Nodos-i: contén a representación interna dun ficheiro, que permite entre outras cousas localizar todos os bloques de disco que conteñen os datos do arquivo.
- Bloques de Datos: almacenan a información. O normal é que un arquivo ocupe máis dun bloque de datos, sen que sexa necesario que estean contiguos.



Dentro da estrutura de directorios de UNIX – Linux existen unha serie de directorios comúns a todas as instalacións que é preciso coñecer:

- /: Directorio raíz, inicio do sistema de arquivos.
- /tmp: Directorio de arquivos temporais.
- /dev : Directorio de dispositivos. Nel atópanse todos os dispositivos de E/S, que se tratan como arquivos especiais.
- /etc: Directorio para arquivos de sistema diversos.
- /bin: Directorio para programas binarios (executables).



- /lib: Directorio de bibliotecas do sistema.
- /usr: Directorio de usuarios.
- /home: Directorio base a partir do cal se sitúan os directorios por defecto das contas de usuario.

LINUX comezou empregando o sistema de arquivos de MINIX, pero estaba limitado a nomes de 14 caracteres e a arquivos de 64 MB de tamaño. A primeira mellora veu da man dun sistema de arquivos denominado Ext que permitía nomes de arquivo de 255 caracteres e 2 GB de tamaño por arquivo, pero era moi lento. A evolución produciuse grazas ao sistema de arquivos Ext2, con nomes de arquivo longos, arquivos grandes e mellor rendemento. Ext2 evolucionou a Ext3, que trouxo principalmente as transaccións (*journaling*) a Ext2. En ext3 almacénase a información necesaria para restablecer os datos afectados pola transacción no caso de que esta falle. A evolución de ext3 é **ext4**, que soporta un tamaño máximo de sistema de arquivos de 1 ExaByte (1 ExaByte = 1024 PetaBytes = 1048576 TeraBytes) e un tamaño máximo por arquivo de 16 TB para os arquivos e, doutra banda, modifica estruturas de datos importantes, como a destinada a almacenar os datos do arquivo utilizando “**extent**”, que é un conxunto de bloques físicos contiguos que mellora o rendemento ao traballar con ficheiros de gran tamaño e reduce a fragmentación.

### **44.3.- SISTEMA OPERATIVO WINDOWS**

#### **44.3.1 Características**

A familia de SO Windows é propiedade de Microsoft. Poderíase dicir que, en esencia, existen dúas versións distintas de SO Windows: aqueles enfocados ao mundo empresarial e aqueles enfocados ao consumidor final ou usuario doméstico.

Dentro desta simple clasificación poderíamos facer outras. Orientados ao usuario doméstico, existen SO para equipos de sobremesa, portátiles,



tablets e dispositivos móbiles. E orientados ao mundo empresarial podemos distinguir SO para servidores e SO para estacións de traballo. Aínda poderíamos dividir máis os SO para servidores (Standard, Enterprise, Datacenter, Web, Storage, Small Business Server)...

Os últimos SO que viron a luz son Windows 7 coas versións Starter (só para 32 bits), Home Basic, Home Premium, Professional, Ultimate, Enterprise. E respecto a SO para servidores, Windows Server 2008 R2.

É salientable o interese por mellorar que mostrou Microsoft cos SO de servidor, comprometéndose a un cambio cada 5 anos (antes do 2008 estaba o 2003) e a unha revisión cada 2 anos aprox. (por iso existe Windows Server 2008 R2).

Windows Server 2008 é o último SO de servidor que inclúe estas melloras:

- Novo proceso de reparación de sistemas NTFS: proceso en segundo plano que repara os arquivos danados.
- Creación de sesións de usuario en paralelo: reduce tempos de espera nos Terminal Services e na creación de sesións de usuario a grande escala.
- Peché limpo de Servizos.
- Sistema de arquivos SMB2: de 30 a 40 veces máis rápido o acceso aos servidores multimedia.
- Address Space Load Randomization (ASLR): protección contra malware na carga de controladores en memoria.
- Windows Hardware Error Architecture (WHEA): protocolo mellorado e estandarizado de informe de erros.
- Virtualización de Windows Server: melloras no rendemento da virtualización.
- PowerShell: inclusión dunha consola mellorada con soporte GUI para administración.



- **Server Core:** o núcleo do sistema renovouse con moitas e novas melloras.

Nos seguintes apartados imos ver características de Windows, e aínda que as máis son comúns para todas as versións, as explicacións estarán máis centradas en Windows 2008 Server R2.

#### **44.3.2 Arquitectura de Windows**

A estrutura modular de Windows 2008 proporciona unha gran flexibilidade. O seu deseño permítelle executarse en múltiples plataformas hardware. Nesta estrutura modular distínguense dúas capas principais:

- **Modo usuario:** Os seus programas e subsistemas están limitados aos recursos do sistema aos que teñen acceso. Está formado por subsistemas que poden pasar peticións de E/S aos controladores apropiados do modo núcleo a través do xestor de E/S.
- **Modo núcleo ou kernel:** Ten acceso total ao hardware da máquina, impedíndolles aos servizos do modo usuario e ás aplicacións accederen ao hardware, que queda totalmente protexido polo sistema operativo. A arquitectura dentro do modo núcleo componse do seguinte:
  - o O micronúcleo: situado entre a capa de abstracción de hardware e o Executive, proporciona a xestión *multiprocesador*: xestión de procesos, fíos e tratamento de interrupcións e de excepcións.
  - o Unha capa de abstracción de hardware (en inglés, *Hardware Abstraction Layer* ou HAL): encárgase de ocultar as diferenzas de hardware e, xa que logo, proporciona unha plataforma única onde se poida executar o SO independentemente do HW.
  - o Controladores, tamén chamados drivers: utilizados para interactuar cos dispositivos hardware.



o Executive: sobre o cal son implementados todos os servizos de alto nivel. Relaciónase con todos os subsistemas do modo usuario. Ocúpase da entrada/saída, a xestión de memoria, Plug&Play, a seguridade e a xestión de procesos.

Dado que o enlace estático dos programas de usuario coas bibliotecas da API Win32 implicaría un tamaño enorme nos programas e un desperdicio de memoria, pois cada programa en execución tería a súa copia destas bibliotecas, todas as versións de Windows manexan bibliotecas compartidas, chamadas bibliotecas de vínculos dinámicos (DLLs; *Dinamic Link Libraries*).

#### **44.3.3 Xestión de procesos**

Os procesos créanse como obxectos, e un proceso pode ter varios fíos. Dado que o proceso é un obxecto, a súa composición será un conxunto de datos só accesibles a través dun conxunto de funcións que os ocultan do resto de aplicacións ou funcións. Estas funcións ou servizos actívanse por medio de mensaxes. O proceso terá polo menos un fío, que, á vez, pode executar outros fíos, podendo facelo en paralelo nun sistema multiprocesador.

Windows mantén dúas listas diferentes coa información de todos os procesos e fíos. Cada proceso ten asociado un bloque ou estrutura de datos EPROCESS, que apunta ao EPROCESS seguinte e ao anterior (dobre lista enlazada); a outra estrutura é ETHREAD para recoller a información dos fíos.

O algoritmo de planificación está baseado en colas de retroalimentación de múltiples niveis con prioridades. Cada cola xestiónase cunha política Round Robin.

A planificación aplícase sobre os fíos, non aos procesos (sen ter en conta a qué proceso pertencen os distintos fíos que se executan), e está baseada



en prioridades; é dicir, sempre se executará o fío de maior prioridade da cola de fíos preparados.

Cando se selecciona un fío para a súa execución, concédeselle un *quantum*, ou intervalo de tempo durante o cal se lle permite ao fío executarse antes de que o faga outro fío do mesmo nivel de prioridade. Os valores do quantum poden variar.

Aínda que se lle conceda un quantum a un fío, este podería non consumilo completamente porque apareza no sistema un novo proceso de maior prioridade, que obrigaría ao que se está executando a abandonar o procesador.

O sistema trata igual a todos os fíos que teñan a mesma prioridade, asignándolle a cada fío de maior prioridade un intervalo de tempo de procesador cun Round Robin. Se ningún destes fíos estivese preparado para executarse, pasarían a executarse, seguindo a mesma regra, os da prioridade inmediatamente inferior.

En cada un destes casos, Windows debe determinar qué fío debe executarse a continuación, e esta decisión é o que se coñece como *dispatcher*.

Cada proceso recibe unha prioridade base para todos os seus fíos. O sistema baséase en 32 prioridades, do 0 (menor prioridade) ao 31 (maior prioridade):

- A prioridade 0 está reservada para o fío de sistema responsable de poñer a cero as páxinas libres cando non as necesiten ningún fío.
- As prioridades 1 a 15 son as reservadas para os procesos de usuario (prioridades variables).
- As prioridades 16 a 31 estánlle reservadas ao sistema operativo (prioridades en tempo real).



O planificador funciona accedendo á táboa polo proceso de prioridade 31 e vendo se ten fíos listos para executar. Se os hai, toma o primeiro da lista e execútao durante un quantum. Namentres existan procesos preparados dunha prioridade superior, o sistema ha concederlles todo o tempo que precisen. Este comportamento repítese para cada unha das entradas da táboa de prioridades.

En certas condicións, un subprocesso pode ver incrementada a súa prioridade base, pero nunca por riba da prioridade 15 e nunca para subprocessos de prioridade maior de 15. Se unha operación de E/S libera un subprocesso, este ve incrementada a súa prioridade base de modo que se poida executar axiña.

Tamén se produce aumento de prioridade se o subprocesso estaba a agardar por un semáforo, mutex ou outro suceso. Estas elevacións de prioridade van diminuindo a medida que un subprocesso beneficiado vai consumindo por completo o seu quantum, ata volver situarse na súa prioridade base.

#### **44.3.4 Xestión de memoria**

A xestión de memoria en Windows é de memoria virtual con paxinación.

As aplicacións de 32 bits teñen un espazo de enderezo do proceso de 4 GB de memoria. Os sistemas operativos de Microsoft Windows proporcionanlles ás aplicacións acceso a 2 GB de espazo de enderezo do proceso, especificamente coñecido como espazo de enderezos virtuais do modo de usuario. Todos os subprocessos pertencentes a unha aplicación comparten o mesmo espazo de enderezos virtuais do modo de usuario. Os 2 GB restantes resérvanse para o sistema operativo (tamén coñecido como espazo de enderezo do modo de kernel).

O espazo de enderezos virtual paxínase por demanda con tamaño fixo de páxinas (mínimo 4KB para arquitecturas x86 e x64bits e 8KB para arquitecturas IA64, e máximo de 4MB para arquitecturas x86, 2MB para x64 e 16MB para IA64).



Windows utiliza un algoritmo de paxinación **por demanda anticipada**; é dicir, cada vez que se produce un fallo de páxina, o sistema copiará en memoria a páxina correspondente á referencia á memoria que causou o fallo de páxina e ademais un conxunto de páxinas próximas a ela, tanto anteriores como posteriores, ao supoñer que, debido á localidade das referencias, é case seguro que nun futuro próximo tamén se fará referencia a estas páxinas, que cando se queiran utilizar xa estarán en memoria e, polo tanto, non se producirán fallos de páxinas adicionais.

O mecanismo de paxinación apóiase moito no concepto de **Conxunto de Traballo (Working Set)** que asegura unha certa cantidade de memoria física para cada proceso.

Windows préstalle especial atención ao momento de arranque dos procesos, xa que, como non teñen ningunha páxina cargada na memoria, ata que carguen todas as páxinas necesarias han producirse moitos fallos de páxina. Para optimizar a carga dos procesos, Windows conta co que se coñece como “Prefetcher”, que ten como misión acelerar o proceso de carga.

Se se produce un fallo de páxina e é necesario substituír algún marco de páxina que está en memoria, Windows emprega o algoritmo LRU (aínda que algunhas versións utilizan tamén FIFO).

Permite compartir páxinas, ao poder protexelas contra lectura ou escritura. Igualmente admite que se poida bloquear unha páxina en memoria que sexa crítica, impedindo que se poida substituír ante unha falta de páxina, facilitando así a implementación de aplicacións en tempo real.

#### **44.3.5 Xestión de E/S**

O sistema de entrada/saída (E/S) de Windows é o que permite utilizar os dispositivos facilitando o acceso a estes e independizando os programas



dos dispositivos, ofrecendo ademais seguridade no seu uso e a escalabilidade do sistema.

As entradas e saídas en Windows poden ser síncronas (o proceso agardará ata que se complete a operación no dispositivo hardware) ou asíncronas (o proceso lanza a operación e segue coa súa execución; cando a operación E/S finaliza, o SO avísao).

En Windows cárganse e descárganse os drivers en calquera momento, evitando que consuman recursos se non se van utilizar.

Isto faise grazas ao Plug and Play (PnP), que permite detectar calquera dispositivo que se conecte ao sistema e cargar o driver correspondente.

O sistema de E/S componse dos seguintes módulos:

- O xestor de E/S: define a infraestrutura que soporta os drivers de dispositivos. Forma parte do sistema operativo.
- O driver de dispositivo: proporciona un interface de E/S para un determinado tipo de dispositivo. Os drivers reciben peticións canalizadas a través do xestor de E/S, diríxenas ao dispositivo concreto e informan o xestor de que se completou a operación de E/S. Estes módulos desenvólvenos os fabricantes.
- O xestor de PnP: detecta os dispositivos hardware ao conectarse ou desconectarse.
- O xestor de enerxía: facilítalle ao sistema, así como aos drivers de dispositivo, os cambios de estado de consumo de enerxía eléctrica de acordo coa actividade do dispositivo.

#### **44.3.6 Xestión de arquivos**

En Windows, a asignación do espazo realízase o subsistema de ficheiros en unidades “cluster” cuxo tamaño depende da capacidade do disco; normalmente oscila desde 512 bytes ata 4 Kbytes. Utiliza 64 bits para direccionar os cluster e permite definir ficheiros de 264TB (16.384



petabytes), aínda que, lxicamente, o tamaño máximo dos ficheiros está limitado pola capacidade dos discos.

Windows xestiona os discos e a información que conteñen sobre a base de particións e volumes.

- **Particións.** Cada disco pódese dividir en particións primarias e estendidas. As particións primarias serán aquelas que poidan conter un SO e, polo tanto, permitan o arranque do SO desde elas. De aquí podemos deducir que o sistema require como mínimo unha partición primaria nalgún disco.

Unha partición nunca poderá exceder dun disco. Só pode haber unha partición estendida por disco, e como máximo só poderá conter 4 particións en total.

Unha vez creada a partición, é necesario darlle formato para que poida conter datos. Un volume é sinónimo de partición formateada.

En Windows, as particións xestiónaas o xestor de particións. Este utiliza o xestor de E/S para identificar as particións e crear os dispositivos que as representen, é dicir, as unidades lóxicas correspondentes.

Este xestor envíalle un comando ao xestor de volumes (descrito máis adiante) para saber se a partición ten un volume asociado, e se ten tal, a partir dese momento calquera acción sobre a partición ha notificarlla ao xestor de volumes.

- **Volume.** Un volume desde o punto de vista do usuario é unha partición formateada. As particións primarias só poderán conter un volume, mentres que as estendidas poderán albergar varios, tendo en conta que un sistema só poderá ter como máximo 24 volumes, xa que se identifican por medio das letras do abecedario, que en inglés só ten 24 letras.

En Windows Server podemos traballar con dous tipos de discos:





- Discos básicos. Son os que se basean exclusivamente en táboas de particións MBR (*Master Boot Record*) ou táboas GPT (*GUID Partition Table*).
- Discos dinámicos. Baséanse en volumes *dinámicos* que permiten a creación de volumes de particións múltiples tales coma simples, distribuídos, espellos, *stripes* e RAID-5. Os discos dinámicos particiónanse co Administrador de discos lóxicos (*LDM - Logical Disk Manager*).

Windows traballa cos seguintes tipos de volumes dinámicos:

- Volumes distribuídos (*spanned*). É un único volume lóxico composto por un máximo de 32 particións libres nun ou máis discos. É unha forma de xuntar o espazo non asignado nun sistema con varios discos nunha única unidade lóxica.
- Volumes Espello. Neste tipo de volume o contido da partición dúplícase nunha partición idéntica noutro disco, aínda que se ven como un único volume e non como dous. Os volumes espello coñécense como RAID de nivel 1 (RAID1) e son tolerantes a fallos.
- Volumes divididos (*Striped*). Semellante ao volume distribuído, utiliza o espazo de varios discos e convérteos nunha única unidade lóxica. Utiliza un tipo especial de formato para escribir no disco e ten máis rendemento ca o volume distribuído. Os fallos de escritura adoitan ser maiores que no caso do volume distribuído. Coñécense como RAID de nivel 0 (volumes RAID-0).
- Volumes RAID-5. Como os volumes *stripped*, pero con tolerancia a fallos, xa que distribúen a información de paridade entre todos os discos membros do volume.

### **Sistemas de ficheiros**

Windows soporta os seguintes formatos de sistemas de ficheiros:



A) CDFS (sistema de ficheiros de CD-ROM): só permite a lectura e soporta os formatos de disco ISO-9660 e Joliet.

B) UDF: é un subconxunto do formato ISO-13346 con extensións para formatos como CD-R e DVD-R/RW. UDF está incluído na especificación DVD e é máis flexible ca o CDFS.

X) FAT12, FAT16, e FAT32: Windows é compatible co sistema de ficheiros FAT por compatibilidade con MS-DOS e outras versións de Microsoft Windows. O formato FAT (*File Allocation Table*) inclúe un mapa de bits que se utilizan para identificar clusters ou bloques no disco.

FAT32 ten unha capacidade teórica para direccionar volumes de 8 terabytes (TB); no entanto limita os volumes a un máximo de 32 GB.

Δ) NTFS (*New Technology File System*): é o formato nativo de Windows Server para os sistemas de ficheiros. NTFS utiliza enderezos de disco de 64 bits, co que podería xestionar volumes de ata 16 exabytes, pero Windows limita o tamaño dun volume NTFS ao que se poida direccionar con 32 bits, que é un pouco menos de 256 TB (con clusters de 64 KB). NTFS admite ficheiros de máximo 16 TB.

NTFS engade características de seguridade de ficheiros e directorios, cotas de disco, compresión de ficheiros, enlaces simbólicos baseados en directorios, e cifrado. Unha das súas características máis significativas é a recuperabilidade: rexistra os cambios que se realizan nos metadatos coma se fosen transaccións coa finalidade de que se poidan recuperar no caso da perda de ficheiros ou dos seus datos.

A estrutura central de NTFS é a táboa mestra de arquivos MFT (*Master File Table*), que é unha sucesión lineal de rexistros de tamaño fixo (1 KB). Cada rexistro de MFT describe un arquivo ou un directorio, contén os atributos do arquivo, como o seu nome e marcas de tempo, e a lista de enderezos de disco onde están os seus bloques. Se un arquivo é



demasiado grande, pode ser necesario empregar máis dun rexistro MFT para conter a lista de todos os bloques. Neste caso o primeiro rexistro denomínase rexistro base, e apunta aos demais rexistros MFT.

#### **44.3.7 Seguridade**

O administrador de seguridade, compoñente do Executive, fai que se respecte o complexo mecanismo de seguridade de Windows 2008, que satisfai os requisitos C-2 do *Libro Laranxa* do Departamento de Defensa de Estados Unidos.

Cada usuario e grupo de Windows 2008 identifícase cun SID (Security ID) único a nivel mundial. Cada proceso leva asociado unha ficha de acceso que especifica o seu SID e outras propiedades.

Cada obxecto ten asociado un descritor de seguridade que indica quen pode realizar qué operacións con el. Un descritor de seguridade está formado por un encabezado, seguido dunha DACL (*Discretionary Access Control List*) cun ou máis elementos de control de acceso (ACE). Os máis importantes son Allow e Deny. Ademais da DACL, o descritor ten unha SACL (*System Access Control List*), que non especifica quen pode usar o obxecto senón qué operacións co obxecto se asentán no rexistro de sucesos de seguridade do sistema (función de auditoría).

Nun sistema autónomo a validación corre por conta do proceso winlogon e a configuración de seguridade almacenada na propia máquina nas claves do rexistro: SECURITY e SAM, onde a primeira establece as políticas globais de seguridade e a segunda a seguridade específica de cada usuario.

Nun sistema en rede, a autenticación dos usuarios está centralizada en certos servidores denominados controladores do dominio. Os equipos organízanse dentro de Dominios, podendo estes estar xestionados mediante o emprego do Active Directory.



Windows 2008 dispón de administración centralizada de certificados. Nas versións anteriores confiábase en que cada aplicación mantiña a súa propia lista de claves ou CA fiables.

O protocolo KERBEROS (RFC 1510), que é un estándar de Internet para autenticación, é o método nativo que empregan os sistemas Windows 2008. Calquera servidor do Directorio Activo, automaticamente, ten o servizo do 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

Colexiado do CPEIG



# **45. SERVIDORES DE MENSAXARÍA. SISTEMAS DE CORREO. SERVIDORES DE APLICACIÓNS E SERVIDORES WEB.**



## **Tema 45. Servidores de mensaxería. Sistemas de correo. Servidores de aplicacións e servidores web.**

### **ÍNDICE**

<b>45.1 Servidores de MensaXería.....</b>	<b>2</b>
45.1.1 Sistema de mensaXería Centralizada.....	3
45.1.2 Sistema de mensaXería Distribuído.....	4
<b>45.2 Sistemas de Correo.....</b>	<b>4</b>
45.2.1 Sistemas de correo electrónico: arquitectura.....	6
45.2.2 SMTP (Simple Mail Transfer Protocol).....	10
45.2.3 POP (Post Office Protocol).....	14
45.2.4 IMAP - Internet Message Access Protocol.....	17
45.2.5 Formato de mensaxes en internet.....	22
45.2.6 Extensións de SMTP.....	23
45.2.6.1 ESMTTP.....	23
45.2.6.2 MIME.....	24
<b>45.3 Servidores de aplicacións e servidores web.....</b>	<b>28</b>
<b>45.4 Servidores de Aplicacións.....</b>	<b>30</b>
45.4.1 Servizos proporcionados por un servidor de aplicacións.....	31
45.4.2 Estándar J2EE.....	33
45.4.3 Estrutura dun servidor de Aplicacións.....	34
45.4.4 BEA WebLogic.....	34
<b>45.5 Servidores Web.....</b>	<b>36</b>
45.5.1 Introducción Cliente/Servidor.....	36
45.5.2 Servidores Web.....	37
45.5.3 Características dos servidores web.....	38
45.5.4 Arquitectura.....	39
45.5.4.1 Funcionamento do servidor web.....	40
45.5.5 Apache.....	42
45.5.5.1 httpd.conf.....	42
Directivas de contorno global.....	42
Directivas de configuración do servidor principal.....	43
45.5.5.2 Módulos de Apache.....	44





45.5.6 Microsoft IIS.....	44
45.5.6.1 Administración de IIS.....	45
45.5.7 Lighttpd.....	45
<b>45.6 Bibliografía.....</b>	<b>46</b>

## 45.1 SERVIDORES DE MENSAXERÍA

Un servidor de mensaxería é unha aplicación que posúe a capacidade para manexar mensaxes entre dúas ou máis entidades, ben sexan aplicacións de usuario ou outros sistemas de xestión de mensaxes. As mensaxes dun servidor de mensaxería son enviadas a través dun *middleware*, que facilita a comunicación entre os distintos elementos do sistema, utilizando normalmente un conxunto de regras e especificacións que posibilitan a comunicación entre as distintas partes. Outra das características dos servidores de mensaxería é a capacidade de almacenaxe das mensaxes, este almacenamento prodúcese normalmente nunha cola ata que é posible o seu envío cara ao seu destinatario, que polo xeral resulta ser outra aplicación.

É moi habitual atoparse nunha empresa ou organización un sistema de mensaxería funcionando nun servidor e agardando o envío de mensaxes á súa cola de entrada. Desde alí, o *middleware* analiza mensaxe a mensaxe determinando o destino de cada unha. Unha vez no servidor, unha mensaxe só ten dúas posibilidades de entrega: ou ser enviada de maneira local, ou que teña que ser redirixida a outro servidor de mensaxería para que sexa el o que realice a entrega. Se a mensaxe vai ser entregada a un destino local, entón é enviada inmediatamente á caixa do correo local. Pola contra, se a mensaxe se determina como remota, o servidor de mensaxería debe enviarlla a outro servidor de mensaxería dentro do seu contorno para que sexa este o que realice a entrega da mensaxe.



Polo xeral, se existen problemas de conexión entre os servidores ou non é posible determinar a localización do servidor de mensaxería remoto, o usuario que realizou o envío da mensaxe é informado da situación a través doutra mensaxe enviada polo servidor de mensaxería. Este tipo de mensaxes adoitan ser só de notificación de que se está a ter problemas co envío da mensaxe, posto que o servidor de mensaxería continuará tentando enviar a mensaxe ata que se esgote o número máximo de intentos de envío ou ata que a mensaxe caduque, é dicir, exceda un límite de tempo de estancia no servidor.

Normalmente, os modelos dos servidores de mensaxería adáptanse a unha arquitectura centralizada ou seguen unha solución distribuída.

#### **45.1.1 SISTEMA DE MENSAXERÍA CENTRALIZADA**

Un sistema de mensaxería centralizada fundaméntase nun núcleo de datos, o cal aloxa todos os recursos e servizos dos servidores que conforman o sistema. Este núcleo de datos permite que calquera usuario do sistema de mensaxería se conecte aos servizos de mensaxería, ben sexa de forma local ou remota.

As características dun sistema de mensaxería centralizado son:

- **Datos:** Todos os datos e a información se atopan albergados e se xestionan desde o núcleo, mesmo cando os usuarios establecen unha conexión remota para a súa utilización. Esta centralización facilita en gran medida a administración dos servizos, xa que a fai máis sinxela.
- **Actualizacións:** As actualizacións débense realizar unicamente no núcleo central, onde se atopa todo o sistema.
- **Localización:** O centro de datos engádelle ao sistema dispositivos de illamento da alimentación ou sistemas de alimentación ininterrompida (SAI). Proporciona ademais a posibilidade de ofrecer



servizos mesmo cando se produce algunha incidencia de carácter grave, xa que posibilita a réplica de todo o sistema dun xeito eficaz.

#### **45.1.2 SISTEMA DE MENSAXERÍA DISTRIBUÍDO**

Un sistema de mensaxería distribuído está formado por unha serie de sucursais repartidas en distintas localizacións conectadas entre si. Cada sucursal posúe un servidor ou servidores de mensaxería con todos os seus servizos de xeito independente do resto de sucursais. Cada un dos servidores de mensaxería realiza o envío das súas mensaxes locais e redirixe aos outros servidores aquelas mensaxes que non son de dominio local e que si son capaces de resolver algún dos outros servidores.

- **Datos:** A información atópase tamén distribuída entre cada unha das sucursais e cada unha delas xestiona e administra esta información e os seus servizos, o que provoca un aumento na complexidade destas tarefas.
- **Actualizacións:** Cada vez que se leva a cabo unha tarefa de actualización esta tense que realizar en cada unha das sucursais para que teña efecto en todo o sistema.
- **Localización:** Cada sucursal posúe o seu propio centro de datos, os cales poden ofrecer os mesmos servizos ca nunha arquitectura centralizada.

#### **45.2 SISTEMAS DE CORREO**

Un servidor de correo é unha aplicación que nos permite enviar mensaxes (correos) duns usuarios a outros, con independencia da rede que os devanditos usuarios estean a utilizar.



Para logralo defínense unha serie de protocolos, cada un cunha finalidade concreta:

- *SMTP, Simple Mail Transfer Protocol*: É o protocolo que se utiliza para que dous servidores de correo intercambien mensaxes.
- *POP, Post Office Protocol*: Utilízase para obter as mensaxes gardadas no servidor e pasarllas ao usuario.
- *IMAP, Internet Message Access Protocol*: A súa finalidade é a mesma que a de POP, pero o funcionamento e as funcionalidades que ofrecen son diferentes.

Así pois, un servidor de correo consta en realidade de dous servidores: un servidor SMTP, que será o encargado de enviar e recibir mensaxes, e un servidor POP/IMAP, que será o que lles permita aos usuarios obter as súas mensaxes.

Para obter as mensaxes do servidor, os usuarios sérvense de clientes, é dicir, programas que implementan un protocolo POP/IMAP. Nalgunhas ocasións o cliente execútase na máquina do usuario, mais existe outra posibilidade: que o cliente de correo non se execute na máquina do usuario; é o caso dos clientes vía web.

O correo electrónico é unha das aplicacións TCP/IP máis utilizadas nestes días. Na súa forma máis sinxela, o correo electrónico, é un xeito de enviar mensaxes ou cartas electrónicas dun computador a outro.

O correo electrónico de internet implementouse orixinalmente como unha función do protocolo FTP. En 1980 Suzanne Sluizer e Jon Postel realizaron traballos cun protocolo que posteriormente se denominaría SMTP (*Simple Mail Transfer Protocol*). Hoxe en día séguese a utilizar este protocolo, cos avances lóxicos que require o tipo de transferencia actual.



O protocolo SMTP foi desenvolvido pensando en que os sistemas que intercambiarían mensaxes serían grandes computadoras, de tempo compartido e multiusuario, conectadas permanentemente á rede Internet. Con todo, coa aparición dos computadores persoais, que teñen unha conectividade ocasional, fíxose necesaria unha solución para que o correo chegase a estes equipos. Para resolver esta limitación, nacen os protocolos POP e IMAP.

Xa que logo, podemos discriminar dous tipos de axentes que están implicados na transferencia de correo, MUA e MTA:

- Axente de usuario (MUA), interface para ler e escribir as mensaxes; son os clientes finais.
- Axente de transporte (MTA ou estafeta), encargado do transporte das mensaxes (SMTP). O primeiro MTA ao que o cliente lle entrega o seu correo chámase MSA (S de *sending*) e o último que o recibe e llo entrega ao cliente destinatario chámase MDA (D de *delivering*).
- Cando un MTA non é o destinatario dun correo, débello entregar a outro, e así ata chegar ao seu destino. Este comportamento é coñecido como **Relay**.

É moi importante configurar ben a función *Relay* dun MTA, porque se se configura de forma aberta, pode terminar sendo unha fonte de SPAM.

#### 45.2.1 **SISTEMAS DE CORREO ELECTRÓNICO: ARQUITECTURA**

Os sistemas de correo electrónico configúranse para que traballen de forma asíncrona na comunicación, de forma que o cliente envía unha mensaxe e non ten que agardar resposta. Deste xeito, as caixas do correo e as



funcións de transmisión e recepción de mensaxes sitúanse nun servidor de correo.

O servidor de correo permanece “á escoita” de conexións doutros servidores de correo para a recepción de mensaxes con destino aos seus usuarios; almacena as mensaxes de correo recibidas nas caixas do correo e realiza a transmisión de mensaxes dos usuarios a outros servidores remotos, é dicir, o servidor de correo actúa como un MTA.

Os usuarios dun servidor interactuarán con el a través dun cliente de correo. Para a recuperación de mensaxes utilízase POP3 ou IMAP4. O envío de mensaxes non se realizará directamente aos servidores remotos, senón que, en primeiro lugar, se lle envía a mensaxe ao servidor de correo que lle dá servizo, mediante SMTP, e será este servidor o que realice a transmisión definitiva da mensaxe ao servidor de correo remoto que albergue a caixa do correo destino.

Tendo en conta isto, a arquitectura dun sistema de correo inclúe:

- Servidores de correo para o envío, recepción e almacenamento da información dos usuarios. Deberán estar correctamente configurados para poder ser alcanzables no DNS.
- Clientes de correo utilizados polos usuarios para, esencialmente, compoñer e ler correos electrónicos.
- Soporte para plataformas de acceso: hoxe en día pódese acceder desde o posto de traballo na intranet da empresa, desde un equipo portátil na extranet, e mesmo desde *smartphones* e PDA.
- Sistema de almacenamento das mensaxes nas respectivas caixas do correo.
- Sistema de Directorio (*Active Directory* ou LDAP): útiles para acceder á información dos usuarios da empresa (nome, cargo, etc.).



## **Cientes de correo**

Os clientes de correo permítenlles aos usuarios interaccionar co sistema para enviar e recibir mensaxes. Desde os inicios do correo electrónico ata hoxe en día, podemos clasificar os clientes en:

- Clientes en modo texto: é o cliente accesible desde a interface de comandos Shell mediante unha conta na máquina que alberga o servidor de correo.
- Clientes pesados: software que se instala no PC do usuario e xestiona todo o ciclo de vida da creación, envío e recepción de mensaxes, así como a posibilidade de almacenamento local; por exemplo, Microsoft Outlook, Lotus Notes Thunderbird, etc.
- Clientes lixeiros: non fai falta instalar ningún software no PC, senón que o usuario accede ao sistema de correo a través dunha interface web cun navegador. A seguridade na comunicación pódese implementar mediante SSL.
- Clientes en *smartphones* ou PDA: permiten acceder ás mensaxes en calquera lugar e a calquera hora. Normalmente configúranse con IMAP4 como protocolo de recuperación para permitir que, posteriormente, as mensaxes procesadas sexan visibles tamén desde o PC.

## **Servidores de correo**

Ata hai poucos anos a un servidor de correo pedíáselle que xestionase correctamente o servizo do correo. Actualmente, téndese a proporcionar unha solución unificada de mensaxería para unha organización que integre mensaxería móbil, mensaxería instantánea, *groupware*, etc., de tal xeito que a fronteira entre servidores puros de correo electrónico e servidores xerais de mensaxería ou contorno colaborativo está pouco clara.



Para escoller un servidor ou outro cumpriría ter en conta as seguintes consideracións:

- O cliente que se asocia ao servidor, xa que algúns servidores esixen un tipo determinado de cliente, como Lotus Notes.
- Estreitamente relacionado co anterior punto está a integración de servidor e cliente e, polo xeral, obtense unha mellor integración cando o servidor e o cliente son do mesmo fabricante, como sucede con Microsoft Outlook e Microsoft Exchange Server, ou Lotus Notes e Lotus Domino.
- O nivel de xestión requirido, a solución de almacenamento e a dispoñibilidade. Para unha organización pode resultar insuficiente tocante a rendemento, flexibilidade e escalabilidade utilizar os discos dunha máquina para almacenar as caixas do correo, así que é máis habitual contar con dispositivos de almacenamento dedicados como solucións NAS ou SAN, con conexións rápidas de fibra óptica e que, ademais, facilitan o uso de clústeres activo/activo ao permitiren que un servidor asuma as caixas do correo xestionadas por outro en caso de caída.
- É importante utilizar técnicas de *benchmarking* para avaliar as seguintes características:
  - o A capacidade de tratar correo concorrentemente.
  - o A velocidade de entrega.
  - o A extensibilidade e funcións implementadas.
  - o A estabilidade.
- Ao tratarse dunha aplicación vital para unha organización, o correo electrónico debe ser configurado como unha solución de alta



dispoñibilidade, o cal require o establecemento de políticas de redundancia adecuadas para garantir o servizo.

Actualmente, os programas servidor de correo máis estendidos son:

- Microsoft Exchange Server.
- Lotus Domino/Notes.
- Sendmail.

#### 45.2.2 **SMTP (SIMPLE MAIL TRANSFER PROTOCOL)**

O significado das siglas de SMTP é “Protocolo Simple de Transmisión de Correo”. Este protocolo é o estándar de internet para o intercambio de correo electrónico. SMTP necesita que o sistema de transmisión poña á súa disposición unha canle de comunicación fiable e con entrega ordenada de paquetes, para o cal o uso do protocolo TCP (porto 25) na capa de transporte é o adecuado. Para que dous sistemas intercambien correo mediante o protocolo SMTP non é preciso que exista unha conexión interactiva, xa que este protocolo usa métodos de almacenamento e reenvío de mensaxes.

Realmente son tres os estándares que se aplican a un envío de correo desta clase. O termo SMTP é frecuente e erroneamente usado para se referir á combinación do grupo dos tres estándares involucrados no envío de correo electrónico. Iso débese a que os tres están estreitamente relacionados, pero falando estritamente, SMTP é un dos tres estándares. Os tres estándares son:

- Un estándar para o intercambio de correo entre dúas computadoras, o cal especifica o protocolo usado para enviar correo entre "host" TCP/IP. Este estándar é SMTP e está definido orixinalmente no RFC 821, e foi actualizado nos RFC 2821 e RFC 5321 (outubro do 2008).



- Un estándar do formato da mensaxe de correo, contido en dous RFC:
  - o RFC 822 describe a sintaxe das cabeceiras do correo electrónico e describe a interpretación do grupo de campos da cabeceira. Este protocolo foi actualizado nos RFC 2821 e 5322.
  - o RFC 1049 describe cómo un conxunto de documentos de tipos diferentes do texto ASCII plano se poden usar no corpo do correo. Os estándares son PostScript, Scribe, SGML, TEX, TROFF e DVI. O nome do protocolo oficial para este estándar é MAIL.
- Un estándar para o encamiñamento de correo usando o DNS (sistema de nomes de dominio), descrito en RFC 974. O nome oficial do protocolo para este estándar é DNS-MX.

### **Funcionamento**

O protocolo SMTP é un protocolo cliente/servidor, polo que sempre é o usuario SMTP o que inicia a sesión e o servidor de correo o que responde.

O protocolo SMTP baséase na entrega de mensaxes extremo a extremo. Cando un servidor de SMTP require transmitir unha mensaxe a outro servidor SMTP, o emisor (servidor que inicia a sesión SMTP) establece unha conexión co receptor (servidor que recibe petición de establecer sesión SMTP). Esta conexión é unidireccional, é dicir, o emisor pódelle enviar correo ao receptor, pero durante esa conexión o receptor non lle pode enviar correo ao emisor. Se o receptor ten que lle enviar correo ao emisor, ten que agardar a que finalice a conexión establecida e establecer outra en sentido contrario, cambiando os papeis de emisor e receptor. Unha vez establecida a conexión, o emisor envía comandos e mensaxes.

O protocolo SMTP funciona con comandos e respostas de texto escritas en ASCII-NVT (estándar USA - 7 bits).





Cada comando envíase ao servidor SMTP, ao porto 25, de maneira predeterminada. A cada comando enviado polo cliente séguelle unha resposta do servidor SMTP composta por un código numérico de tres díxitos, seguido dunha mensaxe descritiva. O número está pensado para un procesamento automático da resposta, namentres que o texto permite que un humano interprete a resposta.

No protocolo SMTP todas as ordes, respostas e datos son liñas de texto, delimitadas polo carácter CRLF. Todas as respostas teñen un código numérico ao comezo da liña.

### **Fluxo**

Os pasos fundamentais para traballar co correo electrónico utilizando este protocolo son os seguintes:

- O cliente SMTP conéctase ao servidor SMTP, realizando un *telnet* polo porto 25 e agarda resposta.
- O servidor SMTP pode responder:
  - o “220 Service Ready”, xunto co nome de dominio do servidor, se o servizo de correo está dispoñible.
  - o “421 Service not available” se o destinatario é temporalmente incapaz de responder.
- Se o servizo está dispoñible, o cliente tense que identificar. Para iso envía o comando HELO seguido polo nome de dominio do seu equipo. Desde abril do 2001, as especificacións para o protocolo SMTP, definidas en RFC 2821, indican que o comando HELO sexa substituído polo comando EHLO.
  - o Un receptor SMTP que non soporte o RFC 2821 responderá cunha mensaxe “*500 Syntax error, command unrecognized*”. O





emisor SMTP debería intentalo de novo con HELO ou, se non pode retransmitir a mensaxe, enviar unha mensaxe QUIT.

- o Se un receptor SMTP soporta as extensións de servizo, responde cunha mensaxe "250 OK" que inclúe unha lista das extensións de servizo que soporta.
- O emisor inicia agora unha transacción enviándolle o comando MAIL FROM: ao servidor. Este comando contén a ruta de volta ao emisor que se pode empregar para informar de erros. Se se acepta o comando, o receptor responderá cunha mensaxe "250 OK". Calquera outro código indica erro.
- O segundo paso do intercambio de correo consiste en darlle ao servidor SMTP o destinatario da mensaxe (pode haber máis dun receptor). Isto faise enviando un ou máis comandos "RCPT TO: <destinatarios>" (Se hai máis dun destinatario estes irán separados por comas. Cada un deles recibirá unha resposta "250 Recipient OK" se o servidor coñece o destino, ou un "550 Non such user here" se non.
- O seguinte paso é informar o servidor de que se vai empezar a introducir o corpo da mensaxe; para iso utilízase a orde DATA.
- O servidor contesta con "354 Start mail input, end with <CRLF>.<CRLF>", onde se indica que a mensaxe debe finalizar cun punto nunha única liña, seguido dun retorno de carro.
- O cliente envía os datos liña a liña, acabando coa liña <CRLF>. <CRLF> que o servidor recoñece con "250 OK" ou a mensaxe de erro apropiada se calquera cousa foi mal.
- Unha vez que o servidor recibe a mensaxe finalizada cun punto pode almacenala se é para un destinatario que pertence ao seu dominio,



ou ben retransmitirla a outro servidor para que finalmente chegue a un servidor do dominio do receptor.

- Agora hai varias accións posibles:
  - o O emisor non ten máis mensaxes que enviar; pechará a conexión cun comando QUIT, que será respondido con "221 Service closing transmission channel".
  - o O emisor non ten máis mensaxes que enviar, pero está preparado para recibir mensaxes (se as hai) do outro extremo. Mandará o comando TURN. Os dous SMTP intercambian os seus papeis e o emisor que era antes receptor pode enviar agora mensaxes.

Se se require autenticación TLS/SSL a conexión realízase aos portos 465 ou 587, en vez do porto 25.

#### 45.2.3 **POP (POST OFFICE PROTOCOL)**

O protocolo de oficina de correo, POP, é un protocolo que ten como misión a entrega final do correo ao destinatario; non serve para enviar correos nin para envialos. O seu obxectivo principal é poder xestionar os correos sen ter que estar conectado a internet, é dicir, permítelles aos usuarios con conexións intermitentes ou moi lentas (p. ex. módem), descargar o correo electrónico mentres teñen conexión e revisalo posteriormente mesmo estando desconectados.

#### **Modelo de comunicacións POP**

A descrición do protocolo POP podémola atopar no RFC 1939. A última versión do POP é a 3, por iso é habitual referirse a este protocolo como POP3.



O protocolo POP3 é un protocolo cliente/servidor, polo que sempre é o usuario POP3 o que inicia a sesión e o servidor de correo o que responde.

O protocolo POP3 funciona con comandos e respostas de texto escritas en ASCII.

O cliente POP conéctase co servidor a través do porto TCP, 110. Para conectarse ao servidor, é necesario unha conta de identificación nesta máquina (o que lle permite ter un espazo reservado para os seus correos). Deseguido cómpre verificar que é dono da conta a través dunha clave. Unha vez conectado ao sistema, o cliente POP pode dialogar co servidor para saber, entre outras cousas, se existen mensaxes na caixa, cantas mensaxes son, ou para solicitar a descarga dalgunha delas.

Cando a conexión TCP está establecida, POP3 continúa con tres fases:

- **Autorización:** Envíase o *login* e *password* para identificar o usuario que quere ler o correo. Cando se verifica que o nome e a clave son correctos, o servidor pasa a un estado de transacción. Antes de pasar a este estado, o servidor POP bloquea a caixa do correo para impedir que os usuarios modifiquen ou borren o correo antes de pasar ao estado seguinte.
- **Transacción:** Prodúcese a manipulación do contido da caixa do correo do usuario.
- **Actualización:** Todas as modificacións se realizan cando o cliente finaliza o servizo (co comando QUIT).

Polo tanto, o protocolo POP3 administra a autenticación utilizando o nome de usuario e o contrasinal. Non obstante, isto non é seguro, xa que os contrasinais, do mesmo xeito que os correos electrónicos, circulan pola rede como texto plano, sen cifrar. En realidade, segundo RFC 1939, é posible cifrar o contrasinal utilizando un algoritmo MD5 e beneficiarse



dunha autenticación segura. Aínda así, debido a que este comando é opcional, hai poucos servidores que o implementen. Ademais, o protocolo POP3 bloquea as bandexas de entrada durante o acceso, o que significa que é imposible que dous usuarios accedan de maneira simultánea á mesma bandexa de entrada.

## **Fluxo**

Os pasos fundamentais para traballar co correo electrónico utilizando este protocolo son os seguintes:

- O cliente establece unha conexión TCP no porto 110 do servidor POP.
- O servidor POP responderá cun indicador de estado e unha palabra clave. Se o servizo está dispoñible, responderá co indicador de estado +OK; en caso contrario, responderá con -ERR.
- Se o servizo está dispoñible, pásase á fase de autorización e o cliente identifícase cos comandos USER e PASS.
- Se a información é correcta, o servidor responderá con +OK e dá acceso exclusivo á caixa do correo.
- O cliente pode interactuar coa caixa do correo; para iso utiliza os seguintes comandos:
  - o LIST mostra os correos que hai na caixa do correo e o seu tamaño.
  - o STAT dá o número de correos non borrados na caixa e a súa lonxitude total.
  - o TOP <nº\_mens> <num\_liñas> mostra  $n$  liñas do correo; o seu número dáse no argumento. No caso dunha resposta positiva do servidor, este enviará de volta os encabezados do correo,





despois unha liña en branco e finalmente as primeiras  $n$  liñas do correo.

- o RETR <nº\_mens> recolle un correo especificado polo seu número.
  - o DELE <nº\_mens> borra un correo especificado polo seu número.
  - o RSET recupera os correos borrados (na conexión actual).
  - o UIDL obtén a listaxe con todos os identificadores únicos de mensaxes. O servidor asígnalle un identificador único a cada mensaxe, de modo que non cambie o seu identificador entre sesións. Este identificador é o UID.
- Para rematar a sesión POP utilízase o comando QUIT. Elimínanse aquelas mensaxes que foron marcadas co comando DELE. Ata que non se invoca a orde QUIT, as mensaxes marcadas non son borradas da caixa do correo.

Se se require autenticación TLS/SSL, a conexión realízase ao porto 995, non ao porto 110.

#### 45.2.4 **IMAP - INTERNET MESSAGE ACCESS PROTOCOL**

O protocolo IMAP (protocolo de acceso a mensaxes de internet) é un método utilizado polas aplicacións cliente de correo electrónico para obter acceso ás mensaxes almacenadas remotamente. Neste caso, as mensaxes non son recuperadas polo xestor de correo, senón que se traballa con elas directamente sobre o servidor.

É un protocolo máis complexo que POP3. Algunhas vantaxes sobre o anterior son:

- As transaccións IMAP poden durar moito máis tempo.





- O servidor garda información do estado dos correos (se foron lidos ou non, se foron gardados nunha carpeta, etc.).
- Pódense definir distintas carpetas para acceder a distintas caixas do correo.
- Pódenselle devolver partes da mensaxe ao cliente, aforrando ancho de banda.
- Pódese conectar máis dun cliente á mesma caixa do correo.
- Posúe buscadores que se executan no servidor.
- A diferenza de POP (onde o cliente debe estar conectado ao servidor para que se realicen os cambios), IMAP permítelles aos clientes realizaren cambios tanto estando estes conectados como desconectados.
- É totalmente compatible con diferentes estándares de mensaxes de internet, como MIME.

Con todo, posúe certas desvantaxes:

- É máis complexo de implementar que POP3.
- O servidor debe ser máis potente para atender a todos os usuarios. Consome máis recursos de CPU, memoria, etc.

O protocolo IMAP é un protocolo cliente/servidor, polo que sempre é o usuario IMAP o que inicia a sesión e o servidor de correo o que responde.

Os clientes IMAP poden acceder seguindo un destes tres modos de conexión:



- **Modo offline.** Periodicamente conéctase ao servidor para descargar mensaxes novas e sincronizar calquera cambio que se poida producir nas diferentes carpetas. Existe a posibilidade de borrar as mensaxes a medida que as descargamos, seguindo un funcionamento moi semellante a POP3.
- **Modo online.** Accédese directamente á copia das mensaxes do servidor exactamente cando fai falta, sincronizando os cambios practicamente ao instante.
- **Modo desconectado.** Neste caso o cliente traballa cunha copia local namentres que non ten acceso a internet, creando/borrando/lendo os seus correos. A próxima vez que se conecte a internet estes cambios han sincronizarse coa copia mestra do servidor.

O protocolo IMAP funciona con comandos e respostas de texto escritas en ASCII. Actualmente a versión operativa é a 4, por iso este protocolo tamén se coñece como IMAP4.

Dado que se parte dun modelo no que as mensaxes se gardan normalmente no servidor despois de ser lidas, IMAP define unha maneira sinxela de administralas: con caixa do correo, é dicir, con carpetas. Estas seguen unha xerarquía de tipo árbore. Seguindo o estándar, sempre existirá unha caixa do correo de entrada que será a principal, pero poderemos crear outras carpetas con diferentes atributos. Por exemplo, existen atributos para especificar que unha carpeta contén só correos (*\NoInferiors*) ou só carpetas (*\NoSelect*), pero tamén poden ter outros atributos que indiquen se existen ou non mensaxes novas desde a última vez que a abrimos (*\Marked* e *\Unmarked*).

Unha clase parecida de etiquetas poden ter os correos que se reciban e/ou envíen. Unha das máis usadas será a que indica se está lido ou non (*\Seen*), pero tamén existen outras que indican que a mensaxe foi contestada



(\Answered), que a mensaxe foi destacada (\Flagged), que é un borrador (\Draft), etc. Toda esta información se garda directamente no servidor e non no cliente, o que permite sincronizar perfectamente estes metadatos entre varios clientes.

Na RFC 2060 (actualmente, a RFC 3501) defínense as instrucións para poder interactuar co servidor de correo e as súas caixas do correo.

Fases dunha sesión IMAP. Do mesmo xeito que no POP3, nunha sesión IMAP existen as seguintes fases:

- *Non-authenticated state*: neste estado o cliente aínda non se autenticou co servidor.
- *Authenticated state*: o cliente foi autenticado polo servidor e debe seleccionar unha caixa do correo para interactuar.
- *Selected state*: o cliente seleccionou unha caixa do correo e pódense realizar accións sobre os correos contidos nela.
- *Logout state*: a conexión foi finalizada.

## **Fluxo**

Os pasos fundamentais para traballar co correo electrónico utilizando este protocolo son os seguintes:

- O cliente establece unha comunicación TCP co servidor IMAP polo porto 143.
- O servidor responde con OK se o servizo está dispoñible; en caso contrario, o servidor responderá con BAD.
- Deseguido, o cliente tense que identificar mediante o comando LOGIN <usuario> <password> para poder acceder ás caixas do correo. Esta é unha forma non segura, porque a *password* non vai





cifrada. Pódese utilizar o comando AUTHENTICATE para autenticar o usuario de forma segura.

- Se os datos son correctos, o servidor responde OK. Se se produce un fallo de autenticación, devolverá un NON. Se os argumentos non son válidos, devolverá un BAD.
- Agora o cliente pode interactuar coas súas mensaxes. Para iso usa os comandos:
  - o LIST para ver as caixas do correo existentes.
  - o SELECT <nome\_caixa do correo> para ver o contido dunha caixa do correo determinada.
  - o CREATE <nom\_caixa do correo\_nova> para crear caixas do correo.
  - o DELETE <nome\_caixa do correo> para borrar caixas do correo.
  - o RENAME <nome\_caixa do correo\_old> <nome\_caixa do correo\_new> para renomear caixas do correo.
  - o FETCH <num\_mens> <parte\_mens> para ver as diferentes partes das mensaxes da caixa do correo que se seleccionou.
  - o CLOSE pecha a caixa do correo e borra as mensaxes marcadas para borrar.
  - o EXPUNGE borra todas as mensaxes marcadas para borrar.
  - o SEARCH busca mensaxes segundo algún criterio de busca.
  - o COPY copia as mensaxes dunha carpeta a outra.
- Co comando LOGOUT remata a sesión.



Se se require autenticación TLS/SSL, a conexión realízase ao porto 993, non ao porto 143.

#### 45.2.5 **FORMATO DE MENSAXES EN INTERNET**

A RFC 2822 define o estándar do formato de mensaxe de internet. O correo electrónico divídese en dúas partes separadas por unha liña en branco.

- A cabeceira da mensaxe.
- O corpo da mensaxe.

**Corpo da mensaxe.** Contén a información que se intercambian o emisor e o receptor. A forma en que está codificada vén determinada polo RFC 2231.

**Encabezados da mensaxe.** É a metainformación colocada antes do corpo da mensaxe. En xeral, o software de transporte de correo non revisa nin altera os encabezados do correo, a excepción da cabeceira *Received*. Están formadas pola tupla Palabra\_clave: valor. As cabeceiras máis importantes son:

- From: Enderezo do emisor da mensaxe.
- Reply-to: Conta de correo a onde se dirixirán as respostas ao correo. En ausencia deste campo as respostas diríxense ao/s enderezo/s indicado/s no campo *From*.
- To: Este campo contén o/s enderezo/s do/s principal/-ais destinatario/s da mensaxe.
- Cc: Copia a destinatarios. Campo que indica o/s enderezo/s ao/s que se lle/s fará chegar unha copia do correo, aínda que o contido da mensaxe poida que non vaia dirixido expresamente a el ou eles.
- Bcc: Copia oculta. Mándaselles unha copia aos destinatarios aquí indicados sen que o resto de destinatarios teñan coñecemento diso.





- Message-ID: É un identificador único de cada mensaxe. Este código é asignado polo servidor de onde sae a mensaxe. Este identificador non se pode cambiar nin modificar.
- Reference: Contén todos os Message-IDE das mensaxes ás que este fai referencia. Este campo é xerado pola aplicación cliente.
- KeyWords: Palabras clave que identifican o contido da mensaxe.
- Return-Path: Contén a traxectoria de regreso ao remitente.
- Received: É a información que se utiliza para comprobar os problemas que aparezan na repartición dunha mensaxe. Nela móstranse os enderezos das máquinas polas que pasou a mensaxe en dirección ao seu destino, xunto coa data e hora en que o fixo.
- Date: Data e hora na que a mensaxe é entregada á cola do servidor SMTP para o seu envío. Este campo establéceo o servidor orixe.
- Subject: Este campo contén un pequeno texto coa descrición do asunto da mensaxe.
- X- : Son campos definidos polo usuario. Sempre teñen que empezar por X-, seguidos do nome que se lle queira asignar ao campo. Por exemplo, X-mailer: “O meu xestor de correo”. Estase utilizando unha cabeceira X-SPAM para marcar correos como presuntos correos lixo.

Só as cabezas subliñadas son obrigatorias segundo o estándar.

## 45.2.6 EXTENSIÓNS DE SMTP

### 45.2.6.1 ESMTP

O protocolo ESMTP é unha extensión do protocolo SMTP, definido na RFC 4954.



Trátase dun mecanismo para autenticar a identidade do cliente que se conecta ao servidor, e ademais permite a negociación dunha capa de seguridade para facer máis segura a comunicación. O protocolo SMTP permanece inalterado; o que se fai é agregar os seguintes comandos:

- EHLO dominio. Fai que o servidor realice unha consulta ao DNS do reverso do dominio indicado para verificar que este exista.
- ETRN dominio (*Extended Turn*). Este comando permite que o cliente lle pida ao servidor que lle envíe todas as mensaxes que posúe destinadas ao cliente. Se hai mensaxes para a máquina cliente, o servidor debe iniciar unha nova sesión SMTP para enviarlle as mensaxes.
- AUTH: Comando que serve para negociar un protocolo de seguridade para o intercambio de datos. Os posibles protocolos, para a capa de seguridade, que se poden negociar dáos como resposta o servidor ao comando EHLO.

#### 45.2.6.2 **MIME**

O protocolo SMTP impón determinadas restricións sobre o contido das mensaxes:

- O contido só debe estar composto de caracteres ASCII; non se poden enviar ficheiros binarios como audio, vídeo, documentos, etc.
- As liñas non poden exceder os 100 caracteres.
- O tamaño total do contido non pode exceder unha determinada dimensión.
- Ademais, tamén existen problemas á hora de enviar mensaxes en linguaxes distintas do inglés:



- o Linguaxes sen alfabetos “occidentais” (chinés, xaponés)
- o Linguaxes con alfabetos non latinos (ruso, árabe)
- o Linguaxes con acentos (alemán, castelán)

Para liquidar estas limitacións definíronse as especificacións MIME (*Multipurpose Internet Mail Extensions*), que son unhas extensións do correo electrónico, utilizadas tamén noutros protocolos coma o HTTP, que permiten a transmisión de datos non ASCII, a través de correo electrónico, no corpo da mensaxe.

MIME non cambia a SMTP nin o substitúe, polo que as mensaxes que se envíen con MIME tamén cumprirán este protocolo. Dado que SMTP utiliza para comandos e respostas o ASCII de 7 bits, o camiño seguido para transmitir calquera ficheiro é transformar (codificar) o ficheiro non ASCII en ASCII de 7 bits (facéndoo compatible con SMTP), transmitilo neste formato e reconvertelo en destino ao formato orixinal (descodificalo).

MIME incorpora as seguintes características ao servizo de correo electrónico:

- Capacidade de enviar múltiples adxuntos nunha soa mensaxe.
- Lonxitude ilimitada da mensaxe.
- Uso de conxuntos de caracteres non pertencentes ao código ASCII.
- Uso de texto enriquecido (deseños, fontes, cores, etc.).
- Adxuntos binarios (executables, imaxes, arquivos de audio ou vídeo, etc.), que se poden dividir de ser necesario.

As cabeceiras descritas na RFC 2822 son suficientes para enviar correo codificado en texto ASCII, pero non son adecuadas para mensaxes



multimedia. Para iso MIME engade unhas cabeceiras que describen o tipo de contido da mensaxe e o tipo de código. Estas son:

- **MIME-Version:** Contén a versión das extensións MIME empregadas na mensaxe.
- **Content-Transfer-Encoding:** Sinala como foi codificada a mensaxe para a súa transmisión por correo electrónico, de forma que poida viaxar sen problemas de que se corrompa desde o destinatario ao receptor a través dos axentes de correo (MUA). Para transferir datos binarios, MIME ofrece cinco formatos de codificación:

- o *7bit*: significa que o ficheiro é SÓ texto ASCII (caracteres non acentuados). As liñas deben ser "curtas", de 100 caracteres ou menos, rematando con CRLF.
- o *Quoted-Printable*: utilizado por texto que é maioritariamente US-ASCII (7 bit) pero cunha pequena porcentaxe de caracteres "estraños" (8 bit). Este é o caso do castelán.

Nesta codificación, cada carácter de 8-bits é codificado en tres caracteres de 7 bits; o primeiro o signo igual (=) e o valor hexadecimal do carácter. Por exemplo, o "ñ", "F1" en hexadecimal, codifícase como "=F1".

- o *base64*: usado para codificar secuencias arbitrarias de octetos de forma que satisfaga as regras de 7bit. Utilízase para enviar binarios.
- o *8bit*: formato de texto de 8 bits.
- o *binary*: envío de binarios.
- **Content-Type:** indica que tipos de datos contén a mensaxe. Un tipo de MIME está composto do seguinte xeito:





tipo\_mime\_principal/subtipo\_mime. Pódense atopar os seguintes tipos:

- o *text*: texto con ou sen formato.
- o *image*: imaxes estáticas.
- o *video*: imaxes dinámicas; pode incluír audio.
- o *audio*: son.
- o *message*: significa que o contido está configurado segundo o estándar RFC 822; isto pode ser usado para reexpedir mensaxes.
- o *application*: emprégase para sinalar que o contido é para serlle enviado a un programa externo; por exemplo, texto para unha impresora PostScript.
- o *multipart*: unha mensaxe tamén pode ter varias partes con varios contidos separados, mesmo de tipos diferentes (texto, audio e imaxes). Incluso cada parte pode ter subpartes (ser, pola súa vez, multiparte), posto que o formato MIME pode ser recursivo.

Á parte do tipo, tamén se pode especificar un subtipo, ambos os dous separados por unha barra inclinada /. Por exemplo, image/gif é unha imaxe en formato GIF; o tipo é image e o subtipo gif; text/html, text/plain, etc.

Se o tipo é *multipart*, os subtipos admitidos son:

- o *mixed*: permite que nunha soa mensaxe conteña varias submensaxes independentes, cada unha co seu tipo e codificación. Desta forma pódese incluír nunha mensaxe imaxes, audio, vídeo...





- o *parallel*: permite incluír nunha mensaxe subpartes que se poden ver simultaneamente; por exemplo, reproducir audio e vídeo.
- o *digest*: permite incluír nunha mensaxe varias mensaxes.
- o *alternative*: permite que nunha mesma mensaxe se poida incluír unha única información pero en diversos formatos. Isto é útil cando os destinatarios teñen distinto hardware e/ou sistema operativo.

Finalmente, pode ter parámetros opcionais empezando por un punto e coma ;. Por exemplo, o parámetro charset= en Content-type: text/plain; charset=iso-8859-1, indica que o corpo da mensaxe utiliza o xogo de caracteres ISO-8859-1.

### 45.3 SERVIDORES DE APLICACIÓNS E SERVIDORES WEB

A diferenza básica entre o servidor web e o servidor de aplicacións é que o servidor web serve para ver as páxinas nun navegador web, mentres que un servidor de aplicacións proporciona os métodos necesarios que poden ser chamados polas aplicacións cliente. Noutras palabras, as peticións http son manexadas polos servidores web e a lóxica de negocio sérveselles aos programas de aplicación a través dunha serie de protocolos no servidor de aplicacións. Nun servidor de aplicacións, un cliente pode utilizar a GUI e os servidores web, mentres que nos servidores web o cliente pode usar HTML e HTTP.

	<b><i>Servidor web</i></b>	<b><i>Servidor de aplicacións</i></b>
<i>Que é?</i>	Un servidor que xestiona conexións HTTP.	Un servidor que lle expón a lóxica do





		negocio ao cliente mediante unha serie de protocolos, pero non exclusivamente HTTP.
<i>Engade funcionalidade?</i>	Un servidor web non engade funcionalidade; simplemente recibe unha petición e envíalle a resposta ao cliente.	Engade funcionalidade, xa que implementa unha lóxica de negocio intermedia.
<i>Que tipo de aplicacións serve?</i>	Só baseadas en web.	Aplicacións baseadas en web, pero tamén outras que non o son, o cal é posible xa que un servidor de aplicacións inclúe internamente un servidor web.
<i>Que tipo de clientes permite?</i>	Navegadores.	



		Navegadores e interfaces pesadas.
<i>Cales son as súas funcións?</i>	Almacenar ficheiros escritos en HTML, PHP, etc., de tal forma que sexan accesibles para os navegadores web cando os sitios web necesiten acceder a eles.	Ofrecer aplicacións a outro sistema.

#### 45.4 **SERVIDORES DE APLICACIÓNS**

Por servidor de aplicacións entendemos aquel que permite a execución dunha serie de aplicacións. Habitualmente trátase dun programa software que xestiona case por completo as funcións de lóxica de negocio e de acceso aos datos da aplicación. O seu propósito é xestionar de forma centralizada a maneira en que os clientes se conectan á base de datos ou aos servizos cos que estes deben interactuar.

Os servidores de aplicacións comezan a xurdir cando se fai patente que as aplicacións cliente/servidor ían presentar problemas de escalabilidade cando se tratase de servir a un gran número de usuarios. Ademais, era necesario trasladar as regras de negocio a un lugar intermedio entre os clientes e a base de datos.



O concepto de servidor de aplicacións está moi ligado ao de sistema distribuído, os cales permiten mellorar 3 aspectos fundamentais nunha aplicación:

- *A alta dispoñibilidade:* refírese á necesidade de que un sistema funcione 24 horas ao día, todos os días. Para poder cumprir con esta característica son necesarias técnicas de equilibrio de carga e de recuperación ante fallos.
- *A escalabilidade:* consiste na capacidade de facer crecer un sistema cando aumenta o número de peticións. Cada sistema pode atender un número limitado de peticións, xa que os seus recursos son finitos; ao engadir novos equipos, a cantidade de recursos multiplícase e, con iso, o número de peticións que poden ser atendidas.
- *O mantemento:* ten que ver coa facilidade para realizar actualizacións, depurar fallos e manter o sistema.

#### 45.4.1 **SERVIZOS PROPORCIONADOS POR UN SERVIDOR DE APLICACIÓNS**

- *Xestión da sesión:* o servidor debe conservar a información entre peticións dun usuario mentres dure a antedita sesión.
  - Esta é unha característica fundamental para as aplicacións de comercio electrónico, que requiren establecer o usuario a través da súa navegación polo sitio web; con todo, o protocolo http é un protocolo sen sesión, polo que non permite manter unha conexión aberta entre cliente e servidor máis aló do que dura a transferencia de información. Por iso son os servidores de aplicacións os que se encargan de todo o relacionado coa xestión da sesión.
- *Equilibrio de carga:* un servidor de aplicacións debe proporcionar





técnicas para equilibrar a súa propia carga, é dicir, debe ser capaz de repartir o procesamento entre diversos servidores, o cal é fundamental para a súa escalabilidade.

- o As peticións que realizan os clientes transmítense á máquina que estea menos ocupada en cada momento, o cal mellorará o rendemento global da aplicación.
  - o Cun bo equilibrio de carga, ademais de conseguir un sistema máis escalable, conséguese unha maior tolerancia a fallos.
- *Acceso aos datos:* un servidor de aplicacións proporciona un acceso sinxelo para realizar a administración das conexións a bases de datos relacionais.
  - o É habitual que tamén permitan o acceso a outros tipos de fontes de datos como:
    - ERP
    - Repositorios XML
    - Sistemas herdados
- *Pooling de conexións:* é habitual que os servidores de aplicacións manteñan de forma permanente conexións coas bases de datos. Estas conexións distribúense entre os procesos de forma transparente, xa que sería moi custoso, ademais de influír negativamente no rendemento da aplicación, abrir unha conexión por cada consulta que se queira realizar.
- *Xestión de transaccións:* as transaccións son fundamentais en calquera software e máis aínda nos de tipo comercial, xa que evitan a aparición de información inconsistente.
  - o Os servidores de aplicacións adoitan contar con esta característica, de forma que, con indicar en que momento se inicia unha transacción e en que momento se finaliza, o propio sistema se encargaría de desfacer os pasos intermedios no



caso de que se produza un erro na aplicación.

#### **45.4.2 ESTÁNDAR J2EE**

As plataformas máis comúns en que se asentán os servidores de aplicacións son J2EE e .NET. J2EE está máis estendida e, ata hai relativamente pouco, era impensable implementar un servidor de aplicacións que non seguisse este modelo.

O estándar J2EE permite desenvolver aplicacións empresariais de forma eficiente e sinxela. O feito de desenvolver unha aplicación con tecnoloxías J2EE permite que esta sexa despregada en calquera servidor de aplicacións que cumpra co devandito estándar. Un servidor de aplicacións é unha implementación da especificación J2EE que se compón de:

1. Cliente web ou contedor de applets: é un navegador web que interactúa co contedor web mediante HTTP.
  - a. Pode executar applets e código javascript.
  - b. Recibe páxinas HTML ou XML.
2. Aplicación cliente: trátase de clientes que non se executan dentro dun navegador.
  - a. Poden utilizar distintas tecnoloxías para comunicarse co contedor web.
  - b. Pódense comunicar directamente coa base de datos.
3. Contedor web ou servidor web: correspóndese coa parte visible dun servidor de aplicacións.
  - a. Emprega os protocolos HTTP e SSL.
4. Servidor de aplicacións: proporciona servizos que lle dan soporte á execución e dispoñibilidade das aplicacións despregadas.



Existen distintas implementacións partindo deste estándar, cada unha cunhas peculiaridades de seu que as poden facer máis adecuadas para un determinado sistema. Algunhas das máis destacadas son:

- BEA WebLogic
- IBM WebSphere
- Sun-Netscape IPlanet
- Sun One
- Oracle IAS
- Borland AppServer
- HP Bluestone

#### 45.4.3 **ESTRUTURA DUN SERVIDOR DE APLICACIÓNS**

Un servidor de aplicacións aséntase nunha estrutura en 3 capas que permite realizar unha estruturación máis eficiente do sistema.

- *Capa cliente*: contén os programas que executan os usuarios, como navegadores web. Estes programas poden estar implementados en calquera linguaxe de programación.
- *Capa media*: contén o servidor de aplicacións e outros que poden ser direccionados polos clientes, como servidores proxy ou servidores web existentes.
- *Capa datos*: contén os recursos, como sistemas de bases de datos, ERP, etc.

#### 45.4.4 **BEA WEBLOGIC**

BEA WebLogic Server é un servidor de aplicacións completo e baseado en estándares, que proporciona o fundamento sobre o cal unha empresa pode construír as súas aplicacións. Presenta un completo conxunto de características, como son o cumprimento dos estándares abertos, a



arquitectura de varios niveis, e o apoio para o desenvolvemento baseado en compoñentes, etc.

Oracle WebLogic Server proporciona todas as funcións básicas esenciais dun servidor de aplicacións e servizos, tales como:

- Equilibrio de carga
- Tolerancia a fallos
- Servizos web
- Transparencia na rede
- Integración de sistemas herdados
- Xestión de transaccións
- Seguridade
- Multi-threading
- Persistencia
- Conectividade con bases de datos
- Agrupación de recursos

Estas funcionalidades axilizan o desenvolvemento de aplicacións e alivian os esforzos dos desenvolvedores.

Ademais de J2EE, Oracle WebLogic Server implementa todos os estándares importantes de programación, integración e traballo en rede que son a base para a construción dunha infraestrutura de aplicacións, incluíndo:

- XML: Oracle WebLogic Server implementa a última versión da API de Java para o procesamento de XML (JAXP), e inclúe un analizador integrado Apache Xerces e un analizador XML de alto rendemento deseñado especificamente para pequenas e medianas empresas.
- SOAP: SOAP é o novo estándar para o intercambio de información nun contorno distribuído. É o protocolo de comunicación para definir



o formato dos datos para os servizos Web que se entregan a través de HTTP.

- WSDL: WSDL é unha linguaxe baseada en XML utilizada para describir un servizo web publicado. BEA WebLogic Server ten soporte incorporado para WSDL e xera un gui3n WSDL de forma autom3tica cando un servizo Web se implementa no servidor WebLogic.
- UDDI: un rexistro UDDI 3 un directorio de servizos web, distribuído e baseado en Web, moi semellante a un caderno de tel3fonos. Oracle WebLogic Server inclúe incrustado un rexistro UDDI e unha API para a procura e a actualizaci3n deste ou calquera outro rexistro UDDI.
- JMX e SNMP: a infraestrutura do servidor WebLogic baséase no est3ndar aberto e extensible JMX. Ademais, o axente SNMP est3 dispoñible para a compatibilidade con sistemas que se baseen en SNMP.
- Os administradores de sistemas poden configurar as s3as pol3ticas de seguridade baseadas en roles de acceso.

## **45.5 SERVIDORES WEB**

### **45.5.1 INTRODUCCI3N CLIENTE/SERVIDOR**

A tecnoloxía Cliente/Servidor consiste no procesamento cooperativo da informaci3n mediante un conxunto de procesadores, no cal m3ltiples clientes, xeograficamente dispersos, poden realizar petici3ns a un ou m3is servidores centrais.

Desde unha perspectiva funcional, podemos definir cliente-servidor como unha arquitectura distribuída que lles permite aos usuarios obter acceso de



forma transparente á información. Este tipo de arquitectura é a máis estendida nos sistemas distribuídos.

Un sistema cliente servidor baséase nas seguintes características:

- *Servizo*: o servidor proporciónaos e o cliente utilízalos.
- *Recursos compartidos*: son moitos os clientes que empregan os mesmos servidores mediante os cales comparten recursos non só lóxicos senón tamén físicos.
- *Protocolos asimétricos*: os clientes son os encargados de iniciar a comunicación co servidor, os cales esperan o establecemento da conexión de forma pasiva.
- *Transparencia de localización*: os clientes non saben onde se localizan fisicamente os recursos que desexan utilizar.
- *Independencia da plataforma*.
- *Sistemas debilmente axustados*: interacción baseada en envío de mensaxes.
- *Encapsulación de servizos*: a implementación que os servidores realizan dos servizos é transparente para os clientes.
- *Escalabilidade horizontal*: incorporar novos clientes.
- *Escalabilidade vertical*: aumentar a potencia dos servidores.
- *Integridade*: tanto os datos como os programas están centralizados en servidores que facilitan a súa integridade e mantemento.

#### 45.5.2 **SERVIDORES WEB**

Un servidor http ou servidor web é un programa que permite procesar as peticións dos distintos navegadores, servindo os recursos que estes soliciten mediante os protocolos HTTP ou HTTPS. De forma xeral, un servidor web funciona de xeito moi simple, executando constantemente as seguintes accións:





1. Agardar peticións no porto TCP indicado.
  - a. Por defecto empregarase o porto 80.
2. Recibir unha petición.
3. Procesar a solicitude.
4. Enviarlle ao cliente a resposta obtida, empregando a mesma conexión pola que se recibiu a petición.
5. Volver agardar novas peticións.

Se un servidor web se cingue ao patrón anterior, cumprirá todos os requisitos básicos dos servidores HTTP, aínda que se verá limitado a servir ficheiros estáticos. Os servidores web existentes actualmente deseñáronse e implementáronse a partir do patrón anterior, onde a variación entre eles só radica no tipo de peticións que van atender, se son ou non multiproceso, etc.

#### 45.5.3 **CARACTERÍSTICAS DOS SERVIDORES WEB**

- *Servir ficheiros estáticos*: un servidor web debe ser capaz de servir ficheiros estáticos que se localicen nalgún lugar do disco (requisito imprescindible).
  - o Debe ser posible especificar que parte do disco se vai servir.
  - o Aínda que un servidor poida especificar un directorio por defecto, non debe obrigar a empregar un concreto.
  - o En moitos servidores é posible especificar outros subdirectorios ou directorios, indicando en que punto do sistema de ficheiros virtual do servidor irán localizados os recursos.
- *Seguridade*: un servidor web debe poder especificar directivas de seguridade, isto é, establecer quen pode acceder a que recursos.
  - o Algúns servidores permiten especificar os ficheiros que se considerarán como índice do directorio.



- *Contido dinámico*: é unha das características fundamentais. Indica a capacidade do servidor para ofrecer contido dinámico.
  - o A gran maioría do contido web que se serve é dinámico.
  - o Punto fundamental á hora de elixir un servidor.
- *Soporte para distintas linguaxes*: a maior parte dos servidores ofrece soporte para algunhas linguaxes de programación como:
  - o PHP
  - o JSP: para que un servidor atenda peticións JSP requirirá algún tipo de software para funcionar, como un contedor de Servlets.
  - o ASP
  - o CGI (sistema máis antigo e sinxelo para xerar contido dinámico.)

Antes de seleccionar unha linguaxe de programación de servidor, é necesario considerar se se desexa unha linguaxe máis estándar que poida ser atendida por calquera servidor xenérico, ou ben se se prefire unha arquitectura concreta, etc.

#### 45.5.4 **ARQUITECTURA**

A arquitectura dun servidor web divídese en:

1. Capa servidor
2. Capa soporte

**Capa servidor**: contén 5 subsistemas, que teñen a función de implementar as funcionalidades do servidor.

- *Subsistema de recepción*: é o encargado de agardar polas peticións do cliente a través da rede.



- o Ten a capacidade de manexar peticións simultáneas, podendo analizalas para determinar se son ou non compatibles co navegador.
- *Analizador de peticións*: asocia ao recurso de rede un arquivo local.
- *Control de acceso*: encárgase de validar e permitir o acceso.
- *Controlador de recursos*: fixa o tipo de recurso que se solicitou, execútao e obtén a resposta.
- *Rexistro de transacción*: a súa función é rexistrar as peticións xunto coas súas respostas.

**Capa soporte:** conforma a interface entre o servidor web e o sistema operativo, e manexa os seguintes subsistemas.

- *Útil*: contén as funcións que utilizan os outros subsistemas.
- *Capa abstracta do sistema operativo*: encapsula o funcionamento do sistema operativo para facilitar a portabilidade do servidor entre as diferentes plataformas.

#### 45.5.4.1 **Funcionamento do servidor web**

Un servidor web execútase agardando peticións por parte dun navegador web (cliente) e atendendo as devanditas peticións de forma adecuada, respondendo mediante unha mensaxe de erro ou unha páxina web coa resposta á petición formulada.

Por exemplo, se tecleamos [www.xunta.es](http://www.xunta.es) nun navegador, o proceso que se desencadea é o seguinte:

- O navegador realiza unha petición HTTP ao servidor do enderezo solicitado.
- O servidor responde enviando o código HTML da páxina solicitada.
- O cliente recibe o código da páxina, interprétao e móstrao en pantalla.



É o cliente o encargado de interpretar o código HTML, mostrar os textos e obxectos da páxina, as súas cores, fontes, etc. O servidor, pola súa banda, limítase a transmitir o código da páxina sen realizar ningún tipo de interpretación da devandita páxina.

Un servidor web, ademais de transmitir código HTML, tamén pode entregar aplicacións web, que son segmentos de código que se executan cando se producen certas peticións ou respostas HTTP.

É necesario distinguir entre:

1. *Aplicacións no lado do cliente:* é o navegador web o encargado de executar estas aplicacións no equipo do usuario (scripts). Nesta categoría atopamos aplicacións como Applets de Java ou Javascript.
  - O servidor proporciónalle o código destas aplicacións ao cliente e é o navegador deste o que as executa.
  - É necesario que o navegador do cliente dispoña da capacidade para executar estas aplicacións.
  - Por defecto, a maioría dos navegadores permite executar scripts de java e javascript, aínda que mediante plugins se poden engadir máis linguaxes.
2. *Aplicacións no lado do servidor:* é o servidor o encargado de executar a aplicación, a cal, unha vez executada, xera un código HTML que o servidor toma e envía ao navegador do cliente mediante HTTP.

Polo xeral, a opción que se escolle é a das aplicacións no lado do servidor, xa que ao executarse no servidor e non no equipo do cliente, este non require ningún tipo de software ou funcionalidade engadida, mentres que no caso das aplicacións no lado do cliente, si que é necesario.



#### 45.5.5 **APACHE**

O servidor Apache é un servidor web de código aberto que se desenvolve dentro do proxecto HTTP Server da Apache Software Foundation, e que pode ser instalado en plataformas Windows, Unix, Mac e outras. Este servidor vén instalado na maioría de distribucións de Linux e en Mac OS X; Apache vén integrado como parte do seu propio servidor web.

Trátase do servidor web máis empregado, e malia que presenta algunhas vulnerabilidades de seguridade, a gran maioría delas só poderían ser explotadas de forma local e non remota.

##### 45.5.5.1 **httpd.conf**

Apache pode ser configurado mediante o ficheiro *httpd.conf*. Cada vez que se introduza unha modificación neste ficheiro, será necesario reiniciar o servizo Apache. Trátase dun servidor altamente configurable, inda que a súa interface gráfica non é demasiado intuitiva.

O ficheiro de configuración *httpd.conf* pódese dividir en varias seccións:

- *Sección 1*: Contorno global. Sección do ficheiro onde se localizan as rutas a outros ficheiros de configuración e se describe o funcionamento xeral do servidor.
- *Sección 2*: Contorno servidor principal. Sección do ficheiro onde se describe a configuración que non atende as peticións dos servidores virtuais. Trátase do comportamento predeterminado do servidor.
- *Sección 3*: Contorno de servidores virtuais. Sección do ficheiro onde se poden configurar servidores virtuais para traballar co mesmo programa.

#### ***Directivas de contorno global***

A configuración realízase mediante directivas, variables almacenadas nun arquivo de texto, que permiten alterar e controlar o funcionamento de Apache en función dos valores que estas tomen.



- *ServerType*: permite indicar como será a resposta do servidor.
  - o *Inetd*: execútase cando hai unha petición.
  - o *Standalone*: sempre existe un proceso httpd en execución e este crea novos fillos para as conexións cos diferentes clientes.
- *ServerRoot*: permite detallar o directorio que actuará como raíz do servidor.
- *Timeout*: permite especificar o número de segundos que se mantén á espera un servidor, desde que se recibe a petición ata que se entende a conexión como inactiva.
- *MaxClients*: limita o número máximo de clientes que se poden conectar de forma simultánea. Se se supera este número, os clientes son bloqueados.
- *Listen*: permítelle a Apache atender peticións noutro enderezo e/ou portos ademais dos establecidos por defecto.
- *BinAddress*: emprégase para especificar que enderezos ou IP se deben atender no servidor. Permite dar soporte a servidores virtuais.
- *LoadModule*: permite cargar un novo módulo para proporcionarlle maior funcionalidade ao servidor.

### ***Directivas de configuración do servidor principal***

- *Port*: permite especificar o porto en que escoitará o servidor. Só pode existir unha directiva *Port*, mentres que se poden especificar varias *Listen*.
- *User e Group*: permite indicar o usuario ou grupo que pode iniciar a execución de httpd.
- *ServerAdmin*: establece o enderezo de correo electrónico onde enviar os problemas que poidan xurdir. Este enderezo mostrarase nas páxinas de erro que xera o servidor.
- *ServerName*: permite asignar o nome do servidor que se lles vai



mostrar aos clientes. Non é aconsellable empregar o nome real da máquina.

- *ServerSignature on/off/email*: emprégase para que, en caso de acceso a unha páxina inexistente, o servidor devolva unha páxina de erro indicando a versión de Apache e o nome da máquina.
- *DocumentRoot*: especifica o directorio onde se localizan os documentos web que o servidor poñerá a disposición dos clientes.

#### 45.5.5.2 **Módulos de Apache**

Apache é un servidor estruturado en módulos cuxa configuración se realiza mediante a modificación das directivas presentes en cada módulo.

Os módulos de Apache poden ser clasificados nos seguintes grupos:

- Módulos base: módulos que engloban as funcións básicas de Apache.
- Módulos multiproceso: módulos que se encargan da interconexión cos portos do ordenador, aceptando as peticións e enviando as peticións aos distintos fíos para seren atendidas. Módulos adicionais: calquera módulo que incorpore unha funcionalidade ao servidor.

#### 45.5.6 **MICROSOFT IIS**

IIS (*Internet Information Services*) é un servidor web específico para o sistema operativo Microsoft Windows. IIS converte un ordenador nun servidor web, que permite publicar páxinas web e facelas accesibles localmente, cara a unha intranet ou cara a internet; ademais, proporciona as funcións e ferramentas necesarias para realizar de forma sinxela a administración dun servidor web seguro.

IIS baséase en diversos módulos que lle proporcionan a capacidade de servir varios tipos de páxinas, como ASP (*Active Server Pages*), ASP.NET, PHP ou Perl.



#### 45.5.6.1 **Administración de IIS**

A última versión de IIS é a 7, aplicable a Windows 7, Windows Server 2008, Windows Server 2008 R2 e Windows Vista.

En IIS7 hai varias ferramentas para realizar a súa administración e configuración, entre as cales se inclúen:

- Administrador de IIS.
- Ferramenta de liña de comandos denominada Appcmd.exe.
- Almacén de configuración de IIS que consta de arquivos ApplicationHost.config e Web.config.
- Espazo de nomes de Instrumental de Administración de Windows (en inglés, WMI ou *Windows Management Instrumentation*).

#### 45.5.7 **LIGHTTPD**

Lighttpd é un servidor web libre, distribuído baixo a licenza BSD, deseñado para ser rápido, seguro, flexible e respectuoso cos estándares. Está deseñado para contornos onde a velocidade é moi importante e se requiren respostas rápidas e de alta escalabilidade. Consome menos memoria e procesador que outros servidores.

Algunhas características de Lighttpd son:

- Permite a comunicación con programas externos mediante SCGI ou FastCGI.
- Ten un módulo de reescritura e de redirección de URL.
- Fixéronse melloras específicas para a súa integración con PHP e Ruby on Rails.
- Permite módulos externos.
- Permite VirtualHosting.



- Pode servir tanto HTTP como HTTPS.
- Autenticación con LDAP, httpasswd ou MySQL.
- Acepta Webdav.

Lighttpd pódese usar só ou combinado con outros; de feito, é habitual empregalo para liberar de carga a outros servidores máis lentos, especialmente cando hai que realizar o envío de ficheiros grandes, que adoita ser moito máis rápido ca no resto de servidores. É común atopar Lighttpd en combinación con instalacións de Apache, para facelo máis escalable e rápido en situacións de carga.

#### **45.6 BIBLIOGRAFÍA**

- Internet y Correo electrónico. Silva Salinas, Sonia y López Sanjurjo, Catherin (aut.). Ideaspropias Editorial, 2007.
- Correo electrónico. Romero Dueñas, Carlos y González Hermonso, Alfredo. Edelsa, 2001.
- Apache: The Definitive Guide, Third Edition. Ben Laurie, Peter Laurie.
- Apache: soluciones y ejemplos para administradores de Apache. Ken Coar e 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 e Bobby Martya.

**Autor:** Francisco Javier Rodríguez Martínez



Subdirector de Sistemas da Escola Superior de Enxeñaría Informática de Ourense

Colexiado do 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 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.**



**Tema 46: Administración e xestión de sistemas e almacenamento. Virtualización de servidores. Virtualización do posto cliente. Computación baseada en servidor (SBC). Grid Computing. Cloud computing. Green IT e eficiencia enerxética. Redes SAN e elementos dun SAN. Virtualización do almacenamento. Xestión do ciclo de vida da información (ILM). Sistemas de backup: hardware e software de backup. Estratexias de backup a disco. Replicación local e remota, estratexias de recuperación.**

---

#### **46.1 Administración e xestión de sistemas e almacenamento**

*46.1.1 Administrador de rede*

*46.1.2 Administración e xestión de redes*

#### **46.2 Virtualización de servidores**

*46.2.1 Funcionamento*

#### **46.3 Virtualización do posto cliente**

*46.3.1 Modos de operación de VDI*

#### **46.4 Computación baseada 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 implantación*

#### **46.7 Green IT e eficiencia enerxética**

*46.7.1 Tecnoloxías verdes*

*46.7.2 Actividades relacionadas con Green IT*

#### **46.8 Redes SAN e elementos dunha SAN**



#### *46.8.1 Estrutura das SAN*

### **46.9 Virtualización do almacenamento**

#### *46.9.1 Virtualización por bloques*

#### *46.9.2 Virtualización a nivel de arquivo*

#### *46.9.3 Diferenzas entre NAS e SAN*

### **46.10 Xestión do ciclo de vida da información (ILM)**

#### *46.10.1.....Xestión do ciclo de vida dos datos*

#### *46.10.2.....Xestión do ciclo de vida da información*

#### *46.10.3.....Algunhas solucións para a xestión*

### **46.11 Sistemas de backup: hardware e software de backup**

#### *46.11.1.....Hardware de backup*

#### *46.11.2.....Software de backup*

### **46.12 Estratexias de backup a disco**

### **46.13 Replicación local e remota, estratexias de recuperación**

#### *46.13.1.....Replicación local*

#### *46.13.2.....A replicación remota*

### **46.14 Replicación local e remota, estratexias de recuperación**

#### *46.14.1.....Replicación local*

### **46.15 Bibliografía**

## **461 ADMINISTRACIÓN E XESTIÓN DE SISTEMAS E ALMACENAMENTO**

### **461.1 Administrador de rede**

Un administrador de rede, como o seu nome indica, "administra" unha



rede, é dicir, encárgase de:

- Instalación e configuración da rede.
- Hardware de rede, conexións físicas, concentradores, conmutadores, encamiñadores, servidores e clientes.
- Software de rede, como os sistemas operativos de rede, servidores de correo electrónico, software para a realización de copias de seguridade, base de datos servidores e software de aplicación.

O máis importante é que o administrador ten coidado dos usuarios da rede, respondéndolles as súas preguntas, escoitando os seus problemas e resolvéndollos.

Cando as tarefas de administración se realizan nunha rede grande e complexa, este conxunto de tarefas débense abranguer de xeito dedicado, é dicir, posuír unha ou varias persoas realizando unicamente as tarefas de administración da rede. Isto debe ser así debido a que as redes tenden a ser volátiles no sentido de:

- Os usuarios da rede cambian constantemente.
- Os equipos fallan.
- Prodúcense conflitos entre as distintas aplicacións.
- En xeral, unha rede complexa sofre continuos estados de crise.

Pola contra, as redes de menor tamaño, e por iso menos complexas, son xeralmente moito máis estables. Adoita ser habitual que unha vez posta en funcionamento unha rede sinxela non teña que sufrir continuas e complexas tarefas de administración xa sexan de hardware ou software.

Neste tipo de redes pequenas os problemas tamén aparecen, pero como



interveñen un reducido número de equipos é normal que sexan sinxelos, poucos e distantes entre si.

Independentemente do tamaño dunha rede, un administrador debe cubrir as seguintes tarefas que son comúns a calquera tipo de rede:

- Involucrarse e formar parte na toma de decisións para a adquisición de novo equipamento, servidores, equipos, impresoras, etc.
- Establecer as accións necesarias para o correcto funcionamento cada vez que se engada un novo equipo, é dicir, un administrador de rede cando se integra un novo elemento na rede encárgase de introducir cambios na configuración da cablaxe, de asignar un nome de rede ao novo equipo, integrar un novo usuario no sistema de seguridade garantindo ademais os seus privilexios.
- Estar ao corrente das actualizacións de software que publiquen os provedores e considerar se as súas novas características son suficientes para xustificar unha posible actualización.

Na maioría dos casos, a parte máis difícil dun proceso de actualización de software é a determinación do camiño a seguir, como levar a cabo a actualización de toda a rede afectando o menos posible ao funcionamento dos usuarios. Isto adoita ser aínda máis crucial se o software que se quere actualizar é o sistema operativo de rede, posto que calquera cambio nel pode afectar a toda a rede.

Dentro deste proceso de actualización tamén interveñen afectando en menor medida á estabilidade do sistema os parches e Service Packs que publican os provedores para actualizar as súas solucións e que solucionan problemas menores.

- Realizar tarefas rutineiras como a realización de copias de



seguridade dos servidores, a administración do historial de datos ou a liberación de espazo nos discos duros. Gran parte das tarefas de administración dunha rede consisten en asegurarse de que todo funcione correctamente, buscando e corrixindo os problemas que poidan ter os usuarios.

- Recompilar, organizar e controlar o inventariado de toda a rede, para poder solucionar no menor tempo posible calquera imprevisto.

#### **461.2 Administración e xestión de redes**

O concepto de administración ten asociado moitos significados. Dende un punto de vista informal, a xestión de redes refírese ás actividades relacionadas co funcionamento dunha rede, xunto coa tecnoloxía necesaria para apoiar estas actividades. Outro aspecto de importancia na xestión dunha rede é a monitorización dela, é dicir, entender en todo momento que é o que está a suceder na rede.

Dende un enfoque software, a xestión de redes fai referencia ao conxunto de actividades, métodos, procedementos e ferramentas que interveñen nas operacións de administración, mantemento e aprovisionamento dos sistemas existentes dentro da rede.

Supón ademais garantir toda a oferta operativa de servizos mantendo a rede en marcha e funcionando sen problemas. Para conseguir isto faise imprescindible a utilización de ferramentas para a monitorización da rede, que ofrezan a detección de problemas tan pronto como sexa posible, mesmo antes de que algún usuario se vexa afectado.

A administración abrangue á súa vez as tarefas de seguimento dos recursos na rede e de como estes se asignan, facendo uso de todos os procesos ou accións de limpeza da rede que sexan necesarias para manter todo baixo o control do administrador ou administradores.

O proceso de *mantemento*, que se ocupa de realizar as operacións de



reparación e mellora, debe levar a cabo tarefas como a substitución dunha tarxeta de rede, actualización do sistema operativo dun encamiñador (router), engadir un novo conmutador (switch) á armazón de rede. O mantemento tamén implica medidas para a corrección e prevención, como por exemplo, o axuste dos parámetros necesarios dun dispositivo en función das necesidades que se soliciten ou intervir cando sexa necesario para mellorar o rendemento da rede en momentos puntuais.

Outro aspecto da administración dunha rede é o *aprovisionamento*, tarefa que concirne á configuración e adaptación dos recursos de rede para dar soporte aos servizos ofertados. Un exemplo de aprovisionamento é o feito de engadir as configuracións necesarias nos sistemas para proporcionar o servizo de voz a un novo usuario.

#### 461.2.1 Tarefas da xestión de rede

As tarefas de xestión dunha rede pódense caracterizar do seguinte xeito:

- QoS e xestión do rendemento: un administrador de rede debe supervisar e analizar periodicamente os encamiñadores, ordenadores anfitrións e o funcionamento das ligazóns e despois, en función dos resultados obtidos, realizar unha redirección do fluxo de datos para evitar a sobrecarga de certos puntos da rede. Para realizar esta tarefa de seguimento da rede, existen ferramentas que detectan rapidamente os cambios que se producen no tráfico dunha rede.
- Xestión de fallos pola rede: calquera fallo na rede, ligazóns, nodos, encamiñadores, fallos de hardware ou software, debe ser detectado, localizado e respondido pola propia rede, é dicir, a propia rede debe posuír mecanismos para intentar solucionar por si mesma o maior número de continxencias que se poidan producir.
- Xestión da configuración: esta tarefa implica o seguimento de



todos os dispositivos baixo xestión e a confirmación de que todos os dispositivos están conectados e funcionan correctamente. Se se produce un cambio inesperado nas táboas de encamiñamento, o administrador ha de descubrir o problema de configuración e solucionalo o antes posible para que ningún servizo nin usuario se vexa afectado.

- **Xestión da seguridade:** o administrador de rede é o responsable da seguridade da rede. Para poder manexar esta tarefa utilízanse principalmente as devasas (firewall) , posto que un firewall pode monitorar e controlar os puntos de acceso á rede informando sobre calquera intento de intrusión.
- **Xestión de facturación e contabilidade:** o administrador especificálles aos usuarios da rede os accesos ou restricións sobre os recursos e encárgase da facturación e dos cargos aos usuarios polo uso destes.

#### 461.2.2 Elementos da xestión de rede

A xestión de rede está composta por tres compoñentes principais:

- **Centro de xestión:** composto polo administrador de rede e as súas oficinas ou centros de traballo. Normalmente o centro de xestión está composto por un grupo humano importante.
- **Dispositivos para xestionar:** conformado polo equipamento da rede, incluído o seu software, que é controlado mediante o centro de xestión. Calquera concentrador, ponte, encamiñador, servidor, impresora ou módem é considerado un dispositivo que debe ser xestionado.
- **Protocolo de xestión da rede:** é o conxunto de políticas que adopta o centro de xestión para controlar e manexar todos os



dispositivos que conforman a rede. O protocolo de xestión de rede permítelle ao centro de xestión coñecer o estado dos dispositivos.

#### **461.2.2.1 Estrutura de xestión da información (SMI, Structure of Management Information):**

Define as regras para nomear os obxectos e para codificalos nun centro de xestión dunha rede, é dicir, é unha linguaxe mediante a que se definen as instancias dentro dun centro de xestión de rede.

A linguaxe SMI tamén ofrece construcións da linguaxe de maior nivel que, habitualmente, especifican os tipos de datos, o estado e a semántica dos obxectos que conteñen a información necesaria para realizar as tarefas de xestión. Por exemplo, a cláusula STATUS especifica se a definición do obxecto é actual ou está obsoleta.

Traballa baixo o protocolo SNMP (Simple Network Management Protocol) no que se definen os conxuntos de obxectos dentro a xestión de información base (MIB).

#### **461.2.2.2 A xestión da información base (MIB, Management Information Base)**

É un medio de almacenamento de información que contén os obxectos que mostran o estado actual dunha rede. Debido a que os obxectos teñen asociada información que se almacena no MIB, este forma coleccións de obxecto, nas que inclúe as relacións entre eles no centro de xestión.

Os obxectos organízanse dunha forma xerárquica e identifícanse pola notación abstracta ASN.1, linguaxe de definición de obxectos. A xerarquía,



coñecida como ASN.1, é unha árbore de identificadores de obxecto no cal cada rama ten un nome e un número, o que lle permite así á xestión de rede identificar obxectos por unha secuencia de nomes ou números dende a raíz ao obxecto.

#### **461.2.2.3 Protocolo SNMP (Simple Network Management Protocol)**

O Simple Network Management Protocol (SNMP) está deseñado para monitorar o rendemento dos protocolos de rede e dos dispositivos. As unidades de datos do protocolo SNMP (PDU) poden ser transportadas nun datagrama UDP, polo que a súa entrega en destino non está garantida. Os dispositivos que se administran como os encamiñadores ou ordenadores anfitrións, son obxectos e cada un ten unha definición formal e MIB adapta unha base de datos de información que describe as súas características. Con este protocolo un xestor de rede pode encontrar onde se localizan os problemas.

Execútase sobre UDP e utiliza unha configuración cliente-servidor. Os seus comandos definen como realizar as consultas sobre a información dun servidor ou como enviar esta cara a un cliente ou cara a outro servidor.

A tarefa principal do protocolo SNMP é a de transportar información entre os centros de xestión e os axentes que se executan en representación dos centros de xestión. Para cada obxecto MIB que se xestiona utilízase unha petición SNMP para obter o seu valor ou para modificala. Se un axente recibe unha mensaxe non solicitada ou se unha interface ou dispositivo deixa de funcionar, daquela o protocolo pode informar o centro de xestión do fallo que se está a producir.

A segunda versión deste protocolo, SNMPv2, corre por enriba de varios protocolos e ten máis opcións de mensaxaría, o que resulta nunha xestión



máis eficaz da rede. Ten sete unidades de PDU, ou mensaxes:

1. **GetRequest.** Utilízase para obter un valor de obxecto MIB.
2. **GetNextRequest.** Utilízase para obter o seguinte valor dun obxecto MIB.
3. **GetBulkRequest.** Recibe múltiples valores, o que equivale a GetRequests múltiples, pero sen necesidade de utilizar múltiples peticións.
4. **InformRequest.** É unha mensaxe de director a director de comunicación que se envían entre si dous centros de xestión a distancia o un do outro.
5. **SetRequest.** É utilizado por un centro de xestión para iniciar o valor dun obxecto MIB.
6. **Response.** É unha mensaxe de resposta a unha petición de tipo PDU.
7. **Trap.** Notifica un centro de xestión dun evento inesperado.

Hai dous tipos de representación de PDU, Get ou Set e Trap.

- O formato de PDU de Get ou Set é o seguinte:
  - o *PDU type*, indica un dos sete tipos de PDU.
  - o *Request ID*, é un ID que se utiliza para verificar a resposta dunha solicitude. Polo tanto un centro de xestión pode detectar peticións perdidas ou duplicadas.
  - o *Error status*, só é usado polas PDU Response para indicar tipos de erros reportados por un axente.
  - o *Error index*, é un parámetro que lle indica a un administrador o nome do obxecto que causou o erro.



Se as solicitudes ou respostas se perden, o protocolo non realiza un reenvío. Os campos Error status and Error index son todo zeros agás para as PDU *GetBulkRequest*

- O formato de PDU de Trap é:
  - o *Enterprise*, para usar en múltiples redes.
  - o *Timestamp*, para realizar as medicións de tempo.
  - o *Agentadress*, para indicar que o enderezo do axente xestor está incluído na cabeceira PDU.

## **462 VIRTUALIZACIÓN DE SERVIDORES**

Podemos definir virtualización como a técnica que consiste basicamente en agrupar diferentes aplicacións e servizos de sistemas heteroxéneos dentro dun mesmo hardware, de forma que os usuarios e o propio sistema os vexan como máquinas independentes dedicadas. Para iso, o sistema operativo virtualizado debe ver o hardware da máquina real como un conxunto normalizado de recursos independentemente dos compoñentes reais que o formen.

Desta forma, para virtualizar un sistema de servidores, os administradores deben, basicamente, optimizar os recursos dispoñibles, incluíndo o número e a identidade dos servidores físicos individuais, procesadores, e sistemas operativos, co obxectivo de producir unha mellora tanto na xestión coma no manexo de sistemas informáticos complexos. O administrador do sistema virtual utilizará un software para a división do servidor físico en ámbitos virtuais illados. Estes ámbitos son o que se coñece tecnicamente como servidores privados virtuais, pero tamén se poden encontrar



referencias como particións, instancias, colectores ou emulacións de sistemas.

En concreto, podemos dicir que un servidor privado virtual é un termo de mercadotecnia utilizada polos servizos de hospedaxe para referirse a unha máquina virtual para o uso exclusivo dun cliente individual do servizo. O termo utilízase para resaltar que a máquina virtual, a pesar de executarse no mesmo equipo físico que as máquinas virtuais doutros clientes, é funcionalmente equivalente a un equipo físico independente, está dedicado ás necesidades individuais do cliente e pode ser configurado para executarse como un servidor da internet (é dicir, para executar software de servidor). O termo VDS ou Virtual Dedicated Server (servidor virtual dedicado) emprégase para o mesmo concepto.

Cada servidor virtual pode executar o seu propio sistema operativo e ser reiniciado de modo independente.

#### **462.1      *Funcionamento***

O servidor físico realiza unha abstracción dos recursos que se denomina hipervisor ou VMM (Virtual Machine Monitor), elemento software que se instala na máquina onde se vai levar a cabo a virtualización e sobre a que se configuran as máquinas virtuais que é onde van residir as aplicacións. É o encargado de xestionar os recursos dos sistemas operativos "aloxados" (guest) ou máquinas virtuais.

Dende un punto de vista lóxico, o usuario percibe que son máquinas independentes e illadas entre si, pero dende unha perspectiva física, todas as máquinas virtuais residen nun único servidor. A estas máquinas virtuais asígnaselles unha porcentaxe dos recursos do servidor físico, que serán os únicos que o cliente coñeza.

Pódense encontrar tres modelos de virtualización: o modelo de máquina virtual ou virtualización completa, o modelo paravirtual ou virtualización parcial; e a virtualización a nivel de sistema operativo.



#### 462.1.1 Virtualización completa

O modelo de máquina virtual está baseado na arquitectura cliente/servidor, onde cada cliente funciona como unha imaxe virtual da capa hardware. Este modelo permite que o sistema operativo cliente funcione sen modificacións. Ademais permítelle ao administrador crear diferentes sistemas cliente con sistemas operativos independentes entre si. A vantaxe principal deste modelo radica no descoñecemento por parte dos sistemas hóspede do sistema hardware real sobre o que está instalado. Non obstante, realmente todos os sistemas virtuais fan uso de recursos hardware físicos. Estes recursos son administrados por un hipervisor que coordina as instrucións CPU, convertendo as peticións do sistema invitado nas solicitudes de recursos apropiados no computador anfitrión, o que implica unha sobrecarga considerable. Case todos os sistemas poden ser virtualizados utilizando este método, xa que non require ningunha modificación do sistema operativo. A pesar disto, é necesaria unha virtualización da CPU como apoio para a maioría dos hipervisores que levan a cabo a virtualización completa.

Exemplos típicos de sistemas de servidores virtuais son VMware Workstation, VMware Server, VirtualBox, Parallels Desktop, Virtual Iron, Adeos, Mac-on-Linux, Win4BSD, Win4Lin Prol, e z/VM, openvz, Oracle VM, XenServer, Microsoft Virtual, PC 2007 e Hyper-V.

#### 462.1.2 Paravirtualización

O modelo de máquina paravirtual (PVM) ou virtualización parcial baséase, como o modelo anterior, na arquitectura cliente/servidor, incluíndo tamén a necesidade de contar cun sistema monitor. Non obstante, neste caso, o VMM accede e modifica o código do sistema operativo do sistema hóspede. Esta modificación coñécese como *porting*. O *porting* serve de soporte ao VMM para que poida realizar chamadas ao sistema directamente. Ao igual que as máquinas virtuais, os sistemas paravirtuais son capaces de soportar diferentes sistemas operativos instalados no hardware real. Esta técnica



utilízase con intención de reducir a porción de tempo de execución empregada polo hóspede empregado en realizar as operacións que son moito máis difíciles de executar nun ámbito virtual en comparación cun contorno non virtualizado. Así permítese que o(s) invitado(s) e o hóspede soliciten e recoñezan estas tarefas, que doutro modo serían executados no dominio virtual (onde o rendemento de execución é peor). Unha plataforma paravirtualizada exitosamente pode permitir que o VMM sexa menos complexo (pola recolocación da execución das tarefas críticas do dominio virtual no dominio do servidor), e/ou reducir a degradación do rendemento global da máquina virtual durante a execución de convidado.

UML, XEN, Xen, Virtuozzo, Vserver e OpenVZ (que é o código aberto e a versión de desenvolvemento de Parallels Virtuozzo Containers) son modelos de máquinas paravirtuais.

#### 462.1.3 Virtualización por S.O.

A virtualización a nivel de sistema operativo diferénciase das anteriores en que, neste caso, non existe un sistema cliente/servidor propiamente dito. Neste modelo o sistema principal exporta a funcionalidade do sistema operativo dende o seu propio núcleo. Por esta razón, os sistemas virtuais usan o mesmo sistema operativo que o nativo (aínda que na maioría dos casos poden instalar distintas distribucións). Esta arquitectura elimina as chamadas do sistema entre capas, o que favorece unha redución importante no uso da CPU. Ademais, ao compartir os ficheiros binarios e librarías comúns do sistema na mesma máquina, a posibilidade de escalado é moito maior, o que permite que un mesmo servidor virtual sexa capaz de dar servizo a un gran número de clientes ao mesmo tempo.

A virtualización do SO mellora o rendemento, a xestión e a eficiencia. Podemos entendelo como un sistema en capas. Na base reside un sistema operativo hóspede estándar. A continuación encontramos a capa de virtualización, cun sistema de arquivos propietario e unha capa de abstracción de servizo de *kernel* que garante o illamento e seguridade dos



recursos entre distintos colectores. A capa de virtualización fai que cada un dos colectores apareza como servidor autónomo. Finalmente, o colector aloxa a aplicación ou a carga de traballo.

Exemplos de sistemas que usan virtualización a nivel de sistema operativo son Virtuozzo e Solaris.

### **463 VIRTUALIZACIÓN DO POSTO CLIENTE**

Esta técnica consiste na separación do ámbito de usuario dun ordenador persoal da máquina física co modelo cliente-servidor. O modelo que segue un servidor para implantar esta característica denomínase VDI (Virtual Desktop Infrastructure, Infraestrutura de Escritorio Virtual), tamén chamada Interface de Escritorio Virtual.

A maioría de implantacións comerciais desta tecnoloxía usan un servidor central remoto para levar a cabo a "virtualización" do escritorio do cliente, en lugar de usar o almacenamento local do cliente remoto. Isto implica que todas as aplicacións, procesos, configuracións e datos do cliente están almacenadas no servidor e execútanse de forma centralizada.

O sistema cliente pode utilizar unha arquitectura de hardware completamente diferente da utilizada polo ámbito de escritorio proxectado, e tamén pode estar baseada nun sistema operativo completamente diferente.

O modelo de virtualización do posto cliente permite o uso de máquinas virtuais para que múltiples subscritores de rede poidan manter escritorios individuais nun único ordenador, o servidor central. Este servidor central pode operar nunha residencia, negocio ou centro de datos. Os usuarios poden estar xeograficamente dispersos, pero todos están conectados á máquina central por unha rede de área local, unha rede de área ampla, ou



a internet.

#### **463.1      *Modos de operación de VDI***

Basicamente existen catro modelos de operación VDI:

- Aloxado (como servizo). Adoitan contratarse provedores comerciais e normalmente proporciona unha configuración do sistema operativo do posto cliente administrado. Os principais subministradores son CITRIX, VMware e Microsoft.
- Centralizado. Neste caso todas as instancias VDI están aloxadas nun ou máis servidores centralizados, os datos están en sistemas de almacenamento conectados a estes. Este modelo á súa vez pode distinguir dous tipos:
  - o VDI estático ou persistente. Existe unha única imaxe de escritorio asignado por cliente e estes deben ser xestionados e mantidos.
  - o VDI dinámico ou non persistente. Existe unha imaxe mestra común para todos os clientes que se clona e personaliza no momento da petición cos datos e aplicacións particulares de cada cliente.
- Remoto (ou sen ataduras). Ten como base o concepto de VDI centralizado pero permite traballar sen a conexión a un servidor central ou á internet. Cópiase unha imaxe ao sistema local e execútase sen necesidade de máis conexión. As imaxes teñen certo período de vida e actualízanse periodicamente. Esta imaxe execútase no sistema local que necesita un sistema operativo e un hipervisor (que executa a instancia VDI). Isto implica que o dispositivo cliente teña maiores necesidades de memoria, espazo en disco, CPU... A vantaxe é a menor dependencia de conexión.

Os modelos aloxado e centralizado necesitan dunha rede que conecte co



servidor onde se executa a instancia VDI. O concepto base deste modelo é similar ao de clientes lixeiros debido a que o cliente só ten que mostrar o escritorio virtual.

No caso do modelo remoto, permítese aos usuarios copiar a instancia VDI no sistema e logo executarase o escritorio virtual sen necesidade de ningún tipo de conexión.

#### **464 COMPUTACIÓN BASEADA EN SERVIDOR**

Tamén coñecida como SBC do inglés Server Based Computing, consiste na separación do procesamento de certas tarefas como a xestión de datos, que será realizado nun servidor central e outras tarefas de procesamento, como a presentación de aplicacións de usuario e impresión de datos no cliente. O único transmitido entre servidor e cliente son as pantallas de información. Esta arquitectura pódelle dar solución aos principais problemas que aparecen cando se executan aplicacións nos clientes. Ademais simplifica procesos como poden ser os contornos hardware, actualizacións de software, despregamento de aplicacións, soporte técnico, almacenamento e respaldo de datos. Centralízase a xestión de todos estes procesos nun único servidor.

Os clientes que actúan nesta arquitectura adoitan chamarse *thin clients*, ou clientes lixeiros, este é un termo xeral para dispositivos que se basean nun servidor para operar. O cliente lixeiro proporciona pantalla, teclado, rato e un procesador básico que interactúa co servidor. Os clientes lixeiros non almacenan ningún dato localmente e requiren de poucos recursos de procesamento. A característica máis destacada destes terminais é a redución de custos asociados co mantemento, administración, soporte, seguridade e instalación de aplicacións comparándoo cun PC tradicional.

Esta tecnoloxía está composta por tres compoñentes principais:

- Sistemas operativos multiusuario que permiten o acceso e execución



de modo concorrente, usando aplicacións diferentes e con sesións de usuario protexidas. Exemplos dalgunhas terminais de servizo son: 2 x 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, 2 X Application Server, HOblink, Propalms TSE (antes Tarantella), Jethro cabina, GraphOn GO-Global, VMware View.

- O cliente lixeiro pódese executar cunha cantidade mínima de software pero necesita polo menos un programa de conexión a servizos de terminal. O cliente lixeiro e o programa de servizos de terminal poden ser executados en sistemas operativos completamente diferentes.
- Un protocolo que lles permita ao programa de servizos de terminal e ao cliente lixeiro comunicarse e enviar as pulsacións de teclado, de rato e as actualizacións de pantalla a través da rede. Os protocolos máis populares son RDP3 (Realtime Desktop protocol), ICA e NX.

Entre as vantaxes da computación baseada en servidor pódense citar:

- Redución dos custos de administración. A xestión de clientes lixeiros está case na súa totalidade centralizada no servidor.
- Redución de custos de hardware. O hardware nos clientes lixeiros é xeralmente máis barato porque non é necesario ter memoria para as aplicacións ou un procesador de grande alcance.
- Seguridade. Pode ser controlada centralmente.
- Menor consumo de enerxía. O hardware especializado no cliente lixeiro ten un consumo moito menor de enerxía que os tradicionais.
- Redución da carga de rede. O tráfico de rede que xeran os terminais lixeiros só é o dos movementos do rato, teclado e información de



pantalla dende / cara ao usuario. No caso de que un cliente pesado abra e gardase un documento xa implicaría o paso deste dous veces pola rede. Usando protocolos eficientes de rede tales como ICA e NX xa é posible usar esta tecnoloxía nun largo de banda de 28,8 Kbps.

- Actualización de hardware simple. Se o uso está por riba dun límite predefinido, é relativamente sinxelo solucionar o problema, abondaría cun disco novo nun rack de servidores, aumentando así o número de recursos exactamente a cantidade necesaria. Se ocorrese isto con clientes pesados habería que substituír un PC completo, o que suporía tanto custos económicos como de recursos humanos.

A pesar do anterior, esta tecnoloxía tamén presenta certos inconvenientes:

- Altos requirimentos de servidor. Ao centrarse a carga de traballo no servidor, o sistema de clientes lixeiros implica maior consumo de recursos nos servidores, mesmo é habitual que se use un gran número de servidores, o que se denomina "granxa de servidores".
- Pobre rendemento multimedia. O envío de datos de audio e vídeo requiren moito largo de banda, polo que estes sistemas son menos útiles para aplicacións multimedia.
- Menos flexibilidade. Non todos os produtos software do mercado poden funcionar correctamente nun cliente lixeiro.

## **465 GRID COMPUTING**

Arquitectura distribuída e paralela, de ámbito extenso xeograficamente, na que se premia a distribución, e a continuación a paralelización. Os seus creadores foron Ian Foster e Carl Kesselman. O seu nome provén do paradigma da rede eléctrica (power grid).

Baséase na compartición, selección e agregación de forma dinámica e en



tempo de execución de recursos autónomos, distribuídos xeograficamente, dependendo de criterios como a dispoñibilidade do hardware, a capacidade transaccional, o rendemento que se poida proporcionar á solución final, o custo e os criterios de calidade do servizo que o demandante poida proporcionar e esixir.

A rede está formada por un conxunto de ordenadores independentes e interconectados que poñen a disposición do *grid* os excedentes do seu procesamento individual, é dicir, os ciclos de reloxo das súas CPU non aproveitados por elas, sen poder superar unha determinada porcentaxe de dedicación configurada individualmente en cada nodo. A partir da porcentaxe proporcionada por cada nodo, virtualízase un recurso computacional único.

Os sistemas baseados en *grid computing* están indicados para atender produtividades sostidas e sustentables, sen poder nunca superar un determinado limiar. Nestes sistemas garántese a escalabilidade como un criterio parametrizable. É posible definir con que criterio engadimos cada novo nodo á solución final.

Actualmente, o único criterio que se ten en conta é a capacidade de procesamento (transaccionalidade), pero no futuro, será posible ter en conta criterios máis finos, referidos á calidade do servizo.

Ademais, estes sistemas están dotados dun comportamento dinámico, segundo o cal, un determinado programa en execución no sistema pode modificar en tempo real o dimensionamento da grid para adaptalo ás súas necesidades.

#### **465.1 Características:**

- Podemos conseguir un máximo aproveitamento dos nodos (100% de utilización da CPU).
- Os nodos non teñen que estar dedicados. Ademais, ao contrario



que no caso do clúster, asegurámonos de que a contribución ao *grid* non vai superar unha determinada porcentaxe de tempo de procesamento en cada nodo.

- Son sistemas heteroxéneos, nos que podemos encontrar diversos HW e SW.
- A escalabilidade parametrizable é a característica máis potente desta arquitectura.

#### **465.2      *Funcionalidades:***

- Localización dinámica de recursos (máquinas con excedente).
- Optimización do acceso a datos, mapeando as estruturas de datos en cachés temporais locais (directorios).
- Autenticación do usuario (usr/pwd, certificados...).
- Monitorización de tarefas e procesos dende calquera nodo da rede, sempre que o usuario teña permisos.
- As máquinas encóntranse en situación paritaria.
- Se é posible, paralelízase. O fundamental é a distribución de procesos debilmente conectados.

#### **465.3      *Arquitectura grid***

Habitualmente descríbese a arquitectura do grid en termos de "capas", executando cada unha delas unha determinada función. Como é habitual neste tipo de enfoque, as capas máis altas están máis preto do usuario, en tanto que as capas inferiores o están das redes de comunicación.

Empezando polos alicerces, encontrámonos coa capa de rede, responsable de asegurar a conexión entre os recursos que forman o grid.

Na parte máis alta está a capa de recursos, constituída polos dispositivos



que forman parte do grid: ordenadores, sistemas de almacenamento, catálogos electrónicos de datos e mesmo sensores que se conecten directamente á rede.

Na zona intermedia está a capa "mediadora", encargada de proporcionar as ferramentas que permiten que os distintos elementos (servidores, almacéns de datos, redes, etc.) participen de forma coordinada nun ámbito grid unificado. Esta capa é a encargada das seguintes funcións:

Encontrar o lugar conveniente para executar a tarefa solicitada polo usuario.

- Optimiza o uso de recursos, que poden estar moi dispersos.
- Organiza o acceso eficiente aos datos.
- Encárgase da autenticación dos diferentes elementos.
- Ocúpase das políticas de asignación de recursos.
- Executa as tarefas.
- Monitora o progreso dos traballos en execución.
- Xestiona a recuperación fronte a fallos.
- Avisa cando remate a tarefa e devolve os resultados.

O ingrediente fundamental do mediador son os metadatos (datos sobre os datos), que conteñen, entre outras cousas, toda a información sobre o formato dos datos e onde se almacenan (ás veces en varios sitios distintos ).

O mediador está formado por moitos programas software. Algúns deses programas actúan como axentes e outros como intermediarios, negociando entre si, de forma automática, en representación dos usuarios do grid e dos provedores de recursos. Os axentes individuais presentan os metadatos



referidos aos usuarios, datos e recursos. Os intermediarios encárganse das negociacións entre máquinas (M2M) para a autenticación e autorización dos usuarios e encárganse de definir os acordos de acceso aos datos e recursos e, no seu caso, o pagamento por eles. Cando queda establecido o acordo, un intermediario planifica as tarefas de cómputo e supervisa as transferencias de datos necesarias para acometer cada traballo concreto. Ao mesmo tempo, unha serie de axentes supervisores especiais optimizan as rutas a través da rede e monitoran a calidade do servizo.

Na capa superior deste esquema está a capa de aplicación onde se inclúen todas as aplicacións dos usuarios, portais e ferramentas de desenvolvemento que soportan esas aplicacións. Esta é a capa que ve o usuario.

Ademais, nas arquitecturas máis comúns do grid, a capa de aplicación proporciona o chamado "serviceware", que recolle as funcións xerais de xestión tales como a contabilidade do uso do grid que fai cada usuario.

Para poder facer todo o anterior, as aplicacións que se desenvolvan para ser executadas nun PC concreto, terán que se adaptar para poder invocar os servizos axeitados e utilizar os protocolos correctos. Igual que as aplicacións que inicialmente se crearon para funcionar illadamente se adaptan para poder ser executadas nun navegador web, o grid requirirá que os usuarios dediquen certo esforzo a "GRIDizar" as súas aplicacións.

Non obstante, unha vez adaptadas ao grid, miles de usuarios poderán usar as mesmas aplicacións, utilizando as capas de mediadores para adaptarse aos posibles cambios no tecido do grid.

## **466 CLOUD COMPUTING**

Modelo que permite acceso a un conxunto compartido de recursos informáticos configurables a través da rede (por exemplo, redes, servidores, almacenamento, aplicacións e servizos) que poden ser



desenvolvidos e despregados rapidamente con mínimo esforzo de xestión ou interacción co provedor de servizos.

Este termo refírese á utilización e ao acceso de múltiples recursos baseados en servidores a través dunha rede. Os usuarios da "nube" poden acceder aos recursos do servidor utilizando un ordenador, netbook, pad computer, smartphone ou outro dispositivo. No cloud computing, o servidor presenta e xestiona as aplicacións; os datos tamén se almacenan de forma remota na configuración da nube. Os usuarios non descargan nin instalan aplicacións no seu sistema, todo o procesamento e almacenamento mantense polo servidor. Os servizos en liña poden ser ofrecidos a partir dun "provedor da nube" ou por unha organización privada.

#### **466.1      *Arquitectura***

Normalmente a arquitectura dos sistemas software implicados no desenvolvemento de cloud computing inclúen múltiples compoñentes denominados "compoñentes cloud" que se comunican mediante mecanismos de baixo acoplamento, tales como as colas de mensaxes.

Os dous compoñentes máis significativos da arquitectura cloud computing coñécense como o frontend e o backend. O frontend é a parte vista polo cliente, é dicir, o usuario do PC. Isto inclúe a rede do cliente e as aplicacións utilizadas para acceder á nube a través dunha interface de usuario, como un navegador web. O backend da arquitectura é a propia nube, que comprende varios ordenadores, servidores e dispositivos de almacenamento de datos.

Dentro desta arquitectura pódense distinguir as seguintes capas:

- Proveedor: Empresa responsable de proporcionar o servizo na nube.
- Cliente: Serán o hardware e software deseñados para cloud computing, que permiten interactuar cos servizos remotos.



- **Aplicación:** Son os servizos na nube ou "Software as a Service" (SaaS), o software proporciónase a través da internet coma se fose un servizo. Deste modo evítase a necesidade de instalar e executar no equipo do cliente a aplicación. Redúcense así o mantemento e o apoio.
- **Plataforma:** Son os servizos de plataforma na "nube", tamén coñecidos como "Platform as Service" (PaaS), proporcionan unha plataforma de procesamento e unha pila de solucións como un servizo, constitúen a base e infraestrutura das aplicacións da nube. Facilita o desenvolvemento de aplicacións evitando o custo e a complexidade de comprar e manter o hardware e as capas de software de base.
- **Infraestrutura.** Servizos de infraestrutura, tamén coñecidos como "Infrastructure as a Service" (IaaS), proporciona a infraestrutura como un servizo, adoita ser unha plataforma virtualizada. En lugar de comprar servidores, software, centro de datos especiais ou equipos de rede, os clientes adquiren eses recursos de servizos externos. A IaaS evolucionou a partir das ofertas de servidores virtuais privados.

#### **466.2      *Modelos de implantación***

- **Nube pública ou external cloud:** É o concepto tradicional onde os recursos se presentan a través da internet en función da demanda, a través de aplicacións ou servizos web.
- **Nube da comunidade:** Dáse cando varias organizacións coas mesmas necesidades comparten recursos. Neste caso existen menos usuarios que na nube pública e ofrécese maior privacidade e seguridade. Un exemplo pode ser o Google's "Gov Cloud".
- **Nube híbrida.** É común que unha empresa use tanto a nube pública como desenvolvementos privados para satisfacer as súas necesidades con respecto ás TI. Existen varias empresas como HP, IBM, Oracle and VMware que ofrece tecnoloxías para manexar a complexidade de



mantemento, seguridade e privacidade consecuencia do uso do conxunto destes servizos.

- Nube combinada. Denomínase ao conxunto formado varios servizos de nube de distintos provedores.
- Nube privada. É trasladar o concepto de nube pública a unha rede de uso privado. É dicir, o uso da nube única e exclusivamente dentro da rede dunha empresa.

#### **467 GREEN IT E EFICIENCIA ENERXÉTICA**

O termo Green Computing acuñouse posiblemente por primeira vez tras o inicio do programa Energy Star en 1992, promovido polo goberno estadounidense.

Tiña por obxectivo etiquetar monitores e equipamento electrónico caracterizados pola súa eficiencia enerxética. O termo quedou rexistrado xa en 1992 nun grupo de noticias. Hoxe en día o programa Energy Star é o motor da eficiencia enerxética nos sistemas electrónicos (non só de procesamento da información, senón tamén do equipamento electrónico doméstico).

A adopción de produtos e aproximacións máis eficientes poden permitir máis equipamento dentro do mesmo gasto enerxético, o que se denomina impresión enerxética, ou energy footprint. As regulacións estanse a multiplicar e poderían limitar seriamente as empresas á hora de construír centros de procesamento datos, xa que o efecto das redes de subministración eléctrica, as emisións de carbono polo incremento de uso e outros impactos ambientais están sendo investigadas. Polo tanto, as organizacións deben considerar as regulacións e ter plans alternativos para o crecemento dos seus centros de procesamento de datos e da súa capacidade.



Co paso dos anos, o número de servidores existentes en todo o mundo crece de forma case exponencial. Consecuencia disto é o crecente gasto enerxético para a refrixeración e xestión dos equipos. Hoxe en día xa se están empezando a formular solucións que optimicen este gasto enerxético.

Este consumo enerxético non é o único problema ambiental relacionado coas TI. A etapa de fabricación de equipos presenta serios problemas relacionados co medio: materiais de refugallo tóxicos, produción de gases contaminantes, etc. A tendencia actual é a de minimizar o impacto contaminante (*carbon footprint*) presente nas tecnoloxías de fabricación dos sistemas electrónicos.

Finalmente, tamén ten un impacto inmediato a eliminación de equipos para as TI, caracterizados por un tempo de vida incrivelmente breve duns dous ou tres anos. Se non se reciclan de forma eficiente, rematan tirados en vertedoiros, e debido á presenza de compoñentes tóxicos, son unha fonte de contaminación terrestre e das augas. Todos estes aspectos deben ser considerados de xeito global polos fabricantes e usuarios de equipos TI. A concienciación da existencia deste problema levou á elaboración de numerosas e ríxidas normativas a todos os ámbitos, o que empeza a obter algúns resultados.

GreenPeace Internacional elabora unha listaxe cos 18 principais fabricantes do sector electrónico (ordenadores persoais, teléfonos móbiles etc.) de acordo coas súas políticas de redución de emisións tóxicas, reciclado ou minimización de impacto no cambio climático, e publícaa na súa Guía para a Electrónica Verde (Guide to Greener Electronics), de publicación trimestral. Como se pode ver nos resultados de decembro de 2010, as empresas do sector obteñen unhas cualificacións realmente baixas, das que a mellor é Nokia cun 7,5 sobre 10.

A metade destas 18 empresas suspenden un estudo que busca que as



empresas analizadas:

- Limpen os seus produtos ao eliminar substancias perigosas. Os produtos químicos perigosos con risco impiden a posterior reciclaxe dos equipos.
- Reciclen equipos/productos baixo a súa responsabilidade despois de quedaren obsoletos.
- Reduzan o impacto climático debido ás súas operacións e produtos.

Por todo o exposto, a resolución efectiva do impacto ambiental das tecnoloxías TI require un enfoque holístico do problema que englobe as catro vías:

- Utilización ecolóxica: principalmente a través da redución do consumo enerxético. A produción de enerxía eléctrica é a principal fonte de xeración de gases de efecto invernadoiro.
- Deseño ecolóxico ou ecodeseño: inclúe deseño de equipos máis eficientes enerxeticamente e respectuosos co medio.
- Fabricación ecolóxica: eliminando completamente ou minimizando o impacto do proceso de fabricación no medio (emisións, materiais de refugallo, etc.).
- Eliminación ecolóxica: unha vez finalizado o período de utilización dun equipo débense poñer en marcha as estratexias denominadas tres R: reutilización e renovación de equipos e, se non son aproveitables, reciclaxe.

A idea principal do enfoque holístico é que se peche o ciclo de vida dos equipos TI de forma que non se prexudique o medio, o que permitiría conseguir unha mellora substancial de cara ao desenvolvemento sustentable.



### **467.1      *Tecnoloxías verdes***

Hoxe en día existen distintos enfoques tecnolóxicos que se achegan a un desenvolvemento sustentable das TI.

- **Monitores LCD.** Co paso dos anos os monitores pasaron de ser CRT a LCD, este cambio non é só estético ou de tamaño, senón que os niveis de consumo diminuíron notablemente. Un monitor CRT medio require 85 W se está activo, fronte aos 15 W dun LCD, 5 W en modo baixo consumo para un CRT mentres que un LCD consumiría 1,5W. Apagados ambos os dous consumirían 0,5W. Nos últimos anos revolucionouse o mercado das pantallas de ordenador coa aparición da tecnoloxía OLED (Organic Light Emitting Diode), baseadas na utilización de díodos LED nos que a capa electroluminiscente se fai cun composto orgánico (un polímero que se ilumina ao aplicarlle unha voltaxe). A vantaxe principal deste tipo de pantallas fronte ás tradicionais de cristal líquido (LCD) é que os díodos OLED non necesitan retroiluminación, polo que o consumo de enerxía que requiren é moi inferior.
- **Discos duros.** O consumo dos discos duros non é para nada desprezable, sobre todo no arranque do sistema. Por exemplo, o disco Seagate Barracuda 7200.8 require ata 2,5 A da liña de alimentación de 12 V. Se a isto lle sumamos 3 W que extrae dende a liña de +5 V pódese chegar a un consumo de pico no arranque de 33 W. Se en lugar de só un disco duro falamos dun equipo con dous ou máis empezamos a falar de cifras moi comprometidas. Isto fixo que os fabricantes de discos duros comecen a ter en conta o consumo nos seus produtos, e creen case todos unha nova gama denominada "verde" ou "ecolóxica"; por exemplo, Western Digital con "Caviar Green", Samsung con Eco Green, ou Hitachi con eco-friendly Deskstar e Travelstar. Como alternativa aos discos tradicionais aparecen os discos en estado sólido (SSD), que presentan menores consumos de enerxía e é a tecnoloxía á que se espera que evolucionen os sistemas de almacenamento.



- CPD. Aquí é onde se aloxa toda a infraestrutura de soporte aos diversos servizos computacionais, e unha estrutura axeitada permitirá bos aforros de enerxía, de espazo e de custos a medio e/ou longo prazo. Buscando a redución de enerxía pódese empezar pola acción máis simple que é apagar o equipo que non se estea a utilizar, a redución do hardware estudando necesidades reais, ou actuacións específicas en función da actividade da empresa.
- Virtualización. A virtualización de servidores permite o funcionamento de múltiples servidores nun único servidor físico. Isto axuda a reducir o impacto contaminante do centro de datos ao diminuír o número de servidores físicos e consolidar múltiples aplicacións nun único servidor co cal se consome menos enerxía e se precisa menos arrefriamento. Ademais lógrase un maior índice de utilización de recursos e aforro de espazo.
- Cliente/Servidor. Estes sistemas manteñen o software, as aplicacións e os datos no servidor. Pódese ter acceso á información dende calquera situación e o cliente non require moita memoria ou almacenamento. Este ambiente consume menos enerxía e arrefriamento.
- Cloud computing. Isto proporciónalles aos seus usuarios a posibilidade de utilizar unha ampla gama de recursos en rede para completar o seu traballo. Ao utilizar computación na nube as empresas vólvense máis ecolóxicas porque diminúen o seu consumo de enerxía ao incrementar a súa capacidade sen necesidade de investir en máis infraestrutura.
- Teletraballo. Definido por Merriam-Webster como o traballo na casa co uso dunha ligazón electrónica coa oficina central. Ao non se desprazar o empregado, a contaminación é menor.



## **467.2      *Actividades relacionadas coas tecnoloxías verdes***

Existen varias actividades que promoven e intentan solucionar as cuestións expostas anteriormente. Estas actividades están patrocinadas ben dende administracións públicas, ben dende empresas, que están a entender que as tecnoloxías verdes, ademais dunha necesidade, poden ser un negocio, dende o punto de vista de consultoría e servizos, ou ben por consorcios de empresas.

The Green Grid (<http://www.thegreengrid.org>) é un consorcio global dedicado a avanzar na eficiencia enerxética dos centros de procesamento de datos e en ecosistemas de computación de negocio. En cumprimento da súa misión, The Green Grid céntrase en:

- Definir métricas e modelos significativos e centrados no usuario.
- Desenvolver estándares, métodos de medida, procesos e novas tecnoloxías para mellorar o rendemento dos centros de procesamento de datos fronte ás métricas definidas.
- Promover a adopción de estándares, procesos, medidas e tecnoloxías enerxeticamente eficientes.

O comité de directores de The Green Grid está composto polas seguintes compañías membros: AMD, APC, Dell, HP, IBM, Intel, Microsoft, Rackable Systems, Sun Microsystems e VMware.

Climate Savers. Iniciada por Google e Intel en 2007, Climate Savers Computing Initiative ([www.climatesaverscomputing.org](http://www.climatesaverscomputing.org)) é un grupo sen ánimo de lucro de consumidores e negocios con conciencia ecolóxica e organizacións conservacionistas. A iniciativa iniciouse baixo o espírito do programa Climate Savers de WWF (<http://www.worldwildlife.org/climate/projects/climateSavers.cfm>), que mobilizou unha ducia de compañías dende 1999 a recortar as emisións de dióxido de carbono, demostrando que reducir as emisións é bo para o



negocio. O seu obxectivo é promover o desenvolvemento, despregamento e adopción de tecnoloxías intelixentes que poidan mellorar a eficiencia de uso da enerxía do computador e reducir o seu consumo cando o computador se encontra inactivo.

SNIA (Storage Networking Industry Association, <http://www.snia.org>) é unha organización global sen ánimo de lucro composta por unhas compañías da industria do almacenamento. SNIA Green Storage Initiative (<http://www.snia.org/green>) está a levar a cabo unha iniciativa para avanzar no desenvolvemento de solucións enerxeticamente eficientes para o almacenamento en rede, incluíndo a promoción de métricas estándares, a formación e o desenvolvemento de boas prácticas enerxéticas ou o establecemento de alianzas con organizacións como The Green Grid.

Energy Star. En 1992 a Axencia de Protección Ambiental dos EUA (U.S. Environmental Protection Agency) lanzou o programa Energy Star, que se planificou para promover e recoñecer eficiencia enerxética en monitores, equipos de climatización e outras tecnoloxías. Aínda que de carácter voluntario inicialmente, resultou pronto de ampla aceptación, pasando a ser un feito a presenza dun modo de espera (sleep mode) na electrónica de consumo.

Directiva europea de ecodeseño. Seguindo a mesma liña que a iniciativa Energy Star dos EUA, a Unión Europea aprobou a Directiva 2005/32/EC para o ecodeseño, novo concepto creado para reducir o consumo de enerxía de produtos que a requiren, tales como os dispositivos eléctricos e electrónicos ou electrodomésticos. A información relacionada coas prestacións ambientais dun produto debe ser visible de forma que o consumidor poida comparar antes de comprar, o cal está regulado pola Directiva de etiquetaxe da Enerxía (Energy Labelling Directive). Os produtos aos que se conceda a Ecoetiqueta serán considerados como cumpridores coa implantación das medidas, de forma moi similar á etiqueta de Energy Star.



O Código de conduta da Unión Europea para centros de datos está sendo creado como resposta ao crecente consumo de enerxía en centros de datos e á necesidade de reducir o impacto ambiental, económico e de seguridade de abastecemento enerxético relacionado. O obxectivo é informar e estimular os operadores ou propietarios dos centros de datos a que reduzan o consumo de enerxía dunha forma rendible sen dificultar o seu funcionamento. Este código de conduta quere conseguir isto mediante a mellora da comprensión da demanda de enerxía dentro do centro de datos, aumentando a concienciación, e mediante a recomendación de prácticas e obxectivos enerxeticamente eficientes.

Grupo de Traballo de Green IT da plataforma INES (Iniciativa Española de Software e Servizos, <http://www.ines.org.es>) é a plataforma tecnolóxica española na área dos sistemas e servizos software e constitúe unha rede de cooperación científico-tecnolóxica integrada polos axentes tecnolóxicos relevantes deste ámbito (empresas, universidades, centros tecnolóxicos, etc.).

Segundo a Axenda Estratéxica de Investigación de INES, o plan de dinamización para o Grupo de Traballo de Green IT consiste nas seguintes accións:

- Análise da influencia e importancia das solucións tecnoloxías verdes.
- Difusión das informacións, noticias e existencia deste grupo de traballo pola internet.
- Fomentar o interese e apoiar o desenvolvemento baixo tecnoloxías verdes.

Big Green Innovations (<http://www.ibm.com/technology/greeninnovations/>) é un programa de IBM. Dentro deste programa, e con fins educativos, IBM presentou un centro de datos virtual ecolóxico denominado Virtual Green Data Center.



A lista Green500 (<http://www.green500.org>) proporciona unha clasificación dos supercomputadores máis eficientes enerxeticamente do mundo, e serve como unha visión complementaria á lista Top500 (<http://www.top500.org>).

Outras empresas, como Google, Dell ou Symantec, están a desenvolver programas de eficiencia enerxética, tanto para os seus propios procesos de TI coma para os dos seus clientes.

#### **468 REDES SAN E ELEMENTOS DUNHA SAN**

Como resumo, unha SAN é unha rede onde se realiza o almacenamento e se xestiona a seguridade dos datos. As SAN (Storage Area Network, redes de almacenamento) son redes nas que se conectan servidores de almacenamento (especialmente arrays de discos). Tamén hai que considerar como parte das SAN as librarías necesarias para o uso dos arrays e os accesos ás redes. De forma contraria ás redes tradicionais, nas SAN empréganse protocolos orientados á recuperación da información dos arrays de disco e inspirados nos propios estándares de comunicación con discos tradicionais (SCSI e SATA).

Normalmente os equipos deseñados para participar nestas redes adoita ser especialmente caro aínda que o seu prezo depende, nunha grande medida, das tecnoloxías e protocolos empregados para a transmisión dos datos. Entre as tecnoloxías dispoñibles na actualidade encóntranse: iSCSI (Internet Small Computer Storage Interconnect), Fibre Channel e AOE (ATA Over Ethernet, Advanced Technology Attachment Over Ethernet).

Entre as vantaxes da interconexión de redes de almacenamento resáltanse



as seguintes:

- Elimina os límites de distancia de discos introducidos por SCSI ou ATA
- Consegue un maior caudal de datos xa que os protocolos están especificamente deseñados para a transferencia de datos de dispositivos de almacenamento.
- Permite un aproveitamento maior dos discos permitindo que máis dun servidor acceda ao mesmo disco.
- Capacidade para o uso de múltiples discos de forma transparente dende un ou varios servidores.
- Adquisición de discos diferida debido ao maior aproveitamento
- Capacidades de recuperación ante desastres. Os arrays de discos empregados nas SAN adoitan dispoñer de discos de reserva (para fallos doutros discos) e permitir distintos esquemas de RAID.
- Recuperación en quente ante desastres
- Mellor capacidade de administración. A administración é máis sinxela e está máis centralizada.
- Redución dos custos de administración e de almacenamento de datos
- Mellora de dispoñibilidade global xa que as SAN teñen menos fallos que os discos internos dos equipos.
- Redución de servidores eliminando servidores de arquitecturas antigas (NFS, SMB, etc).
- Redución do caudal das redes convencionais, xa que as copias de seguridade se poden facer dende as SAN.
- Incremento da rapidez das operacións de entrada/saída
- Redución dos custos de administración de *backups*
- Protección de datos críticos
- Incremento da capacidade de forma transparente
- Desenvolvemento e proba de aplicacións de forma máis eficiente mediante o uso de copias dos datos de produción realizadas na SAN.



- Facilita o emprego de clúster de servidores que teñen que dispoñer dun almacenamento común.
- Permite o almacenamento baixo demanda de forma que calquera servidor pode solicitar espazo de almacenamento segundo as súas necesidades.

Dentro dunha organización, debería incluír nunha SAN a seguinte información:

- A información almacenada por SXBD (sistemas xestores de bases de datos). De feito, algúns sistemas xestores como Oracle, Sybase, SQLServer, DB2, Informix ou Adabase recomendan esta alternativa
- A información almacenada por servidores de arquivos. Os servidores de arquivos funcionarán mellor e con menos recursos se os arquivos están almacenados nunha SAN.
- Servidores de backup. Se os servidores de backup están conectados a unha SAN conseguíase reducir os tempos de copia de seguridade con respecto a facelos nunha LAN (Local Area Network, rede de área local) e reducir o tráfico da LAN.
- Arquivos de servidores de voz e vídeo para *streamming*. Debido a que este tipo de servizos require grandes cantidades de disco, unha SAN pode reducir os custos asociados ao almacenamento e desprazar o máximo posible o custo (incluír novos discos nos arrays cando sexan necesarios).
- Caixas de correo de usuario (mailboxes) de servidores de correo permitindo que os servidores de correo funcionen mais rápido e que se poida realizar unha restauración rápida en caso de que algún arquivo se corrompa.
- Servidores de aplicacións de alto rendemento. As SAN poden mellorar o rendemento de calquera aplicación incluíndo xestores documentais, aplicacións científicas, aplicacións de datawarehouse e cadros de mando



integrais, aplicacións para xestionar as relacións cos clientes (CRM), etc.

- Solucións de Virtualización.

Así mesmo, non é conveniente usar unha SAN para:

- Servidores web que non requiran grandes necesidades de almacenamento (a maioría)
- Servidores con servizos de rede básicos como DNS, DHCP, WINS (Windows Internet Name Servers) e controladores de dominio de Windows (DC). Este tipo de servidores non requiren das capacidades de almacenamento permitidas polas SAN
- PC de escritorio
- Servidores que necesitan menos de 10 xigabytes de almacenamento
- Servidores que non necesitan un acceso rápido á información
- Servidores que non comparten arquivos

#### **468.1      *Estrutura das SAN***

Habitualmente as SAN concíbense e estrutúranse en tres capas:

1. A capa de ordenadores anfitrións: Constituída na súa maioría polos servidores, os drivers e software necesarios para a conexión á rede e os HBA (Host Bus Adapters) que son dispositivos (tarxetas) que se conectan a cada servidor para acceder ao almacenamento (nalgúns solucións concretas son adaptadores Ethernet simples e no caso Fibre Channel levan un conector GBIC-Gigabit Interface Connector).
2. A capa de estrutura (fabric layer): Constituída por concentradores, conmutadores, pasarelas e encamiñadores se fose necesario. Se se emprega a tecnoloxía Fibre Channel, todos estes dispositivos empregan



GBICs (Gigabit Interface Conectors) para a interconexión dos dispositivos das capas superiores e inferiores.

3. A capa de almacenamento (storage layer): Constituída por todo tipo de dispositivos de almacenamento.

Un conxunto de discos situados no mesmo sitio e sen funcionalidades adicionais coñécese como JBOD (Just a Bunch Of Disks). Dentro da capa de almacenamento, os arrays non son simplemente JBOD, senón que inclúen certas funcionalidades interesantes implantadas no firmware da controladora como o RAID.

## **469 VIRTUALIZACIÓN DO ALMACENAMENTO**

Este tipo de virtualización permite unha maior funcionalidade e características avanzadas no sistema de almacenamento. Consiste en abstraer o almacenamento lóxico do almacenamento físico e adoita usarse nas SAN (Storage Area Network, rede de área de almacenamento).

Este sistema de almacenamento coñécese tamén como "storage pool", matriz de almacenamento, matriz de disco ou servidor de arquivos. Estes sistemas adoitan usar hardware e software especializado, xunto con unidades de disco co fin de proporcionar un almacenamento moi rápido e fiable para o acceso a datos. Son sistemas complexos, e poden ser considerados como un ordenador de propósito especial deseñado para proporcionar capacidade de almacenamento xunto con funcións avanzadas de protección de datos. As unidades de disco son só un elemento dentro do sistema de almacenamento, xunto co hardware e o software de propósito especial incorporado no sistema.

Os sistemas de almacenamento poden ser de acceso por bloques, ou



acceso por ficheiros. O acceso por bloques adoita levarse a cabo por medio de Fibre Channel, iSCSI, SAS, FICON ou outros protocolos. Para o acceso por arquivos úsanse os protocolos NFS ou CIFS.

Dentro deste contexto podémonos encontrar con dous tipos principais de virtualización: a virtualización por bloques e a virtualización por arquivos.

### **469.1      *Virtualización por bloques***

Este tipo de virtualización baséase na abstracción (diferenciación) entre o almacenamento lóxico e o almacenamento físico, conseguindo que o acceso non teña en conta o almacenamento físico ou estrutura heteroxénea.

Existen tres tipos de virtualización por bloques: baseada en ordenador anfitrión, baseada en dispositivos de almacenamento, baseada en rede.

#### **469.1.1      Virtualización baseada en ordenador anfitrión**

Esta virtualización require software adicional que se executa no ordenador anfitrión. Nalgúns casos a administración de volumes está integrada no sistema operativo, e noutros casos ofrécese como un produto separado. Os volumes (LUN) dispoñibles no sistema son manexados por un controlador de dispositivos físicos tradicional. Por riba deste controlador encóntrase unha capa software (o xestor de volumes) que intercepta as peticións de E/S, e proporciona a busca de metadatos e mapeamentos de E/S.

Os sistemas operativos máis modernos teñen algún tipo de xestor de volumes lóxicos integrado (MVI en UNIX / Linux, ou administrador de discos lóxicos ou LDM en Windows), que realiza tarefas de virtualización.

Existen varias tecnoloxías que implantan este tipo de virtualización, como poden ser a xestión de volumes lóxicos (Logical Volume Management, LVM ), os sistemas de arquivos (CIFS, NFS) ou a montaxe automática (autofs)



#### 469.1.2 Virtualización baseada en dispositivos de almacenamento

Pódese levar a cabo a virtualización baseada en medios de almacenamento masivo utilizando un controlador de almacenamento primario que proporcione os servizos de virtualización e permita conexión directa dos controladores de almacenamento. En función da implantación é posible usar modelos de distintos fabricantes.

O controlador primario proporcionará a posta en común e os metadatos de servizo de xestión. Tamén pode ofrecer servizos de replicación e migración a través dos controladores que se virtualizan.

Unha nova xeración de controladores de serie do disco permite a inserción posterior dos dispositivos de almacenamento.

Os sistemas RAID poden ser un exemplo desta técnica. Estes sistemas combinan varios discos nunha única matriz.

As matrices avanzadas de disco contan a miúdo con clonación, instantáneas e replicación remota. En xeral, estes dispositivos non ofrecen os beneficios da migración de datos ou de replicación a través de almacenamento heteroxéneo, xa que cada fabricante tende a utilizar os seus propios protocolos propietarios.

#### 469.1.3 Virtualización baseada en rede

Esta é unha virtualización de almacenamento operando nun dispositivo baseado en rede (polo xeral un servidor estándar ou un smart switch) e o uso de redes iSCSI ou FC de Fibre Channel para conectar como SAN (Storage Area Network). Este é o tipo de virtualización de almacenamento máis común.

O dispositivo de virtualización encóntrase na SAN e proporciona a capa de abstracción entre os ordenadores anfitrións, que permiten a entrada/saída, e os controladores de almacenamento, que proporcionan capacidade de almacenamento.



Hoxe en día existen dúas implantacións distintas, a baseada no **dispositivo** e a baseada en **conmutación**. Ambos os dous modelos proporcionan os mesmos servizos: xestión de discos, busca de metadatos, migración e replicación de datos. Igualmente, ambos os dous modelos necesitan dun hardware específico que permita ofrecer eses servizos.

A baseada en dispositivos consiste en establecer o hardware especializado entre os anfitrións e a parte de almacenamento. As solicitudes de entrada/saída rediríxense ao dispositivo, que realiza a asignación de metadatos mediante o envío das súas propias ordes de E/S á solicitude de almacenamento subxacente. O hardware usado tamén pode proporcionar almacenamento de datos en caché, e a maioría das implantacións proporcionan algún tipo de agrupación de cada un dos dispositivos para manter un punto de vista atómico tanto dos metadatos coma dos datos da caché.

Este tipo de almacenamento tamén pode clasificarse en in-band (simétrica) ou out-of-band (asimétrica).

#### **469.1.3.1 In-band (simétrica)**

Neste caso os dispositivos de virtualización aséntanse entre o ordenador anfitrión e o almacenamento. Todas as peticións de E/S e datos pasan a través do dispositivo. Os anfitrións nunca interactúan co dispositivo de almacenamento senón co dispositivo de virtualización.

#### **469.1.3.2 Out-of-band (asimétrica)**

Os dispositivos usados neste tipo de virtualización tamén son chamados servidores de metadatos. A única finalidade destes dispositivos é proporcionar a asignación de metadatos. Isto implica o uso de software adicional no anfitrión, que é coñecedor da situación real dos datos. Deste modo, intercétase a petición antes de que saia do anfitrión, solicítase



unha busca de metadatos no servidor (pode ser a través dunha interface que non sexa SAN) e devólvese a situación real dos datos solicitados polo anfitrión. Finalmente recupérase a información a través dunha solicitude de E/S común ao dispositivo de almacenamento. Non se pode dar un almacenamento en caché xa que os datos nunca pasan a través do dispositivo de virtualización.

### **469.2      *Virtualización por arquivos***

Con este tipo de virtualización preténdese eliminar as dependencias entre o acceso a datos por arquivos e a situación física deles. Esta técnica, coñecida como NAS (Network-Attached Storage) ou almacenamento conectado á rede, adoita ser un equipo especializado pensado exclusivamente para almacenar e servir ficheiros. Os equipos que funcionan como dispositivo NAS adoitan incluír un sistema operativo específico para o propósito, como pode ser FreeNAS ou FreeBSD.

Estes sistemas poden conter un ou máis discos duros, dispostos a miúdo en colectores lóxicos redundantes ou arrays RAID.

NAS utiliza protocolos baseados en arquivos como NFS (sistemas UNIX), SMB / CIFS (Server Message Block/Common Internet File System) (sistemas MS Windows), ou AFP (Apple Filing Protocol, sistemas Apple Macintosh). As unidades NAS non adoitan limitar os clientes a un único protocolo. FTP, SFTP, HTTP, UPnP, rsync e AFS (Andrew File System) tamén o soportan.

Deste modo conséguese optimizar a utilización do almacenamento e as migracións de arquivos sen interrupcións.

### **469.3      *Diferenzas entre NAS e SAN***

NAS proporciona almacenamento e un sistema de arquivos, o que adoita contrastar con SAN, que soamente proporciona almacenamento baseado en bloques e deixa do lado do cliente a xestión do sistema de arquivos.

NAS aparece no sistema cliente como un servidor de arquivos (pódense



asignar unidades de rede ás accións do servidor) mentres que un disco a través dunha SAN se presenta ao cliente como un disco máis do sistema operativo, que podemos montar, desmontar, formatar...

	<b>NAS</b>	<b>SAN</b>
<b>Tipo de datos</b>	Arquivos compartidos	Datos por bloques, por exemplo, bases de datos.
<b>Cablaxe utilizada</b>	Ethernet LAN	Fibre Channel dedicado
<b>Cientes principais</b>	Usuarios finais	Servidores de aplicacións
<b>Acceso a disco</b>	A través do dispositivo NAS (IP propia)	Acceso directo

## **4610 XESTIÓN DO CICLO DE VIDA DA INFORMACIÓN (ILM)**

### **4610.1 Xestión do ciclo de vida dos datos**

A xestión do ciclo de vida dos datos ou DLM (Data Lifecycle Management) é un enfoque da xestión da información dende o punto de vista do manexo do fluxo dos datos dun sistema de información durante todo o seu ciclo de vida, dende que se crean e se produce o seu primeiro almacenamento, ata que son declarados obsoletos e eliminados do sistema.

Os produtos para a xestión do ciclo de vida dos datos tratan de automatizar os procesos que forman parte deste ciclo de vida. Organizan os datos en distintos niveis seguindo unhas políticas especificadas, e automatizan a migración ou intercambio dos datos entre uns niveis e outros baseándose para iso nos criterios especificados de cada un.

Como norma xeral, os datos máis recentes e aqueles aos que se accede con máis frecuencia téndense a almacenar en medios de almacenamento máis rápidos, pero tamén máis caros, mentres que os datos dun nivel



menos crítico almacénanse nos dispositivos máis baratos e máis lentos.

As arquitecturas que xestionan o ciclo de vida dos datos adoitan incluír un sistema de arquivos que indexa toda aquela información crítica e aquela considerada non tan crítica pero que garda a mesma relevancia ou relación que esta. Con esta información crea copias de respaldo, almacénnaas en localizacións seguras para evitar manipulacións pero que poidan ser accesibles dun xeito seguro e fiable.

Estas arquitecturas tamén se encargan das posibles duplicacións de datos e da comprensión destes para garantir un correcto e eficiente uso do espazo de almacenamento dispoñible.

Desafortunadamente, moitas implantacións de DLM de negocios estancáronse, principalmente porque as empresas non lograron definir nin as políticas de migración axeitadas nin o arquivado de datos. Dado que esas políticas necesitan reflectir as prioridades de regulación e de negocio, nas súas definicións é necesario unha colaboración que involucre non só membros do departamento de tecnoloxías da información, senón tamén a membros de diferentes departamentos do negocio.

Por outro lado, o criterio máis sinxelo para realizar unha migración da información a un sistema de almacenamento máis económico é o temporal, é dicir, os datos máis antigos nos sistemas máis lentos e baratos. Non obstante, as empresas en industrias altamente reguladas a miúdo queren ir máis lonxe, establecendo a clasificación dos datos en función da rapidez coa que se precisen, ou a frecuencia coa que se accede a eles, ou sobre a base de quen os enviou ou os recibiu, ou sobre a base dun conxunto de palabras clave ou cadeas numéricas, etc. Daquela o reto está en conseguir definilos de tal maneira que sexa viable realizalo no tempo e mediante a menor intervención humana.



## **4610.2 Xestión do ciclo de vida da información**

A xestión do ciclo de vida da información ou ILM (Information Lifecycle Management) é un enfoque integral para o manexo do fluxo dos datos dun sistema de información e os metadatos asociados dende a súa creación e almacenamento inicial ata o momento en que estes se volven obsoletos e son borrados.

A diferenza de anteriores enfoques para a xestión de almacenamento de datos, ILM abrangue todos os aspectos nos que se tratan os datos, partindo das prácticas dos usuarios, en lugar da automatización dos procedementos de almacenamento e en contraste cos sistemas máis antigos, ILM permite criterios moito máis complexos para a realización da xestión do almacenamento que a antigüidade dos datos ou a frecuencia de acceso a eles.

É importante destacar que ILM non é só unha tecnoloxía senón que integra os procesos de negocio e TI co fin de determinar como flúen os datos a través dunha organización, permitíndolles aos usuarios e administradores xestionar os datos dende o momento que crean ata o instante no que xa non son necesarios.

Aínda que os termos xestión do ciclo de vida dos datos (DLM) e xestión do ciclo de vida da información (ILM) se utilizan ás veces indistintamente, ILM considérase un proceso máis complexo.

A clasificación dos datos en función de valores do negocio é unha parte integral e moi importante do proceso ILM. Isto quere dicir que ILM recoñece que a importancia dos datos non se basea unicamente na súa antigüidade ou na súa frecuencia de acceso, senón que ILM espera que sexan os usuarios e os administradores os que especifiquen distintas directivas para que os datos vaian variando dun xeito decrecente a súa relevancia ou grao de importancia para a organización, ou que poidan conservar a súa importancia durante todo o seu ciclo de vida, etc.



Para unha implantación eficiente e con éxito de IML necesítase que a organización identifique requisitos de seguridade dos datos críticos e incluílos nos seus procesos de clasificación. Os usuarios dos datos, tanto os individuos coma as aplicacións, deben de ser identificados e categorizados en función das necesidades asociadas coas súas tarefas.

Algunhas das mellores prácticas relacionadas coa implantación de IML comparten enfoques como:

- Céntranse na produtividade do usuario co fin de obter unha vantaxe estratéxica a través do acceso aos datos necesarios.
- Protexer os datos contra o roubo, a mutilación, a divulgación involuntaria, ou a eliminación.
- Crear múltiples capas de seguridade, sen crear unha xestión excesivamente complexa.
- Asegurarse que os procesos de seguridade están incorporados nos procesos xerais do negocio e nos procesos de TI.
- Utilizar estándares e modelos de referencias co fin de satisfacer unicamente as necesidades de seguridade da organización.

Por suposto, cada organización deberá desenvolver e implantar a súa propia solución de seguridade de almacenamento, que debe seguir evolucionando, adaptándose ás novas oportunidades, ameazas e capacidades.



### **4610.3    *Algunha solucións para a xestión***

#### **4610.3.1    Microsoft**

Microsoft Identity Lifecycle Manager ofrece unha solución integrada e completa para a xestión do ciclo de vida das identidades de usuario e as súas credenciais asociadas. Esta solución proporciona a sincronización de identidades, os certificados e administración de contrasinais e subministración de usuarios. A solución funciona baixo plataformas Windows e outros sistemas organizacionais.

#### **4610.3.2    IBM**

As solucións de IBM para a xestión do ciclo de vida da información agrupáronse en cinco categorías (IBM, 2008):

- *Arquivo de correo electrónico* (IBM DB2 CommonStore, VERITAS Enterprise Vault, OpenText-IXOS Livelink)
- *Aplicación e base de datos de arquivo* (Arquivo Activo de Princeton Softech),
- *Xestión do ciclo de vida dos datos* (TotalStorage de IBM SAN File System)
- *Xestión de contidos* (repositorio de administración de contido, DB2 Content Manager)
- *Xestión de rexistros* (IBM DB2 Records Manager).

#### **4610.3.3    Oracle**

Oracle ILM Assistant é unha ferramenta que se basea nunha interface gráfica de usuario para a xestión de ámbito de ILM. Ofrece a posibilidade



de crear definicións de ciclo de vida, que se asignan ás táboas na base de datos. Posteriormente baseándose nas políticas establecidas sobre o ciclo de vida, ILM Assistant informa cando é o momento para mover, archivar ou suprimir os datos. Tamén mostra as necesidades de almacenamento e o aforro de custos asociados co cambio de situación dos datos.

Outras capacidades de Oracle ILM Assistant inclúen a habilidade de mostrar como realizar particións nunha táboa baseada nunha definición do ciclo de vida, e poder simular os eventos para comparar o resultado en caso de que a táboa fose particionada.

## **4611 SISTEMAS DE BACKUP: HARDWARE E SOFTWARE DE BACKUP**

Un factor importante en todo sistema de backup é a elección dos sistemas hardware e software que o compoñen.

### **4611.1 Hardware de Backup**

Na categoría de elementos hardware de backup temos:

#### **4611.1.1 Cintas**

Tradicionalmente, os cartuchos de cinta magnética son os medios de comunicación máis habituais nos sistemas de *backup*. Como soporte de almacenamento dos respaldos de datos, a cinta magnética ten unha longa historia de uso e é o medio de copia de seguridade con maior nivel de madurez. A cinta magnética, ou dunha forma máis abreviada, a cinta, é un compoñente baseado en cartuchos que se fai tipicamente dalgún tipo de plástico ríxido. Contén unha ou máis bobinas de plástico flexible que se impregnaron cun material con comportamento magnético.



Os cartuchos de cinta están fabricados en varios formatos. Cada formato ten unhas características diferentes que responden ás diferentes necesidades de almacenamento físico e de tempo de preservación da copia de seguridade, tanto en termos da cantidade de datos almacenados, como de vida útil dos medios de almacenamento ou o seu custo. Os formatos de cinta de uso común son os seguintes:

- DLT/ SDLT
- LTO
- AIT
- STK 9840/9940/T10000

Segundo o tipo de cada cartucho este posúe distintas capacidades ou características como a velocidade de funcionamento. O mercado está a renovar continuamente este tipo de dispositivos co fin de mellorar ambos os dous aspectos. Non obstante, existen tres formatos que podemos considerar dos máis comúns e teñen características particulares que se describen aquí como exemplos de elementos arquitectónicos de deseño: *DLT, LTO, T10000 e STK*.

O resto de formatos, aínda que sexan formatos comúns, utilízanse normalmente para ámbitos especializados, como o arquivado e almacenamento intermedio (nearline storage) empregado entre o almacenamento en liña e o almacenamento de backups.

#### **4611.1.1.1 Digital Linear Tape (DLT)**

Digital Linear Tape (DLT) é o formato de cinta máis antigo e polo tanto un dos produtos máis maduros do mercado. Orixinalmente foi deseñado e



implantado por DEC en 1984, para posteriormente ser adquirida por Quantum e redistribuído en 1994.

DLT é o primeiro cartucho de cinta compacta para copias de seguridade de sistemas abertos na empresa. Mentres que outros tipos de medios se encontraban en uso (como a cinta media polgada, 4 mm/8 mm, e outros), DLT proporciona o mellor compromiso entre todos os factores debido ao seu tamaño, a fiabilidade do seu almacenamento, a capacidade, e dispoñibilidade relativa.

A conectividade de DLT límtase aos tradicionais de SCSI, e está limitado a 300 xigabytes de capacidade nativa de almacenamento e 160 Mb/s de velocidade de transferencia (SDLT600). Existían outras variantes dispoñibles, pero nunca chegaron a popularizarse con carácter xeral. Hoxe en día, DLT encóntrase normalmente como copia de seguridade de longa duración en ámbitos pequenos que non requiren maior capacidade.

#### **4611.1.1.2 Linear Tape Open (LTO)**

Linear Tape Open (LTO) foi deseñado e concibido como unha evolución e alternativa aos formatos DLT e outros xa existentes, e estaba destinado a proporcionar unha plataforma común para os backups en cinta.

Seagate, HP e IBM foron os iniciadores orixinais do consorcio LTO, encargado de realizar o desenvolvemento inicial e o cal mantén a licenza da tecnoloxía e a certificación do proceso. En teoría, deberíase de ter producido un formato estándar de cinta, co cal os fabricantes poderían seguir traballando co estándar no mercado e incorporando as súas propias características e funcións adicionais.

Non obstante, entre o orixinal LTO-1 e os formatos de LTO-2 houbo



problemas de compatibilidade. Estes problemas abranguían dende bloqueos nas cintas cando se utilizan medios adquiridos a dous provedores distintos ata a incapacidade dunha unidade LTO dun fabricante a ler os datos escritos nun cartucho doutra.

O LTO-1 inicial proporcionaba 100 xigabytes de almacenamento nativo e 15 Mb/s; cos actuais sistemas de LTO-4 proporciónanse 400 Xb de almacenamento nativo de 160 Mb/s. Pola súa banda, o LTO-5 proporciona 800 Xb de capacidade de almacenamento nativo a 160 Mb/s.

#### **4611.1.1.3 Sun StorageTek T10000 (T10k)**

O T10000 / StorageTek (T10k) de Sun representa unha das tecnoloxías de almacenamento en cinta que mellor se comportou en termos de capacidade. O T10k é un formato propietario producido unicamente por StorageTek e encóntrase normalmente en ámbitos nos que se empregaban as tecnoloxías anteriores de Sun como o STK (9840/9940). Tamén se utilizaron en sistemas abertos de servidores ou mainframe. O T10k está deseñado para 500 Xb de almacenamento nativo de 120 Mb/s.

#### **4611.1.1.4 Características de almacenamento en cinta**

Aínda que todos os datos anteriores indican un valor interesante en canto ao rendemento, todos os dispositivos de cinta con características similares de rendemento deben terse en conta á hora de deseñar contornos de backup.

A primeira e máis importante delas é o feito de que todas as unidades de



cinta son contornos serie. A diferenza dos dispositivos de disco, os dispositivos de cinta escriben os bloques de datos de forma lineal, un tras outro. As unidades de cinta só teñen unha cabeza de escritura que escribe un bloque de datos de cada vez na cinta, a medida que esta se move por ela. Os dispositivos de disco teñen unha serie de dispositivos de escritura, ou cabezas, que se moven a varios puntos do disco xiratorio para situar os datos dun xeito óptimo. Isto permite que os dispositivos de disco poidan ler calquera anaco de información solicitada. Dado que os discos teñen varias cabezas para obter bloques de datos en paralelo, varios sistemas poden acceder ao disco ao mesmo tempo

A lectura dos datos dunha cinta realízase mediante o proceso inverso: A cinta debe rebobinarse ata o principio, cara a diante ata o bloque que se necesita, e ler así o bloque de datos. Ao poder devolverse unicamente un segmento de datos con cada lectura, os dispositivos de cinta non se poden compartir de forma paralela entre sistemas sen un mecanismo para transferir o control entre os sistemas que usan ese dispositivo.

O tipo de conectividade tamén ten influencia sobre a utilización de dispositivos de cinta. As unidades de cinta dependen dunha conexión directa co ordenador anfitrión para o transporte dos datos. Unha vez máis, isto débese ao feito de que as unidades de cinta son dispositivos de serie que só aceptan unha única conexión á vez.

#### *4611.1.2    Disco*

A cinta proporciona un método moi maduro, moi coñecido, e de baixo custo para almacenar copias de seguridade. Non obstante, as debilidades, tales como a natureza secuencial da cinta, a complexidade mecánica, e a gran variabilidade do rendemento dos dispositivos de cinta están rapidamente relegando á cinta a medio de almacenamento secundario ou terciario en moitos ámbitos.

Con todos os problemas coa cinta, os administradores buscaban un medio



que permitise un rápido acceso ás copias de seguridade e que proporcionase unha forma de ter un almacenamento rápido e fiable: o *disco*.

As copias de seguridade a disco son simples sistemas de arquivos que foron situados á parte para que o software de backup os use. Aínda que isto parece sinxelo, a implantación e xestión das solucións baseadas en disco poden ser moi complexas.

O almacenamento en disco supera algunhas das desvantaxes propias das cintas. Pola capacidade de recibir datos de forma rápida, ten múltiples fluxos para almacenar copias de seguridade ao mesmo tempo, e ten a capacidade de presentar o almacenamento dun número de maneiras diferentes, dependendo da necesidade do sistema, por iso, o disco é moi empregado como un medio de almacenamento de copia de seguridade primario.

Pero o disco tamén ten as súas debilidades: o custo dos medios de comunicación, a falta de portabilidade, e a dificultade de asegurar a plena utilización dos medios de comunicación fan que o disco non sexa tan satisfactorio como parece a priori.

#### 4611.1.3 Medios virtuais

Os medios virtuais emulan o hardware físico de cinta co obxectivo de reducir ou eliminar os problemas de xestión asociados aos medios físicos. Mediante a eliminación do hardware cunha alta complexidade mecánica e de xestión e a eliminación dos seus sistemas asociados e substituíndoos por unidades de disco, os medios virtuais tamén teñen a vantaxe de aumentar a fiabilidade xeral do contorno de backup. Os medios virtuais ofrecen estas vantaxes sen cambiar os procedementos operativos ou esixir modificacións do software de copia de seguridade. Ademais, nalgúns casos,



o rendemento pode aumentarse a través dun mellor uso do largo de banda nos medios de comunicación utilizados para conectar os medios virtualizados cos servidores de backup.

Os medios virtuais de copia de seguridade asóciáanse tradicionalmente de forma exclusiva con bibliotecas de cintas virtuais (VTL) pero recentemente realizáronse novas implantacións a través de protocolos que permiten a virtualización doutros tipos de sistemas de almacenamento.

#### *4611.1.4 Medios ópticos*

Os medios ópticos sitúanse entre as vantaxes das cintas e as do disco. Sobresaen nas áreas de fiabilidade, flexibilidade, ciclo de traballo e inamobilidade, mentres que os seus retos os encontramos nas áreas de rendemento, capacidade e custo.

##### **4611.1.4.1 CD**

CD, ou compact disk (disco compacto), é un soporte dixital óptico que se utiliza para o almacenamento de practicamente calquera tipo de datos. Na actualidade o uso do CD está a decaer a favor do aumento do uso dun novo medio de similares características como o DVD.

O CD serviu e segue servindo como medio de almacenamento de copias de seguridade grazas á súa fiabilidade e inamobilidade. Proporciona en comparación con outros medios como a cinta magnética, maior seguridade e protección dos datos, dado que o propio medio é moito máis robusto fronte a interaccións físicas externas (por exemplo os campos magnéticos).

Ademais de ser un medio habitual para o almacenamento de pistas de audio, os CD utilízanse habitualmente para a xeración de copias de seguridade relacionadas coa recuperación dos sistemas.



Os sistemas de CD utilizan un dispositivo hardware específico para gravar información, coñecido como gravadora/regravadora de CD. Existen tamén dispositivos hardware similares que soamente permiten a lectura deste medio.

As capacidades habituais dos CD estándar abranguen dende os 650 Mb ata os 900 Mb.

#### **4611.1.4.2 DVD**

Os DVD veñen a ser a evolución da tecnoloxía dixital óptica dos CD.

Ao igual que os CD, existen dous tipos de dispositivos para o uso dos DVD que son as gravadoras e os lectores. Existen diferentes tipos de DVD e diferentes categorizacións, das que a máis importante é a relativa ao número de capas, factor que determina a capacidade final do dispositivo.

As capacidades actuais abranguen dende os 4,3 Xb ata os 17 Xb. Os DVD utilizan dous tipos de sistemas de ficheiros que substitúen o antigo ISO 9660 dos CD, e que son o UDF e o Joliet.

### **4611.2      *Software de Backup***

Na categoría de elementos software de backup temos ferramentas de código aberto ou software libre e software privativo ou comercial. As ferramentas máis comúns en canto a software son:

#### **4611.2.1      Ferramentas de código aberto - AMANDA**

Amanda (Advanced Maryland Automated Network Disk Archiver) é o software de código aberto de copia de seguridade máis coñecido. Amanda



desenvolveuse inicialmente na Universidade de Maryland en 1991 co obxectivo de protexer os arquivos dun gran número de estacións de traballo cliente cun servidor de copia de seguridade único. James da Silva foi un dos seus desenvolvedores orixinais.

O proxecto Amanda rexistrouse en SourceForge.net en 1999. Jean-Louis Martineau, da Universidade de Montreal foi o líder do desenvolvemento de Amanda nos últimos anos. Durante anos, máis de 250 desenvolvedores contribuíron ao código fonte de Amanda, e miles de usuarios achegan probas e comentarios, o que o converte nun paquete robusto e estable. Amanda inclúese coa maior parte das distribucións Linux.

Nun principio, Amanda foi utilizado maioritariamente nas universidades, laboratorios técnicos, e departamentos de investigación. Hoxe, coa ampla adopción de Linux nos departamentos de informática, Amanda encóntrase en moitos outros lugares, sobre todo cando a atención se centra en aplicacións LAMP (Linux+Apache+MySQL+PHP). Cos anos, Amanda recibiu múltiples premios dos usuarios.

Amanda permite configurar un único servidor backup mestre para realizar múltiples copias de seguridade de equipos Linux, Unix, Mac OS X e Windows nunha ampla variedade de dispositivos: cintas, discos, dispositivos ópticos, bibliotecas de cintas, sistemas RAID, dispositivos NAS, e moitos outros.

As principais razóns para a adopción xeneralizada de Amanda son:

- Pódese configurar un único servidor de copia de seguridade de varios clientes en rede con calquera dispositivo de almacenamento: unha cinta, disco ou sistema de almacenamento óptico.
- Está optimizado para o backup en disco e cinta, permitindo escribir simultaneamente a copia de seguridade a cinta e disco.



- Non utiliza drivers propietarios, calquera dispositivo soportado por un sistema operativo tamén poderá funcionar en Amanda.
- Utiliza ferramentas estándar, como dump e tar. Posto que non son formatos propietarios, os datos pódense recuperar con esas mesmas ferramentas.
- Utilízase un planificador que optimiza niveis de seguridade para os diferentes clientes, de tal maneira que o tempo total do backup é aproximadamente o mesmo para cada execución.
- Existe unha ampla e activa comunidade de usuarios que crece día a día.
- O custo total de propiedade (TCO) dunha solución de backup baseada en Amanda é significativamente menor que o TCO de calquera solución que utilice software privativo.

#### *4611.2.2 Ferramentas de código aberto - BackupPC*

BackupPC é un sistema de alto rendemento que permite realizar copias de seguridade de sistemas Unix, Linux, Windows e MacOS nun disco. É polo tanto unha ferramenta baseada totalmente en disco.

Ofrece unha serie de vantaxes como son:

- **Soporta calquera sistema operativo cliente.** Isto débese a que se utilizan ferramentas estándar que ou veñen co SO ou se poden engadir a el, sen necesidade de instalar cliente. Así resulta máis doado integrar un novo cliente.
- **Interface web** con control de usuario para acceder a copias de seguridade. A maioría dos SO traen un navegador web, así que usar unha



interface web é outro xeito de acelerar o proceso de incorporación de novos clientes con diferentes sistemas operativos. A interface web está deseñada para dar o máximo control posible ao cliente de forma segura. O usuario pode solicitar restauracións, navegar doadamente e restaurar arquivos individuais. Non obstante, o usuario non poderá ver as máquinas doutro usuario.

- **Soporte de clientes DHCP.** Mediante o uso de servizos estándar, BackupPC soporta clientes DHCP, sempre e cando o cliente estea rexistrado cun servizo de nomes como DNS, Active Directory ou LDAP.

### **Funcionamento de BackupPC**

O modelo de BackupPC ten un usuario por cliente. Isto é así porque BackupPC foi especificamente deseñado para realizar copias de seguridade dos PC de varios usuarios (de aí o nome).

Normalmente, o usuario é o propietario dos datos da máquina. Se se traballa cun servidor de ficheiros, o usuario deberá ser un administrador.

BackupPC envía mensaxes de correo electrónico ao propietario se non pode realizar a copia de seguridade despois dun tempo configurable; o propietario pode xestionar as restauracións das copias a través dunha interface web.

Nos seguintes puntos descríbense algunhas das características proporcionadas por BackupPC:

- **Directo ao disco.** BackupPC almacena todas as súas copias de seguridade directamente no disco. Os arquivos idénticos en calquera directorio ou cliente gárdanse só unha vez, o que reduce drasticamente os requisitos de almacenamento do servidor. Estes arquivos almacénanse nun conxunto de discos. Ademais do conxunto de discos, as copias de seguridade están nunha árbore de directorios organizados por anfitrión.

BackupPC tamén ten un proceso (que se lanza polas noites) que



recupera espazo do conxunto de discos que non está referenciado por ningún backup, o que evita un uso inadecuado do espazo en disco. Este é un proceso automático que o administrador non ten que configurar.

- **Sistema operativo do servidor.** A parte do servidor de BackupPC está deseñada para executarse nun sistema tipo Unix con Perl e mod\_perl. Ofrece o mellor rendemento con Apache, pero pódese executar en calquera servidor web que soporte Perl (requírese mod\_perl ou Perl setuid.) O servidor debe ter un disco con gran capacidade ou RAID para almacenar os backups.
- **Sistema operativo do cliente.** Como se comentou anteriormente, soporta calquera SO. As versións máis modernas das variantes comerciais de Unix (Solaris, AIX, IRIX, HP-UX) traen na propia distribución as ferramentas tar, compress, gzip, rsync, e rsh e / ou ssh. Outros sistemas operativos tipo Unix (Linux, FreeBSD, OpenBSD, NetBSD, Mac OS X) tamén contan con estas ferramentas.

Os clientes de Windows poden facer copias de seguridade de diferentes formas dependendo de se as políticas locais permiten ou non a instalación de software. Se non se permite, BackupPC utiliza parte da suite Samba (<http://www.samba.org>) para facer backup da información compartida mediante SMB ou CFIS. Se se permite instalar software, utilízase rsync xunto co conxunto de ferramentas Cygwin (<http://www.cygwin.com>).

- **Soporte para ferramentas nativas.** BackupPC utiliza as ferramentas estándar de Unix para o seu funcionamento interno. Isto inclúe programas como Perl, tar, rsync, compress, gzip, bzip2, zip, apache e samba.

BackupPC non utiliza unha base de datos ou catálogo para almacenar a información de respaldo. No seu lugar, utiliza a árbore de directorios para almacenar esta información. Isto simplifica as actualizacións do



sistema operativo do servidor de BackupPC ou da propia aplicación BackupPC.

- **Control dos backups e restauracións a través de interface web.** A web é a interface principal de BackupPC. Tras a configuración inicial, non é necesario acceder ao servidor mediante liña de comandos para administrar BackupPC. A interface web está escrita en Perl e foi deseñada para funcionar tanto con mod\_perl coma con CGI ou con Perl setuid.

A interface permite aos usuarios identificarse, acceder e controlar os respaldos e as restauracións.

O usuario pode solicitar copias de seguridade de tipo one-time, de tipo completa, ou de tipo incremental.

Pódense utilizar varias opcións para recuperar ficheiros:

- o Os arquivos individuais recupéranse mediante selección.
- o Os grupos de arquivos ou directorios pódense restaurar á súa situación orixinal.
- o O usuario pode descargar os arquivos como un arquivo tar ou zip.

O usuario ten control absoluto sobre que arquivos ou directorios se restauran e onde hai que restauralos. Un histórico mostra que arquivos se modificaron durante cada copia de seguridade en cada directorio.

- **Soporte para clientes DHCP.** Os clientes BackupPC referéncianse por nome de anfitrión. Se a rede da copia de seguridade utiliza DHCP e se permite a resolución de nomes dinámica, non hai que facer nada máis para que o servidor BackupPC avale os clientes DHCP. Se este non é o caso, e os clientes son máquinas Windows, BackupPC pódese configurar para buscar un conxunto de enderezos dos clientes, localizándoos mediante SMB.

Se o cliente non está en liña durante o período de copia de seguridade



normal, o servidor BackupPC xera un erro a menos que transcorrese un período de tempo establecido dende a última copia de seguridade. Neste punto, o servidor envíalle un correo electrónico ao propietario do cliente e recórdalle que se asegure de que a máquina está na rede para facer unha copia de seguridade. (O servidor tamén lle pode enviar calquera erro ao administrador.)

Os clientes que residen noutra LAN poden ser xestionados nun ámbito local asumindo que hai conectividade de entre as redes. Isto significa que se pode facer backup dos clientes conectados a través dunha rede privada virtual (VPN).

Se o usuario non desexa realizar copias de seguridade nun momento dado, conectaríase a través da interface web para cancelar a copia de seguridade.

- **Pool de Backups.** Cando os clientes utilizan o mesmo sistema operativo dúplícanse os arquivos avalados. Se se quere manter múltiples copias de seguridade completas aumenta o número de arquivos duplicados, o que aumenta os requisitos de capacidade de almacenamento para o servidor. BackupPC almacena unha árbore de directorios por cliente avalado, pero comproba se os arquivos se almacenaron antes. Se é así, BackupPC utiliza unha ligazón que apunta ao ficheiro existente no conxunto de discos común, aforrando unha gran cantidade de espazo. Ademais, BackupPC pode comprimir opcionalmente para aforrar máis espazo.
- **Doadada configuración por cliente.** Unha vez que o administrador defina cales deberían de ser as políticas de backup do sitio, é moi doado anular calquera opción de configuración sobre a base dun cliente. Isto permite unha gran flexibilidade sobre que, cando, e como facer copia de seguridade dun cliente. Non hai clases de clientes por si mesmo.



### *4611.2.3 Ferramentas de código aberto - Bacula*

Bacula é un conxunto de programas Open Source, listos para ser utilizados nun contorno doméstico e profesional, que permiten administrar os backups, restauración e verificación de datos nunha rede heteroxénea. Bacula é relativamente doado de usar e eficiente, á vez que ofrece moitas funcionalidades avanzadas para a administración dos datos almacenados, o cal facilita atopar e recuperar arquivos perdidos ou danados. En termos técnicos, Bacula é un sistema de backups Open Source, orientado á rede e listo para a empresa.

É capaz de realizar copias de seguridade en disco, cinta ou medios ópticos. Bacula foi escrita orixinalmente por John Walker e Kern Sibbald no ano 2000. John deixou o proxecto non moito tempo despois da súa creación, e Kern traballou nel dende mediados do 2000 ata o primeiro lanzamento público de Bacula en abril de 2002. Dende aquela, outros desenvolvedores contribuíron ao seu desenvolvemento.

Bacula está dispoñible baixo licenza AGPL versión 3. A páxina web do proxecto encóntrase en <http://www.bacula.org>, e os arquivos descargables e un repositorio CVS alóxanse en SourceForge.

#### **Bacula Arquitectura**

Bacula é unha solución distribuída de backups. Isto significa que Bacula está composto por varios elementos, que poden ou non residir no mesmo ordenador anfitrión. Por exemplo, pódese ter un anfitrión co catálogo e noutro o storage.

Baséase nunha arquitectura cliente-servidor que resulta eficaz e doada de manexar, dada a ampla gama de funcións e características que brinda: copiar e restaurar ficheiros danados ou perdidos. Ademais, debido ao seu desenvolvemento e estrutura modular, Bacula adáptase tanto ao uso persoal como profesional.



Pódese utilizar TLS (Transport Layer Security) para protexer os datos durante a transmisión.

Os compoñentes principais desta arquitectura son:

- **Director (DIR)** é o encargado de xestionar de forma centralizada a lóxica dos procesos de backup e os demais servizos. Traballa sobre a base dunha unidade básica denominada JOB (un cliente, un conxunto de arquivos...) de tal forma que o director planifica, inicia e supervisa todos os jobs.

Tamén é o encargado de manter o catálogo, polo que o servidor da base de datos debe estar accesible dende a máquina que executa o director.

- **Storage** é o encargado de xestionar os dispositivos de almacenamento; isto esixe que estea instalado na máquina que posúa a conexión física aos dispositivos de almacenamento, tales como: discos locais, gravadoras, unidades de cinta, volumes NAS ou SAN, autocargadores, ou librarías de cinta.
- **File Daemon** é o axente que corre do lado do cliente, é dicir, na máquina da que se van gardar os datos, e que ten como obxectivo empacar os datos e envialos ao Storage, onde serán almacenados.
- **Consola** é a ferramenta que lle permite ao usuario ou administrador controlar Bacula. Comunícase co director vía rede, iniciando os jobs, revisando a saída do job, facendo consultas e modificacións no catálogo.

Existen consolas en modo texto, modo GUI para Windows e Linux/UNIX e interfaces web.

- **Catálogo** é unha base de datos onde se garda información sobre os jobs e sobre os datos gardados. O catalogo permite dúas cousas:

o Por un lado, como garda información dos jobs, pools e volumes, Bacula úsao para saber se hai un backup completo para un job, e se non



o hai, realizará para ese backup unha copia completa.

o Por outro lado, o catálogo ten todos os nomes de arquivo (e os seus atributos, como data de última modificación, etc.) que se gardaron, e iso é o que permite facer unha recuperación selectiva, é dicir, seleccionar (marcar, na xerga de Bacula) individualmente que arquivos e/ou directorios restaurar.

#### 4611.2.4 Software Propietario CommVault Simpana

Simpana comezou como un proxecto dentro de AT&T Labs en 1987 e posteriormente foi adquirido pola empresa CommVault.

Simpana é un software de backup que realiza copias de seguridade de contornos Unix, Windows, Linux, servidores de correo Exchange, Lotus Notes, bases de datos Oracle, MySQL, SQLServer e máquinas virtuais VMware. Ademais permite funcións avanzadas como pode ser o arquivado, a deduplicación e a replicación de ficheiros.

O funcionamento da aplicación baséase no uso dos bloques de disco, polo que todos os módulos non utilizan a información do arquivo, se non que traballa a máis baixo nivel. Con iso consegue mellores porcentaxes de compresión e unha importante redución da ventá de backup, ao utilizar unicamente os bloques modificados e non o ficheiro enteiro para realizar estas operativas.

Outra característica que incide no uso de almacenamento de baixo custo é a capacidade de xerar políticas como as de arquivado, mediante as que automaticamente permite mover ficheiros dun almacenamento a outro con maior capacidade a menor custo. Desta forma, por exemplo poderíanse pasar os datos dunha cabina de fibra a outra con discos SATA, e pódese chegar a un terceiro nivel a cinta, sobre a base duns requisitos (data do arquivo, último acceso ao arquivo, etc.). Todos os movementos realízanse de forma transparente para o usuario, tanto no arquivado coma na súa recuperación (se fose necesario).



A estas funcionalidades hai que sumar a capacidade de deduplicación, que realiza unha compresión dos datos aproveitando as duplicidades dos datos por bloques, conseguindo alcanzar porcentaxes de ata o 50% de aforro no uso de almacenamentos en datos de segundo nivel e ata o 90% nos de terceiro nivel.

Para rematar o repaso ás principais funcionalidades, a replicación, permite a utilización de “instantáneas” (snapshots) nas cabinas permitindo volver o almacenamento replicado a un estado anterior ou montar a imaxe instantánea como un recurso compartido.

Adminístrase todo dende unha única consola centralizada, que simplifica toda a administración da plataforma. Adicionalmente o motor de busca ofrece a opción de buscar rapidamente e recuperar datos sen necesidade de saber onde se sitúan.

#### 4611.2.5    *Software Propietario Symantec NetBackup*

Symantec NetBackup é actualmente o titular da maior cota de mercado do contorno de software de copia de seguridade.

Netbackup 7 é a nova versión da solución de copia de seguridade e recuperación de datos orientada a grandes corporacións. Esta ferramenta tenta simplificar a xestión da información reducindo o volume de almacenamento de datos con técnicas de deduplicación nos ordenadores cliente da rede ademais do propio servidor, ofrecendo protección para contornos virtualizados. Todo iso co único propósito de axilizar os procesos de backup e recuperación de datos.

A nova ferramenta inclúe eliminación de datos duplicados nativos dentro do cliente NetBackup e permítelles aos clientes multiplicar por dez a velocidade das copias de seguridade en oficinas remotas, o propio centro de datos e os contornos virtuais. Esta eliminación de datos duplicados no cliente e no destino ofrece unha maior cobertura con menos ferramentas.



O proceso de deduplicación considérase para todos os sistemas físicos e virtuais, independentemente do método de copia de seguridade. Deste modo intégrase unha maior protección para os cada vez máis estendidos contornos virtualizados baixo as plataformas Hyper-V e VMware. É no caso desta última na que se puido observar un incremento de velocidade de ata o 50% á vez que diminúe o volume de almacenamento necesario nun 40%.

Outro dos aspectos notablemente mellorados en Netbackup 7 é a velocidade de recuperación de datos ante desastres. Permitindo a restauración de grandes volumes de información en poucos segundos dende calquera lugar e punto no tempo. Esta xestión facilítase ao administrador de TI mediante un sistema centralizado de supervisión e alerta, que integra a administración de varios dominios de arquivos coas súas respectivas políticas de salvagarda de datos.

A tecnoloxía incluída en NetBackup acelerará a transición a un contorno virtual para as organizacións empresariais que instalen un gran número de máquinas virtuais ou que decidan crear unha infraestrutura de nube privada.

A solución NetBackup tamén ofrece unha elaboración de informes simplificada e un maior soporte ás aplicacións de bases de datos de Oracle e MySQL.

Algunhas das prestacións e beneficios incluídos na última versión da ferramenta son:

- A tecnoloxía Virtual Machine Intelligent Policy incorpora a automatización á localización e a protección de máquinas virtuais e minimiza os esforzos de administración necesarios para realizar copias de seguridade de máquinas virtuais VMware de alto rendemento.
- Un 50% máis de rapidez en copias de seguridade de máquinas virtuais grazas a que a tecnoloxía Granular Recovery Technology (GRT)



está agora dispoñible para sistemas Linux en contornos VMware. Isto permítelles aos clientes reducir os tempos comparables de copias de seguridade de máquinas virtuais nun 50%, ademais de simplificar a administración e mellorar a velocidade de recuperación de arquivos individuais.

- Recuperación "á carta" dende calquera lugar coa nova tecnoloxía de replicación de imaxe que permite aos clientes que replican datos entre múltiples sitios ou dominios de NetBackup realizar backup de datos nun sitio alternativo.
- Recuperación acelerada: NetBackup RealTime ofrece soporte a contornos VMware para eliminar o espazo de tempo entre copias de seguridade, ademais de reducir o impacto para grandes hosts de VMware e permitir a recuperación case instantánea de sistemas completos.
- Satisfacer os requisitos normativos e de cumprimento para seguimento de auditorías.
- Incorpora informes mellorados das políticas do ciclo de vida do almacenamento, do seguimento das auditorías e do estado das licenzas.
- Desduplicación para Oracle mellorando o rendemento das copias de seguridade.
- Engádese un novo axente que presta soporte a MySQL para centralizar e automatizar as copias de seguridade e a recuperación de datos das bases de datos de MySQL.
- Actualización simplificada de clientes con LiveUpdate que permite melloras en equipos cliente para UNIX, Linux e Windows respecto á versión NetBackup 6.5 e posterior dende unha política única controlada polo administrador de NetBackup.







## **4612 ESTRATEXIAS DE BACKUP A DISCO**

As estratexias de backup definen o plan que cómpre seguir para garantir a integridade da información. Os motivos polos que se debe establecer unha correcta estratexia antes de comezar a realizar as copias de seguridade poden ser moi diversos, pero en esencia consiste en determinar o mellor xeito para asegurar a información tendo en conta as posibles dificultades de recuperación de parte dos datos, o custo dos medios que se empregaran e o tempo que se necesitara.

Como non todos os sistemas son iguais, non todas as estratexias de backup son adecuadas para todos os sistemas. Partindo dunhas características comúns, algunhas das propiedades básicas dunha estratexia backup son:

- **Tempo de almacenamento.** Define o tempo máximo que unha copia permanece almacenada nun dispositivo. Ao finalizar este tempo a copia pode cambiar de dispositivo ou ser borrada para liberar espazo no medio de almacenamento e poder facer uso deste.
- **Almacenamento alternativo.** Posibilita realizar unha ou varias copias de seguridade nunha situación externa ao sistema e á localización xeográfica deste, manténdoa durante un elevado período de tempo, aumentando a seguridade ante calquera catástrofe, xa sexa a nivel de software ou de hardware.
- **Protección ante fallo dos dispositivos.** Establece o número de medios que se empregan. Canto maior é o número de medios utilizados, maior é a seguridade contra posibles perdas de información producidas por un fallo no dispositivo de almacenamento.
- **Tempo de restauración.** Esta característica especifica o tempo de



rexeneración do sistema en caso de producirse algún fallo.

- O custo. Aadoita ser un factor determinante á hora de seleccionar a estratexia a realizar.

As estratexias para a realización de copias de seguridade poden ser moi distintas, dependendo do sistema en cuestión sobre o cal se realizan.

Nalgúns casos, efectúase só un backup de todo o contido. Isto prodúcese por algún motivo especial e moi específico ou por algún motivo técnico, cuestións de tempo ou por que existe un elevado risco para os datos. Algún destes casos especiais poden ser:

- Non dispoñer do software orixinal.
- Descoñecemento da situación dos ficheiros de configuración.
- Cambiar un disco de almacenamento ríxido.
- Realizar cambios nas particións dun ou máis discos de almacenamento ríxidos.

É habitual que este tipo de situacións concretas se produzan á hora de levar a cabo tarefas de reparación ou actualización sobre sistemas non controlados.

Cando hai que cubrir algún destes casos, a estratexia de backup que se debe seguir é sinxela, realizar un resgardo ou copia de seguridade de todo o contido das unidades involucradas para así garantir que non se perderá ningunha información e que será posible realizar a restauración completa do sistema.

Por outro lado, cando realmente cómpre deseñar un plan estratéxico para a realización das copias de seguridade dun sistema propio ou dunha organización externa, débense ter en conta unha serie de pautas que axudan a que o plan estratéxico de backups sexa o máis conveniente e



conseguir a mellor relación custo/beneficio posible.

Estas pautas achegan unha redución no tempo de resposta á hora de realizar unha recuperación en caso de que se producira se calquera tipo de continxencia.

Ao intentar definir un plan de backups, xorden unha serie de dúbidas:

***Que datos se deberían resgardar en cada backup?*** Datos que resgardar.

É un factor determinante para unha estratexia de backup que se determine o grao ou graos de importancia da información, é dicir, establecer que información resulta de maior valor para a organización. Non teñen a mesma transcendencia un documento de traballo que unha copia de respaldo da configuración dunha aplicación.

***Cada canto se debería efectuar unha copia de seguridade dos datos?*** Frecuencia do backup.

Para determinar a periodicidade coa que se deben realizar as copias de seguridade non existe un criterio claramente definido. Non obstante téñense en conta factores como:

- Tempo empregado na creación da información.
- Custo investido na creación da información.
- Posibles consecuencias derivadas da súa perda.

***Canto tempo deberían permanecer gardadas as copias de seguridade?*** Tempo de almacenamento.

O período máximo de tempo de estancia dunha copia de seguridade nun dispositivo, é dicir, o tempo de retención, está directamente relacionado cos medios de almacenamento dispoñibles, e por conseguinte polo orzamento da estratexia de backup.



Outra das decisións importantes que hai que tomar durante a elaboración dunha estratexia para a realización de copias de seguridade é a de seleccionar e planificar os distintos tipos de copias de seguridade.

Os backups son copias exactas da información. Pódense definir como instantáneas dos datos nun momento determinado, almacenados nun formato estándar, pódese realizar un seguimento ao longo do seu período de utilidade e con cada nova copia mantense a independencia con copia inicial. Pódense crear múltiples niveis de backups, sendo os principais:

- **Copias de seguridade completas (Full backups):** representan unha copia exacta nun momento dado, dos datos que se pretende protexer. Proporcionan a base para todos os demais niveis de backup.

- Por outro lado, están dous niveis de backup que capturan unicamente os cambios realizados sobre unha copia de seguridade completa.

o **Copia de seguridade diferencial**, tamén coñecida como a *copia de seguridade incremental acumulativa*, captura copias de seguridade que se produciron dende o último backup completo e adoita utilizarse en ámbitos nos que non se produce un elevado número de cambios. A copia de seguridade diferencial débese utilizar con coidado debido a que pode crecer con rapidez e igualar e mesmo superar o tamaño da copia de seguridade completa.

A vantaxe de utilizar as copias de seguridade diferenciais vén dada no momento da restauración posto que no momento de restaurar unha copia de seguridade diferencial só se necesita o backup completo e a última copia diferencial realizada. Debido a que unicamente se precisan dúas imaxes para a restauración, a probabilidade de que ambas as dúas imaxes sufran algún contratempo, perda, corrupción, etc., redúcese significativamente.



o ***Copia de seguridade incremental***, é capaz de capturar os cambios que se produciron dende a última copia de seguridade realizada, independentemente do tipo que sexa. É a forma máis utilizada para a realización de copias de seguridade, evidentemente combinada cunha copia de seguridade completa.

Este tipo de copia de seguridade contén a menor cantidade de datos necesarios durante cada ciclo de backup, reducindo a cantidade de datos que se transfiren e o tempo que se necesita para a creación dunha copia de seguridade.

Non obstante as copias de seguridade incrementais teñen aspectos negativos. Se se está a recuperar un grupo de arquivos dun conxunto de copias de seguridade completas e incrementais, é probable que se requiran máis de dúas imaxes de copias de seguridade diferentes para completar a restauración, o que aumenta a probabilidade de que algunha destas partes sufra algún tipo de problema e non se poida completar a restauración.

#### **4613 REPLICACIÓN LOCAL E REMOTA, ESTRATEXIAS DE RECUPERACIÓN**

A replicación é o proceso de creación dunha copia exacta dos datos. A creación dunha ou varias réplicas dos datos de produción é unha das maneiras de proporcionar continuidade ao do negocio (BC).

Estes modelos poden ser utilizados para operacións de recuperación e reinicio dos sistemas en caso de que se produza unha perda de datos.

Unha réplica debe proporcionar:



- **A capacidade de recuperación:** permite a restauración dos datos dos volumes de produción en caso de que se produza unha perda dos datos. Débese proporcionar un mínimo de RTO e un RPO concreto que nos garantan o reinicio das operacións comerciais nos volumes de produción.
- **A capacidade de reinicio:** garante a coherencia dos datos da réplica, posibilitando o reinicio das operacións de negocio utilizando para iso a información contida nas réplicas.

A replicación pódense clasificar en dúas grandes categorías: ***locais e remotas***



### **4613.1      *Replicación local***

A replicación local fai referencia ao proceso creación de réplicas dentro do mesmo array de discos ou o mesmo centro de datos.

#### **4613.1.1      Tecnoloxías de replicación local**

As replicacións host-based (baseadas en replicación en ordenador anfitrión local) e storage-based (baseadas en almacenamento) son as dúas principais tecnoloxías adoptadas para a replicación local. A replicación de arquivos do sistema e a replicación baseada en LVM son exemplos da tecnoloxía host-based de replicación local. A replicación de almacenamento baseada en matrices de disco pode levarse a cabo con solucións distintas, a duplicación de todo o volume, a replicación pointer-based de todo o volume, e a replicación baseadas en punteiros e virtual.

##### **4613.1.1.1      Baseada en replicación en anfitrión local**

Neste tipo de replicación, os administradores do sistema levan a cabo o proceso de copia e restauración na propia máquina, podendo basearse a recuperación nunha replicación integral do volume mediante LVM (Logical Volume Manager), ou ben mediante instantáneas do sistema de ficheiros.

- Replicación do volume mediante LVM: O LVM encárgase de crear e controlar o volume de host a nivel lóxico e está formado por tres compoñentes: os discos físicos, os volumes lóxicos e os grupos de volumes. Na replicación de volumes baseado en LVM, cada partición lóxica nun volume asígnase a dúas particións físicas en dous discos diferentes. Desesta forma conséguese un espello que permite redundancia e recuperación directa en caso de necesitar replicar.
- Instantánea de arquivos do sistema: Consiste en crear unha réplica a base de instantáneas do sistema de ficheiros mediante a utilización de metadatos almacenados nun mapa de bits. Estes metadatos van reflectindo o cambio que se vai producindo no sistema



de ficheiros e van almacenando un rexistro dos enderezos accedidos mediante operacións de lectura/escritura. Este sistema require dunha fracción do espazo utilizado polo sistema de ficheiros orixinal.

#### **4613.1.1.2 Baseada en arrays de discos**

Neste tipo de replicación faise uso de matrices de discos que poden estar distribuídas dentro do CPD. O contorno operativo é o que leva a cabo o proceso de replicación dun determinado sistema de ficheiros, sen necesidade de que os recursos de acollida (CPU e memoria) do anfitrión interveñan no proceso de replicación.

### ***4613.2 A replicación remota***

A replicación remota consiste no proceso de creación de replicas do conxunto de datos en lugares con outra situación física. As réplicas remotas axudan ás organizacións a mitigar os riscos asociados ás interrupcións rexionais do servizo, que poden estar provocadas por diferentes causas, por exemplo, desastres naturais. A infraestrutura na que os datos se almacenan inicialmente chámase fonte. A réplica, ou infraestrutura remota na que se almacena a copia chámasele branco.

#### **4613.2.1 Tecnoloxías de replicación remota**

A máis habitual é a tecnoloxía de replicación baseada en ordenador anfitrión remoto, que utiliza un ou máis compoñentes da máquina para realizar e xestionar a operación de replicación. Existen dous enfoques fundamentais para a replicación baseada en anfitrión remoto: Replicación remota baseada en LVM e replicación de bases de datos a través de transvasamento de rexistros.



#### **4613.2.1.1 Replicación remota baseada en LVM**

Neste modelo, a replicación efectúase e xestiona por grupo de volumes. O LVM da máquina orixe é o encargado de xestionar e transmitir a información do volume ao LVM da máquina remota. O LVM da máquina remota encárgase de recibir os datos e realiza a operación de réplica do volume.

Antes do inicio da replicación, débense configurar os sistemas fonte e remoto para que os sistemas de arquivos, os volumes e a agrupación de volumes sexa idéntica en ambos os dous. O punto de partida, ou sincronización inicial, pódese realizar de diferentes formas, sendo a máis frecuente a restauración no punto remoto dunha copia de seguridade dos datos de orixe.

Na replicación remota baseada en LVM sopórtanse dous modos de transferencia de datos, que son o sincrónico e o asíncrono. No modo asíncrono, as operacións de escritura vanse almacenando nunha cola de rexistros xestionada polo LVM e vanse enviando ao anfitrión remoto na orde na que son recibidas. En caso de fallo da rede, as operacións seguen acumulándose na cola de rexistros.

Na replicación síncrona, as operacións de escritura deben estar comprometidas tanto en orixe como en destino. As operacións de escritura consecutivas non poden ocorrer en fonte nin destino ata que as operacións previas finalicen. Isto garante que os datos da fonte e destino son exactamente os mesmos en todo momento. Isto fai posible que o RPO en caso de fallo sexa cero ou próximo a cero. Non obstante, como contraprestación ao nivel de seguridade, o tempo de resposta é moito maior. O grao de impacto no tempo de resposta depende da distancia entre ambos os dous sitios (fonte e destino), do largo de banda dispoñible e da infraestrutura de conectividade de rede.



#### **4613.2.1.2 Baseada en transvasamento de rexistros**

A replicación de bases de datos a través de transvasamento de rexistros consiste na captura das transaccións realizadas na base de datos fonte, que son almacenadas en rexistros que se transmiten periodicamente dun ordenador anfitrión fonte a un anfitrión destino. O host destino recibe o conxunto de rexistros e realiza as operacións oportunas na base de datos replicada. O proceso inicial de produción e reprodución require que todos os compoñentes importantes da base de datos se repliquen no sitio remoto.

Os sistemas xestores de bases de datos permiten definir un intervalo de tempo para o envío dos ficheiros de rexistro, ou ben configurar un tamaño predeterminado deles. Cando un rexistro supera o intervalo de tempo establecido ou alcanza o seu tamaño máximo, péchase, e ábrese un novo ficheiro para rexistrar as transaccións. Os rexistros pechados van sendo enviados dende a fonte ao destino garantindo que a base de datos replicada en destino sexa consecuente coa fonte ata o último rexistro pechado. O RPO no sitio remoto dependerá do tamaño do ficheiro de rexistro e da frecuencia de cambio de rexistro na fonte.

#### ***4614 REPLICACIÓN LOCAL E REMOTA, ESTRATEXIAS DE RECUPERACIÓN***

A replicación é o proceso de creación dunha copia exacta dos datos. A creación dunha ou varias réplicas dos datos de produción é unha das maneiras de proporcionar continuidade ao do negocio (BC).

Estes modelos poden ser utilizados para operacións de recuperación e reinicio dos sistemas en caso de que se produza unha perda de datos.

Unha réplica debe proporcionar:



- **A capacidade de recuperación:** permite a restauración dos datos dos volumes de produción en caso de que se produza unha perda dos datos. Débese proporcionar un mínimo de e RTO e un RPO concreto que nos garantan o reinicio das operacións comerciais nos volumes de produción.
- **A capacidade de reinicio:** garante a coherencia dos datos da réplica, posibilitando o reinicio das operacións de negocio utilizando para iso a información contida nas réplicas.

A replicación pódense clasificar en dúas grandes categorías: ***locais e remotas***



### **4614.1      *Replicación local***

A replicación local fai referencia ao proceso creación de réplicas dentro do mesmo array de discos ou o mesmo centro de datos.

#### **4614.1.1      Tecnoloxías de replicación local**

As replicacións host-based (baseadas en replicación en ordenador anfitrión local) e storage-based (baseadas en almacenamento) son as dúas principais tecnoloxías adoptadas para a replicación local. A replicación de arquivos do sistema e a replicación baseada en LVM son exemplos da tecnoloxía host-based de replicación local. A replicación de almacenamento baseada en matrices de disco pode levarse a cabo con solucións distintas, a duplicación de todo o volume, a replicación pointer-based de todo o volume, e a replicación baseadas en punteiros e virtual.

##### **4614.1.1.1      Baseada en replicación en ordenador anfitrión local**

Neste tipo de replicación, os administradores do sistema levan a cabo o proceso de copia e restauración na propia máquina, podendo basearse a recuperación nunha replicación integral do volume mediante LVM (Logical Volume Manager), ou ben mediante instantáneas do sistema de ficheiros.

- Replicación do volume mediante LVM: O LVM encárgase de crear e controlar o volume de anfitrión a nivel lóxico e está formado por tres compoñentes: os discos físicos, os volumes lóxicos e os grupos de volumes. Na replicación de volumes baseado en LVM, cada partición lóxica nun volume asígnase a dúas particións físicas en dous discos diferentes. Desesta forma conséguese un espello que permite redundancia e recuperación directa en caso de necesitar replicar.
- Instantánea de arquivos do sistema: Consiste en crear unha réplica a base de instantáneas do sistema de ficheiros mediante a utilización de metadatos almacenados nun mapa de bits. Estes metadatos van reflectindo o cambio que se vai producindo no sistema



de ficheiros e van almacenando un rexistro dos enderezos accedidos mediante operacións de lectura/escritura. Este sistema require dunha fracción do espazo utilizado polo sistema de ficheiros orixinal.

#### **4614.1.1.2 Baseada en arrays de discos**

Neste tipo de replicación faise uso de matrices de discos que poden estar distribuídas dentro do CPD. O contorno operativo é o que leva a cabo o proceso de replicación dun determinado sistema de ficheiros, sen necesidade de que os recursos de acollida (CPU e memoria) do anfitrión interveñan no proceso de replicación.

#### **4614.1.2 A replicación remota**

A replicación remota consiste no proceso de creación de replicas do conxunto de datos en lugares con outra situación física. As réplicas remotas axudan ás organizacións a mitigar os riscos asociados ás interrupcións rexionais do servizo, que poden estar provocadas por diferentes causas, por exemplo, desastres naturais. A infraestrutura na que os datos se almacenan inicialmente chámase fonte. A réplica, ou infraestrutura remota na que se almacena a copia chámasele branco.

#### **4614.1.3 Tecnoloxías de replicación remota**

A máis habitual é a tecnoloxía de replicación baseada en ordenador anfitrión remoto, que utiliza un ou máis compoñentes da máquina para realizar e xestionar a operación de replicación. Existen dous enfoques fundamentais para a replicación baseada en anfitrión remoto: Replicación remota baseada en LVM e replicación de bases de datos a través de transvasamento de rexistros.

##### **4614.1.3.1 Replicación remota baseada en LVM**

Neste modelo, a replicación efectúase e xestiona por grupo de volumes. O



LVM da máquina orixe é o encargado de xestionar e transmitir a información do volume ao LVM da máquina remota. O LVM da máquina remota encárgase de recibir os datos e realiza a operación de réplica do volume.

Antes do inicio da replicación, débense configurar os sistemas fonte e remoto para que os sistemas de arquivos, os volumes e a agrupación de volumes sexa idéntica en ambos os dous. O punto de partida, ou sincronización inicial, pódese realizar de diferentes formas, e a máis frecuente é a restauración no punto remoto dunha copia de seguridade dos datos de orixe.

Na replicación remota baseada en LVM sopórtanse dous modos de transferencia de datos, que son o sincrónico e o asíncrono. No modo asíncrono, as operacións de escritura vanse almacenando nunha cola de rexistros xestionada polo LVM e vanse enviando ao anfitrión remoto na orde na que son recibidas. En caso de fallo da rede, as operacións seguen acumulándose na cola de rexistros.

Na replicación síncrona, as operacións de escritura deben estar comprometidas tanto en orixe como en destino. As operacións de escritura consecutivas non poden ocorrer en fonte nin destino ata que as operacións previas finalicen. Isto garante que os datos da fonte e destino son exactamente os mesmos en todo momento. Isto fai posible que o RPO en caso de fallo sexa cero ou próximo a cero. Non obstante, como contraprestación ao nivel de seguridade, o tempo de resposta é moito maior. O grao de impacto no tempo de resposta depende da distancia entre ambos os dous sitios (fonte e destino), do largo de banda dispoñible e da infraestrutura de conectividade de rede.

#### **4614.1.3.2 Baseada en transvasamento de rexistros**

A replicación de bases de datos a través de transvasamento de rexistros



consiste na captura das transaccións realizadas na base de datos fonte, que son almacenadas en rexistros que se transmiten periodicamente dun ordenador anfitrión fonte a un anfitrión destino. O anfitrión destino recibe o conxunto de rexistros e realiza as operacións oportunas na base de datos replicada. O proceso inicial de produción e reprodución require que todos os compoñentes importantes da base de datos se repliquen no sitio remoto.

Os sistemas xestores de bases de datos permiten definir un intervalo de tempo para o envío dos ficheiros de rexistro, ou ben configurar un tamaño predeterminado deles. Cando un rexistro supera o intervalo de tempo establecido ou alcanza o seu tamaño máximo, péchase, e ábrese un novo ficheiro para rexistrar as transaccións. Os rexistros pechados van sendo enviados dende a fonte ao destino garantindo que a base de datos replicada en destino sexa consecuente coa fonte ata o último rexistro pechado. O RPO no sitio remoto dependerá do tamaño do ficheiro de rexistro e da frecuencia de cambio de rexistro na fonte.

#### **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 e 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 e Alok Shrivastava.
- Backup & Recovery. W. Curtis Preston e O'Reilly Media.
- Redes de área de Almacenamento na Wikipedia.  
[http://es.wikipedia.org/wiki/Red\\_de\\_área\\_de\\_almacenamiento](http://es.wikipedia.org/wiki/Red_de_área_de_almacenamiento)
- Cristopher Poelker e Alex Nikitin. Storage Area Networks for Dummies.
- G. Somasundaram, Alok Shirvastava "Information, Storage and Management: Storing, Managin and Protecting Digital Information". John Wiley & Sons. 6 de abril de 2009.
- · Jason Buffington "Data protection for Virtual Data Centers". Sybex. 2 de agosto de 2010.
- · Doug Lowe "Networking for Dummies". John Willey & sons. 29 de maio de 2007.

**Autor:** Francisco Javier Rodríguez Martínez

Subdirector de Sistemas Escola Superior Enxeñaría Informática Ourense

Colexiado do CPEIG