

# Administració del web

Jordi Serra Ruiz  
Miquel Colobran Huguet  
Josep Maria Arqués Soldevila  
Eduard Marco Galindo

PID\_00190185



*Els textos i imatges publicats en aquesta obra estan subjectes –llevat que s'indiqui el contrari– a una llicència de Reconeixement-NoComercial-SenseObraDerivada (BY-NC-ND) v.3.0 Espanya de Creative Commons. Podeu copiar-los, distribuir-los i transmetre'ls públicament sempre que en citeu l'autor i la font (FUOC. Fundació per a la Universitat Oberta de Catalunya), no en feu un ús comercial i no en feu obra derivada. La llicència completa es pot consultar a <http://creativecommons.org/licenses/by-nc-nd/3.0/es/legalcode.ca>*

# Índex

<b>Introducció</b> .....	5
<b>Objectius</b> .....	6
<b>1. Els servidors web i l'organització</b> .....	7
1.1. Servidor web i aplicacions .....	7
1.2. Servidor web de pàgines .....	7
1.3. Servidor web i servidor d'aplicacions .....	8
<b>2. L'administrador i el servidor</b> .....	10
2.1. Estructura d'un servidor web .....	10
2.1.1. Servidor/tallafoc .....	11
2.1.2. Ports .....	11
2.2. L'administrador .....	11
2.3. Responsabilitats genèriques de l'administrador .....	13
2.4. L'administrador i la creació de pàgines .....	14
<b>3. Recursos per a crear pàgines</b> .....	16
3.1. HTML .....	16
3.2. XML/XHTML .....	17
3.3. CSS .....	17
3.4. Galetes ( <i>cookies</i> ) .....	17
3.5. Pàgines dinàmiques .....	18
3.6. DHTML de client .....	19
3.6.1. JavaScript i JScript .....	19
3.6.2. <i>Applets</i> .....	20
3.6.3. VBScript .....	20
3.7. DHTML de servidor .....	21
3.7.1. SSI .....	21
3.7.2. CGI .....	22
3.7.3. Perl .....	23
3.7.4. Java .....	23
3.7.5. <i>Servlets</i> .....	24
3.7.6. PHP .....	24
3.7.7. ASP .....	25
3.7.8. JSP .....	25
3.8. Aplicacions web .....	25
3.8.1. CMS .....	26
3.8.2. Wikis .....	27
3.8.3. RSS .....	28
3.9. Aplicacions d'exemple .....	29

<b>4. Seguretat</b> .....	31
4.1. Seguretat de publicació .....	31
4.2. Seguretat del programari .....	31
4.3. Protocols de comunicació encriptats (HTTPS) .....	31
4.4. Registres de les connexions .....	32
4.5. Còpies de seguretat .....	32
4.6. Atacs al servidor .....	32
4.6.1. Falsificació de dades HTTP .....	33
4.6.2. Manipulació URL .....	34
4.6.3. Injecció de codi .....	34
4.6.4. Injecció SQL .....	35
4.7. Mètodes per a accedir a bases de dades de l'organització .....	35
4.7.1. Arquitectura a tres nivells .....	35
4.7.2. Restricció d'accessos .....	36
<b>5. Aspectes legals</b> .....	37
5.1. Propietat intel·lectual i societat de la informació .....	37
<b>6. Tasques/responsabilitats</b> .....	39
<b>Resum</b> .....	40
<b>Activitats</b> .....	41
<b>Exercicis d'autoavaluació</b> .....	41
<b>Solucionari</b> .....	43
<b>Glossari</b> .....	44
<b>Bibliografia</b> .....	46

## Introducció

En aquest mòdul veurem què és un servidor web, els components que el formen, els protocols que pot tenir i les maneres segures d'instal·lar-lo, ja que pot contenir informació sensible de l'organització.

També en veurem la utilitat dins la intranet o d'Internet i com es relacionen tots aquests elements entre si per a formar una part activa de l'organització.

El web pot ser la nostra imatge pública o una eina per a crear aplicacions intranet. També té el potencial d'accedir a bases de dades. Veurem maneres segures de fer-ho.

Actualment un web pot ser la base per a aplicacions i usos molt diversos. Cal tenir-ho present quan es dissenya l'entorn web de l'organització.

Finalment, parlarem de la figura més important: el *webmaster* o administrador del web. Potser no es dedica a omplir el web de contingut, sinó a controlar-lo. En aquest cas són els usuaris els que hi posen el contingut, i han de conèixer un mica els elements amb què es poden construir les pàgines per a poder ser realment creatius. En aquest cas s'ha d'anar amb compte, ja que la informació publicada al web de l'empresa pels usuaris pot estar en contradicció amb alguna política de privacitat de l'empresa o no respectar alguna de les lleis de privacitat de dades. Per tant, sempre és millor que els continguts tinguin alguna mena de control abans de fer-los públics.

## Objectius

Els materials d'aquest mòdul contenen les eines necessàries perquè l'estudiant assoleixi els objectius següents:

- 1.** Conèixer els components d'un servidor web.
- 2.** Saber les possibilitats que pot oferir un servidor web, tant si és a Internet com si forma part de la intranet.
- 3.** Saber com es pot instal·lar un servidor web.
- 4.** Conèixer les tecnologies i aplicacions més importants associades als entorns web.
- 5.** Saber com s'han d'instruir els creadors de continguts d'un servidor web, a partir dels recursos i els elements que poden fer servir.
- 6.** Conèixer les qüestions de seguretat i legals associades als servidors web.

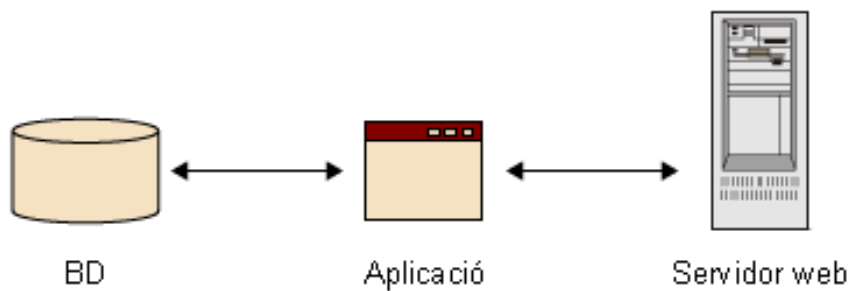
## 1. Els servidors web i l'organització

Quan es parla d'administrar un web es parla de ser responsable del funcionament d'un servidor web. Ara bé, un servidor web es pot veure des de dos vessants: com a interfície d'una aplicació o com a servidor de pàgines.

### 1.1. Servidor web i aplicacions

Normalment, quan un servidor web s'utilitza formant part d'una aplicació, simplement fa d'interfície per a l'aplicació, de manera que és un element més. Aquesta aplicació utilitza un navegador com a enllaç per a presentar les pantalles de sortida i com a mecanisme per a demanar informació a l'usuari. Les dades es guarden en una base de dades. Genèricament, l'esquema és el següent:

Servidor web com a element d'una aplicació



La distribució d'aquests elements en servidors físics depèn de molts factors com, per exemple, la mida i la complexitat de l'aplicació, que segurament estarà dividida en mòduls. Molt possiblement l'estructuració estarà condicionada al servidor web que es faci servir.

No podem oblidar que aquí hi haurà seguretat perquè aquesta aplicació no ha d'arribar necessàriament fins a Internet. Possiblement només interessarà que funcioni dins la intranet. La seguretat també ens marcarà molt l'estructura dels servidors.

En aquesta situació, doncs, el servidor web és un mitjà, tot i que molt flexible, per a fer funcionar l'aplicació, que és el centre principal de tot.

### 1.2. Servidor web de pàgines

Si fa de servidor web amb entitat com a tal, ho pot fer dins la intranet, a Internet o, fins i tot, una part a la intranet i l'altra a Internet.

## Terminologia

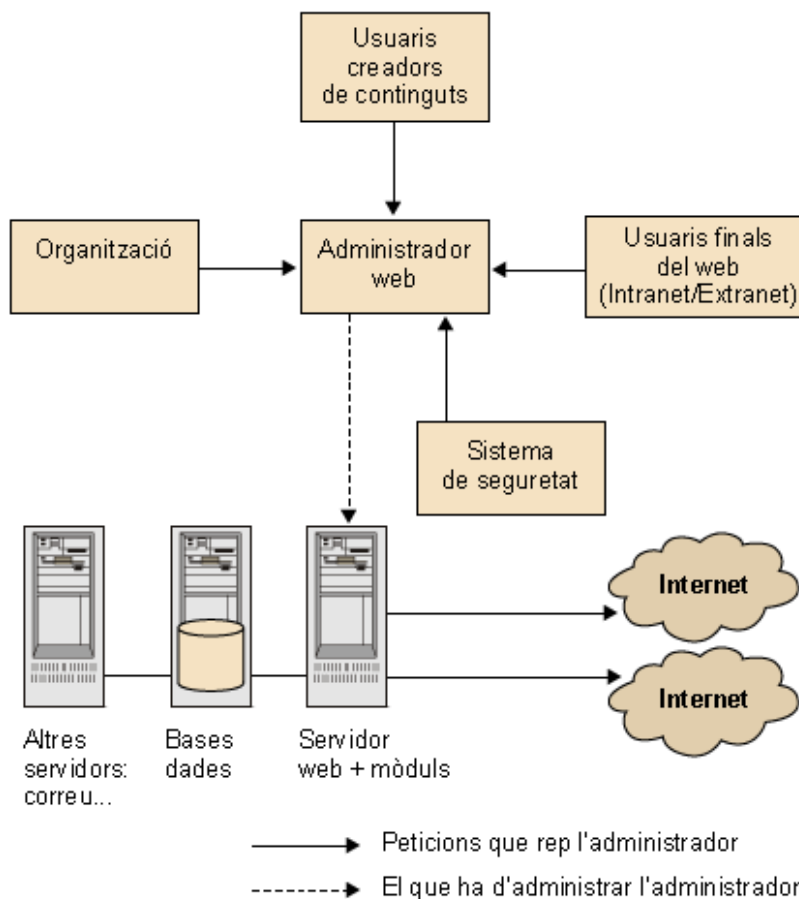
S'usa el terme intranet per a referir-se a una xarxa usada internament en una organització. El terme intraweb per a una web que abasta els nodes d'aquesta intranet. Però sovint s'usa intranet com a sinònim d'intraweb.

Quan s'utilitza una intranet i es publica informació a Internet, de vegades es fa servir el terme *extranet*.

La primera qüestió que cal tenir en compte és que exportem la imatge de l'organització a l'exterior i, per tant, s'ha de tenir cura de com presentem el web, com el mostrem i quina informació hi introduïm.

El web mostra la imatge corporativa de l'organització a la comunitat Internet.

Esquema de la relació entre l'administrador, el servidor web, l'organització i els usuaris



### 1.3. Servidor web i servidor d'aplicacions

Inicialment el servei web provenia només del servidor web, què proporcionava tots els recursos necessaris. Actualment, els sistemes han evolucionat i ja no podem afirmar que tots els recursos necessaris per a un servei web vinguin donats pel servidor web.



Amb l'arribada dels servidors d'aplicacions s'inicia una autèntica revolució en el tractament dels serveis web. Tot i que el punt d'entrada i primer tractament de les peticions es realitza en el servidor web, els sistemes complexos de generació de pàgines que utilitzen processament i accés a dades necessiten els recursos que ofereixen els servidors d'aplicacions.

Així doncs, actualment tenim dues possibles vies de servei segons els nivells d'accés necessaris:

- **En dos nivells (serveis estàtics).** Servits des d'un servidor web. En el primer nivell situaríem les estacions de treball, ja siguin locals o remotes, i en el segon nivell el servidor web. Estaríem parlant d'una estructura client/servidor aplicada a un entorn web.
- **En tres nivells (serveis dinàmics).** Servits des d'un servidor web i un servidor d'aplicacions. En el primer nivell trobarem les estacions de treball i els usuaris (locals o remots). En el segon nivell trobarem el servidor web (porta d'entrada) i el servidor d'aplicacions (proporciona capacitat de processament i gestió de dades). Finalment en el tercer nivell trobaríem el servidor de base de dades (dades centralitzades del servei web).

S'ha de tenir en compte, doncs, que, tot i que en general sempre parlarem de servidor web en relació amb els serveis web, pot ser que a causa de les necessitats de l'organització, el sistema es compongui de servidor web i de servidor d'aplicacions (aquests dos últims, no han de ser pas màquines físiques independents).

#### Vegeu també

Sobre l'arquitectura a tres nivells, consulteu el subapartat 4.7.1. d'aquest mateix mòdul.

## 2. L'administrador i el servidor

El binomi administrador-servidor ha d'estar al màxim de ben coordinat per a obtenir el millor rendiment possible.

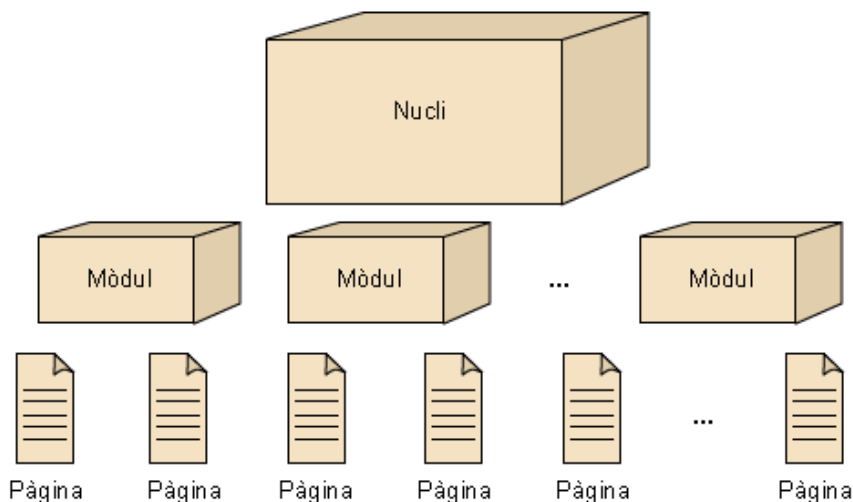
Conèixer, doncs, l'arquitectura de programari del servidor web que s'utilitza és fonamental per a l'administrador. En general, tots els servidors treballen d'una manera semblant i podem fer un esquema genèric de com són. També són bastant generals les funcions de l'administrador, malgrat que és la cultura i la història de l'organització concreta qui marca les tasques i les necessitats reals que ha de cobrir.

### 2.1. Estructura d'un servidor web

Els servidors web es componen bàsicament dels elements següents:

- **Nucli.** És el servidor com a tal, i és el centre del servidor web. Sempre està carregat i funcionant quan el servidor està engegat.
- **Mòduls.** Actualment els servidors web poden ser bastant complexos. Això és possible perquè s'ofereixen mòduls per a augmentar la funcionalitat del servidor. Aquests mòduls, coneguts també com a extensions del servidor, s'instal·len quan es configuren els servidors web. Llavors informem de quins mòduls s'han de fer córrer. En general, i sense entrar en detalls dels mòduls, un servidor pot fer moltes més funcions que dedicar-se exclusivament a lliurar pàgines HTML que estiguin guardades al disc dur.
- **Pàgines.** Les pàgines són el lloc on hi ha guardada la informació que presenta el servidor. Estan estructurades en forma d'arbre de directoris, tot i que en alguns casos hi ha mòduls que modifiquen aquesta informació.

## Estructura d'un servidor web



Un servidor web és, a grans trets, el nucli, els mòduls i les pàgines.

### 2.1.1. Servidor/tallafo

L'arquitectura de connexió física més comuna quan hi ha un servidor web que proporciona informació és tenir-lo darrere d'un tallafo<sup>1</sup>. El concepte és que el tallafo només deixa passar les peticions de dades que el servidor ha de proporcionar, simplement s'ha d'instal·lar el servidor d'acord amb unes normatives de seguretat que l'organització ja ha de tenir. Com que el servidor web mantindrà una certa interacció amb l'usuari, el fet d'estar protegit per un tallafo obligarà a tenir un control rigorós de quines portes d'accés ha d'oferir i quines no.

<sup>(1)</sup>En anglès, *firewall*.

#### Vegeu també

Vegeu el mòdul "Administració de la xarxa" on es parla més dels tallafocs.

### 2.1.2. Ports

El port estàndard de petició de documents o altres recursos web és el 80. En principi, un servidor web pot escoltar peticions per qualsevol port. Els servidors cau<sup>2</sup> en canvi, poden estar escoltant pel mateix port 80 o pel 8080, i en tots dos casos es tracta d'una configuració estàndard.

<sup>(2)</sup>En anglès, *proxy server*.

#### Servidor cau

Un servidor cau actua com a intermediari entre una xarxa interna i una connexió externa a Internet. Així es pot compartir la mateixa connexió per rebre informació des de servidors web. S'usa també per a filtrar continguts.

## 2.2. L'administrador

L'administrador és qui manté el servidor en funcionament, i també qui administra la seguretat i els permisos, i per tant, qui ha de conèixer millor que ningú totes les seves possibilitats.

En concret l'administració ha de fer el següent:

- **Instal·lar i administrar el servidor web.** El servidor web és un recurs amb el qual l'organització dóna accés a informació que vol publicar. Aquest servidor web, per exemple, podria donar servei només a usuaris d'una xarxa local, o potser d'una intranet, o com en la majoria de casos, donar una porta d'accés lliure per a qualsevol usuari connectat a la xarxa Internet. Depenent del servei al qual estigui destinat aquest servidor web, les configuracions i polítiques de disseny variaran.

Un servidor web no planteja gaires problemes a l'hora d'instal·lar-lo, ja que com a concepte només es tracta d'indicar on hi ha la informació i com ha d'acceptar les connexions dels usuaris que hi vulguin accedir. A més a més, el servidor ja proporciona eines de control que permeten de monitoritzar i posteriorment analitzar-ne el rendiment. Per exemple, és molt comú que un servidor web tingui disponible la quantitat d'accessos simultanis que té i quin és el seu rendiment.

- **Conèixer el programari.** Hi ha dues classes de programari que l'administrador del web ha de fer servir. D'una banda, el que s'utilitza per a configurar el servidor i, de l'altra, el que proporciona les eines de preparació, modificació i organització de les diferents pàgines que formen el web que cal servir. Un tema molt important respecte al programari relacionat amb un servidor web és el que correrà a la màquina client, ja que no tots els navegadors interpreten el codi de la mateixa manera. Així, doncs, és important prendre consciència dels clients que visitaran el web ofert pel nostre servidor. En cas de desenvolupar una intranet, potser podem ser més lliures, atès que és molt probable que hi hagi un navegador estàndard per a tota l'organització.
- **Administrar els directoris.** Quan s'ha de dissenyar un nou web, també s'ha de tenir en compte com s'organitzaran els diferents recursos. El més habitual, i que el mateix programari de desenvolupament de les pàgines ja gestiona, és tenir-los classificats en carpetes o directoris segons el tipus. Així, doncs, és habitual que hi hagi un directori d'imatges, al qual totes les pàgines facin referència. Això permet de compartir, per exemple, fitxers de logos de l'organització, ja que no s'han d'incloure totes les ubicacions on hi hagi documents que els requereixin. L'estructura en directoris també permet, si mai és necessari, una migració molt menys costosa.
- **Administrar les pàgines.** Normalment, les pàgines queden fora de l'abast de l'administrador del web, ja que com veurem més endavant hi ha una figura que s'acostuma a dedicar a generar-les. De tota manera, és important ser conscient de la importància d'arribar amb "pocs passos" on sigui necessari. Així, doncs, amb l'experiència d'organitzar informació, fitxers, directoris i aquest coneixement abstracte de com cal moure's per dins d'una màquina, un administrador ha de donar suport al creador de pàgines i

aconsellar com cal disposar els enllaços per a accedir d'una manera més ràpida a tot el web des de qualsevol lloc.

L'administrador instal·la el web i en té cura, no del contingut.

### 2.3. Responsabilitats genèriques de l'administrador

Un servidor web proporciona una porta d'accés a una gran quantitat d'informació d'una manera molt còmoda per al client i, per tant, es tracta d'un servidor que, tot i que durant la instal·lació no presenta gaires problemes, sí que s'ha de plantejar molt bé com cal administrar cadascuna de les parts que el componen, i tots els temes relacionats amb el web, ja que el fet de proporcionar informació és una responsabilitat que recau en mans de l'administrador.

Podem destacar les següents responsabilitats de l'administrador:

- **Responsabilitat del servidor.** Un servidor web no és excessivament complex d'administrar. El més habitual és que el mateix servidor enregistri les dades que es necessita avaluar. És aconsellable tenir-lo (com la resta de servidors de l'organització) sota una alimentació elèctrica ininterrompuda i que disposi d'una connexió amb uns mínims de cabal garantits perquè un increment de trànsit en la xarxa local no afecti els clients remots que consultin el web. Habitualment, els servidors web es configuren amb dues línies alternatives de connexió a Internet (ADSL i fibra òptica, per exemple) i se'ls proporciona un mecanisme que permeti un canvi instantani en la connexió a Internet de còpia de seguretat quan es detecti algun problema, i un cop s'hagi arreglat torni a servir el web pel canal habitual.
- **Responsabilitat dels continguts.** És important adonar-se dels dos aspectes del contingut en un servei web:
  - Hi ha d'haver un control de quines dades es proporcionen a cada instant, de manera que no hi hagi continguts antiquats, repetits, sensibles o que no siguin coherents entre ells.
  - En relació amb l'accés a aquestes dades, hi ha d'haver un control de com es gestiona, consulta i modifica la base de dades a la qual pot estar accedint el servidor web.
- **Responsabilitat de l'homogeneïtat.** Un aspecte més de disseny que no pas d'administració, però que recau molt sovint en l'administració del web, i que és molt important, és el control de la imatge que reflecteix el web. Com que es tracta d'una porta oberta a molts usuaris potencials, que poden no ser coneguts, s'ha de tenir molta cura de com es presenten els recursos. Així, doncs, és important tenir en compte que en totes les pàgines hi ha d'haver un estil constant i que, per exemple, no ens trobem amb una pàgina amb el fons blau, la següent amb el fons negre, i si anem a un

altre apartat del web, trobéssim que totes les pàgines que el formen tenen un fons comú amb el logo de l'organització.

- **Generació d'estadístiques.** Segons les necessitats de cada organització, aquesta tasca pot ser més o menys important. En general, és important tenir un registre de qui ha visitat el web. Tant què ha vist com quan i des d'on ho ha fet. A part, hi ha altres tipus d'estadístiques molt interessants com, per exemple, a quina velocitat s'ofereixen els documents o la quantitat de peticions que hi ha sobre les diferents pàgines, i el temps total que un client es passa navegant per tot el web.

L'administrador no ha de fer necessàriament les pàgines, però es preocupa de totes, i també del disseny global del web.

## 2.4. L'administrador i la creació de pàgines

Actualment, el paper de l'administrador web<sup>3</sup> està dividit. Recordeu que l'estructura d'un web està formada pel nucli, els mòduls i les pàgines. L'administrador del web s'encarrega que el web funcioni bé, que els aspectes de seguretat siguin com necessita l'usuari i l'organització, que se segueixin les directrius que marca l'organització i que el contingut global sigui homogeni.

Però pot no tenir cura dels continguts. La tendència és que les pàgines les mantingui directament la mateixa font d'informació. Això es fa donant plantilles i formant les persones que se n'hagin d'encarregar. Per tant, l'administrador del web té dues classes d'usuaris:

- **Consumidors del material del web.** Els consumidors del material del web són els usuaris finals, els que es connecten per mitjà d'un navegador i consulten el web per saber alguna cosa.
- **Creadors del material del web.** Ja hem dit que l'administrador del web supervisa els continguts, però no els fabrica. Pot aconsellar sobre com fer-los, donar plantilles, donar eines, explicar les possibilitats del web i facilitar els mecanismes del servidor web perquè els creadors del material del web puguin fer la pàgina de la millor manera possible.

En el seus inicis, el creador de pàgines web era una persona amb un perfil bàsicament artístic. No tenia coneixements tècnics, però sí que sabia bé com combinar elements per a crear efectes vistosos. Nosaltres li havíem de proporcionar els coneixements dels elements que integren els webs: HTML, Java, JavaScript, etc.

<sup>(3)</sup>Administrador web en anglès s'expressa com a *webmaster*.

### Vegeu també

Recordeu que al subapartat 2.1 teniu un esquema on es mostra l'estructura d'un servidor web.

Podríem considerar que l'administrador web hauria de seguir pautes com les que es detallen tot seguit:

- Com que hi ha uns propietaris que tenen la responsabilitat de la informació que el servidor web està servint, se'ls ha d'involucrar en la manera en què aquesta informació es presenta. Així, doncs, és important que l'administrador s'encarregui d'orientar i introduir en el món de l'HTML, l'FTP, que ensenyi a fer servir escàners, a convertir formats de fitxers, a buscar per Internet o a utilitzar les propietats del navegador.
- Per a estalviar temps, ajuda molt disposar de plantilles de disseny (habitualment HTML) que es puguin proporcionar en cas que algun creador demani informació.
- Cal tenir creada una política d'ús del servidor web. Així, com que hi ha un grup que ha acceptat la política, no hi pot haver malentesos sobre continguts.
- Verificar els enllaços. Els documents es mouen constantment d'un lloc a l'altre i això representa un problema molt seriós, ja que és molt important que un servidor web no ofereixi enllaços "caducats".

L'administrador és qui promou la utilització dels recursos del web entre els creadors de pàgines. Els proporciona tots els recursos i coneixements necessaris perquè puguin fer pàgines web i posar-les en el servidor.

### 3. Recursos per a crear pàgines

El manteniment de les pàgines cada vegada migra més cap a la mateixa font d'informació. Això permet que s'actualitzi al moment, no hi hagi tants errors, se'n simplifiqui el manteniment i es distribueixi la càrrega de treball.

S'ha de formar els creadors de pàgines en l'ús d'aquestes tecnologies, ja que possiblement a mesura que les facin servir hi trobaran més possibilitats i aniran essent més exigents.

Com que moltes vegades aquest creador és l'usuari final, se l'ha de formar des de la perspectiva de les possibilitats que li ofereix el servidor. Pot fer pàgines molt simples (una pàgina HTML) o molt complexes (programació JSP<sup>4</sup>). El creador ha de saber quines eines i quins recursos hi ha per a decidir si els necessita o no per a la seva necessitat en concret. És l'administrador del web qui, de la manera més simple i didàctica possible, orienta l'usuari final en les possibilitats del servidor web, perquè les faci servir com més li convingui. Fem, doncs, un repàs a totes les característiques de creació de les pàgines web.

<sup>(4)</sup>“JSP” és l'abreviatura de *Java Server Pages*.

#### 3.1. HTML

L'HTML<sup>5</sup> és el cor de totes les pàgines web. És el “llenguatge” amb què estan fetes totes les pàgines i és el llenguatge que interpreten tots els navegadors.

<sup>(5)</sup>HTML és la sigla d'*hypertext markup language*, en català, “llenguatge de marcatge d'hipertext”.

L'HTML és un llenguatge basat en marques. Sempre van entre '<' i '>' i no és important si estan escrites en minúscules o en majúscules. Les marques van en parelles de la forma: <marca> </marca> i tot el que engloben s'escriu al mig.

L'HTML conté el text que es vol presentar a l'usuari, juntament amb informació sobre el format com, per exemple, la mida de la lletra, el color, la informació sobre si hi ha fotografies, les imatges de les fotografies, la informació sobre taules, marges, etc. L'HTML és un llenguatge molt complet i actualment anem camí de la versió 5.

##### HTML 5, la nova versió

Actualment estem usant la versió 4.01 de l'HTML, que fou aprovada el 24 de desembre de 1999 pel World Wide Web Consortium (W3C) –el consorci que marca l'estàndard d'HTML, entre moltes altres coses– (l'especificació es pot consultar en línia).

La nova versió del llenguatge d'Internet (en desenvolupament pel W3C), l'HTML 5, incorpora molts canvis com per exemple les etiquetes *article*, *dialog*, *figure*, *footer*, *header*, *section*. S'espera tenir una revisió estable cap al 2012.



En aquest format es posen els famosos i coneguts enllaços<sup>6</sup>, que permeten d'anar d'un punt a un altre de la mateixa pàgina, a una altra pàgina dins el mateix servidor o a una altra pàgina d'un altre servidor qualsevol. La marca per a fer-ho és `<A href="URL">Text</A>`.

### Veure una pàgina

És possible veure el codi font d'una pàgina web. Per exemple, amb el navegador Firefox hi ha l'opció "Visualitza>Codi font de la pàgina" (o la combinació CONTROL + U), que obre una finestra on es veu tota la pàgina que mostra el navegador amb el format HTML.

Conèixer l'HTML és bàsic per a tot administrador de web.

## 3.2. XML/XHTML

En virtut de l'existència de l'SGML<sup>7</sup>, i en un intent de tenir un llenguatge de marques únic, el World Wide Web Consortium (W3C) defineix l'XML<sup>8</sup>. És un estàndard ajustat a les necessitats d'Internet, ja que l'SGML és massa complex de fer servir. Com que l'HTML 4.01 no és XML compatible, el W3C va oferir un estàndard HTML compatible amb XML, l'XHTML.

## 3.3. CSS

Tot el que hem fet fins ara ens permet de fabricar pàgines, però ens hem de preocupar de donar un estil uniforme a cadascuna de les pàgines. I si volem, per exemple, que totes les pàgines tinguin la mateixa font de lletres i els mateixos colors?

Fins ara això ho havíem de fer d'una a una. Amb l'aparició del CSS<sup>9</sup> ja no és necessari. És un fitxer d'estil en què s'inclou tota la informació rellevant de l'estil de la pàgina, i totes les pàgines el seguiran.

La separació de la presentació del contingut en els documents simplifica la creació de pàgines i el manteniment del web.

Com que era un gran problema per als dissenyadors de pàgines, la idea es va acceptar ràpidament i el 1998 en va sortir la versió 2, el CSS2.

## 3.4. Galetes (cookies)

Si volem fer un web dinàmic, tenim la possibilitat de fer servir galetes<sup>10</sup>, que són petits fitxers que alguns webs guarden a l'ordinador de l'usuari. D'aquesta manera podem saber les pàgines que visita el client per tal d'oferir-li, des del servidor, pàgines a mida segons les que hagi visitat.

<sup>(6)</sup>Enllaç és el terme català. També es fa servir molt el terme original anglès *link* o *hiperlink*.

### Exemple d'enllaç

Per a anar a la pàgina de la UOC posaríem `<A href=http://www.uoc.edu> Connectar-vos a la UOC </A>`.

<sup>(7)</sup>"SGML" és la sigla de *standard generalized markup language*, en català, llenguatge de marcatge generalitzat estàndard.

<sup>(8)</sup>"XML" és la sigla d'*eXtensible Markup Language*, en català, 'llenguatge d'etiquetatge extensible'.

<sup>(9)</sup>"CSS" és la sigla de *cascading style sheets*, estàndard proposat el 1996 pel W3C.

### CSS versió 3

La versió 3 de CSS sembla accidentada. La versió 2, apareix el 12 maig de 1998. Amb els problemes d'adopció per part dels navegadors, ha passat per diversos estadis de revisió, i el 19 de juliol de 2007 apareix la CSS 2.1.

De la versió 3 es va iniciar el desenvolupament el 1998, i deu anys més tard encara no està enllestida.

<sup>(10)</sup>En anglès, *cookies*.

### 3.5. Pàgines dinàmiques

Tot el que hem fet fins ara ens permet de fabricar pàgines, però aquestes són “estàtiques”. Resideixen en el servidor i s’entreguen a través de la xarxa al client a petició d’aquest. El client sempre mostra els elements en el mateix ordre i posició. Si volem canviar quelcom (ordre, posició i/o elements), hem de reescriure el codi HTML. Així doncs, si volguéssim fer un buscador o un gestor de correu estaríem molt limitats amb HTML.

Arran d’aquestes limitacions va aparèixer el que es coneix com a DHTML<sup>11</sup>, que és el conjunt de tècniques per a crear pàgines dinàmiques. DHTML és molt ampli i inclou moltes tècniques i molts llenguatges de programació. Per tant, DHTML, realment no és una nova especificació de l’HTML. De fet és una manera de veure i controlar els codis i comandaments de l’HTML estàndard.

<sup>(11)</sup>“DHTML” és la sigla de *Dyna-mic HTML*, en català, ‘HTML dinàmic’.

A diferència de l’HTML estàtic, una pàgina de DHTML conté elements (com per exemple *scripts*) en el costat del client que canvien l’HTML del document. Això afecta l’aparença i funcions dels objectes de la pàgina. D’aquesta manera obtenim més control sobre la pàgina. Això és possible perquè els navegadors actuals inclouen aquestes capacitats.

Els components d’una pàgina d’HTML dinàmic són:

- **Model en objectes per a la representació de documents ( DOM<sup>12</sup>).** És un model orientat a objectes que els programes i *scripts* poden usar per a accedir i modificar dinàmicament el contingut, estructura i estil dels documents HTML i l’XML.
- ***Scripts*.** Són petits programes (escrits habitualment en JavaScript) incrustats en l’HTML que permeten la interactivitat del codi i el control dels objectes especificats en el DOM.
- **Pàgines d’estil (CSS).** S’usen en DHTML per a controlar l’aparença dels documents. Les pàgines d’estil defineixen els colors, fonts de text, color de fons, imatges i el posicionament d’objectes.
- **HTML 4.x.** Usat per a crear la pàgina i els elements que usaran les pàgines d’estil i el model DOM.

<sup>(12)</sup>“DOM” és la sigla de *Document Object Model*.

#### Adreça recomanada

El model en objectes per a la representació de documents ha estat desenvolupat pel W3C i es pot trobar l’especificació a <http://www.w3.org/DOM>.

Hi ha dues classes diferenciades de DHTML, el DHTML de client i el DHTML de servidor.

### 3.6. DHTML de client

El DHTML de client inclou el conjunt de tècniques per a la construcció de pàgines dinàmiques que es desenvolupen dins l'àmbit d'una pàgina web, quan la pàgina s'està veient en la pantalla de l'usuari, és a dir, en el navegador. En aquesta situació, per a fer qualsevol efecte o interactivitat el recurs és el navegador. Per això es parla de *client*.

Aquest tipus de programació s'utilitza per dotar d'efectes, so, vídeo, menús interactius i resposta les accions de l'usuari a les pàgines. Permet alliberar el servidor de tasques que pot resoldre localment l'ordinador client (com la validació de part de les dades d'un formulari per exemple).

#### 3.6.1. JavaScript i JScript

Fins ara les pàgines eren estàtiques. L'usuari les veia tal com les fèiem i no es modificava res en cap moment. A partir d'aquest moment intentem fer pàgines més dinàmiques, amb més possibilitats.

Volem fer algun càlcul, una pàgina interactiva, o potser una pàgina que faci coses depenent de la posició del ratolí a la pantalla de l'usuari. Sigui el cas que sigui, una possible solució és fer servir JavaScript. Programació dins la pàgina, que s'executa en el client.

JavaScript és un llenguatge per a escriure *scripts* (guions) que va ser desenvolupat per Netscape. Permet de crear pàgines interactives d'una manera molt senzilla. JavaScript no és Java. El que es necessita per a executar JavaScript és simplement un navegador amb l'opció d'executar JavaScript activada.

El JavaScript s'introdueix directament en el codi HTML, entre les marques `<script language="JavaScript">` i `</script>`.

Un dels punts més importants en la programació en JavaScript és la captura d'esdeveniments. Així, doncs, es pot crear una pàgina que tingui en compte els moviments i cada vegada que es fa clic amb el ratolí. A part, també es poden definir funcions dins la pàgina i, per exemple, crear botons que en pressionar-se fan que s'executi aquesta funció.

Finalment, tingueu present que JScript és la versió de Microsoft per al seu navegador de Javascript, amb algunes incompatibilitats.

Però s'ha d'anar amb molta cura a l'hora d'habilitar i fer servir el JavaScript, ja que el fet de poder executar des del servidor programes en les màquines client el fa molt interessant per a executar codi maliciós, ja que només cal carregar

una pàgina web amb un JavaScript que el client es baixará i executarà en el seu ordinador sense que li demani permís. Hi ha tècniques en què es poden aconseguir les galetes (*cookies*).

### 3.6.2. Applets

Un *applet*<sup>13</sup> és una aplicació que s'executa en el context d'un altre programa, típicament un navegador web. L'*applet* s'ha d'executar en un entorn anomenat contenidor a través d'un connector<sup>14</sup>. A diferència d'un programa, un *applet* no pot executar-se de manera independent i té uns privilegis de seguretat molt restringits.

<sup>(13)</sup> *Applet* en català s'expressa com a "miniaplicació".

<sup>(14)</sup> En anglès, *plug-in*.

El principal avantatge d'utilitzar *applets* resideix en el fet que són molt menys dependents del navegador que els *scripts* i són independents del sistema operatiu de l'ordinador en el qual s'executen. Per contra, el seu principal desavantatge en relació amb els *scripts* és que són més lents de processar i no tenen accés a tots els altres components de la pàgina.

Els *applets de Java* són una manera d'incloure codi que s'executa en els clients que visualitzen una pàgina web. Es tracta de petits programes fets en Java, que es transfereixen junt amb les pàgines web i que el navegador executa automàticament en l'espai de la pàgina. Els *applets* de Java estan programats en Java i precompilats. Per aquest motiu funcionen diferent que els llenguatges de *script* com Javascript. Els *applets* són més difícils de programar i requereixen uns coneixements mitjans del llenguatge Java.

Les restriccions de seguretat (l'entorn de seguretat) en què corren són molt fortes, de manera que el programa Java no pot fer tot el que vulgui. De fet, no pot obrir connexions en altres ordinadors que no siguin el *host* del qual prové ni pot esborrar els fitxers del disc dur, per exemple. Tot el Java està orientat i pensat per a córrer per Internet, de manera que encara que es fabriquin programes maliciosos, no puguin operar obertament en els ordinadors objectiu.

En ser Java és més potent que Javascript, el nombre d'aplicacions potencials dels *applets* pot ser més gran.

Els *applets* Java es copien del servidor al client i s'executen sobre l'ordinador client, amb fortes restriccions de seguretat.

#### Exemples d'applets

Exemples d'*applets* són les *Java applets*, les animacions Flash i el Windows Media Player usat per mostrar vídeo als navegadors.

### 3.6.3. VBScript

VBScript o Visual Basic Scripting és similar al llenguatge Visual Basic, però té bàsicament la mateixa estructura que JavaScript.

VBScript és el llenguatge de *scripting* desenvolupat per Microsoft per a pàgines web pel seu navegador. VBScript no té relació amb Javascript. Són dos llenguatges competint, però que poden fer les mateixes funcions.

### 3.7. DHTML de servidor

Quan un client sol·licita una pàgina, el servidor executa els *scripts* i genera una pàgina resultat, que és enviada al client. La pàgina resultat només conté codi HTML, de manera que pot ser interpretada per qualsevol navegador sense errors, independentment de la seva versió. Hi ha una sèrie de llenguatges basats en servidor per executar els seus *scripts*. Genèricament s'anomenen *embedded scripting language* (ESL).

Només la independència del navegador ja és un avantatge que justifica l'existència d'aquesta programació, però ara comptem amb tots els recursos del servidor on estan allotjades les pàgines. Aquests recursos, com podrien ser gestors de bases de dades, servidors de correu o el sistema d'arxius del servidor, ens permeten construir tota mena d'aplicacions.

Cal destacar que el codi de les pàgines mai no arriba al client. El navegador només rep HTML, i això implica que els visitants d'una pàgina mai no poden accedir al nucli de les aplicacions que s'hagin desenvolupat, és a dir, als *scripts* del costat del servidor.

#### 3.7.1. SSI

El Server Side Includes (SSI<sup>15</sup>) no es pot considerar un llenguatge. Són un conjunt de funcions del servidor que permeten modificar el document demanat abans de ser enviat al navegador del client. La finalitat bàsica és inserir contingut com ara la sortida de comandes del sistema.

<sup>(15)</sup>SSI és la sigla de l'expressió anglesa *server side includes*, que es pot traduir per 'inclusions al costat del servidor'.

El funcionament bàsic de les SSI és:

- 1) El client fa una petició d'una pàgina per mitjà del navegador.
- 2) El servidor rep la petició i executa les *server side includes* que s'indiquen en el fitxer, i com a resultat d'això obté un fitxer HTML que guarda en memòria.
- 3) Envia al navegador de l'usuari aquest fitxer HTML perquè el mostri.

Les inclusions al costat del servidor estan escrites com a comentaris HTML de la forma `<! funcio atribut="argument" >`.

### 3.7.2. CGI

CGI<sup>16</sup> és de les primeres extensions que van aparèixer. La idea originària es basava a fer córrer un programa en el servidor que fos capaç de fer qualsevol cosa i generar uns resultats que fossin una pàgina HTML. Va ser el primer “truc” per a accedir a bases de dades i per a crear pàgines que no estiguessin estàtiques en el servidor (pàgines *on line* o pàgines en línia).

<sup>(16)</sup>“CGI” és la sigla de Common Gateway Interface.

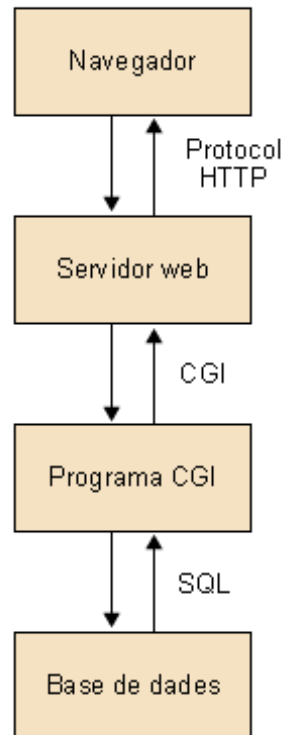
Cal destacar que amb CGI el servidor executa un programa arbitrari, fet amb qualsevol llenguatge de programació. El mètode, a grans trets, és el següent:

- 1) El client informa que cal que s’executi l’aplicació.
- 2) El servidor posa en marxa el programa CGI que li ha demanat el client.
- 3) Aquest programa s’executa (pot consultar una base de dades, per exemple) i treu el resultat per la sortida estàndard (*standard output*) ja preparat i formatat en format HTML. De fet, és una pàgina HTML.
- 4) El servidor captura aquesta sortida i l’envia al client.

Els inconvenients dels CGI són:

- Carrega molt el servidor, ja que s’hi executa sempre.
- En cada petició s’ha de carregar i executar el programa i, per tant, és molt costós.

Com que hi ha tecnologies més actuals, ha caigut bastant en desús.



CGI és un recurs utilitzat àmpliament, ja que va ser el primer a executar codi arbitrari en el servidor i a enviar el resultat al client.

### 3.7.3. Perl

El CGI<sup>17</sup> és un enllaç per a executar programes arbitraris en un servidor web. El Perl és un dels llenguatges usats habitualment per programar els CGI, tot i que es poden usar altres llenguatges com C, C++ o VisualBasic. Tot i això, el Perl s'utilitza en molts altres llocs, i no solament per a CGI.

Perl ve de *practical extraction and report language* i té unes funcions de manipulació de text molt potents. Amb aquest llenguatge es poden definir tot tipus d'algorismes per a tractar fitxers en el sistema i generar les sortides que es volen.

### 3.7.4. Java

Java va aparèixer com un llenguatge de propòsit general però molt orientat a Internet. No està pensat per a cap plataforma concreta, sinó per a una plataforma inexistente, una "plataforma virtual" i, per tant, funciona en qualsevol plataforma sobre la màquina virtual corresponent, sense necessitat de recompilar el codi generat pel compilador. Aquesta independència de plataforma és el que l'ha fet un dels llenguatges preferits per Internet.

<sup>(17)</sup> Recordeu que CGI és la sigla de *common gateway interface*.

#### Origen de Perl

Perl va ser creat per Larry Wall. Volia simplificar les tasques d'administració del sistema operatiu Unix. Ara és un llenguatge de propòsit general.

#### Adreça recomanada

Java és un llenguatge creat per Sun. Trobareu més informació a la següent adreça de Sun: <http://java.sun.com>.

Hi ha diferents edicions de la plataforma Java amb funcionalitats diferents. Trobem *Java 2 Standard Edition* (J2SE), que és l'entorn de desenvolupament d'aplicacions Java orientat a les aplicacions individuals i els *applets*. *Java 2 Enterprise Edition* (J2EE) és la plataforma Java per crear aplicacions Web. I l'edició *Java 2 Micro Edition* (J2ME), que és una versió reduïda de la plataforma Java 2 per a usar en petits dispositius com a telèfons mòbils o agendes electròniques.

De fet, Java s'usa gairebé per a qualsevol cosa que es pugui fer en qualsevol llenguatge (moltes vegades amb grans avantatges). Podem programar pàgines web dinàmiques, amb accessos a bases de dades. La programació dinàmica amb Java s'ha estès tant que no és difícil trobar-nos *servlets* al costat del servidor i *applets* de Java al costat del navegador.

### 3.7.5. *Servlets*

Els *servlets* són la resposta de la tecnologia Java a la programació CGI. Són programes que s'executen en un servidor web i construeixen una pàgina web. Així, doncs, es pot construir una pàgina amb dades que l'usuari acaba d'introduir, o potser amb el resultat dels motors de cerca. Si hem de tenir pàgines amb informació de bases de dades, es pot fer servir JDBC<sup>18</sup> per a accedir-hi i després treure per la sortida estàndard el codi HTML que el servidor web capturarà i enviarà al navegador del client. Un avantatge important respecte als CGI és que cada petició s'arrenca com un fil d'execució lleuger de la màquina virtual, mentre que amb CGI s'arrenca cada vegada un procés, i això pot comportar una sobrecàrrega del sistema.

Els *servlets*, comparats als CGI, tenen molts avantatges. Així poden manejar i sincronitzar moltes peticions concurrents (es podria fer un sistema de conferència *on-line* per exemple). La primera generació de *servlets* era pesada i requeria força coneixements. La solució va passar per JSP, que permet incloure codi Java dins de les pàgines HTML.

### 3.7.6. PHP

El PHP<sup>19</sup> és una eina que permet de crear pàgines web dinàmiques. Són pàgines que es tracten com si fossin HTML. Per defecte, l'extensió és "php" i cal que el servidor ho suporti. En aquest cas, el servidor les interpretarà abans d'enviar-les al client. No s'ha de compilar res ni instal·lar eines addicionals. La marca especial per al PHP és `<?php instruccions?>`.

Una de les característiques més potents del PHP és la manera com tracta els formularis HTML, ja que un element en un formulari es converteix directament en una variable amb el mateix nom, amb el valor que l'usuari ha posat en el formulari.

#### Vegeu també

Recordeu que hem tractat els *applets* al subapartat 3.6.2 d'aquest mateix mòdul.

<sup>(18)</sup>JDBC és la sigla de *Java Database Connectivity*. Permet interactuar amb una base de dades amb un programa Java.

<sup>(19)</sup>PHP és la sigla d'*hypertext pre-processor*, en català, 'preprocessador d'hipertext'.

#### Adreça recomanada

La versió actual de PHP és gratuïta i està sota llicència GPL. Es pot trobar a: <http://www.php.net>.



El llenguatge és com una barreja del millor de C, Java i Perl, amb característiques pròpies de PHP. El resultat està orientat a crear pàgines web dinàmiques ràpidament.

### 3.7.7. ASP

ASP<sup>20</sup> és una DLL que estén el Microsoft Internet Information Server (servidor web de Microsoft). Genèricament, els passos són els següents:

- 1) El client fa una petició d'una pàgina ASP per mitjà del navegador.
- 2) El servidor llegeix el fitxer ASP i l'executa, amb la qual cosa obté un fitxer HTML que guarda en memòria.
- 3) Envia al navegador de l'usuari aquest fitxer HTML perquè el mostri.

### 3.7.8. JSP

JSP<sup>21</sup> és una tecnologia que ens permet de barrejar HTML estàtic amb HTML generat dinàmicament, i integrar codi Java dins la pàgina JSP. A part, tenim l'avantatge respecte a ASP<sup>22</sup> que s'escriu en un llenguatge que no és propietat de Microsoft i que es pot dur a altres sistemes operatius i servidors web.

#### Accedir a altres programes per mitjà del servidor

Amb tot el que hem vist, sembla que una de les coses més importants sigui l'accés a altres programes per mitjà del servidor. Per què?

Doncs, senzillament, perquè la majoria de vegades la petició d'informació que faci un client navegador al nostre servidor implicarà un seguit d'accions que cal dur a terme per tal de preparar la informació corresponent. En la pràctica, la informació no acostuma a ser estàtica, sinó que s'ha d'anar a buscar a bases de dades i, per tant, el resultat ha de ser diferent segons el moment en què es faci la consulta. Per exemple, si un client navegador pregunta per mitjà d'una pàgina web que proporciona informació sobre entrades de concerts, és evident que s'ha de fer una consulta per a obtenir el resultat, i després presentar-lo al client.

## 3.8. Aplicacions web

L'avanç en les tecnologies web ha provocat que s'utilitzin els servidors web i les bases de dades com a tecnologies de base per a oferir serveis i resoldre les noves necessitats que ha provocat la situació. Així ens trobem amb noves aplicacions que utilitzen tecnologia web no com a fi, sinó com a mitja per a proporcionar un nou servei a l'organització i a l'usuari.

<sup>(20)</sup>ASP és la sigla d'*active server page*, en català, servidor de pàgines actiu.

#### ASP.net

La darrera versió treta per Microsoft, l'anomenada plataforma.net incorpora molts canvis que la fan incompatible. Així ja no pot usar VBScript, sinó VB.net. ASP.net és orientat a objectes.

<sup>(21)</sup>JSP són les sigles de *Java server pages*; en català, 'servidor de pàgines Java'.

<sup>(22)</sup>ASP són les sigles d'*active server page*.

### 3.8.1. CMS

Un sistema de gestió de continguts (CMS<sup>23</sup>) és un programa que permet crear una estructura de suport per a la creació i administració de continguts per part dels participants. És una aplicació de base de dades que automatitza el procés d'administrar i mantenir la publicació de contingut. Un bon sistema d'administració de contingut separa el contingut de la presentació.

<sup>(23)</sup>CMS és la sigla de *content management system*.

La divisió entre contingut i presentació porta diferents avantatges al llarg del cicle de vida de la publicació i mes enllà d'aquest. El disseny pot ser concebut, estès i de fet relançat sense la necessitat d'extreure el contingut existent o reinserir-lo.

Un sistema de gestió de continguts consisteix en una interfície que controla una o diverses bases de dades on s'allotja el contingut (texts, imatges, etc.) que veurem en el lloc web. El sistema permet manejar de manera independent el contingut i el disseny. Així, és possible conservar el contingut i canviar en qualsevol moment el disseny sense haver de modificar el format de tot el contingut. A més a més, permet la publicació fàcil i controlada al lloc web. Un exemple clàssic és el d'editors que carreguen el contingut al sistema i un altre de nivell superior que permet que aquests continguts siguin visibles a tot el públic.

Un CMS té dues funcions principals: facilitar la creació de continguts i la presentació d'aquests continguts. En relació amb la primera, proveeix una sèrie d'eines perquè publicar contingut sigui tan fàcil com omplir un formulari, i existeixi, a més, una sola font. Respecte a la segona, facilita la publicació de continguts en múltiples formats a partir d'una sola font, i hi afegeix metadades per a facilitar la navegació.

Un CMS proveeix les eines necessàries per a gestionar el cicle de vida dels continguts: creació, gestió, presentació i manteniment i actualització.

Els avantatges dels CMS són els següents:

- **Manteniment descentralitzat.** L'actualització del contingut es pot dur a terme des de qualsevol navegador. Això permet fer les tasques des de qualsevol estació de treball, en qualsevol moment i per diversos editors simultàniament.
- **Autors de contingut amb pocs coneixements tècnics.** Els autors només necessiten un coneixement estàndard d'aplicacions de processador de text

per a crear contingut. No és necessari cap coneixement d'HTML o eines addicionals.

- **Restriccions d'accés configurables.** S'assignen rols i permisos als diferents usuaris. D'aquesta manera es protegeix el contingut de ser modificat només pels elements autoritzats.
- **Consistència del disseny.** En separar el contingut del disseny, aquest darrer s'aconsegueix preservar molt millor. El contingut de tots els autors es presenta en un estil predefinit, conservant la unitat, identitat i consistència de la informació.
- **Vincles, menús i navegació generats automàticament.** Els menús són generats automàticament basat en el contingut de la base de dades. Els vincles també i s'eviten els vincles erronis o que no enllacen a cap lloc.
- **Contingut emmagatzemat en una base de dades relacional.** L'emmagatzematge central i independent de la presentació es tradueix en la possibilitat d'usar el mateix contingut de diferents maneres (diferents webs per exemple) i en diferents mitjans (WAP, PDA...).
- **Contingut dinàmic.** Les extensions com per exemple fòrums, enquestes, aplicacions de compra, cerca o notícies, poden afegir-se de manera modular. Fins i tot el CMS permet la creació d'extensions a mida que es poden afegir de manera modular.
- **Actualitzacions diàries.** No és necessari personal per a les modificacions rutinàries.
- **Coordinació de tasques.** Simplifica les actualitzacions, registra els canvis i activitat feta per cada autor (*log*) i proporciona eines de col·laboració i treball en grup entre autors.
- **Programació del contingut segons calendari.** El contingut pot tenir una vigència per a ser publicat o requerir contrasenya. Mentrestant pot estar amagat per ser vist només pels editors.

### 3.8.2. Wikis

Les eines per a la creació i manteniment de *wikis* són un tipus de programari col·laboratiu. Un *wiki*<sup>24</sup> és un web que permet llegir, editar i crear pàgines que al final són visualitzades en format HTML. Els *wikis* són usats com a suport

<sup>(24)</sup>El terme *wiki wiki* és d'origen hawaïà i significa *ràpid*.

per a la col·laboració, per a documentació, recopilació d'informacions, etc. Les pàgines s'emmagatzemen en una base de dades i s'inclouen enllaços generats dinàmicament. Per l'usuari del *wiki* representa la creació de pàgines web de forma ràpida i eficaç sense haver de tenir gaires coneixements d'informàtica ni programació, poden incloure de forma molt intuïtiva text, enllaços, documents digitals i altres elements multimèdia.

La tecnologia *wiki* permet que les pàgines *wiki* siguin escrites de forma col·laborativa a través d'un navegador web, conservant un historial de canvis que permet recuperar fàcilment qualsevol estat anterior de la pàgina. És a dir, que estan dissenyats amb la filosofia que sigui fàcil de corregir els errors, en comptes que sigui difícil de cometre'ls.

En els *wikis* es pot consultar (des del registre de canvis) els "Canvis Recents", on trobem una llista amb els canvis fets durant un període de temps. L'"Historial de revisió" mostra versions anteriors d'una pàgina, i es poden veure els canvis entre dues revisions. A través de l'Historial un editor pot veure i decidir restaurar una versió anterior de l'article.

Els avantatges dels *wikis* són els següents:

- Llibertat per a l'usuari. Provoca molta més participació en el seu ús, a diferència dels sistemes tradicionals.
- Els continguts s'actualitzen ràpidament (a temps real).
- El document final acostuma a ser producte de la comunitat d'usuaris.
- Hi ha un registre i control de canvis.
- Les pàgines permeten un mecanisme de comentaris. De manera que es pot arribar a acords sense presència en temps i espai.
- Les eines d'edició i actualització són senzilles i no requereixen coneixements molt tècnics.

### 3.8.3. RSS

Els webs que ens interessin però que canvien molt sovint (per exemple llocs de notícies o weblogs), els hem de visitar regularment. Pot arribar a ser un problema. De la mateixa manera, si la nostra web genera continguts de forma continuada forçarà els usuaris a visitar molt sovint algunes pàgines.

#### Viquipèdia

La Viquipèdia (2001) és una *wiki*. Conté més d'un milió d'articles en anglès. És una enciclopèdia que permet als usuaris consultar i modificar els continguts.

El primer programari per a *wikis* va ser creat per Ward Cunningham cap a 1995. En l'actualitat, existeix una gran varietat d'eines per a *wikis*, moltes d'ells en llicència GPL.



Icona per a indicar que una web té contingut RSS

La solució resideix a crear unes pàgines que contenen la informació d'una pàgina web amb les últimes actualitzacions que s'hi han fet. Aquest tipus d'arxius són els RSS<sup>25</sup> i han esdevingut molt útils per a qualsevol web.

A l'usuari (consumidor) de continguts RSS només li cal recollir informació de diverses fonts utilitzant uns programes, anomenats *agregadors* o lectors de RSS. L'*agregador*, amb una interfície única, permet l'accés immediat a les diverses fonts d'informació, i evita de visitar els diversos llocs web. Els *agregadors* comproven, sota demanda o periòdicament, si hi ha hagut canvis en el contingut de les fonts RSS a què estan subscrits i actuen en conseqüència. Mostrant un avís visual o sonor a l'usuari (per exemple).

El fitxer RSS és una especificació de format de fitxer, basada en XML 1.0, utilitzada a Internet per a difusió de notícies i titulars. És molt habitual en llocs web de premsa, blogs i de continguts tècnics. Igual com l'XML, és independent de la plataforma i s'utilitza en les arquitectures i sistemes operatius més populars.

Aquesta forma de distribució d'informació té diversos avantatges sobre la subscripció clàssica a butlletins de notícies mitjançant correu electrònic:

- L'usuari no ha de facilitar una adreça electrònica, i no se li omplirà cap bústia de correu en cas que no el consulti.
- La disponibilitat de continguts és immediata. Està disponible en RSS, en format web i no cal esperar a missatges en cap llista de correu.
- No hi ha problemes de disponibilitat en servidors de correu o filtres antispam que identifiquin erròniament missatges de llistes de correu.

Pot ser molt útil per a una empresa tenir un gestor d'RSS a la seva pàgina web on va inserint notícies i actes relacionats amb els productes que ofereix, ja que té projecció externa més enllà de la web i dels seus productes.

### 3.9. Aplicacions d'exemple

Vegem algunes aplicacions d'ús d'un servidor web.

#### 1) Accedir a informació de l'organització per mitjà del web

L'organització pot aprofitar el servidor web per a proporcionar una porta d'accés a tota la informació que els seus integrants necessitin. Amb una política d'usuaris i grups, es pot muntar un servidor de documentació que permeti d'accedir a tota la informació de l'organització d'una manera ordenada

<sup>(25)</sup>RSS és la sigla de *rich site summary* (RSS 0.91), *RDF site summary* (RSS 0.9 i 1.0) o *really simple syndication* (RSS 2.0)

#### Adreça recomanada

L'especificació completa de RSS 2.0 es pot trobar a: <http://cyber.law.harvard.edu/rss>.

#### Blogs

No entrarem en detalls, però els blocs s'han tornat tan habituals com el correu electrònic. En essència un bloc (de l'anglès *web log*) o bitàcola és un web que recopila cronològicament articles d'un o de diversos autors, com un diari personal. Habitualment els lectors poden escriure comentaris i l'autor, respondre-hi. Hi ha blogs de caire general i temàtics.

i jerarquitzada, si és necessari. Per exemple, es podria utilitzar el servidor web de manera que quan algú necessiti omplir un formulari de petició de vacances només hagi d'accedir al web i imprimir-lo.

Es pot anar més enllà i fins i tot muntar un sistema de compartició d'informació i actes de l'organització. Si s'agrupa per projectes, o per àrees, es pot tenir tota la informació centralitzada i fer-la accessible per a tothom.

## 2) Creació d'aplicacions

Un dels problemes amb els quals es troben les organitzacions és que cada vegada que s'ha de desenvolupar una nova aplicació, cal formar els usuaris per què hi sàpiguen interactuar. La solució a aquest problema seria simplement utilitzar sempre la mateixa interfície entre l'usuari i l'aplicació, de manera que l'usuari ja estigués acostumat a moure's dins de l'aplicació. D'una manera natural, va sorgir la idea d'aprofitar el navegador, que és independent de plataforma, com a eina entre l'usuari i l'aplicació, i per tant, basar-se sempre en un servidor que acaba servint pàgines HTML. Així, doncs, com que els usuaris ja saben com triar entre diverses opcions, saben com afegir dades, modificar-ne, consultar-ne, etc. Desenvolupar una aplicació amb format web és una opció molt atractiva amb vista a l'eficàcia d'interacció amb l'usuari final.

## 3) Correu web (*webmail*)

El correu web permet a un usuari de consultar el correu electrònic sense haver d'estar sempre en la mateixa ubicació. A la majoria d'usuaris això els representa un increment important de la comoditat, ja que fer servir programes específics de correu implica, d'una banda, tenir-los instal·lats a l'ordinador i, per tant, dependre'n per a mirar el correu, i de l'altra, tot un procés d'aprenentatge per a poder utilitzar les funcionalitats que proporciona.

Es fa evident pensar en el fet que un mitjà d'accés comú a totes les aplicacions facilita molt les coses i, per tant, cada cop sorgeixen més aplicacions que es basen en un client (navegador) i un servidor (servidor web).

### Vegeu també

Vegeu el mòdul "El sistema informàtic dins l'organització", que parla de la creació d'aplicacions des del punt de vista del responsable d'informàtica.

## 4. Seguretat

La seguretat és un aspecte molt complex, i més quan es tracta d'una aplicació orientada específicament cap a l'exterior. Aquí, la seguretat es pot veure des de molts vessants.

### 4.1. Seguretat de publicació

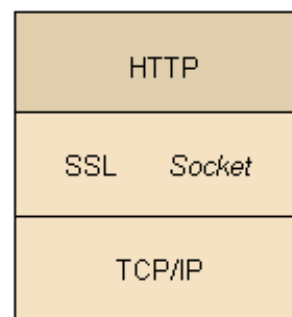
Com que el web publica informació de l'organització, s'ha de vigilar de no publicar per error informació que sigui confidencial. Això pot passar, per exemple, quan el servidor accedeix a una base de dades: per un problema de permisos relaxats, potser es pot accedir a més informació de la necessària.

### 4.2. Seguretat del programari

Tots els servidors web són aplicacions que escolten peticions per uns ports de terminats i donen la informació agafant-la dels servidors i proporcionant-la als clients. Això els fa interessants i potencialment vulnerables per a ser atacats per pirates informàtics, que constantment en busquen les debilitats per a fer-les servir com a porta per a poder accedir als sistemes informàtics.

### 4.3. Protocols de comunicació encriptats (HTTPS)

Per a comunicar-se amb el servidor web que conté informació sensible és recomanable fer servir protocols que utilitzin encriptació. El més utilitzat és l'HTTPS (HTTP segur), que fa servir encriptació en la comunicació, a diferència del protocol HTTP, en què la informació circula visible, per la qual cosa amb un *sniffer* (detector) es pot veure tota la sessió. Per tant, en cas d'una aplicació de correu electrònic, transaccions electròniques, informació personal o informació privada de l'organització, és molt recomanable instal·lar el mòdul d'HTTPS.



Esquema de l'estàndard HTTPS

HTTPS és una versió encriptada d'HTTP, implementada per mitjà de l'estàndard SSL/TLS<sup>26</sup>. La connexió encriptada es duu a terme simplement amb una connexió HTTP ordinària per sobre d'una connexió SSL encriptada. Per a la resta, HTTPS i HTTP són iguals. El port estàndard de connexió HTTPS és el 443.

<sup>(26)</sup>SSL és la sigla de *secure socket layer* i TLS, de *transport layer security*.

#### 4.4. Registres de les connexions

Els servidors fan un log de les connexions, però, a més, amb poca feina afegida, és possible dur un registre de quines màquines són les que s'han connectat, o fins i tot verificar contra un servidor de noms de domini (DNS<sup>27</sup>) el nom de la màquina. Si no és a cap DNS, una opció és no permetre l'entrada, i si és al DNS, guarda l'adreça IP, el nom IP i l'adreça MAC de la màquina.

<sup>(27)</sup>DNS és la sigla de *domain name server*.

#### 4.5. Còpies de seguretat

El servidor web té pocs fitxers de configuració però, atesa la seva importància dins l'organització, es fa indispensable disposar d'un sistema de còpies de seguretat que proporcioni tot el que és necessari en cas d'una fallada. El primer que s'hauria de tenir guardat és tot el procés per a instal·lar-lo i com està organitzat internament. A part, és important prendre nota de tots els canvis que es van fer i totes les polítiques de registres que s'han establert. S'acostuma a tenir tota aquesta documentació per escrit i a part, i a fer a diàriament o setmanalment una còpia de seguretat íntegra de tots els components que formen el servidor web.

#### 4.6. Atacs al servidor

Un servidor està exposat a una gran varietat d'atacs, per la qual cosa ha de ser un element important dins de l'esquema de seguretat del sistema informàtic. Un atac (amb èxit) a un servidor pot tenir conseqüències que cal tenir presents:

- Degradació de la imatge de l'organització.
- Funcionament incorrecte o destrucció de part o de tot el web.
- Robatori d'informació.
- Modificació de dades, i en especial de dades personals i/o sensibles dels usuaris i de l'organització.
- Intrusió en el servidor web.

En haver molts atacs diferents, es poden classificar de la següent manera:

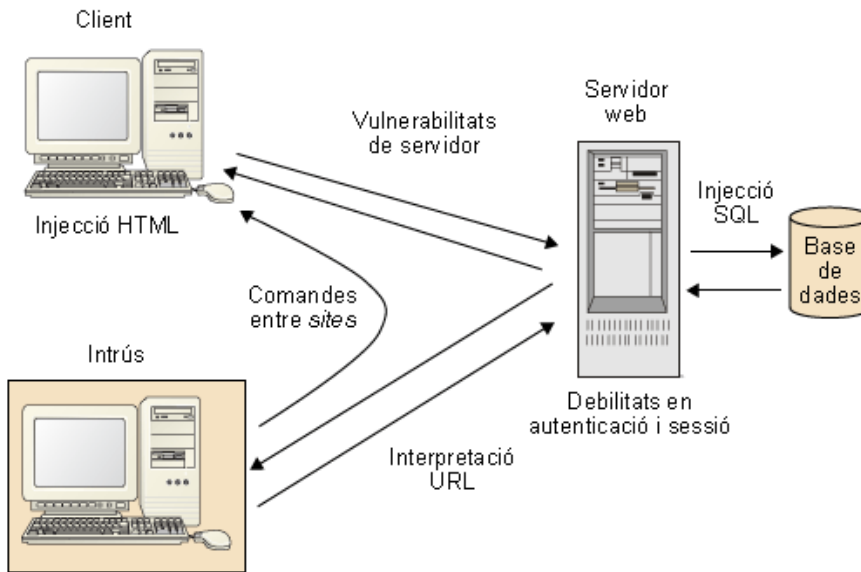
- Vulnerabilitats del servidor web. Ja no és habitual. Els servidors web cada vegada són més segurs.
- Manipulació d'URL. Modificació manual de paràmetres d'URL per a modificar el comportament esperat del servidor web.
- Debilitats dels identificadors de sessió i sistemes d'autenticació.
- Injecció de codi HTML i seqüències de comandes entre *sites*.



- Injecció de comandes SQL<sup>28</sup>.

<sup>(28)</sup>SQL és la sigla de *structured query language*; en català, llenguatge de consultes estructurat.

Resum d'atacs a un servidor



#### 4.6.1. Falsificació de dades HTTP

La falsificació de dades HTTP és un atac que consisteix a enviar al servidor web dades introduïdes manualment per a crear un context anormal.

S'envien dades al servidor a través dels formularis, de les URL i de les capçaleres HTTP. De manera que es poden fer atacs diferents segons com s'enfoquin les dades que es modifiquin. Així doncs, com que tota aquesta informació pot ser manipulada fàcilment per un usuari, no es pot considerar fiable des de la perspectiva de la seguretat. La seguretat no pot estar basada en verificacions del client. Cal tenir present que establir una connexió SSL<sup>29</sup> no protegeix contra la manipulació de dades enviades cap al servidor, sinó que certifica la confidencialitat de la informació entre el web i l'usuari final.

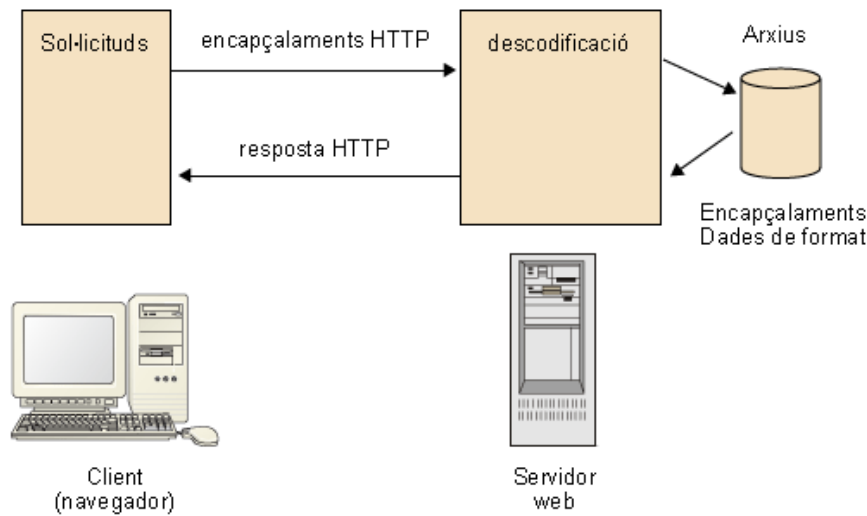
<sup>(29)</sup>SSL és la sigla de *secure socket layer*.

#### Adreça recomanada

L'especificació del protocol http 1.1 es pot consultar a: <http://www.ietf.org/rfc/rfc2616.txt>.

Dins d'aquest tipus d'atacs podem trobar la manipulació d'URL, injecció SQL, modificació de capçaleres HTTP o la modificació de formularis. En veurem alguns per a entendre com funcionen.

## Esquema de falsificació de dades HTTP

**4.6.2. Manipulació URL**

Una URL té aquesta estructura:

Estructura d'una URL

<b>http://</b>	<b>Usuari:password@</b>	<b>www.uoc.edu</b>	<b>:80</b>	<b>/portal/castellano/inde5.html</b>
Protocol	Contrasenya (opcional)	Nom del servidor		Ruta
		Port (opcional si és 80)		

La URL permet enviar paràmetres al servidor afegint el signe d'interrogació després de la ruta. Per exemple, *http://www.uoc.edu/forum/?cat=5&page=1*.

La manipulació d'algunes parts d'una URL pot produir que un servidor web permeti l'accés a pàgines que suposadament no hi tenia accés. També pot provocar el processament d'un cas no previst (paràmetre inexistent o un valor no previst) amb un comportament anòmal de la pàgina web tot oferint informació.

**4.6.3. Injecció de codi**

La injecció de codi és coneguda habitualment com a *cross site scripting* (XSS). La tècnica es basa a injectar i fer córrer *scripts* (habitualment *javascript*) que no estan en el servidor atacat. Es pot fer de diverses maneres. Una d'aquestes es basa a enviar el codi des d'un formulari, aprofitant que el formulari no transformi els caràcters en seqüències HTML.

#### 4.6.4. Injecció SQL

Els atacs d'injecció SQL<sup>30</sup> es fan a webs que tenen bases de dades relacionals. En aquest tipus d'atac, les pàgines web reben paràmetres que es passen a la base de dades com una consulta SQL<sup>30</sup>. Si no es verifiquen els paràmetres, es pot modificar la consulta per a accedir a tot el contingut o modificar-lo.

<sup>(30)</sup>Recordeu que SQL són les sigles de *structured query language*, en català, 'llenguatge de consulta estructurat'.

#### 4.7. Mètodes per a accedir a bases de dades de l'organització

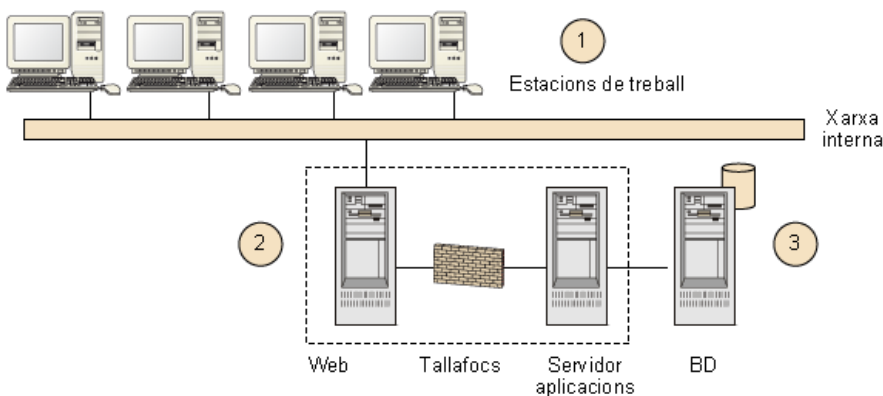
El servidor web és la part pública de la xarxa de l'organització. Per tant, la seguretat en aquesta part és especialment delicada. Vegem possibles maneres d'assegurar-la.

##### 4.7.1. Arquitectura a tres nivells

La idea que hi ha darrere de l'arquitectura a tres nivells és la de tractar diferent la part del web de la part de les dades. Els nivells són els següents:

- 1) Són els clients que accedeixen al web per a consultar i/o modificar informació.
- 2) Pot ser un sol servidor físic o dos, com és el cas representat en l'esquema. És recomanable que siguin dos si la informació és especialment sensible. El servidor web fa de visualitzador.

Esquema de l'arquitectura a tres nivells



El tallafoc està configurat perquè només el servidor web pugui arribar fins al servidor d'aplicacions. D'aquesta manera, només una màquina pot fer servir les aplicacions que tenen accés a les bases de dades. També tenim separades la part de la visualització de la part d'aplicació.

Una altra de les raons de separar la part d'aplicació de la part de web és per a no deixar en cap moment la màquina inconsistent per problemes del visualitzador.

La comunicació entre el servidor web i el servidor d'aplicacions es pot fer utilitzant, per exemple, *servlets*, JSP o IAS.

3) El servidor de bases de dades té les dades. És un servidor diferent, de manera que les dades estan segures i són independents de l'aplicació. Això permet que fins i tot els usuaris de la base de dades les puguin mantenir independentment.

La comunicació entre el servidor d'aplicacions i la base de dades es pot fer, per exemple, utilitzant JDBC<sup>31</sup>.

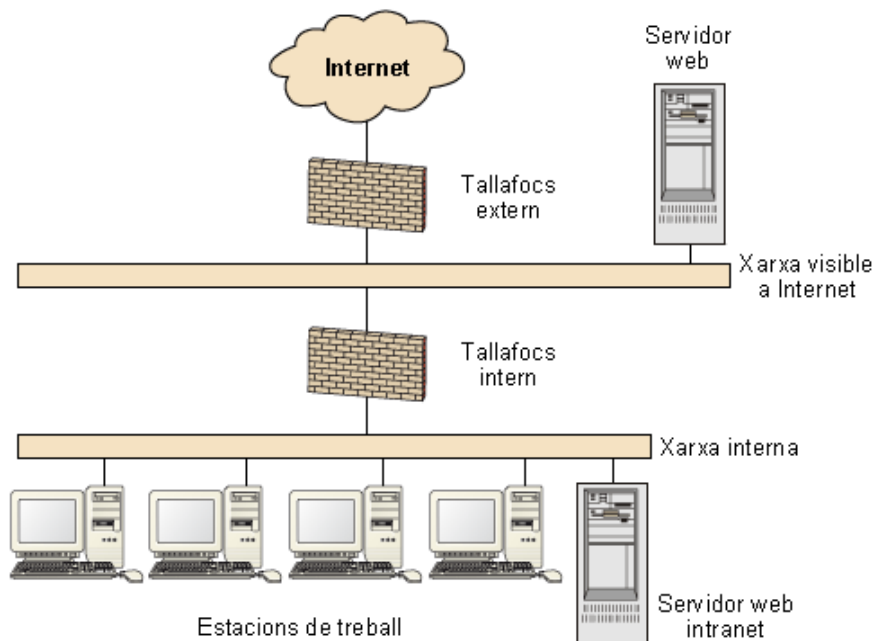
<sup>(31)</sup> Recordeu que JDBC és la sigla de *java database connectivity*.

### 4.7.2. Restricció d'accessos

Una altra arquitectura per a controlar l'accés és la que es mostra al següent esquema.

**Vegeu també**  
 Consulteu el mòdul "Administració de la xarxa" les arquitectures de seguretat i veureu com estem implementant una zona desmilitaritzada (DMZ) per a la nostra xarxa.

Esquema de l'arquitectura de restricció d'accessos



Aquest esquema evita, tant com sigui possible, els atacs per Internet, ja que els tallafocs han de tenir seguretat a nivell de màquina per a assegurar que no hi entra ningú no autoritzat. El servidor web també ha d'estar configurat perquè màquines (a nivell d'adreça) que no siguin de l'organització no puguin accedir a la informació.

La seguretat dels servidors web es pot veure des de moltes òptiques, ja que un servidor es dedica a publicar informació de l'organització possiblement a Internet.

## 5. Aspectes legals

Els aspectes legals dels servidors web, i en aquest cas del seu administrador, se centren en un punt molt discutit: la propietat de la informació que es fa pública.

Una vegada més, i a costa de fer-nos reiteratius, la legislació és molt canviant en aquests punts, per la qual cosa s'ha de consultar un assessor legal en cas que se sospiti que hi pot haver problemes sobre aquesta matèria.

Què contestaríeu davant les preguntes següents?

- “Podem agafar qualsevol cosa d'Internet i posar-la a la web?”.
- (Si abans heu contestat que no) “I si diem a la nostra pàgina d'on l'hem tret?”
- “Si volem posar alguna cosa a la nostra web però no volem que ens ho copiïn, ho hem de registrar per a evitar-ho?”
- “Si tot el que hi ha a Internet és de tots, podem agafar el que vulguem. Què discutim?”

### 5.1. Propietat intel·lectual i societat de la informació

Sota un punt de vista legal, es fan servir els termes següents: la **informació** és un concepte que inclou dades i fets, però també imatges, sons, textos, etc. Per exemple, dades meteorològiques, una pel·lícula, una cançó, un logotip, un quadre o una obra literària. La informació és un **bé valuós** i és un **bé immaterial** que està en un **suport físic (material)** del qual es pot ser **propietari**.

Es comet un error greu en pensar que la propietat d'un suport dóna drets sobre la informació que s'hi fixa. Per exemple, comprar un llibre dóna dret a llegir-lo, però no a fer-ne còpies i a vendre-les.

### Exemple sobre propietat intel·lectual

S'ha dissenyat el web de l'organització i, a més, s'ha decidit que els usuaris tindran pàgines personals en què podran posar el que vulguin. Aquesta acció pretén motivar el personal a fer servir les noves tecnologies. Al cap de poc temps, com que no feu el web vosaltres, com a administradors del sistema, us adoneu que té text, imatges, fotografies, animacions, sons i fins i tot música agafada d'Internet. Què en penseu?

a) Això s'ha d'arreglar de seguida. Aquestes coses no hi poden ser, ja que és il·legal que hi apareguin elements que no ha fet l'autor del web.

b) Tot això ja era a Internet, i Internet és lliure. Per tant, es pot agafar tot el que es vulgui i tornar-ho a penjar a Internet. Quin problema hi ha? Si baixem fotografies d'Internet, les podem posar a la nostra pàgina web, oi? Si són a Internet, bé que hi ha algun tipus de "llicència implícita". També podem fer enllaços sense demanar permís (com que podem accedir a la pàgina, també podem fer-hi anar). I la música? Per un trosset de cançó no ha de passar res. I les estadístiques de la revista del sector també són al web de l'organització (total, si apareixien en una revista, qualsevol les pot tenir, comprant-la o fent-ne fotocòpies).

La resposta, com passa sempre en aquestes situacions, no és simple, però sí clara: la informació té propietari.

### Un cas extrem

Posem un exemple d'un cas extrem. Podem ser propietaris d'un suport i no tenir ni tan sols dret a fer servir la informació que conté. Si comprem un programari copiat il·lícitament, ningú no posarà en dubte la propietat del disquet (el suport), però si instal·lem el programa, cometem un acte il·lícit.

Qualsevol creació, producte de la creativitat humana, s'intenta protegir mitjançant la propietat intel·lectual, i el cas del ciberespai no n'és cap excepció.

### Framing o linking

El *framing* o *linking* consisteix a mostrar continguts d'una pàgina diferent com a propis (segons la percepció de l'usuari). El *framing* es considera una vulneració de la propietat intel·lectual.

## 6. Tasques/responsabilitats

L'administrador del web té un conjunt de funcions molt ampli, especialment si també ha de tenir cura dels continguts:

- Fer que el web sempre estigui actiu.
- Procurar que el temps de resposta estigui dins els marges acceptats.
- Fer que el web sigui homogeni.
- Recollir totes les consultes, queixes i propostes que es facin sobre el web.
- Controlar la seguretat del servidor.
- Generar les estadístiques.
- Preparar la formació de les persones que han de mantenir els continguts.
- Responsabilitzar-se (encara que parcialment) dels continguts.

## Resum

L'administrador del web és una figura que ha de formar un binomi amb el programari del servidor web, que es compon del nucli, els mòduls i les pàgines.

L'administrador té cura del servidor i de la seguretat que s'hi associa. També té cura de la imatge que transmet, ja que és la imatge de l'organització.

Com a responsable del web, ha de saber comunicar els recursos de què disposa el web perquè els creadors de contingut, generalment els usuaris finals, puguin dissenyar les pàgines aprofitant tot el potencial del servidor.

Històricament, primer van aparèixer les pàgines estàtiques, posteriorment, les pàgines estàtiques amb codi inserit, en un intent de fer-les més interactives, i avui dia hi ha les pàgines amb codi que s'interpreta en el servidor, que per mitjà d'algun mecanisme busca informació i modifica la pàgina abans de donar-la al navegador.

Actualment hi ha moltes aplicacions associades a servidors web, i per tant el seu potencial d'ús és molt gran. Això fa que sigui necessari tenir una cura especial als aspectes de seguretat.



## Activitats

1. Connecteu-vos a un web i mireu el codi font de la pàgina per a veure l'HTML amb què està feta. Veieu les marques `<...>` ?

## Exercicis d'autoavaluació

1. Quines d'aquestes frases són certes i quines són falses?

Posaríeu el servidor web en el mateix servidor on hi ha el servidor de bases de dades de l'organització?

- a) Sí, perquè així és molt més senzill accedir a les dades per a presentar-les via web.
- b) No, hi ha un risc de seguretat.
- c) Sí, perquè si no es podria carregar molt la xarxa.
- d) No, perquè si es penja el servidor web hi ha perill que es pengi el servidor de bases de dades.
- e) Si és per a una intranet sí que l'hi posaria.
- f) En cas que hi hagi protecció de tallafoc sí, perquè no hi podrà entrar cap ordinador no autoritzat.

2. Quines d'aquestes accions considereu encertades i quines no en la situació següent:

Heu vist que en una pàgina de l'organització hi ha una imatge que està agafada d'Internet. Què feu?

- a) Parlar amb la persona que l'ha posat per a veure d'on l'ha tret i demanar permís per a posar-la. Si us diuen que sí, la poseu; si us diuen que no, la traieu.
- b) La deixeu: Internet és de tots i per una imatge no cal perdre gaires hores.
- c) Treure-la immediatament. Qualsevol informació que apareix a Internet té propietari.
- d) Si us diu que l'ha agafat d'un lloc que es pot fer servir lliurement no cal fer res.
- e) Alliçoneu l'usuari sobre els perills legals de posar informació agafada d'Internet en un web.

3. Un usuari us comenta el següent: "He entrat en un web i m'ha carregat una pàgina `www/.../tauler.php`. És molt estrany, perquè sé que totes les pàgines acaben amb "htm". M'han provat d'instal·lar un virus a l'ordinador?". Què li explicariéu?

- a) És un llenguatge que es barreja amb l'HTML i que interpreta un mòdul del servidor abans d'enviar la pàgina. PHP vol dir *hypertext preprocessor* (preprocessador d'hipertext).
- b) No, és un llenguatge diferent de l'HTML, per això té una altra extensió.
- c) No, és una qüestió de seguretat. PHP vol dir *PHirewall protection*.
- d) És un llenguatge que, igual que el Java, executa codi en la vostra màquina. Vigileu amb aquesta web, perquè us pot fer coses estranyes a l'ordinador.

4. Una d'aquestes tasques no és responsabilitat de l'administrador de web.

- a) Recollir totes les consultes, queixes, propostes que es facin del web.
- b) Que el web estigui sempre actiu.
- c) Assegurar la disponibilitat de les dades per a tots els usuaris.
- d) Controlar la seguretat del servidor.
- e) Generar les estadístiques.
- f) Preparar la formació de les persones que han de mantenir els continguts.

5. Relacioneu les tecnologies amb la base de funcionament.

JavaScript
CGI
Perl
PHP
ASP
JSP
Java
Servlets

Executa en el client
Executa en servidor
Codi barrejat HTML Interpreta client
Codi barrejat HTML Interpreta servidor

## Solucionari

### Exercicis d'autoavaluació

1. Argumentem una mica cada resposta.

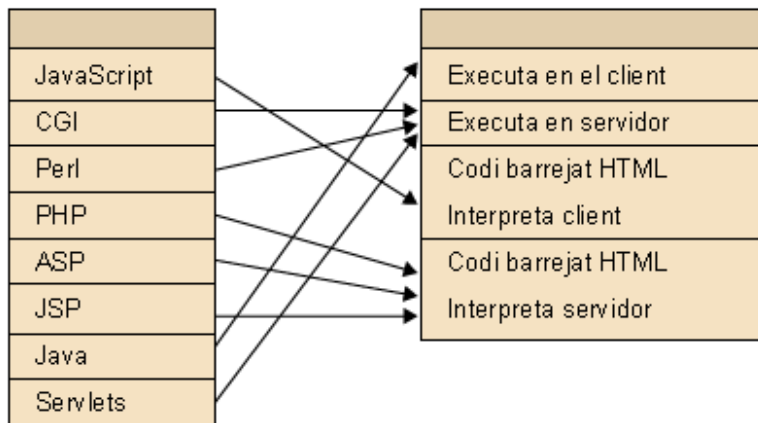
- a) Fals. D'entrada, per a accedir a una base de dades des d'un servidor web, no té importància on és la base de dades. És igual de complex si és en el mateix servidor o en un altre servidor físic.
- b) Certa. Sempre és un risc de seguretat. Qualsevol intrusió que deixi obert el servidor web permetrà d'accedir fàcilment a les dades, atès que seran en el mateix ordinador.
- c) Fals. Per a presentar pàgines viatja poca informació per la xarxa. Si es preveu molt de trànsit, hi ha dissenys de xarxa que l'aïllen entre el servidor web i el servidor de bases de dades. També es poden fer servir connectors en lloc de concentradors per no carregar la xarxa amb aquest trànsit.
- d) Cert, malgrat que depèn dels servidors i els operatius. Per exemple, els Unix, si el servidor dona problemes, és molt estrany que puguin arribar a penjar tot el sistema. Tanmateix, és una possibilitat que no es pot descartar. Segons la funció que faci el servidor de base de dades, és un risc que no es pot córrer. Lògicament, per a b) no és aconsellable.
- e) Cert, sempre que a més a més no contingui dades sensibles. També s'hauria de tenir en compte d), el perill de penjar l'ordinador.
- f) Fals. A més, no té gaire sentit un servidor web obert a tot Internet amb protecció d'accés a només uns ordinadors determinats.

2. Excepte b, totes les altres són coherents.

3. a

4. c "Assegurar la disponibilitat de les dades per a tots els usuaris no pertany a l'administrador web sinó a l'administrador de les dades."

5.



## Glossari

**CMS** Sigla de *content management system*. Aplicació per a crear i administrar continguts.

**CSS** Sigla de *cascading style sheets*, estàndard proposat el 1996 pel W3C.

**cookie** Vegeu **galeta**.

**detector** *m* Programa destinat a mirar d'una manera il·lícita la informació que passa per una xarxa informàtica.  
*en* sniffer.

**DHTML** És la sigla de *dynamic HTML*, 'HTML dinàmic'.

**enllaç** *m* Mecanisme específic per a accedir a un recurs d'Internet. L'utilitzen especialment les pàgines web, en un format definit `<A> <href="URL">text</A>`.  
*en* link.

**extranet** *f* Nom per a designar Internet. Neix per oposició a *intranet*.

**Framing** Vegeu **linking**.

**galeta** Les galletes (*cookies*) són petits fitxers que alguns webs guarden a l'ordinador de l'usuari.

**HTML** *m* Vegeu **llenguatge d'etiquetatge d'hipertext**.

**hypertext markup language** *m* Vegeu **llenguatge d'etiquetatge d'hipertext**.

**intranet** *f* Xarxa interna d'una organització que pot funcionar amb TCP/IP, i a més pot fer servir els recursos d'Internet, com poden ser navegadors, servidors web, etc.

**link** *m* Vegeu **enllaç**.

**linking** *m* Consisteix a mostrar continguts d'una pàgina diferent com a propis. Es considera una vulneració de la propietat intel·lectual.

**llenguatge d'etiquetatge d'hipertext** *m* Llenguatge estàndard per a etiquetar documents en format d'hipertext a fi d'indicar a un navegador com ha de visualitzar un document en la pantalla de l'ordinador.  
*en* hypertext markup language.  
sigla: **HTML**.

**localitzador uniforme de recursos** *m* Manera única d'especificar un recurs a Internet.  
*en* uniform resource locator.  
sigla: **URL**.

**Model en objectes per a la representació de documents** Model orientat a objectes que els programes i scripts poden usar per a accedir i modificar dinàmicament el contingut, estructura i estil dels documents HTML.  
sigla: **DOM**.

**RSS** *m* Format de dades usat per a difondre continguts a subscriptors d'un web.  
sigla: **rich site summary**.

**servidor web** *m* Aplicació que funciona en un ordinador que escolta per un port i serveix pàgines web sota demanda.

**sniffer** *m* Vegeu **detector**.

**uniform resource locator** *m* Vegeu **localitzador uniforme de recursos**.

**URL** *m* Vegeu **localitzador uniforme de recursos**.

**webmaster** *m i f* Responsable del web. Sota aquest nom hi ha la persona que té cura dels continguts (no ha de ser necessàriament l'administrador del web).

**webmail** *m* Correu basat en web.

**W3C** World Wide Web Consortium. El consorci que marca l'estàndard d'HTML, entre moltes altres coses. <http://www.w3c.org>.

**Wiki** Web on col·laboren diversos autors.

**XML** *m eXtensible markup language* o 'llenguatge d'etiquetatge extensible'. És un estàndard ajustat a les necessitats d'Internet.

## Bibliografia

**Coar, K.** (1998). *Apache Server for Dummies*. Estats Units: IDG Books Worldwide, Inc.

**Dwivedi** (2007). *Hacking Exposed Web 2.0: Web 2.0 Security Secrets and Solutions*. Estat Units: McGraw-Hill

**Garfinkel, S.; Spafford, G.** (1997). *Web Security & Commerce*. Estats Units: O'Reilly & Associates, Inc.

**Morales, O.** (2001). *Informàtica Jurídica*. Barcelona: Universitat Oberta de Catalunya.

**Musciano, C.; Kennedy, B.** (2000). *HTML & XHTML The Definitive Guide*. Estats Units: O'Reilly & Associates, Inc.

**Pfleeger, C.** (1997). *Security in Computing*. Estats Units: Prentice Hall.

**Pressman** (2008). *Web Engineering: A Practitioner's Approach*. Estat Units: McGraw-Hill

**Spainhour, S.; Quercia, V.** (1996). *WebMaster in a Nutshell A Desktop Quick Reference*. Estats Units: O'Reilly & Associates, Inc.