

1. Implantació d'arquitectures web

L'arquitectura d'aplicacions en entorns web difereix força de la d'aplicacions d'escriptori, en la qual un programa s'executa en alguna de les modalitats vistes (interpretat, executat directament sobre una plataforma, o bé executat amb una màquina virtual) directament sobre la màquina en la qual treballa l'usuari.

El model d'arquitectura bàsic que hi ha en tota aplicació web és el model anomenat client-servidor, en el qual entren en joc diverses màquines o plataformes, cadascuna de les quals desenvolupa un rol diferenciat en l'execució de l'aplicació. Segons les necessitats i la complexitat de l'aplicació, aquest model bàsic d'arquitectura es pot complicar més o menys per tal d'aconseguir una millor distribució de tasques, millor rendiment, fiabilitat, augment de la capacitat de procés, etc.

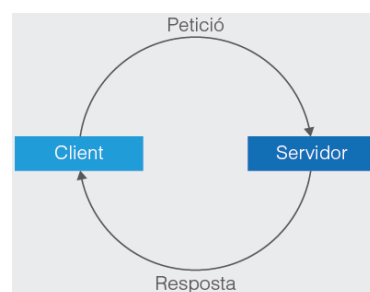
1.1 Arquitectures web. Models

Una aplicació distribuïda està formada per una col·lecció d'ordinadors autònoms enllaçats per una xarxa d'ordinadors i suportats per un programari que fa que la col·lecció actuï com un servei integrat.

1.1.1 El model client-servidor

El model client-servidor és un model d'arquitectura d'aplicacions en el qual es defineixen o s'assignen principalment dos rols als ordinadors, que són, com el nom del model indica, els rols de client i de servidor (vegeu la figura 1.1).

FIGURA 1.1. Estructura client-servidor



Client-servidor

En el model client-servidor hi ha dos tipus de components:

- Clients: fan peticions de servei. Normalment els clients inicien la comunicació amb el servidor.
- Servidors: proveeixen serveis. Normalment els servidors esperen rebre peticions. Un cop han rebut una petició, la resolen i retornen el resultat al client.

Els servidors poden ser amb estat o sense estat. Un servidor sense estat no manté cap informació entre peticions, mentre que un servidor amb estat pot recordar informació entre peticions. Per exemple, un servidor sense estat podria ser aquell que conté pàgines web estàtiques. En canvi, un servidor que tingui una pàgina web amb contingut dinàmic seria un exemple de servidor amb estat.

El model client-servidor bàsic de la figura anterior és vàlid per a aplicacions web petites, senzilles i que no tinguin una gran càrrega de treball, és a dir, un nombre petit de clients connectats simultàniament.

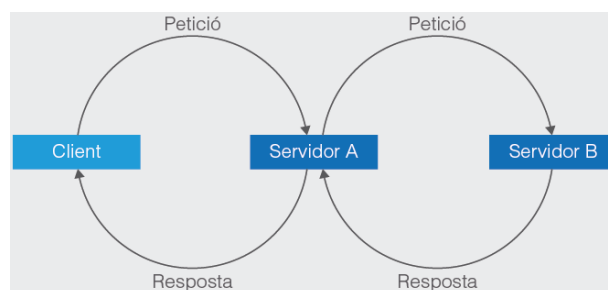
En entorns reals és habitual que no es donin aquestes tres característiques i, per tant, s'hagi d'implementar una arquitectura més complexa basada en el model client-servidor però que pot presentar diferències o ampliacions al model per tal de garantir un bon rendiment de les aplicacions web, la seva fiabilitat i/o la capacitat d'atendre un nombre elevat de peticions dels clients de forma simultània en aplicacions web de mida mitjana o gran i d'un nivell de complexitat mitjà/alt.

Un servidor també pot ser client d'altres servidors. Per exemple, els servidors web i els altres serveis disponibles a internet són clients del servei de resolució de noms (DNS).

1.1.2 El model client-servidor amb servidors encadenats

Quan en una aplicació el servidor ha de realitzar tasques molt complexes o costoses de processar poden distribuir-se subtasques en diversos servidors, de tal manera que un servidor pot actuar com a client d'un altre servidor per tal de delegar-hi determinades responsabilitats (vegeu la figura 1.2).

FIGURA 1.2. Estructura client-servidor encadenada



Per exemple, quan un client d'una entitat bancària accedeix als serveis en línia del seu banc amb un navegador web (client), el client inicia una sol·licitud al servidor web del banc. Les credencials d'inici de sessió del client estan emmagatzemades en una base de dades i el servidor web accedeix al servidor de base de dades com a client. Un servidor d'aplicacions interpreta les dades retornades aplicant la lògica de negoci del banc i proporciona la sortida al servidor web. Finalment, el servidor web retorna el resultat al navegador web del client per a la seva visualització.

1.1.3 Aplicacions basades en el web

Un cas particular d'aplicacions client-servidor són les aplicacions que s'executen aprofitant l'arquitectura del web. Aquestes aplicacions es basen en el fet de tenir tota la capacitat de processament en un servidor web (o conjunt de servidors) al qual s'accedeix des d'un navegador web.

Quan un usuari clica sobre un enllaç d'una pàgina web del seu navegador, aquest genera una petició al servidor que conté la informació. Un cop el servidor rep la petició, retorna el contingut. La comunicació entre client i servidor es fa mitjançant el protocol HTTP.

1.1.4 El model d'igual a igual

Hi ha un tipus d'arquitectura en què tots els ordinadors es comporten al mateix temps com a clients i com a servidors. Aquests tipus de xarxes s'anomenen **d'igual a igual** (*peer-to-peer*).

D'igual a igual

Un sistema d'igual a igual es caracteritza per ser un sistema distribuït en què tots els nodes tenen les mateixes capacitats i responsabilitats, és a dir, tots són clients i servidors al mateix temps i, per tant, tota la comunicació és simètrica.

1.2 Servidors web i d'aplicacions. Instal·lació i configuració bàsica

Durant les fases de desenvolupament, de posada en producció i de manteniment d'una aplicació web podem trobar-nos amb diversos tipus de servidors que duen a terme tasques concretes dins el funcionament global.

1.2.1 Servidors web

Un servidor web és un servidor que permet l'accés a recursos mitjançant el protocol HTTP (*HyperText Transfer Protocol*) d'internet.

La definició original i estricta del concepte de servidor HTTP fa referència a aquells servidors capaços de donar accés i de permetre la gestió d'un conjunt de recursos estàtics com a resposta a peticions rebudes pels clients. És a dir, que permeten consultar, carregar i eliminar recursos del servidor. Aquests recursos solen ser documents d'HTML o variants d'aquest format i continguts adjunts o relacionats amb aquests documents, com poden ser imatges, vídeos, etc.

Aquests recursos solen estar guardats en forma d'arxius a dispositius d'emmagatzematge propis del servidor.

El concepte original de servidor web no contempla la possibilitat de generar de forma dinàmica els continguts a partir de l'execució de codi com a resposta de les peticions. Però, en l'actualitat, la majoria de servidors web admeten la instal·lació de mòduls que permeten que es generin continguts dinàmics a partir de l'execució de programes escrits en diversos llenguatges de programació (PHP, Javascript, Python, Perl, etc.), tot i que aquesta característica és més pròpia dels servidors d'aplicacions.

Alguns exemples de servidors web són Apache HTTP Server, per a sistemes operatius Linux, i Microsoft Internet Information Server, per a Windows.

1.2.2 Servidors d'aplicacions

Un servidor d'aplicacions en general és un servidor que ofereix als clients un servei d'execució d'aplicacions. Si ens centrem en les aplicacions web, un servidor d'aplicacions és un programari que controla l'execució de programes. Els clients, des d'un navegador (usant el protocol HTTP), accedeixen a una interfície web des d'on executaran l'aplicació. Normalment, els servidors d'aplicacions s'utilitzen en aplicacions web amb un grau de complexitat elevat.

Un servidor d'aplicacions web es pot entendre com un servidor orientat a l'execució de programes que pot rebre les peticions de servei i retornar els resultats utilitzant els mateixos protocols (HTTP) i formats de dades que els servidors web (HTML). Si el mateix servidor no té la capacitat d'interactuar amb aquests protocols pot treballar conjuntament amb el suport d'un servidor web que faci d'intermediari entre el servidor d'aplicacions i el client. Els servidors d'aplicacions, a més, acostumen a proporcionar un ampli conjunt de serveis complementaris orientats a la persistència de dades, la seguretat, el control de transaccions i concurrència, entre d'altres.

Alguns exemples de servidors d'aplicacions són GlassFish (servidor Java EE, Oracle) o Microsoft Internet Information Server (servidor .NET).

1.2.3 Servidors de bases de dades

Un servidor de bases de dades s'utilitza per emmagatzemar, recuperar i administrar les dades d'una base de dades. El servidor gestiona les actualitzacions de dades, permet l'accés simultani de molts servidors o usuaris web i garanteix la seguretat i la integritat de les dades.

Entre les seves funcions bàsiques, el programari de servidors de bases de dades ofereix eines per facilitar i accelerar l'administració de bases de dades. Algunes

funcions són l'exportació de dades, la configuració de l'accés dels usuaris i el suport de dades.

Alguns exemples de servidors de bases de dades són Oracle Database, MySQL, Microsoft SQL Server, PostgreSQL, MongoDB o Firebase.

1.2.4 Servidors de fitxers

Un servidor de fitxers és un servidor que permet gestionar a través de xarxa la càrrega, descàrrega, actualització i eliminació de fitxers emmagatzemats en els seus dispositius des d'ordinadors client.

En l'àmbit de les aplicacions web, els servidors de fitxers s'utilitzen principalment per desplegar les aplicacions sobre el servidor on s'executaran. El desplegament d'una aplicació web sobre els servidors de producció comporta habitualment la càrrega de grans quantitats de fitxers sobre aquests servidors. Com que el desenvolupament i manteniment d'aquestes aplicacions es fa en les màquines dels programadors, cal algun sistema de transferència d'arxius cada cop que es vol actualitzar la versió de producció d'una aplicació.

Un dels protocols més usats per a la transferència de fitxers en el desplegament d'aplicacions web és el protocol FTP (*file transfer protocol*), amb les seves variants FTPS i SFTP per adaptar-se a les necessitats actuals de seguretat.

Alguns exemples de servidors de transferència de fitxers són ProFTPD o vsftpd, per a sistemes operatius Linux, i Microsoft Internet Information Server, per a Windows.

1.2.5 Servidors de directori

Un servidor de directori és un servidor que permet gestionar informació administrativa respecte a l'entorn d'una aplicació web, com poden ser, per exemple, els usuaris autoritzats amb els seus rols o permisos, etc.

La utilitat principal dels servidors de directori és facilitar la gestió d'informació relativa a l'explotació d'aplicacions web. L'avantatge de gestionar aquesta informació mitjançant aquest tipus de servidors és la centralització de dades i la facilitat d'accés mitjançant protocols estàndard com LDAP.

Alguns exemples de servidors de directori són OpenLDAP, per a Linux, i Active Directory, per a Windows.

1.3 Estructura i recursos d'una aplicació web

Les aplicacions web, a més de presentar un arquitectura client-servidor (fet que no és necessari en el cas de les aplicacions d'escriptori), solen estar estructurades amb una gran quantitat d'arxius i recursos de tipus diferents.

És per això que cal establir unes directrius per tal d'organitzar la ubicació d'aquests components i la seva interrelació durant la fase de desenvolupament, així com també en el moment de posar l'aplicació en producció. En cas contrari, el desenvolupament i manteniment d'una aplicació de mida mitjana o gran es convertirà en una tasca gairebé impossible de gestionar.

Oblidant-nos de l'organització o estructura que imposa el fet d'escollir unes determinades eines de desenvolupament o un determinat servidor web o d'aplicacions, aquestes aplicacions es poden estructurar segons diversos models d'organització dels seus components i recursos. Alguns dels models d'estructuració d'aplicacions web que podem trobar més habitualment són els que es descriuen a continuació.

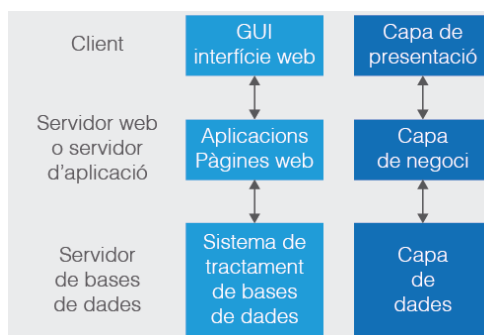
1.3.1 Arquitectura multinivell

L'arquitectura multinivell (*multitier architecture*) és un tipus concret de l'arquitectura client-servidor en la qual els components i recursos d'una aplicació se separen segons la seva funció. Una de les divisions més utilitzades és la que separa el nivell de presentació, el nivell de lògica d'aplicació i el nivell de gestió de dades.

En aquest cas, l'estructura concreta seria de tres nivells (*3-tier architecture*). El model es defineix com a *N-tier architecture* (multinivell), ja que proposa una divisió flexible de les aplicacions en els nivells que calgui per tal de fer més eficient el seu desenvolupament, manteniment i explotació.

En aquest model, la divisió per nivells es fa de forma lineal: el nivell 1 interactua de forma directa i única amb el nivell 2, el nivell 2 interactua amb el 3, i així successivament (vegeu la figura 1.3).

FIGURA 1.3. Arquitectura multinivell



Cal diferenciar entre el concepte multinivell (*multitier N-tier*) i multicapa (*multi-layer N-layer*). En aquest cas, es considera que en el model multinivell cada nivell, a més d'implementar una funció concreta, és executat per un maquinari diferent de la resta de nivells. En el model multicapa, cada capa desenvolupa una funció concreta que pot ser executada per un mateix ordinador que s'encarrega, també, de l'execució d'altres capes.

1.3.2 Arquitectura model-vista-controlador

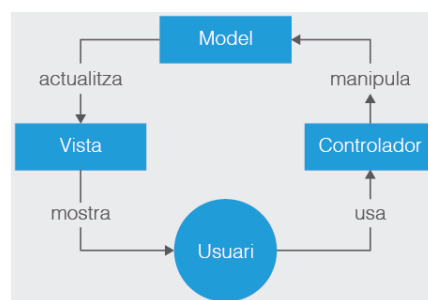
L'arquitectura model-vista-controlador (*model view controller*) és una arquitectura que separa la representació de la informació i la lògica d'una aplicació de la interacció de l'usuari.

Els tres elements que defineix aquesta arquitectura són:

- **Model:** conté les dades de l'aplicació, les regles de negoci o la lògica de l'aplicació i les seves funcions.
- **Vista:** és la representació visible de l'aplicació, la sortida de les dades cap a l'usuari, és a dir, la interfície.
- **Controlador:** controla la interacció de l'usuari (entrada de dades) i converteix aquesta interacció en ordres o comandes per al model o la vista.

La interrelació entre els elements d'aquesta arquitectura no es fa seguint un model lineal com el model multinivell, sinó que es tracta d'un model circular (vegeu la figura 1.4).

FIGURA 1.4. Arquitectura mvc



Paral·lelament a l'estructura de l'aplicació, cal tenir en compte que cada nivell, capa o mòdul pot estar format per un gran nombre de components i recursos de diversos tipus: fitxers HTML, CSS, imatges, etc.

Per això és convenient establir un sistema d'organització coherent i eficient per tal d'estructurar tots aquests components que s'acaben generant durant el desenvolupament d'una aplicació web. La majoria de plataformes de desenvolupament avançades imposen mecanismes per tal d'organitzar i descriure de manera

sistemàtica la localització, les característiques i la configuració dels components i recursos de les aplicacions.

Entre aquests mecanismes en destaquen dos:

- Estructura de directoris: les plataformes avançades de desenvolupament d'aplicacions web acostumen a definir una estructura de directoris mínima que tota aplicació ha de tenir a partir de la qual es despleguen els diversos tipus de components. Els desenvolupadors han de seguir les directrius de cada plataforma.
- Descriptor de desplegament: hi ha un fitxer de configuració on es pot especificar el nom, la ubicació i els paràmetres de configuració dels diversos components que formen una aplicació per tal de tenir aquesta informació centralitzada, accessible i actualitzable sense necessitat de fer modificacions en el codi font de l'aplicació. Aquest descriptor descriu com s'ha de desplegar l'aplicació en el servidor.