

Servidors web i de transferència de fitxers

Roger Girbes Balagué, Miguel Ángel Pérez Pérez

Desplegament d'aplicacions web

Índex

Introducció	5
Resultats d'aprenentatge	7
1 Implantació d'arquitectures web	9
1.1 Arquitectures web. Models	9
1.1.1 El model client-servidor	9
1.1.2 El model client-servidor amb servidors encadenats	10
1.1.3 Aplicacions basades en el web	11
1.1.4 El model d'igual a igual	11
1.2 Servidors web i d'aplicacions. Instal·lació i configuració bàsica	11
1.2.1 Servidors web	11
1.2.2 Servidors d'aplicacions	12
1.2.3 Servidors de bases de dades	12
1.2.4 Servidors de fitxers	13
1.2.5 Servidors de directori	13
1.3 Estructura i recursos d'una aplicació web	14
1.3.1 Arquitectura multinivell	14
1.3.2 Arquitectura model-vista-controlador	15
2 Administració de servidors web	17
2.1 L'URL	17
2.2 Configuració avançada del servidor web	24
2.2.1 Configuració d'Apache	25
2.3 Mòduls: instal·lació, configuració i ús	28
2.3.1 Instal·lació de mòduls en Apache HTTP Server	29
2.3.2 Activació i desactivació de mòduls en Apache HTTP Server	29
2.4 Servidors virtuals. Creació, configuració i utilització	31
2.5 Autenticació i control d'accés	33
2.5.1 Els mòduls de control d'accés	35
2.5.2 Autenticació bàsica amb fitxers	36
2.6 El protocol HTTPS	38
2.7 Certificats. Servidors de certificats	39
2.7.1 Configuració d'Apache per usar SSL	40
2.7.2 Configuració de la seu web amb SSL	41
2.7.3 Verificació de les connexions SSL	42
3 Instal·lació i administració de servidors de transferència de fitxers	43
3.1 Configuració del servei de transferència de fitxers. Permisos i quotes	46
3.1.1 Configuració del servei de transferència de fitxers	46
3.2 Configuració de ProFTPD	47
3.3 Permisos	51
3.4 Quotes	52
3.4.1 Quotes generals del servidor ProFTPD	53

3.4.2	Quotes d'usuari o grups de treball a ProFTPD	53
3.5	Tipus d'usuaris i accessos al servei	56
3.6	Creació d'usuaris a ProFTPD	57
3.6.1	Usuari de sistema a ProFTPD	57
3.6.2	Usuari virtual a ProFTPD	58
3.7	Modes de connexió del client	59
3.7.1	Mode FTP actiu	60
3.7.2	Mode FTP passiu	61
3.8	Protocol de transferència de fitxers segur	62
3.8.1	Configuració FTPS a ProFTPD	64
3.9	Utilització d'eines gràfiques	65
3.9.1	Client gràfic Filezilla	65
3.9.2	Client gràfic dels sistemes operatius	68
3.9.3	Utilització del servei de transferència de fitxers des del navegador	71
3.10	Utilització del servei de transferència de fitxers en el procés de desplegament de l'aplicació web	72

Introducció

El servei HTTP és el servei que permet la creació i visualització de llocs web. El protocol de transferència d'hipertext (HTTP) és un protocol d'aplicació per a sistemes d'informació distribuïts, col·laboradors i hipermèdia. HTTP és la base de la comunicació de dades per a la World Wide Web.

El desenvolupament de l'HTTP va ser iniciat per Tim Berners-Lee al CERN l'any 1989. La primera versió va ser la 0.9, que va aparèixer l'any 1991. Cap a finals del 1996, la Internet Engineering Task Force (IETF) i el World Wide Web Consortium (W3C) van culminar amb la publicació d'una sèrie de *requests for comments* (peticions de comentaris) (RFC). La primera definició d'HTTP/1.1, la versió d'HTTP en ús comú, es va produir a la RFC 2068 el 1997. Durant aquests anys s'han anat fent revisions fins a l'última, l'HTTP/2, l'any 2015. Una de les aplicacions més utilitzades per a aquest protocol és el servidor HTTP Apache.

En primer lloc, dintre del bloc “Implantació d'arquitectures web” veurem els aspectes generals de les diferents arquitectures web, patrons de disseny, tipus de servidors web i d'aplicacions i l'estructura i recursos que componen una aplicació web.

A l'apartat “Administració de servidors web” es descriuen els fonaments i protocols en els quals es basa el funcionament d'un servidor web, el protocol HTTP. S'explica la sintaxi d'aquest protocol. També es mostra com instal·lar i configurar servidors web, gestionar l'accés als llocs web, saber qui té o no té permís per accedir a on, i configurar els mecanismes d'autenticació i control d'accés del servidor.

El 30 d'abril de 1993 Tim Berners-Lee desenvolupa la primera pàgina web. A partir d'aquest moment, durant la dècada dels 90, es produeix un creixement exponencial en el desplegament de les pàgines web. Una de les eines més importants per poder desplegar una pàgina web és el servidor FTP, que ens permet penjar contingut a un servidor remot web en un altre punt del món.

Quan el servei FTP va néixer, el seu objectiu inicial no era associar-se com una eina més per treballar al món web, sinó que es va desenvolupar com un sistema per accedir a recursos en diferents servidors repartits pel món. I això va ser durant el període de l'any 1973 fins als inicis de la dècada dels noranta. Actualment el servei FTP ha evolucionat, però continua tenint l'essència inicial i permet actualitzar els continguts de les pàgines web o aplicacions web que es fan servir actualment.

Com qualsevol aplicació que segueix un protocol estàndard, hi ha diferents alternatives, de programari lliure o privat:

A la banda del servidor hi ha, com a exemples:

- ProFTPD

- Filezilla Server
- VSFTPd
- IIS FTP (*internet information services*)

A la banda del client hi ha:

- Client FTP dels navegadors
- Client FTP dels sistemes operatius
- Client de programari lliure i privatiu
- Clients integrats dins dels IDE de desenvolupament web

En aquesta unitat veureu:

- Desplegament d'un servidor FTP
- Instal·lació i configuració
 - Configuració de permisos i quotes
 - Configuració del servidor amb seguretat
- Utilització d'eines gràfiques
- Desplegament d'aplicacions web mitjançant un client FTP

Resultats d'aprenentatge

En finalitzar aquesta unitat, l'alumne/a:

1. Implanta arquitectures web analitzant i aplicant criteris de funcionalitat.

- Analitza aspectes generals d'arquitectures web, les seves característiques, els avantatges i els inconvenients.
- Realitza la instal·lació i la configuració bàsica de servidors web.
- Classifica i descriu els principals servidors d'aplicacions.
- Realitza la instal·lació i configuració bàsica de servidors d'aplicacions.
- Fa proves de funcionament dels servidors web i d'aplicacions.
- Analitza l'estructura i els recursos que componen una aplicació web.
- Descriu els requeriments del procés d'implantació d'una aplicació web.
- Documenta els processos d'instal·lació i de configuració realitzats sobre els servidors web i d'aplicacions.

2. Gestiona servidors web avaluant i aplicant criteris de configuració per a l'accés segur als serveis.

- Reconeix els paràmetres d'administració més importants del servidor web.
- Amplia la funcionalitat del servidor mitjançant l'activació i la configuració de mòduls.
- Crea i configura llocs virtuals.
- Configura els mecanismes d'autenticació i control d'accés del servidor.
- Obté i instal·la certificats digitals.
- Estableix mecanismes per a assegurar les comunicacions entre el client i el servidor.
- Fa proves de funcionament i de rendiment del servidor web.
- Elabora documentació relativa a la configuració, administració segura i recomanacions d'ús del servidor.
- Realitza els ajustaments necessaris per a la implantació d'aplicacions en el servidor web.

3. Administra servidors de transferència de fitxers avaluant i aplicant criteris de configuració que garanteixin la disponibilitat del servei.

- Instal·la i configura servidors de transferència de fitxers.
- Crea usuaris i grups per a l'accés remot al servidor.
- Configura l'accés anònim.
- Comprova l'accés al servidor, tant de manera activa com passiva.
- Fa proves amb clients en línia d'ordres i clients en mode gràfic.
- Utilitza el protocol segur de transferència de fitxers.
- Configura i utilitza serveis de transferència de fitxers integrats en servidors web.
- Utilitza el navegador com a client del servei de transferència de fitxers.
- Elabora documentació relativa a la configuració i administració del servei de transferència de fitxers.
- Documenta els procediments d'instal·lació i de configuració.

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 diferenciats 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.

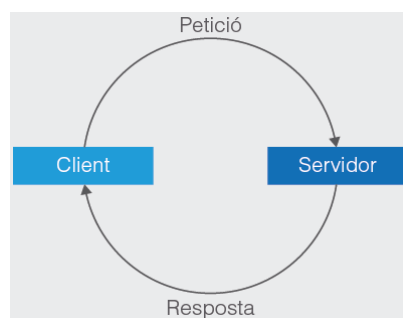
1.1 Arquitectures web. Models

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.

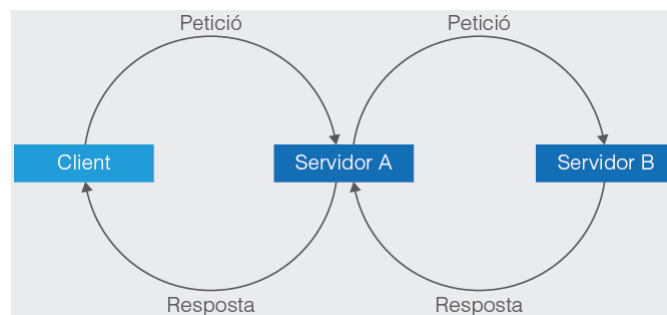
En entorns reals és habitual que no es donin aquestes tres característiques i, per tant, s'hagi d'implementar una arquitectura més complexa basada en el model client-servidor però que pot presentar diferències o ampliacions al model per tal de garantir un bon rendiment de les aplicacions web, la seva 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.

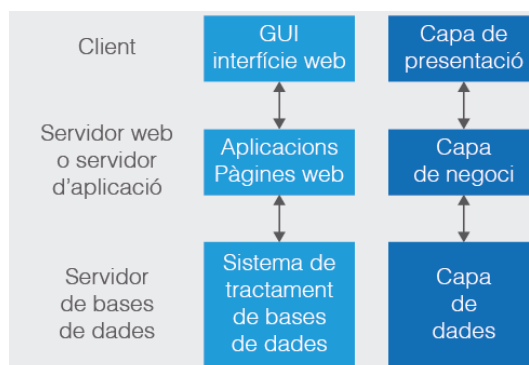
1.3.1 Arquitectura multinivell

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

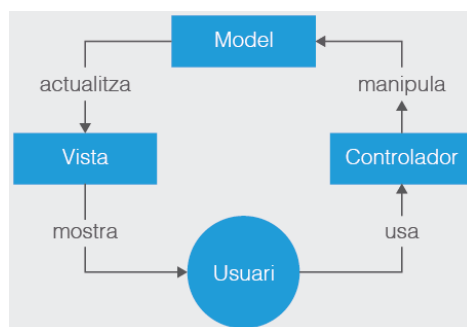
L'arquitectura model-vista-controlador (*model view controller*) és una arquitectura que separa la representació de la informació i la lògica d'una aplicació de la interacció de l'usuari.

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.

2. Administració de servidors web

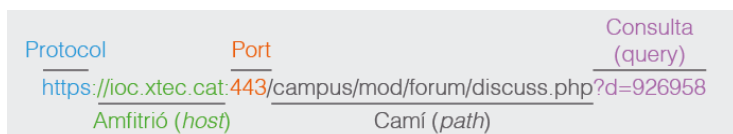
Abans de començar amb la instal·lació, configuració i administració de servidors web és bo conèixer les qüestions fonamentals del protocol HTTP. El protocol de transferència d'hipertext o HTTP (*HyperText Transfer Protocol*) estableix el protocol per a l'intercanvi de documents d'hipertext i multimèdia al web. L'HTTP disposa d'una variant xifrada mitjançant SSL anomenada HTTPS. Tenint en compte que HTTP és el protocol utilitzat en les comunicacions web, conèixer els aspectes bàsics d'aquest protocol ajuda a entendre algunes de les qüestions que tenen a veure amb la configuració i administració de servidors web.

2.1 L'URL

Els localitzadors uniformes de recursos (URL, *Uniform Resource Locator*) són el mecanisme que permet, com indica el seu nom, localitzar recursos a internet mitjançant els diversos protocols disponibles. Existeix també el concepte d'identificador uniforme de recursos (URI, *Uniform Resource Identifier*) que, a la pràctica, es considera equivalent al concepte d'URL, ja que un URI equival al conjunt d'un URL més un nom uniforme de recursos (URN, *Uniform Resource Name*). Com que el concepte URN a la pràctica no s'usa, els termes URL i URI són equivalents i intercanviables en la major part dels casos.

Un URL està format per diversos elements cadascun dels quals ens descriu un aspecte diferent sobre la localització d'un recurs. Els elements en el cas dels URL utilitzats en els protocols HTTP/HTTPS són els que presenta la figura 2.1:

FIGURA 2.1. URL



- Esquema (*scheme*): indica el protocol que s'utilitzarà per accedir al recurs especificat per la resta de l'URL. Segons el protocol indicat (HTTP i HTTPS), la resta de l'URL pot tenir diferències en la seva estructura.
- Informació d'usuari (*user info*): aquest element no forma part de la definició estàndard dels URL, però molts servidors web contempnen l'opció d'ajuntar informació d'usuari (nom, i opcionalment *password*) per tal de facilitar processos bàsics d'identificació i validació d'usuaris.
- Allotjador (*host*): identifica el servidor web. Pot ser un nom de domini o una adreça IP.

- **Port:** es tracta d'un element opcional que serveix per indicar quin port TCP/IP s'utilitzarà per establir la connexió per accedir al recurs. Amb protocol HTTP, si no s'indica, agafa per defecte el port 80 i amb HTTPS s'utilitza per defecte el port 443.
- **Camí (*path*):** indica la localització del recurs dins del servidor. El camí assignat en un URL no té perquè representar un camí físic de disc amb el mateix nom o seqüència de directoris ja que els servidors web permeten fer un mapeig entre camins d'URL i directoris de disc (directoris virtuals).
- **Consulta (*query*):** permet passar paràmetres addicionals útils especialment quan el recurs al qual accedim és un *script* o un altre tipus d'element que executa codi en el servidor. Està format per parells “nom=valor”, els quals, en cas d'haver-n'hi més d'un, se separen amb el caràcter &.
- **Fragment:** permet especificar una secció específica dins del recurs identificar per l'URL. Els navegadors no envien aquesta part de l'URL als servidors web. El navegador, un cop rep la resposta del servidor, si s'ha especificat un fragment, intenta localitzar la secció identificada pel fragment dins del contingut sencer que ha rebut i normalment el focalitza en la visualització a l'usuari.

Molts dels elements d'un URL són opcionals i es poden ometre si el context on s'utilitza l'URL permet assumir-ne uns valors per defecte. Un exemple és a la barra d'adreces d'un navegador, on no és necessari escriure “http://” a l'inici de l'URL, ja que és el valor per defecte per al navegador.

També passa en URL per fer referència a recursos o enllaços dins d'un document HTML. En aquest cas, a més, es pot ometre el nom de *host* i, fins i tot, una part del camí dels URL interns del document HTML.

Podem diferenciar entre:

URL absolut

URL que inclou el nom d'allotjador (*host*) i el camí (*path*) complet del recurs.

Alguns exemples d'URL absoluts són:

https://ca.wikipedia.org/wiki/Localitzador_uniforme_de_recursos

<https://ioc.xtec.cat/educacio/>

URL relatiu

URL que no inclou l'esquema (*scheme*), el *host* i el port. Quan un URL és relatiu s'acostuma a anomenar camí (*path*) i es pot dividir en camins complets i camins relatius.

- **Camins complets (*full paths*):** comencen sempre amb el caràcter / i especifiquen tota la seqüència de directoris que cal recórrer fins arribar al recurs dintre del mateix servidor d'on s'ha descarregat el document HTML, partint del directori arrel del lloc web. Per exemple, /index.html.

- Camins relatius (*relative paths*): comencen amb un caràcter diferent de / i indiquen la seqüència de directoris que cal recórrer per arribar al recurs dintre del mateix servidor d'on s'ha descarregat el document HTML, partint del directori d'on s'ha descarregat el document. Per exemple, ../ioc/imagenes/hdr_bg.gif.

Els caràcters que es poden utilitzar sense restriccions en els URL són:

TAULA 2.1. Caràcters sense restriccions URL

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
0	1	2	3	4	5	6	7	8	9	-	.	~													

També es poden usar altres caràcters, però han d'estar codificats per poder ser interpretats. Per fer-ho, s'usa la codificació percentual (*percent-encoding* o *URL encoding*):

TAULA 2.2. Caràcters sense restriccions URL

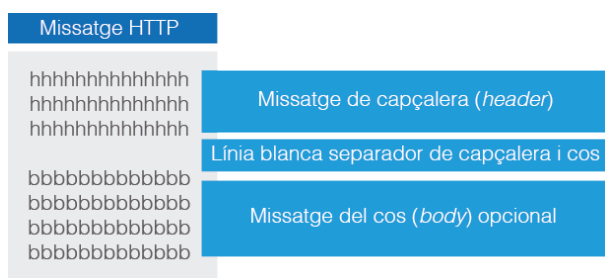
!	#	\$	&	'	()	*	+	,	/	:	;	=	?	@	[]
%21	%23	%24	%26	%27	%28	%29	%2A	%2B	%2C	%2F	%3A	%3B	%3D	%3F	%40	%5B	%5D

HTTP és un protocol de petició/resposta (*request-response*). El cicle de comunicació entre el client i el servidor es basa en dos passos on la resposta per part del servidor ve precedida d'una petició per part del client.

Tot i que les peticions i les respostes contenen informació diferent, totes dues tenen una mateixa estructura formada per una capçalera i un cos. La capçalera conté informació sobre el missatge (metadades) i el cos és el que du el contingut del missatge. El contingut de les peticions i respostes pot ser buit en algunes ocasions, per tant, pot haver-hi peticions o respostes sense contingut, només amb capçalera.

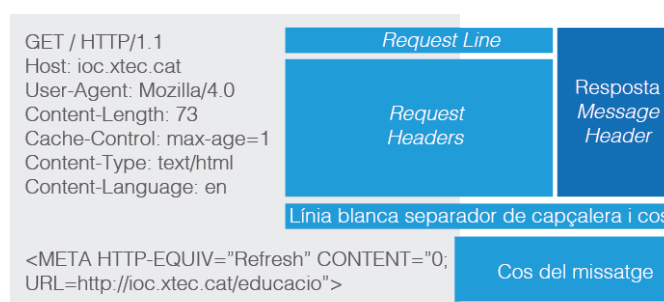
Els missatges HTTP, tant si són peticions com respostes, tenen una estructura formada per una capçalera i un cos. La capçalera està formada per informació alfanumèrica i està separada del cos per una línia en blanc (vegeu la figura 2.2). El contingut del cos (que pot ser opcional) és alfanumèric o binari.

FIGURA 2.2. Missatge HTTP



Les peticions HTTP, a la primera línia de la capçalera, contenen la línia de petició (*request line*). La resta de línies (opcionals) contenen les capçaleres de petició (*request headers*) tal com es veu a la figura 2.3.

- Línia de petició: primera línia de la capçalera d'una petició. Està formada per tres camps:
 - Mètode de petició (*request method*): pot ser qualsevol dels mètodes de petició que defineix el protocol HTTP.
 - URL de petició (*request URL*): és l'URL del recurs que es demana.
 - Versió HTTP: és la versió de protocol HTTP que s'utilitzarà (HTTP/1.0 o HTTP/1.1).
- Capçalera de petició: cada capçalera va en una línia diferent i està formada per un parell "nom: valor". Si hi ha diversos valors, aquests se separen amb comes.

FIGURA 2.3. Petició HTTP

El protocol HTTP té definits 8 mètodes de petició (*request methods*), que són el primer que s'especifica en la línia de les peticions.

Una petició feta amb el mètode POST oculta els paràmetres, però això no implica que sigui un mecanisme de seguretat. Si el protocol utilitzat és HTTP, la informació s'envia en clar a través de xarxa i pot ser interceptada durant el procés d'enviament.

1. GET: el mètode GET s'utilitza per fer la petició d'un recurs o document al servidor (un document HTML, una imatge...), especificat en el camp URL de la línia de petició (*request line*). Les peticions realitzades amb el mètode GET només haurien de fer que el servidor torni algun tipus de resposta sense que això provoqui la modificació de les dades del servidor ni de l'aplicació web sobre el que s'ha executat amb el GET.
2. HEAD: el mètode HEAD s'usa per simular una petició d'un recurs al servidor. Funciona exactament igual que el GET, amb la diferència que el servidor només respon amb les capçaleres de la resposta (no envia el contingut). S'utilitza per poder fer un pronòstic de resposta d'una hipotètica petició GET. És emprat habitualment pels navegadors per comprovar les capçaleres d'un recurs determinat i així decidir si cal descarregar-lo o no en cas de tenir-ne una còpia en la memòria *cache* local (les capçaleres ens poden dir la mida del document, la data d'última modificació, etc.).
3. POST: és el mètode habitual per enviar les dades de formularis HTML que poden provocar modificacions a les dades del servidor o de l'aplicació web. Pot semblar igual que el mètode GET, però té algunes diferències importants: 1) Els paràmetres d'un GET van concatenats a l'URL de la petició i és visible a la barra d'adreces del navegador. En canvi amb POST s'adjunten com a contingut del cos de la petició i, per tant, no són tant visibles. Com a conseqüència, les peticions GET es poden memoritzar als

marcadors del navegador i es poden representar, també, com a URL en un enllaç dintre d'un document HTML. Amb POST, això no és possible. 2) Les peticions GET es consideren idempotents (així ho defineix HTTP), la qual cosa significa que s'han de poder executar tantes vegades com es vulgui sense que es produeixin modificacions de l'estat o les dades del servidor. Per tant, els navegadors permeten l'execució repetida de peticions GET. Per exemple, actualitzant una pàgina carregada amb GET o tirant enrere a una pàgina prèviament carregada amb GET sense donar cap avís a l'usuari. En canvi, amb el POST és a l'inrevés. Les peticions POST es consideren no idempotents (pot provocar modificacions) i els navegadors, cada cop que es pretén repetir una petició POST, informen l'usuari amb un missatge de confirmació.

4. PUT: el mètode PUT s'utilitza per poder crear o reemplaçar un determinat recurs o document del servidor. Amb el mètode PUT podem sol·licitar que el contingut present en el cos de la petició s'emmagatzemi en el servidor i passi a estar accessible a través d'un URL que indiquem a la línia de petició. Si ja existeix, se substituiria l'actual pel que donem amb el PUT. Si no existeix, se'n crearia un de nou. La utilització d'aquest mètode està restringit en la configuració per defecte dels servidors web ja que permetria la modificació no autoritzada dels seus continguts (Si ho provem, podem rebre un error del tipus "*405 Method Not Allowed*").
5. DELETE: el mètode DELETE s'utilitza per poder eliminar un determinat recurs o document del servidor. Amb el mètode DELETE es pot sol·licitar al servidor que elimini l'arxiu associat a un URL del recurs indicat a la línia de petició. La utilització d'aquest mètode està restringida en la configuració per defecte dels servidors web ja que permetria la modificació no autoritzada dels seus continguts (si ho provem, podem rebre un error del tipus "*405 Method Not Allowed*").
6. CONNECT: aquest mètode s'utilitza per poder passar connexions segures SSL a través de connexions HTTP i per poder gestionar connexions HTTP a través de servidors intermediaris (*proxies*).
7. OPTIONS: el mètode OPTIONS demana al servidor quins mètodes HTTP es poden utilitzar sobre el recurs identificat amb un URL de la línia de petició.
8. TRACE: el mètode TRACE demana al servidor que retorni una còpia de les capçaleres de la petició tal com les ha rebut. Aquest mètode HTTP sol estar desactivat per defecte.

Un cop especificada la línia de petició d'una petició HTTP, pot haver-hi una sèrie de capçaleres de petició (opcionals) cadascuna de les quals ocupa una línia i té el format "nom: valor".

Les capçaleres més habituals en les peticions HTTP enviades als navegadors són:

- Allotjador (*host*): capçalera obligatòria en HTTP/1.1. El client ha d'especificar el nom del *host* al qual envia la petició (una mateixa màquina pot tenir

Els factors de qualitat en les capçaleres *Accept (q)* són nombres reals entre 0 i 1 que indiquen la preferència del client sobre un determinat aspecte de la resposta esperada del servidor. Per defecte, el factor de qualitat val 1.

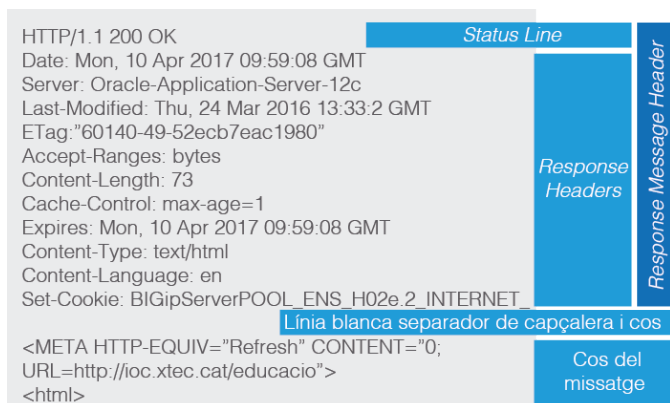
diversos noms de domini associats). Un mateix servidor amb una única IP pot servir diversos llocs/aplicacions web amb noms de domini diferents. Per poder identificar en quin d'ells volem enviar la petició usem aquesta capçalera.

- **Usuari-agent (*user-agent*):** indica quin programari client (habitualment un navegador) utilitza l'usuari. Sol ser una cadena que inclou un nom identificador del navegador, un descriptor de versió i un descriptor de sistema operatiu i plataforma sobre la qual s'executa el navegador. Per exemple: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/29.0.1547.76 Safari/537.36
- ***Accept*:** el client pot utilitzar aquesta capçalera per indicar al servidor els tipus MIME que és capaç de gestionar i quin és el seu ordre de preferència. Si el servidor té diverses versions del recurs sol·licitat pot consultar aquesta capçalera per decidir quina s'enviarà al client. Aquest procés s'anomena negociació de tipus de continguts (*content-type negotiation*). Per exemple: *Accept: image/png,image/*;q=0.8,*/*;q=0.5*
- ***Accept-Language*:** el client pot utilitzar aquesta capçalera per indicar al servidor els idiomes preferits per a les respostes obtingudes. Si el servidor té el recurs sol·licitat en diversos idiomes, pot consultar aquesta capçalera per decidir quina versió enviarà al client. Aquest procés s'anomena negociació d'idioma (*language negotiation*). Per exemple: *Accept-Language: ca,es*
- ***Accept-Charset*:** el client pot utilitzar aquesta capçalera per indicar al servidor el joc de caràcters preferit per a les respostes obtingudes que incloguin informació alfanumèrica. Aquest procés s'anomena negociació de joc de caràcters (*charset negotiation*). Per exemple: *Accept-charset: utf-8, iso8859-1*
- ***Content-Type* i *content-Length*:** quan la petició té un cos, és a dir, un contingut, a la capçalera s'ha d'especificar el tipus (amb un identificador MIME) i la mida (en bytes) mitjançant aquestes dues capçaleres.
- ***Referer*:** aquesta capçalera permet al client especificar al servidor l'adreça (URL) del recurs d'on s'ha obtingut l'URL de la petició que se li està enviant.
- ***Accept-Encoding*:** el client pot utilitzar aquesta capçalera per indicar al servidor els tipus de codificació que suporta. Si el servidor té el recurs comprimit o és capaç de comprimir-lo al vol en algun dels formats que accepta el client, ho pot utilitzar per reduir el temps de transmissió. El servidor ha d'indicar en la capçalera *Content-Encoding* de la resposta el tipus de compressió que ha aplicat.
- ***Connection*:** el client pot utilitzar aquesta capçalera per indicar al servidor si ha de tancar la connexió un cop enviada la resposta a la petició rebuda, o deixar la connexió oberta a l'espera de rebre noves peticions sense haver de repetir el procés d'establiment de la connexió. Pot tenir dos valors: *close* o *keep-alive*.

Les respostes HTTP, a la primera línia de la capçalera, contenen la línia d'estatus. La resta de línies (opcionals) contenen les capçaleres de resposta tal com es veu a la figura 2.4.

- Línia d'estatus (*status line*): primera línia de la capçalera d'una resposta. Està formada per tres camps:
 - Versió HTTP: versió de protocol HTTP que utilitza el servidor (HTTP/1.0 o HTTP/1.1).
 - Codi d'estatus: codi de tres dígitos que indica el resultat de la petició.
 - *Reason phrase*: petita explicació del significat del codi d'estatus.
- Capçalera de resposta (*response header*): cada una va en una línia diferent i està formada per un parell “nom: valor”. Si hi ha diversos valors, se separen amb comes.

FIGURA 2.4. Resposta HTTP



El protocol HTTP defineix 5 codis d'estat (*status codes*) que permeten indicar diversos tipus de situacions que es poden donar com a resposta a una petició d'un client.

- 1xx: són de tipus informatiu, informen el client que la petició ha estat rebuda i que el servidor continua processant la resposta.
- 2xx: indiquen la petició ha estat correcta i s'ha processat satisfactòriament.
- 3xx: indiquen alguna forma de redirecció. Amb un codi d'aquesta sèrie es dona a entendre al client que la petició és correcta però que la resposta s'ha d'obtenir d'algun altre lloc.
- 4xx: indiquen que hi ha hagut una errada en el processament de la petició perquè el client ha fet alguna cosa malament, l'error ha estat causat pel client.
- 5xx: indiquen que hi ha hagut una errada en el processament de la petició a causa d'una fallada en el servidor, l'error ha estat causat pel servidor.

Un cop especificada la línia d'estat d'una resposta HTTP, hi ha una sèrie de capçaleres de resposta (opcionals) cadascuna de les quals ocupa una línia i té el format “nom: valor”. Les capçaleres més habituals en les respostes HTTP són:

- *Content-Base*: capçalera que conté un URL per ser utilitzat com a base per als URL relatius que hi hagi als documents HTML que s'enviïn amb la resposta.
- *Content-Length*: capçalera de resposta que indica la mida en bytes del cos associat a la resposta.
- *Content-Type*: capçalera que indica el tipus de contingut del cos associat a la resposta. Els *media types* reconeguts estan definits per la Internet Assigned Numbers Authority (IANA).
- *Date*: aquesta capçalera, obligatòria en totes les respostes en HTTP/1.1, indica la data i hora d'enviament de la resposta, és a dir, quan surt del servidor.
- *Last-Modified*: capçalera que indica la data i hora en la qual el recurs demanat ha estat actualitzat per últim cop. L'objectiu d'aquesta capçalera és permetre la gestió de *cache* de recursos. No funciona correctament per a recursos dinàmics. Per solucionar-ho es va introduir la capçalera ETag.
- *ETag*: permet que el servidor pugui enviar un *hash* o *checksum* del cos de la resposta etiqueta d'entitat (*entity tag*) per ser utilitzat per al control de *cache* de recursos HTTP en els clients i servidors proxies. Es pot considerar que totes les còpies d'un recurs amb el mateix URL i la mateixa etiqueta d'entitat són idèntics. Per tant, si se'n té una còpia en memòria *cache* no cal descarregar-los un altre cop.
- *Server*: és l'equivalent a la capçalera de petició *User-Agent*, però del costat del servidor. Serveix perquè el servidor web pugui indicar el seu nom i versió.
- *Location*: indica al client cap on ha d'anar a buscar un recurs que hagi demanat i que no es troba a l'URL de la petició (resposta amb un codi d'estat de la sèrie 3xx).

2.2 Configuració avançada del servidor web

En aquesta unitat fareu servir el servidor HTTP anomenat Apache. Apache és un servidor HTTP (de pàgines web) de codi obert multiplataforma desenvolupat per Apache Software Foundation. Apache presenta, entre altres característiques, missatges d'error altament configurables, bases de dades d'autenticació i negociació de continguts, però s'ha criticat per la manca d'una interfície gràfica que ajudi a configurar-lo.

2.2.1 Configuració d'Apache

El sistema on realitzareu la instal·lació i configuració està basat en un sistema Debian (Ubuntu o Lubuntu).

El conjunt d'instruccions d'instal·lació o configuració mitjançant ordres de sistema és dependent d'aquest (si aneu a un sistema REDHAT, CENTOS, etc. heu de mirar el procediment d'instal·lació i les ordres específiques del sistema).

La configuració interna és independent del sistema operatiu que fem servir, qualsevol configuració que realitzeu durant aquests apartats és compatible amb altres sistemes operatius basats en Unix.

Per iniciar el procés d'instal·lació d'Apache, primer de tot obriu un terminal i executeu les ordres d'actualització dels repositoris d'Ubuntu:

```
1 sudo apt-get update
```

```
1 sudo apt-get install apache2
```

APT

APT o Advanced Package Tool és un programari gratuït dins de Debian que funciona amb les llibreries del nucli, per tractar les instal·lacions, configuracions i eliminar programari.

APT permet de forma fàcil agafar la versió més nova d'un paquet d'instal·lació referent a un programari i aplicar les configuracions necessàries depenent de la versió del sistema.

Una vegada accepteu la instal·lació, el procés APT instal·la Apache.

Una vegada finalitzada la instal·lació procediu a verificar el resultat d'executar l'ordre següent:

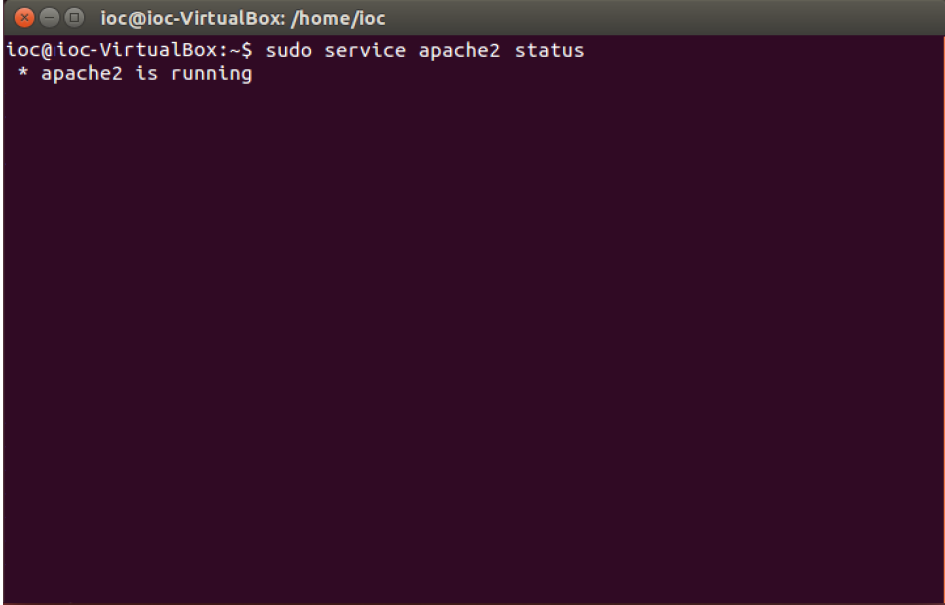
```
1 sudo service apache2 status
```

Us sortirà la següent sortida per a terminal:

```
1 ioc@ioc:/home/ioc# sudo service apache2 status
2 * apache2 is running
```

A la figura [2.5](#) podeu veure la finestra amb l'estat del servidor Apache.

FIGURA 2.5. ApacheStatus



```
ioc@ioc-VirtualBox: /home/ioc
ioc@ioc-VirtualBox:~$ sudo service apache2 status
* apache2 is running
```

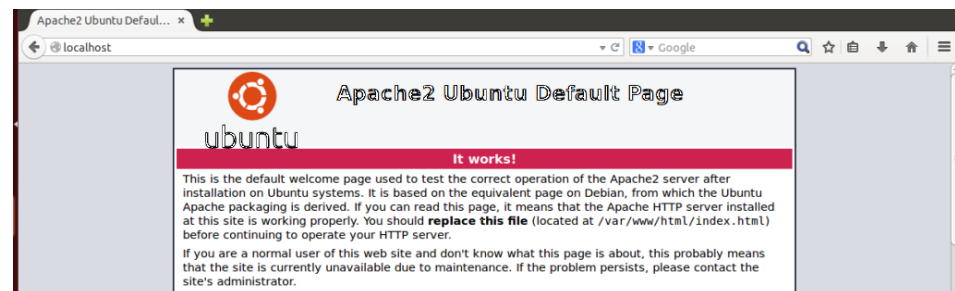
Com podeu veure, indica que el servidor està en funcionament.

A continuació, des d'un navegador, executeu:

1 localhost

Si tot ha anat bé ha de mostrar una pàgina web per informar de la correcta instal·lació del servidor Apache, tal com podeu veure a la figura 2.6.

FIGURA 2.6. ItWorks



Una vegada verificat que Apache funciona, la configuració del servidor es fa mitjançant un fitxer de text pla, de la mateixa manera que se sol fer amb molts serveis d'Unix, com per exemple vsftpd.

El directori on hi ha les configuracions és dins */etc/apache2*, i el fitxer de configuració principal és el fitxer *apache2.conf*.

Editar el fitxer original pot comportar modificar el fitxer i perdre les dades inicials de configuració. Sempre és bo fer una còpia de seguretat del fitxer amb l'ordre `cp` (per copiar fitxers).

Navegueu fins al directori de configuracions d'Apache */etc/apache2* i executeu:

```
1 ls -l
```

Amb aquesta ordre es llista tot el contingut del directori, amb el resultat següent:

```
1 ioc@ioc:/home/ioc# ls -l
2 total 80
3 -rw-r--r-- 1 root root 7115 gen 7 2014 apache2.conf
4 drwxr-xr-x 2 root root 4096 abr 11 11:10 conf-available
5 drwxr-xr-x 2 root root 4096 abr 11 11:10 conf-enabled
6 -rw-r--r-- 1 root root 1782 gen 3 2014 envvars
7 -rw-r--r-- 1 root root 31063 gen 3 2014 magic
8 drwxr-xr-x 2 root root 12288 abr 11 11:10 mods-available
9 drwxr-xr-x 2 root root 4096 abr 11 11:10 mods-enabled
10 -rw-r--r-- 1 root root 320 gen 7 2014 ports.conf
11 drwxr-xr-x 2 root root 4096 abr 11 11:10 sites-available
12 drwxr-xr-x 2 root root 4096 abr 11 11:10 sites-enabled
```

Com podeu veure, el directori */etc/apache* conté diferents fitxers, incloent-hi *apache2.conf*. Els fitxers que hi ha dins fan referència a fitxers de configuracions de modularitats que amplien la capacitat d'Apache, com pot ser: afegir mòduls com PHP, configurar la seguretat per xifrar la informació, crear servidors virtuals, etc.

Inicieu l'edició del fitxer *apache2.conf*.

Dins del directori feu:

```
1 sudo nano apache2.conf
```

Caràcter

Abans de detallar configuracions, fixeu-vos en el fitxer *apache2.conf*. Moltes de les línies del fitxer de configuració estan iniciades amb el caràcter # (*hashtag* o etiqueta).

El caràcter # es fa servir en múltiples configuracions de serveis del sistema. Permet comentar línies de configuració i fer comentaris, ja que l'interpret de l'aplicació no els executa.

Vegeu a "Annexos" el punt de directives d'Apache.

Com podeu comprovar, el fitxer de configuració *apache2.conf* és bastant extens. Tot seguit es detallen les opcions de configuració més rellevants. Per verificar sintaxis i configuracions us recomanem que visiteu l'annex on s'expliquen les sintaxis i les configuracions permeses.

Dins de la configuració inicial, es poden trobar algunes de les directives més importants:

- **Timeout**: nombre de segons abans que rep i envia el temps d'espera.
- **Keep-alive**: permet acceptar connexions persistents.
- **MaxKeepAliveRequests**: nombre màxim de sol·licituds de permís durant una connexió persistent.
- **KeepAliveTimeout**: nombre de segons d'espera per a la següent petició del mateix client en la mateixa connexió.
- **ErrorLog**: ubicació de l'arxiu de registre d'errors.

Apache

Apache té una gran extensió de directives per configurar el servidor HTTP. Per la seva extensió natural no es pot detallar tot. Per a més detall dirigiu-vos a la documentació oficial.

- ***Include module configuration***: camí on es troba el fitxer de mòduls disponibles i actius.
- ***Include list of ports to listen***: camí on es troba el fitxer dels ports d'escolta del servidor.
- ***DocumentRoot***: estableix el directori de publicació del seu web principal o per defecte.
- ***Directory***: per a cada directori que calgui configurar es pot definir un bloc d'opcions de configuració agrupades en aquesta directiva.
- ***DirectoryIndex***: documents que es mostren per defecte quan se sol·licita un URL i no s'especifica el document.
- ***AccessFileName***: nom de l'arxiu que ha de buscar en cada directori per a directives de configuració addicionals.
- ***Include the virtual host configurations***: camí on es troba el fitxer de configuració dels servidors virtuals.
- ***User***: compte d'usuari que el servidor web Apache utilitzarà mentre s'executi (determina els permisos d'accés que tindrà sobre directoris i arxius).
- ***Group***: grup d'usuaris que el servidor web Apache utilitzarà mentre s'executi (igual que amb l'usuari, determina els permisos d'accés que tindrà sobre directoris i arxius).

Amb les opcions d'instal·lació per defecte, ja tindreu el lloc web per defecte a la carpeta:

```
1 /var/www
```

Aquest lloc web, per defecte, és l'ofert per a qualsevol connexió que es faci sobre el port 80 en qualsevol de les adreces IP o noms de domini establerts sobre el servidor. Per modificar el port cal modificar el fitxer *ports.conf*.

2.3 Mòduls: instal·lació, configuració i ús

La gran majoria de servidors web permeten la instal·lació de mòduls per ampliar les seves funcionalitats. Tenir funcionalitats en forma de mòdul permet adaptar millor el consum de recursos del servidor web a les nostres necessitats de producció (el servidor web només carregarà i executarà els mòduls que com a administradors tenim instal·lats i configurats).

El servidor web Apache HTTP Server permet afegir funcionalitats addicionals a les que ens ofereix la seva configuració bàsica en forma de mòduls instal·lables.

2.3.1 Instal·lació de mòduls en Apache HTTP Server

Treballant en una plataforma amb un sistema Debian, la instal·lació de mòduls es fa normalment mitjançant el gestor d'instal·lació de paquets APT. La instal·lació d'un mòdul afegeix els arxius següents en el sistema:

- El fitxer del mòdul (habitualment el nom sol començar per *mod* i acabar en *.so*) es guarda al directori */usr/lib/apache2/modules*. Si feu un *ls -l* de */usr/lib/apache2/modules* podeu veure el contingut de la carpeta similiar a:

```
1 ioc@ioc:/usr/lib/# ls -l
2 total 2732
3 -rw-r--r-- 1 root root 13937 jul 15 2016 httpd.exp
4 -rw-r--r-- 1 root root 10256 jul 15 2016 mod_access_compat.so
5 -rw-r--r-- 1 root root 10248 jul 15 2016 mod_actions.so
6 ...
7 -rw-r--r-- 1 root root 14352 jul 15 2016 mod_usertrack.so
8 -rw-r--r-- 1 root root 10256 jul 15 2016 mod_vhost_alias.so
9 -rw-r--r-- 1 root root 22536 jul 15 2016 mod_xml2enc.so
```

- El fitxer que conté la directiva per carregar del mòdul (LoadModule) té com a extensió *.load*, i es guarda al directori */etc/apache2/mods-available*.
- El fitxer que conté les directives de configuració del mòdul (en cas que en tingui), amb una configuració per defecte, acostuma a tenir una extensió *.conf*, i està emmagatzemat al directori */etc/apache2/mods-available*. No obstant això, no tots els mòduls venen amb el corresponent fitxer de directives de configuració.

2.3.2 Activació i desactivació de mòduls en Apache HTTP Server

Un cop instal·lats els mòduls, s'han d'activar. Amb servidors Apache es poden tenir mòduls instal·lats però no activats, és a dir, que no s'estan utilitzant.

Per activar un mòdul cal crear un enllaç simbòlic dins del directori */etc/apache2/mods-enabled* sobre el fitxer *.load* i sobre el fitxer *.conf* (si té fitxer associat) que conté la directiva de càrrega del mòdul i les directives de configuració respectivament. Aquests fitxers són */etc/apache2/mods-available*. Per facilitar la configuració i l'administració del servidor s'acostuma a posar el mateix nom del fitxer a l'enllaç simbòlic.

També es poden activar els mòduls usant les eines que faciliten el servidor Apache i el sistema operatiu. Disposeu de l'ordre *a2enmod*, que permet activar un mòdul sempre que estigui instal·lat. Si li doneu com a paràmetre el nom del mòdul que s'ha d'activar, crea l'enllaç o enllaços simbòlics necessaris dins el directori */etc/apache2/mods-enabled*.

Per exemple:

```
1 a2enmod rewrite
```

Un cop instal·lats i activats els mòduls, ja poden ser utilitzats. A partir d'aquí, podeu canviar la configuració manipulant el fitxer *.conf* que estigui associat al mòdul. Si no existeix aquest fitxer, pot ser perquè el mòdul no té directives de configuració o perquè per defecte no porta el fitxer (hi ha la possibilitat que es pugui crear i posar-hi algunes directives). Per poder canviar la configuració d'un mòdul, heu de consultar-ne la documentació tècnica per saber quines directives de configuració ofereix i quines opcions possibles hi ha per a cada directiva.

Per desactivar un mòdul hi ha dues opcions, igual que abans. O bé esborreu els enllaços simbòlics que heu creat o bé useu l'ordre *a2dismod*.

Per exemple:

```
1 a2dismod rewrite
```

Els mòduls d'Apache són opcionals (no tenen perquè estar carregats), les seves directives de configuració d'un mòdul en concret poden provocar errors en cas que el mòdul no estigui activat. Per evitar això, Apache disposa del bloc *<IfModule>*, que es pot incloure en els seus fitxers de configuració que permeten indicar que unes determinades directives de configuració tinguin efecte sobre el servidor només si el mòdul que s'indica està actiu.

```
1 <IfModule nom_mòdul.c>
2   # Directives de configuració del mòdul, en cas que estigui activat.
3   ...
4 </IfModule>
```

Per saber quins mòduls estan actius, podeu llistar el contingut del directori amb */etc/apache2/mods-enabled*.

```
1 ioc@ioc:/etc/apache2# ls /etc/apache2/mods-enabled/*.load
2 /etc/apache2/mods-enabled/access_compat.load
3 /etc/apache2/mods-enabled/alias.load
4 /etc/apache2/mods-enabled/auth_basic.load
5 /etc/apache2/mods-enabled/authn_core.load
6 /etc/apache2/mods-enabled/authn_file.load
7 /etc/apache2/mods-enabled/authz_core.load
8 /etc/apache2/mods-enabled/authz_host.load
9 /etc/apache2/mods-enabled/authz_user.load
10 /etc/apache2/mods-enabled/autindex.load
11 /etc/apache2/mods-enabled/deflate.load
12 /etc/apache2/mods-enabled/dir.load
13 /etc/apache2/mods-enabled/env.load
14 /etc/apache2/mods-enabled/filter.load
15 /etc/apache2/mods-enabled/mime.load
16 /etc/apache2/mods-enabled/mpm_event.load
17 /etc/apache2/mods-enabled/negotiation.load
18 /etc/apache2/mods-enabled/setenvif.load
19 /etc/apache2/mods-enabled/status.load
```

L'ordre *apache2ctl -l* llista els mòduls compilats que el servidor Apache porta integrats (mòduls que no es poden descarregar o desactivar).

Per saber quins mòduls hi ha disponibles per instal·lar al servidor, amb el gestor de paquets APT podeu executar la comanda *apt-cache search libapache2-mod*. Es mostrarà un llistat amb el nom del mòdul i una petita descripció de cadascun.

2.4 Servidors virtuals. Creació, configuració i utilització

El protocol DNS permet tenir diversos noms de domini que apunten sobre un mateix servidor (pot ser una mateixa IP o poden ser adreces IP diferents que corresponguin a un mateix servidor).

Mitjançant el protocol DNS es pot aconseguir que un servidor ofereixi llocs webs diferents en funció del nom de domini que s'hagi utilitzat per establir la connexió, gràcies, també, a la capçalera *host* que proporciona el protocol HTTP. Això és el que es coneix com a servidors virtuals (un mateix servidor que actua com si fossin diversos servidors web).

En la terminologia d'Apache s'anomena *virtual host* o *vhost* cada un dels servidors virtuals que hi ha en funcionament a banda del servidor principal o per defecte:

- Quan s'assignen servidors virtuals diferents a adreces IP diferents es parla de **servidors virtuals basats en IP** o *IP-based vhosts*.
- Quan s'assignen múltiples seus virtuals a una mateixa adreça IP es parla de **servidors virtuals basats en nom** o *Name-based vhosts*.

Els passos que s'han de seguir per crear i posar en marxa un servidor virtual són:

1. Aneu al directori */etc/apache2/sites-available*. Aquesta carpeta conté els fitxers de configuració per a cadascun dels servidors virtuals disponibles al servidor.
2. Creeu un nou fitxer de configuració del lloc web amb el nom que vulgueu, dins del mateix directori.
3. Afegiu les opcions per adaptar-lo a les necessitats del nou servidor virtual dintre d'aquest fitxer.
4. Creeu l'enllaç simbòlic dintre la carpeta */etc/apache2/sites-enabled* que apunti cap al fitxer de configuració del servidor virtual creat a la carpeta */etc/apache2/sites-available*. O bé useu l'ordre *a2ensite nom_fitxer_servidor_virtual.conf*.
5. Executeu l'ordre *service apache2 reload*. Aquesta ordre força al servidor Apache a tornar a carregar la seva configuració.
6. Comproveu com respon el servidor facilitant el lloc web corresponent a cada servidor virtual.

Tot procediment de configuració s'ha de dur a terme amb privilegis de superusuari (*root*).

Vegeu un exemple de configuració d'un servidor virtual:

```
1 <VirtualHost *:80>
2
3 # Aquest servidor virtual serà el que actuarà quan a la capçalera Host d'una
4 # petició HTTP hi trobem "www.iocdaw.cat".
5
6 ServerName www.iocdaw.cat
7
8 # Directori arrel del lloc web (directori físic de disc).
9
10 DocumentRoot /daw/m08
11
12 # Opcions de configuració per al directori arrel del lloc web.
13 <Directory /webs/enciams>
14     Options -FollowSymLinks -Indexes +MultiViews AllowOverride None
15     Order allow,deny
16     Allow from all
17 </Directory>
18
19 # Opcions de configuració dels logs del servidor virtual.
20 ErrorLog ${APACHE_LOG_DIR}/daw.error.log
21 LogLevel warn
22 CustomLog ${APACHE_LOG_DIR}/daw.access.log combined
23
24 </VirtualHost>
```

Vegeu l'anàlisi detallada de les principals opcions de configuració necessàries per definir una seu virtual:

- **VirtualHost:** aquesta directiva és la que fa el lligam amb l'adreça IP i port assignats a la seu virtual. Tot i que, per claredat, en l'exemple s'ha indicat un nom d'amfitrió (*host*) en lloc d'una adreça IP, Apache recomana usar sempre l'adreça IP.
- **ServerAdmin:** indica el nom de l'administrador de la seu virtual. De fet, n'indica el correu electrònic.
- **DocumentRoot:** defineix el directori de publicació de la seu web virtual. El directori que s'indica és una ruta absoluta del sistema físic de fitxers, no una ruta relativa del servidor web.
- **ServerName:** és el nom virtual amb el qual es reconeix aquest web, el nom que els clients han de referenciar per poder accedir al web.
- **errorLog i CustomLog:** aquestes dues directives especifiquen la ubicació dels fitxers de registre o logs de monitoratge de l'activitat d'aquesta seu web. Les rutes que s'hi indiquen són relatives i s'utilitza el directori de logs definit en la configuració global.

Cal tenir presents algunes qüestions relatives als permisos d'accés a carpetes i fitxers:

Al directori `/etc/apache2` hi ha un fitxer anomenat `envvars` on, entre d'altres aspectes, es defineix el nom d'usuari i el grup amb el qual treballarà el servidor Apache.

Per defecte trobarem:

```
export APACHE_RUN_USER=www-data
```

```
export APACHE_RUN_GROUP=www-data
```

Això indica que el servidor Apache accedirà a arxius i directoris utilitzant l'usuari `www-data` i el grup `www-data`. Tenint en compte aquest tema, caldrà donar permís de lectura als fitxers (`r-`) i accés als directoris (`r-x`), com a mínim, a aquest usuari/grup a tots els directoris i fitxers que formin part dels llocs web que serveixi Apache.

2.5 Autenticació i control d'accés

Fins ara heu vist com crear i configurar diverses seus web accessibles per a tothom que tingui accés al servidor. Hi ha ocasions en què es vol restringir l'accés a una part del web o a tot el web però només per a uns usuaris concrets. En aquest apartat es descriuen diverses formes de fer-ho.

El servei web incorpora mecanismes bàsics per verificar els usuaris que volen accedir a àrees restringides. Però, a més a més, la flexibilitat dels mòduls fa que es puguin afegir nous mecanismes que puguin sorgir més endavant tot i que no hagin estat desenvolupats per Apache. Així, per validar l'accés a un directori amb material dels professors en un web d'una escola segurament n'hi ha prou amb el mecanisme bàsic de verificació d'usuaris i grups. En canvi, per accedir a un web ultrasecret d'una agència governamental potser cal incorporar mecanismes addicionals, basats per exemple en l'empremta òptica i el registre de veu.

Primerament cal analitzar els mecanismes de validació d'usuaris generals que permet el servidor web:

- **Autenticació bàsica amb fitxers:** el mecanisme més simple per implementar el control d'accés a recursos d'una seu web és utilitzar fitxers d'**usuaris** i **grups** propis del servidor. Apache proporciona eines per crear-los. L'avantatge principal d'aquest mètode és la facilitat d'administració. L'inconvenient és que comporta una gestió diferenciada dels usuaris del servei web i dels del sistema. De fet, això pot ser un inconvenient o un avantatge, si el que interessa és tenir-los segregats.
- **Autenticació mitjançant PAM:** en els sistemes GNU/Linux actuals l'autenticació dels usuaris es fa via PAM (*Pluggable Authentication Module*). El PAM comprova el directori `/etc/passwd`, el LDAP, el Kerberos, les empremtes dactilars o el que calgui. Usar el lligam amb el mòdul del PAM

és un bon mecanisme per validar els usuaris del servei web igual que es validen els usuaris del sistema.

- **Autenticació mitjançant LDAP:** un dels mecanismes més populars actualment per a l'autenticació és el LDAP. El mòdul del LDAP permet passar la validació dels usuaris a l'encarregat de gestionar l'autenticació LDAP dels usuaris del sistema. També es pot tenir en funcionament un servei LDAP específic per a les validacions del servei web.

Alguns conceptes clau relacionats amb el control d'accés al servidor són:

- **Autenticació:** el procés d'autenticació és el que determina si un usuari és qui diu ser. En cap moment governa quins drets té, què pot fer i què no, simplement s'encarrega de comprovar que l'usuari és qui diu que és. Per implementar l'autenticació hi ha innumerables sistemes, des dels fitxers d'usuaris i contrasenyes fins a sofisticats mecanismes d'empremtes dactilars, òptiques, dades biomètriques o llapis USB (sense el llapis l'usuari no es pot identificar).
- **Autorització:** un cop s'ha identificat un usuari (és qui diu ser), què pot fer?, a quins recursos pot accedir?, a quins no? Això és l'autorització: determinar els drets d'utilització dels recursos.
 - **Recurs amb accés restringit:** el control d'accés al servidor busca determinar quins recursos són accessibles per a quins usuaris. Pot restringir l'accés a tota una seu web de manera que només els usuaris autoritzats puguin accedir als seus continguts. Sovint es restringeixen àrees concretes de la seu web, per exemple directoris que són accessibles només per a un conjunt d'usuaris (els empleats, o els professors, en el web de l'escola). En aquest cas parlem de directoris amb accés restringit.
- **Reialme:** en una seu web hi poden haver diverses àrees restringides a perfils d'usuari diferents. Els reialmes permeten definir quines àrees restringides comparteixen el mateix grau d'accés.

Exemple de seu web d'una escola

Tornem a l'exemple d'una seu web d'una escola on hi ha tot de continguts públics accessibles per a tothom. El directori Notes és un recurs restringit on només hi poden accedir els alumnes de l'escola. Els directoris Programacions i Registres de Treball són accessibles només per als professors. Un professor que, per exemple, s'autentica per entrar a l'àrea Programacions introduint el seu identificador d'usuari i contrasenya, si vol entrar a l'àrea Registres de Treball s'hauria de tornar a identificar entrant de nou l'usuari i la contrasenya. Els reialmes permeten declarar que diversos llocs restringits tenen el mateix nivell d'accés, de manera que si un usuari s'ha autenticat en un està autenticat en tots els recursos que formen part del reialme.

- **Web amb inici de nom d'usuari/contrasenya:** un error molt típic és confondre l'autenticació de servidor amb l'autenticació de programari que fan les seus webs. Quan un usuari es valida en un entorn web com per exemple Yahoo o Google, no està usant l'autenticació amb el servidor

web. Està usant un usuari i una contrasenya de l'empresa web a la qual es connecta i la gestió d'aquesta sessió d'usuari per consultar el seu correu es fa mitjançant la programació en les mateixes pàgines web que visita. Això no té res a veure amb el control d'accés al servidor que es tracta en aquest apartat.

Autenticar els usuaris és determinar de forma veraç si un usuari és qui diu ser. **Autoritzar** és indicar quins usuaris tenen dret a accedir a quins recursos. Les seues web i els directoris que limiten l'accés a un conjunt restringit d'usuaris s'anomenen **recursos restringits**. Els recursos restringits que implementen la mateixa política de seguretat es poden agrupar en reialmes.

2.5.1 Els mòduls de control d'accés

Apache gestiona l'autenticació i el control d'accés al servidor mitjançant mòduls propis (i també es poden incorporar mòduls externs). Cada mòdul consta d'un conjunt de directives que permeten configurar el funcionament de l'autenticació i control d'accés implementats. Aquests mòduls es poden classificar en tres categories segons la seva funcionalitat:

1) Tipus d'autenticació: l'autenticació pot ser de tipus *basic* o *digest*. En aquests exemples s'utilitza autenticació bàsica. L'autenticació *digest* implica comunicacions xifrades. Aquests mòduls s'implementen amb la directiva *AuthType*.

```
1 AuthType Basic
```

Podeu observar que els mòduls identifiquen en el seu nom la cadena **auth** d'*authentication*.

```
1 mod_auth_basic
2 mod_auth_digest
```

2) Proveïdor d'autenticació: indica quin és el mecanisme usat per realitzar l'autenticació. Són els mòduls que permeten autenticar usant fitxers de contrasenyes o el mòdul PAM o el de LDAP. Es poden identificar els mòduls d'aquesta família perquè inclouen en el seu nom la cadena **authn** d'*authentication*.

```
1 mod_authn_file
2 mod_authn_alias
3 mod_authnz_ldap
4 ...
```

3) Autorització: els mòduls d'aquesta família proporcionen autorització d'usuari, de grups, del LDAP o del que convingui. Aquests mòduls es determinen segons el valor que prengui la directiva *require*.

```
1 Require user -validuser
```

Podeu observar que els mòduls identifiquen en el seu nom la cadena **authz** d'*authorization*.

```
1 mod_authz_user
2 mod_authz_group
3 mod_authz_owner
4 mod_authz_ldap
5 ...
```

2.5.2 Autenticació bàsica amb fitxers

El mecanisme més senzill per implementar l'autenticació en el servidor és l'autenticació bàsica amb fitxers d'usuaris i grups específics per al servidor web. Això es pot interpretar com un desavantatge perquè obliga a portar una gestió d'usuaris a més de la gestió d'usuaris del sistema. Però al mateix temps és un avantatge si el que volem és segregar aquets dos conjunts d'usuaris i administrar-los per separat.

Amb l'autenticació bàsica utilitzant fitxers es poden validar els usuaris utilitzant un **fitxer d'usuaris**, que conté els comptes d'usuaris i les seves contrasenyes.

També es poden validar grups d'usuaris amb un **fitxer de grups**, que indica quins usuaris formen part de cada grup.

El procés més simplificat per implementar la verificació d'usuaris i grups mitjançant fitxers de text pla amb contrasenyes requereix els passos següents:

1. Crear el fitxer d'usuaris en què s'indica la contrasenya corresponent a cada usuari.
2. Crear el fitxer de grups assignant a cada grup els usuaris que en formen part.
3. Identificar (o crear) el recurs que ha de tenir l'accés restringit.
4. Definir les directives apropiades per restringir l'accés al recurs als usuaris autoritzats.

L'exemple següent crearà un directori anomenat *m08* en el web www.iocdaw.cat, al qual només podran accedir els usuaris autoritzats (els professors).

Primer cal crear el fitxer d'usuaris. Es tracta d'un fitxer de text pla en el qual s'emmagatzemen l'identificador i la contrasenya, que pot ser en text pla o xifrada, de cada usuari. Per crear el fitxer i cada nou usuari s'utilitza l'ordre **htpasswd**, proporcionada pel paquet del servidor. En el primer exemple s'utilitza l'opció **-c**, que crea el fitxer de nou.

```
1 ioc@ioc:/var/www# htpasswd -c passwd/passwd alumne1
2 New password:
3 Re-type new password:
```

```

4 Adding password for user alumne1
5 ioc@ioc:/var/www# htpasswd passwd/passwd alumne2
6 ioc@ioc:/var/www# htpasswd passwd/passwd professor
7 ioc@ioc:/var/www# cat passwd/passwd
8 alumne1:$apr1$pEDPMqo8$ITpyGspQph.EjTvHpZIm1
9 alumne2:$apr1$zeHdfxng$iKxwbzVAGItP/2Fq8BniM.
10 professor:$apr1$Uje7o6eZ$B7c5ACGE8WH2bdZqzc0kz1

```

A continuació cal posar en cada grup (de moment no n'hi ha cap) els usuaris que n'han de formar part. De fet, és tan senzill com crear un fitxer de text pla cada línia del qual consta del nom del grup, el delimitador dos punts (:) i la llista d'usuaris separats per espais. Podeu observar que la usuària *anna* està en tots dos grups.

```

1 ioc@ioc:/var/www# cat passwd/group
2 alumnes: alumne1 alumne2
3 profes: professor

```

Ara cal generar el **recurs restringit**, l'accés al qual només es permetrà als usuaris autoritzats. En aquest cas serà un directori anomenat *m08* en el web www.iocdaw.cat.

```

1 ioc@ioc:/var/www# mkdir m08
2 ioc@ioc:/var/www# nano m08/index.html
3 ... crear una pàgina ....

```

Finalment s'ha d'assignar al directori local les directives apropiades per convertir-lo en un recurs d'accés restringit. Caldrà modificar el fitxer de configuració global `httpd.conf` i definir un bloc de configuració usant la directiva **Directory**. En aquesta directiva cal indicar la ruta absoluta corresponent al sistema de fitxers real del servidor (no és possible usar rutes relatives al servei web).

```

1 <Directory --pathabsolutfilesystem>
2 ... opcions de configuració ...
3 </Directory>

```

Un exemple complet de configuració és el que es mostra a continuació, en el qual únicament es permet accedir al recurs a usuaris del grup dels professors anomenat *profes*:

```

1 <Directory /var/www/m08>
2   AuthType Basic
3   AuthName "Restringit a professors"
4   AuthBasicProvider file
5   AuthUserFile /var/www/passwd/passwd
6   AuthGroupFile /var/www/passwd/group
7   Require group profes
8 </Directory>

```

Repassem les directives que s'hi utilitzen:

AuthType: indica que el tipus d'autenticació és bàsica (en lloc de *digest*).

AuthName: declara el reialme al qual pertany el recurs restringit. Això permet que si hi ha altres recursos restringits associats a aquest reialme l'usuari que ja s'ha autenticat en un d'ells no ho hagi de fer en els altres. El nom del reialme el posa l'administrador web.

AuthBasicProvider: indica el mètode d'autenticació que s'ha d'usar. Pot prendre valors tipus *ldap*, *pam*, *dbm*, *bdb*, *file* i d'altres. El valor *file* significa que s'utilitzarà un fitxer d'usuaris i opcionalment un de grups.

AuthUserFile: indica quin és el fitxer que conté els comptes dels usuaris locals del servidor Apache. És el fitxer que s'ha creat en l'exemple anterior.

AuthGroupFile: indica quin és el fitxer de grups en el qual consta quins grups d'usuaris hi ha i quins usuaris pertanyen a cada grup.

Require: aquesta directiva és la que determina quina és l'autorització que s'ha de fer. En l'exemple es permet que qualsevol usuari del grup *profes* tingui accés al recurs.

Finalment, cal verificar que l'accés al directori local és concedit únicament als membres del grup *profes*. Evidentment el mecanisme més senzill és verificar des d'un navegador l'accés al recurs www.iocdaw.cat/m08 i observar que es demana l'autenticació.

Exemples de mecanismes d'autorització

La directiva *require* és la que defineix l'autorització d'accés al recurs, és a dir, qui pot accedir-hi. Aquests en són alguns exemples d'ús:

1. *Require user valid-user:* permet l'accés a qualsevol usuari autenticat.
2. *Require user alumne1 alumne2:* permet l'accés als usuaris indicats (*alumne1* i *alumne2*).
3. *Require group profes alumnes:* permet l'accés als usuaris que són membres d'algun dels grups indicats.

2.6 El protocol HTTPS

El protocol HTTP pateix els mateixos problemes de seguretat que els seus companys dels inicis d'internet (FTP, TFTP, SMTP...). Tota la informació viatja en text net i és fàcilment monitorable per altres. Quan un usuari es connecta a un web i indica l'usuari i la contrasenya, aquestes dades viatgen sense cap mena de protecció i qualsevol les pot capturar. Si el que es transmet són dades bancàries, llistes d'amistats íntimes o qualsevol tipus de dada privada, és desaconsellable fer-ho per HTTP.

El primer mecanisme de seguretat que es va implementar per a HTTP va ser el protocol SSL (*Secure Socket Layer* o capa de sòcol segur), desenvolupat per Netscape. L'SSL proporciona una capa entre la capa de transport TCP i la capa d'aplicació HTTP en què les dades viatgen xifrades. L'HTTPS solament és un esquema URI que indica la utilització d'HTML més algun mecanisme de transport xifrat, com SSL o TLS.

Quan s'utilitza **HTTP amb un protocol xifrat** com SSL o TLS s'anomena **HTTPS** (*secure HTTP*). Utilitza el port 443.

El protocol SSL es va enviar a l'IETF (Internet Engineering Task Force o equip d'enginyeria d'internet, l'òrgan rector d'internet) per a l'estandardització i, després de diversos canvis, va sorgir el protocol TLS (*Transport Layer Security*, seguretat de capa de transport). El TLS proporciona les mateixes condicions de confidencialitat i autenticació en les transmissions HTTP que SSL.

Un dels avantatges de l'HTTPS és que permet la confidencialitat entre tots dos extrems de la comunicació, encara que només sigui un dels extrems el que s'ha autenticat. Aquest model és molt pràctic quan, per exemple, un client anònim compra en un web autenticat. Quan es volen pagar els bitllets d'avió, interessa que les dades de la targeta de crèdit viatgin xifrades i que el receptor sigui la companyia aèria i no un web fals.

L'ús dels certificats no és exclusiu per autenticar el servidor. Si cal, els clients poden ser autenticats. Per exemple, un web pot requerir que els clients disposin del certificat que els atorga dret a accedir-hi (expedit, per exemple, per la mateixa entitat).

Els passos necessaris per implementar comunicacions segures que permeten a un navegador client (o un client, sigui qui sigui) connectar-se via HTTPS a una seu web són:

- **Certificats digitals:** el servidor web ha de disposar d'una clau privada i d'un certificat digital.
- **Mòdul *mod_ssl*:** cal tenir instal·lat el paquet de programari que proporciona les prestacions SSL al servidor i que la configuració activa en carregui els mòduls pertinents.
- **Configuració de la seu web segura:** finalment, cal establir les directives SSL apropiades a la seu web que es vol configurar per fer-la accessible via SSL.

2.7 Certificats. Servidors de certificats

L'objectiu de les explicacions següents és implementar connexions segures HTTPS al web www.iocdaw.cat utilitzant SSL com a mecanisme de transport xifrat.

Suposarem que el servidor disposa ja d'una clau privada i d'un certificat, amb independència de com s'hagin obtingut. En concret, en el subdirectori certs del directori base del servei web hi ha:

- **server.crt**: el fitxer corresponent al certificat o clau pública del servidor. Aquest fitxer assegura als clients que es connecten a la seu web que el servidor és qui realment diu ser.
- **server.key**: és el fitxer amb la clau privada del servidor. Aquest fitxer s'ha codificat amb una *passphrase* o frase de pas de manera que cada vegada que s'inicialitzi el servidor web caldrà entrar aquesta frase.

Cal recordar que els navegadors clients validaran la confiança que els mereix el certificat contrastant el seu emissor amb la llista d'entitats certificadores que tenen carregada. Si l'emissor del certificat no hi és, caldrà fer passos per incorporar el certificat al navegador.

Aquests passos poden ser:

- Admetre el certificat com a vàlid quan el navegador presenta l'excepció de seguretat.
- Obtenir el certificat de l'entitat CA (*certification authority* o autoritat de certificació) que l'ha generat i incorporar l'entitat a la llista d'entitats en què el navegador confia.

Generar un certificat autosignat

A mode de recordatori ràpid, es pot generar una clau privada i un certificat autosignat fent:

```
1 # openssl req new x509 nodes out server.crt keyout server.key
```

2.7.1 Configuració d'Apache per usar SSL

El servidor web podrà usar SSL si disposa dels mòduls que en proporcionen la capacitat. En cas de no tenir-los, cal buscar en els repositoris de programari habitual un paquet que proporcioni el mòdul apropiat, instal·lar-lo i examinar-ne el contingut. Usualment, tant el paquet com el mòdul que proporcionen les prestacions de trànsit segur SSL s'anomenen *mod_ssl*.

```
1 # Buscar el paquet mod_ssl i instal·lar lo.
2 ioc@ioc:/# apt-cache search mod_ssl
3 libapache2-mod-gnutls - Apache module for SSL and TLS encryption with GnuTLS
4 libapache2-mod-nss - NSS-based SSL module for Apache2
5 python-mod-pywebsocket - WebSocket extension for Apache HTTP Server
6
7 ioc@ioc:/# apt-get install libapache2-mod-gnutls
8
9 ioc@ioc:/# a2enmod ssl
10 ioc@ioc:/# service apache2 restart
```

Com podeu veure, el paquet conté, entre d'altres, un fitxer de configuració específic anomenat *ssl.conf* i un únic mòdul anomenat *mod_ssl*. Tant l'un com l'altre estan en el directori *mods-available*. El fitxer de configuració específic del mòdul SSL conté tot de directives que configuren el funcionament global del

trànsit SSL. Com diu Apache, val més no tocar res. Es pot observar que el mòdul està carregat a la carpeta *mods-enabled*:

```
1 ioc@ioc:/# ls -l /etc/apache2/mods-enabled/ssl.*
2 lrwxrwxrwx 1 root root 26 abr 15 10:38 /etc/apache2/mods-enabled/ssl.conf ->
  ../mods-available/ssl.conf
3 lrwxrwxrwx 1 root root 26 abr 15 10:38 /etc/apache2/mods-enabled/ssl.load ->
  ../mods-available/ssl.load
```

2.7.2 Configuració de la seu web amb SSL

Finalment cal aplicar a la seu web *www.iocdaw.cat* les directives SSL apropiades per fer possible l'accés a aquesta seu web per HTTPS. El llistat de la directiva *VirtualHost* és:

```
1 <VirtualHost www.iocdaw.cat:443>
2   ServerAdmin admin@ www.iocdaw.cat
3   DocumentRoot /var/www/m08
4   ServerName www.iocdaw.cat
5   ErrorLog logs/m08 error_log
6   CustomLog logs/m08 access_log common SSLEngine On
7   SSLProtocol all SHAv2
8   SSLCertificateKeyFile /var/www/certs/server.key
9   SSLCertificateFile /var/www/certs/server.crt
10  #SSLCACertificateFile /var/www/certs/ca.crt
11 </VirtualHost>
```

Les directives usades són:

- **Port 443:** aquest és el port usual per a les connexions segures HTTP. Si la seu web només escolta per aquest port només es podrà accedir al seu contingut per HTTPS. Si es volen seus diferents per al trànsit xifrat i per al no xifrat n'hi ha prou de crear una altra seu virtual amb un altre port.
- **SSLEngine On:** aquesta directiva indica que cal activar el trànsit SSL per a aquesta seu web.
- **SSLProtocolall-SHAv2:** en aquesta directiva s'indiquen quins protocols es poden usar per generar el trànsit xifrat. Les opcions *all* i *-SHAv2* indiquen que s'accepten tots els protocols vàlids excepte el protocol SHA versió 2.
- **SSLCertificateKeyFile <clau privada del servidor>:** aquesta directiva indica el fitxer amb la clau privada del servidor.
- **SSLCertificateFile <certificat>:** indica quin és el fitxer que conté el certificat del servidor. Aquest és el certificat que els navegadors clients veuran i del qual hauran de decidir si hi confien o no.
- **SSLCACertificateFile <certificat-CA>:** aquesta directiva és opcional i permet indicar quin és el fitxer que conté el certificat públic que ha emès l'entitat de certificació CA. Recapitem, si el certificat del servidor ha estat emès per una entitat externa (per exemple, VeritatAbsoluta), aquesta

directiva permet que el client obtingui el certificat de l'entitat emissora, estalviant-li la cerca. Ara bé, encara falta que el navegador client confiï en aquesta entitat.

Un cop configurada apropiadament la seu virtual, cal posar de nou en funcionament el servei web (moment en què es demanarà la frase de pas del servidor per a la clau privada de `www.iocdaw.cat`). Des de qualsevol navegador s'ha de poder accedir a la seu usant HTTPS. Ara bé, es generarà una excepció de seguretat perquè el navegador desconeix la procedència del certificat. Si l'usuari accepta confiar en la seu web, el certificat s'incorporarà al navegador i accedirà de forma xifrada a la seu web.

Una altra opció és carregar prèviament en el navegador el certificat de l'entitat CA VeritatAbsoluta, que és qui ha actuat en l'exemple com a entitat certificadora local. Si això es fa **abans** de contactar amb la seu web, quan el navegador hi accedeixi per HTTPS ja no es produirà una excepció de seguretat. Com que el certificat del servidor està signat per una entitat en la qual el navegador confia (forma part de la seva llista de *trusted CAs*) s'acceptarà automàticament.

2.7.3 Verificació de les connexions SSL

Els problemes principals que es poden trobar els navegadors en connectar amb seus web amb certificats són:

- Amb un certificat autosignat no cal definir cap CA. El navegador client mostrarà la típica pantalla d'excepció de seguretat i caldrà indicar que s'accepta el certificat de servidor per a l'entitat `www.iocdaw.cat`. És un certificat emès per la mateixa entitat.
- Amb un certificat generat per una CA local cal incorporar manualment el certificat al navegador. Un cop fet això el navegador serà capaç de validar el certificat del servidor amb la CA que l'ha expedit (*issuer*). En el nostre exemple, el certificat del servidor l'expedeix la CA VeritatAbsoluta.

A més dels navegadors, hi ha eines d'entorn de text per verificar connexions SSL, de la mateixa manera que s'utilitza *telnet host 80* per verificar connexions HTTP. D'una banda, es pot usar el mateix **OpenSSL** i, de l'altra, es pot instal·lar la utilitat **Curl**, que permet fer un ampli seguiment del diàleg SSL.

```
1 OpenSSL> s_client -connect www.iocdaw.cat:443 -state -debug
2 curl https://www.iocdaw.cat -kv
```

3. Instal·lació i administració de servidors de transferència de fitxers

Les aplicacions de transferència de fitxers van ser una de les primeres eines en desenvolupar-se en l'expansió de les xarxes d'internet. La necessitat de poder accedir a diferents sistemes i intercanviar informació va originar un dels sistemes que actualment es fan servir.

Actualment hi ha diferents formes d'intercanvi d'informació de forma distribuïda en format fitxer:

- Sistemes de fitxers en xarxes
- Programari de missatgeria
- Programari de distribucions de fitxers P2P (*peer-to-peer*)

P2P

El P2P és un concepte de xarxes de computadors referent a la transmissió entre dos equips dins la xarxa. Concepte aplicat en les aplicacions com Napster, eMule, Bittorrent, en la transmissió de fitxers en les xarxes d'internet entre clients.

Windows 10 fa servir tecnologia P2P per realitzar la distribució de les actualitzacions del sistema operatiu dins d'una xarxa local.

L'**FTP** (*file transfer protocol*) o **protocol de transferència de fitxers** és un protocol que proporciona el servei de transferència de fitxers entre sistemes de diferent naturalesa, és a dir, es poden interconnectar clients de Linux cap a un sistema de Microsoft o d'altres.

La implementació de l'FTP es remunta a l'any 1971, quan es va desenvolupar un sistema de transferència de fitxers, definit dins la **RFC** (*request for comments*) **141**, entre equips de l'Institut Tecnològic de Massachusetts (MIT, Massachusetts Institute of Technology). Durant els anys posteriors es van fer diferents innovacions al protocol bàsic, que es van incloure l'any 1973.

El protocol FTP, tal com es coneix actualment com a estàndard, s'especifica dins la RFC 959 l'any 1985 i defineix el funcionament del protocol. Posteriorment, el protocol FTP s'ha anat revisant amb algunes noves característiques, però la seva base de funcionament ha estat mantinguda.

RFC

Els *request for comments* o **documents RFC** són una sèrie de publicacions que descriuen diversos aspectes del funcionament d'internet, xarxes de computadors, protocols i procediments. La seva creació, per part de Steve Crocker l'any 1969, estava destinada al registre dels dissenys del grup de treball de xarxa per a ARPANET. El registre s'efectua basat en un esquema per realitzar el contingut i usant text en format ASCII (American Standard Code for Information Interchange).

El protocol FTP es basa en l'arquitectura client/servidor i fa ús del protocol de control de transport, TCP (*transport control protocol*) per realitzar el canal de

Per obtenir més informació sobre l'especificació del protocol FTP en la RFC 959, aneu a la secció "Annexos" del web d'aquest mòdul.

Protocol d'internet

Conjunt de regles de comunicació de xarxa en què es basa internet i que, a més, faciliten l'intercanvi de dades entre ordinadors connectats a la xarxa.

transmissió entre el client i el servidor, amb la garantia que la informació que s'envia o es llegeix arribarà al seu destí.

TCP

El TCP (*transmission control protocol*) és un protocol de control de transmissió dissenyat l'any 1973 i 1974 per Vint Cerf i Robert Kahn que es defineix dins la capa de transport en la pila de protocols TCP/IP. Les especificacions es troben dins la RFC 793 i la RFC 1323.

Es fan servir dos canals de comunicació dins del protocol FTP, el canal de control i el canal de dades:

- El **canal de control** envia totes les ordres de comunicació, com poden ser iniciar la sessió de treball i ordres d'execució com llegir, escriure, llistar, esborrar, etc.
- El **canal de dades** envia el contingut d'aquells fitxers a treballar, que pot ser tant per llegir el contingut del fitxer com per fer l'escriptura del fitxer.

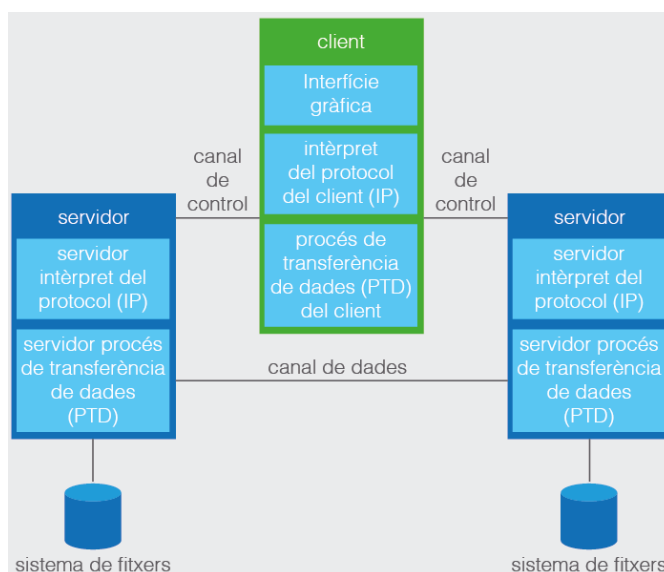
No confongueu les sigles IP (intèrpret de protocol) amb l'Internet Protocol.

Tant el client com el servidor gestionen dos processos:

- **PTD** (procés de transferència de dades): és l'encarregat d'establir la connexió i administrar el canal de dades. Tant el client com el servidor tenen el seu propi PTD.
- **IP** (intèrpret del protocol): interpreta el protocol i permet que el PTD pugui ser controlat mitjançant ordres rebudes pel canal de control.

L'IP és diferent en el client i servidor, cadascú s'encarrega d'unes funcions específiques. A la figura 3.1 podeu observar l'estructura de funcionament.

FIGURA 3.1. Gràfic del funcionament del client i servidor FTP juntament amb el procés IP i PTD



L'IP del servidor:

- Escolta les ordres que provenen de l'IP de l'usuari mitjançant el canal de control per un port de dades.
- Estableix la connexió del canal de control.
- Rep les ordres FTP de l'IP de l'usuari, les respon i executa al PTD del servidor.

L'IP del client:

- És el responsable d'establir la connexió amb el servidor FTP.
- Envia ordres FTP.
- Rep les respostes del servidor IP.
- Controla el PTD de l'usuari.

Quan un client connecta al servidor FTP, l'IP de l'usuari inicia la connexió amb el servidor amb el protocol Telnet (RFC 854).

Telnet

Telnet és un protocol que permet connectar-se a sistemes remots anomenats *hosts* fent servir la xarxa TCP/IP (xarxes LAN o internet). Mitjançant un programa anomenat client de Telnet, es pot fer una connexió a un servidor Telnet remot.

Una vegada iniciada la comunicació amb el servidor Telnet, s'obté un terminal virtual que permet la comunicació amb el servidor Telnet o *host* remot.

Per poder accedir a un *host* remot es necessita un usuari i una contrasenya que ha de ser dins del sistema remot.

L'especificació del protocol Telnet es troba dins la RFC 854.

El client envia ordres FTP al servidor, el servidor les interpreta, executa el PTD i respon amb un format estàndard. Una vegada establerta la connexió, l'IP del servidor proporciona el port pel qual s'enviaran les dades al PTD del client, per on escoltarà i rebirà les dades del servidor. El sistema d'emmagatzematge dels fitxers dependrà del sistema on sigui i el seu sistema de fitxers. Per exemple, sistemes de fitxers ext4 per a Unix o NTFS per a Microsoft Windows.

Tota la comunicació que es fa en el canal de control segueix les recomanacions del protocol Telnet. Les ordres FTP són cadenes de caràcters Telnet amb format NVT-ASCII (*Network Virtual Terminal-American Standard Code for Information Interchange*) per on s'envien les ordres de l'FTP.

NVT-ASCII

Sistema estàndard definit dins del sistema Telnet per a l'enviament d'ordres i dades fent servir el sistema de codificació de caràcters ASCII.

Les ordres FTP permeten especificar:

- El port que es farà servir.
- El mètode de transferència de dades.
- L'estructura de dades.
- L'acció que es durà a terme (llegir, eliminar, llistar, emmagatzemar, etc.).

Hi ha tres distribucions d'ordres FTP:

- **Ordres de control d'accés:** especifiquen els identificadors del control d'accés. La majoria d'aquests controlen qui accedeix al servidor FTP i quins privilegis tindrà l'usuari.
- **Ordres de paràmetres de transferència:** tenen un valor per defecte del servidor FTP, i només es fa ús d'aquests paràmetres si han estat modificades.
- **Ordres de servei FTP:** són les ordres més usades. Defineixen la transferència de fitxers i la navegació dels directoris remots per a l'usuari. L'argument principal de les ordres de servei sol ser el nom d'un directori. Totes les dades que s'envien per a una ordre de servei sempre s'envien pel canal de dades.

3.1 Configuració del servei de transferència de fitxers. Permisos i quotes

Actualment hi ha moltes aplicacions que implementen el protocol FTP tant per la banda del client com per la del servidor. D'aquestes implementacions del protocol FTP n'hi ha de font pública i que es poden baixar gratuïtament en sistemes propietaris o lliures. La decisió d'una aplicació o una altra que implementi el protocol FTP ve donada per les possibilitats que ofereix i el sistema de treball on s'exerceix la feina. En el cas del desplegament d'aplicacions web, qualsevol servidor o client FTP s'ajusta a les necessitats del desplegament web.

Per defecte la majoria de sistemes operatius porten un client FTP gràfic o terminal, per poder accedir remotament als servidors externs.

3.1.1 Configuració del servei de transferència de fitxers

En aquesta unitat fareu servir el servidor FTP anomenat **ProFTPD** (*short for PRO FTP Daemon*). ProFTPD és un servidor FTP amb llicència GPL (*general public license*) per a Linux que permet fer una customització del seu funcionament depenent de les necessitats de configuració de l'entorn de treball per part de l'administrador.

Les característiques principals d'aquest programari són:

- El seu sistema de configuració es basa en un fitxer de configuració amb directrius intuïtives molt semblants a les que podeu haver fet en les configuracions dins del servidor Apache Web Server.
- Permet la configuració de servidors virtuals.
- Permet l'execució del servei com a servidor independent (*standalone*) o *inetd*.

Per a més detall del funcionament del servei Apache Web server, dirigiu-vos al punt 2.2. d'aquesta unitat, "Configuració avançada del servidor web".

- Permet mantenir l'arrel del directori anònim.
- El codi font està disponible per als desenvolupadors.
- Permet amagar els fitxers i directoris, basant-se en els permisos que fa servir Linux.

Dimonis 'standalone' i 'inetd'

El dimoni *standalone* escolta les peticions del servei i llança diferents processos per tractar-les. Es fa servir per a trànsits elevats de dades i usuaris. El servei sempre està actiu.

El dimoni *inetd* escolta les peticions del servei dels ports. Si detecta comunicació en el port configurat, inicia el servei passant-li la connexió. Es fa servir en situacions de poc trànsit.

Hi ha una gran varietat de ProFTPD en distribucions basades en Unix excepte plataformes Microsoft, encara que és possible que es pugui executar gràcies a l'acord Ubuntu Microsoft, que permet executar el Terminal Bash al sistema Windows 10 des de l'any 2016, amb el programa desenvolupador.

3.2 Configuració de ProFTPD

El sistema on es fa la instal·lació i configuració està basat en un sistema Debian (Ubuntu o Lubuntu). El conjunt d'instruccions d'instal·lació o configuració mitjançant ordres de sistema en són dependents (si anem a un sistema REDHAT, CENTOS, etc., haureu de mirar el procediment d'instal·lació i les ordres específiques del sistema).

La configuració interna és independent del sistema operatiu usat, qualsevol configuració que feu durant aquests apartats és compatible amb altres sistemes operatius basats en Unix.

Per iniciar el procés d'instal·lació de ProFTPD, primer de tot obriu un terminal i executeu l'ordre d'actualització dels repositoris d'Ubuntu:

```
1 sudo apt-get update
```

APT

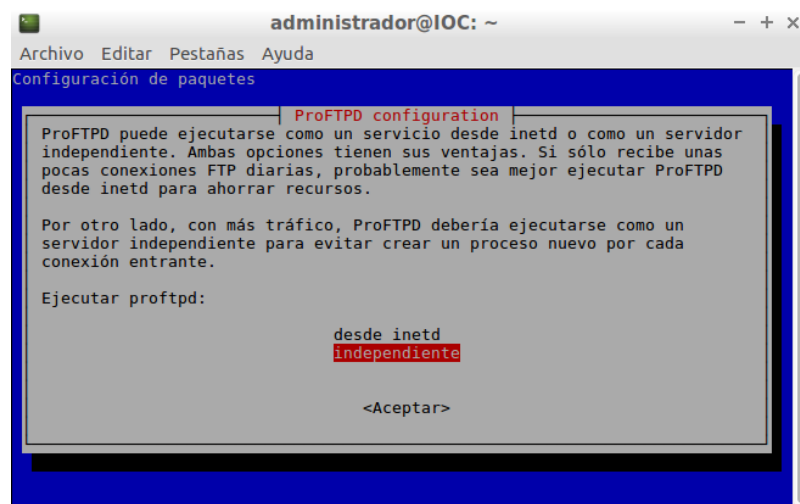
APT (*Advanced Package Tool*) és un programari gratuït dins de Debian que funciona amb les llibreries del nucli, per tractar les instal·lacions, configuracions i eliminar programari.

APT permet de forma fàcil agafar la versió més nova d'un paquet d'instal·lació referent a un programari i aplicar les configuracions necessàries depenent de la versió del sistema.

```
1 sudo apt-get install proftpd
```

Una vegada accepteu la instal·lació, el procés APT instal·la proFTPD i us fa escollir entre dues opcions referents a la modalitat d'inici del servei.

A la figura 3.2 podeu veure la finestra amb les dues opcions. Per defecte, escolliu proFTPD com a procés independent (*standalone*).

FIGURA 3.2. Opcions de modalitat d'inici de servei

El fitxer `/etc/shadow` conté els noms dels usuaris del sistema on treballem.

Una vegada finalitzada la instal·lació, procediu a verificar el resultat d'executar unes ordres. Executeu l'ordre següent:

```
1 sudo cat /etc/shadow
```

L'ordre que acabeu d'executar llista tots els usuaris del sistema on treballem. Si us hi fixeu bé trobareu una línia anomenada ftp i proftpd. Aquestes línies fan referència a dos usuaris que ha creat el procés de configuració inicial de proFTPD. Els usuaris ftp i proftpd són usuaris d'accés per realitzar accessos anònims.

Podeu comprovar el mateix resultat amb les ordres següents:

```
1 sudo cat /etc/shadow | grep proftpd
2 sudo cat /etc/shadow | grep ftp
```

O amb una ordre única:

```
1 sudo cat /etc/shadow | grep -E "proftpd|ftp"
```

Vegeu el resultat d'aquesta ordre dins la :figura 3.3.

FIGURA 3.3. Usuaris ProFTPD i FTP dins de `/etc/shadow`

A continuació comproveu l'estat dels processos realitzats amb ProFTPD:

```
1 sudo ps -ef | grep proftpd
```

Us sortirà la següent sortida per terminal:

```
1 root@I0C:/etc/proftpd# ps -ef | grep proftpd
2 proftpd 35075 1 0 10:01 ? 00:00:00 proftpd: (accepting
   connections)
3 root 36023 11190 0 11:18 pts/1 00:00:00 grep --color=auto proftpd
```

Veureu que hi ha un procés anomenat proftpd i que està acceptant les connexions. A continuació, executeu l'ordre:

```
1 sudo netstat -ltn
```

Aquesta ordre us mostrarà aquells processos que estan escoltant el port 21 (port del servei FTP).

La línia que us interessa reconèixer és l'última, ja que confirma que hi ha un procés que està escoltant el port 21, que és el que es fa servir dins del protocol FTP.

Per a més detall, vegeu l'apartat 3.7.1 "Mode FTP actiu".

```
1 root@I0C:/etc/proftpd# netstat -ltn
2 Conexiones activas de Internet (solo servidores)
3 Proto Recib Enviad Dirección local Dirección remota Estado
4 tcp 0 0 0.0.0.0:10000 0.0.0.0:* ESCUCHAR
5 tcp 0 0 127.0.1.1:53 0.0.0.0:* ESCUCHAR
6 tcp6 0 0 :::80 :::* ESCUCHAR
7 tcp6 0 0 :::21 :::* ESCUCHAR
```

Una vegada heu verificat els processos i els ports de proFTPD, la configuració del servei FTP es realitzarà mitjançant un fitxer de text pla, tal com se sol fer amb molts serveis d'Unix, com per exemple Apache Web Server.

El directori on es troben les configuracions són dins de */etc/proftpd/* i el fitxer de configuració principal és el fitxer *proftpd.conf*.

Editar el fitxer original pot comportar haver de modificar el fitxer i perdre les dades inicials de configuració. Sempre és bo fer una **còpia de seguretat** del fitxer. Amb l'ordre *cp* (per copiar fitxers) o navegant al directori */usr/share/proftpd/templates/* teniu les còpies dels fitxers originals, en cas de voler recuperar les configuracions inicials.

Navegueu fins al directori de configuracions de ProFTPD */etc/proftpd* i executeu:

```
1 ls -l
```

Amb aquesta ordre llistareu tot el contingut del directori, amb el resultat següent:

```
1 root@I0C:/etc/proftpd# ls -l
2 total 1324
3 -rw-r--r-- 1 root root 1310700 abr 5 2016 blacklist.dat
4 drwxr-xr-x 2 root root 4096 abr 5 2016 conf.d
5 -rw-r--r-- 1 root root 9420 abr 5 2016 dhparams.pem
6 -rw-r--r-- 1 root root 701 ene 31 16:26 ldap.conf
```

```

7 -rw-r--r-- 1 root root 2882 ene 31 17:54 modules.conf
8 -rw-r--r-- 1 root root 5356 ene 31 17:29 proftpd.conf
9 -rw-r--r-- 1 root root 862 ene 31 16:26 sql.conf
10 -rw-r--r-- 1 root root 2082 ene 31 16:26 tls.conf
11 -rw-r--r-- 1 root root 832 ene 31 16:26 virtuals.conf

```

LDAP

Servei de directori organitzat que emmagatzema dades per ser consultades en un entorn de xarxes. Per a més informació, cerqueu a internet o dirigiu-vos l'apartat d'aquesta unitat "Servidors web i d'aplicacions".

Com podeu veure, el contingut del directori */etc/proftpd* conté diferents fitxers, inclòs el *proftpd.conf*. Els fitxers que hi ha dins fan referència a fitxers de configuracions de modularitats que amplien la capacitat de ProFTPD, com poden ser vinculacions de dades amb SQL, configuracions de seguretat per xifrar la informació, creació de servidors virtuals, vinculació amb el servei de directori LDAP (*Lightweight Directory Access Protocol*), etc.

Inicieu l'edició del fitxer *proftd.conf*. Dins del directori feu:

```
1 sudo nano proftpd.conf
```

En les següents ordres dins d'aquest apartat es fa servir l'editor de text Nano, però podeu editar els fitxers de configuracions amb diferents editors dins del terminal o gràfics.

Abans de detallar configuracions, fixeu-vos en el fitxer *proftpd.conf*. Moltes de les línies del fitxer de configuració estan iniciades amb el caràcter # (*hashtag* o etiqueta). El caràcter # es fa servir en múltiples configuracions de serveis del sistema i permet comentar línies de configuració i fer comentaris. L'interpret de l'aplicació no els executarà.

Com podeu comprovar, el fitxer de configuració *proftpd.conf* és bastant extens. Tot seguit es detallen les opcions de configuració més rellevants.

Dins de la configuració inicial, les directives més importants són:

- **Servername:** defineix el nom del servidor que mostrarà, en aquest cas "Debian". Podeu canviar el nom per un que identifiqui el nostre servei.
- **TimeoutIdle:** defineix el temps que un usuari pot estar connectat sense fer cap acció.
- **TimeoutStalled:** defineix el temps que una connexió de dades pot estar aturada.
- **DisplayLogin:** defineix el fitxer de text on es mostrarà el missatge de benvinguda.
- **DisplayChdir:** defineix el fitxer de missatge que es mostrarà a cada canvi de navegació de directori.
- **ListOptions:** defineix l'ordre "-l" per a llistar els directoris.
- **DefaultRoot:** encapsula els usuaris en els directoris *home* de cada usuari.
- **RequireValidShell:** si el valor és *on* obliga que els usuaris de sistema tinguin definit un *shell* vàlid a la seva configuració.
- **Port:** defineix el número de port que es farà servir per a les connexions de control, per defecte el 21.
- **Umask:** màscara de permisos que tindrà per defecte.

ProFTPD té una gran extensió de directives i per la seva extensió no es pot detallar tot.

- ***AllowOverWrite***: permet sobre escriure el fitxer si el valor està a `\\on\\`.
- ***QuotaEngine***: permet activar el motor de quotes del servidor FTP.

Aquestes directives estan actives sense el símbol *tag* i funcionen per defecte en el servidor.

Les següents directives amb el símbol *tag*:

- ***#Include /etc/proftpd/virtuals.conf***: fitxer on s'habilitaran les configuracions de servidors virtuals alternatius que es poden configurar.
- ***#Anonymous***: habilita el servidor anònim.
- ***#Include /etc/proftpd/tls.conf***: fitxer on es configurarà el servidor per suportar connexions segures.
- ***#Directory***: configura un directori específic amb una configuració específica, com pot ser habilitar accés als usuaris, permetre llegir, escriure, llistar, etc.

Vegeu a "Annexos" les directives ProFTPD, per a més detall de les directives FTP.

Per ampliar la teoria, vegeu l'apartat "Protocol de transferència de fitxers segur".

3.3 Permisos

Dins d'un servei FTP, un dels passos importants és el de concedir permisos determinats per controlar l'accés al servidor o als diferents directoris.

Dins de ProFTPD la directiva **<LIMIT>** permet fer les configuracions de permisos dins del servidor FTP.

La directiva **<LIMIT>** la podem configurar dins de les directives. Dins de la configuració general del servidor:

- **<VirtualHost>**
- **<Directory>**
- **<Anonymous>**
- **<Global>**

I els permisos generals que podem configurar són:

- **ALL** (tots els permisos excepte de LOGIN).
- **DIRS** (inclou **CDUP**, **CWD**, **LIST**, **MDTM**, **MLSD**, **MLST**, **NLST**, **PWD**, **RNFR**, **STAT**, **XCUP**, **XCWD**, **XPWD**)
- **LOGIN** (accés d'usuaris)
- **READ** (inclou **RETR**, **SIZE**)

Vegeu a "Annexos" les ordres i permisos de ProFTPD per a més detall dels permisos FTP.

- **WRITE** (inclou **APPE**, **DELE**, **MKD**, **RMD**, **RNTO**, **STOR**, **STOU**, **XMKD**, **XRMD**)

Vigileu quan gestioneu permisos de directoris de sistema amb CHMOD. Revisau pertinences de grups i permisos dels usuaris. Una màscara 770 ens donarà permís total (escriptura, lectura i execució) a propietaris i al grup, la 774 permet la lectura a la resta d'usuaris i la 777 donaria permisos a tothom, eviteu fer servir aquest últim. El principal receptor de la gestió d'aquests permisos és la directiva *<Directory>*.

Aquestes paraules clau (ALL, DIRS, LOGIN, READ, WRITE) permeten configuracions generals, però podem també configurar permisos específics que estan englobats dins les generals.

Altres directives útils que podeu fer servir dins de *<LIMIT>* són:

- **AllowUser nomUsuari**: dona el permís a un usuari específic.
- **DenyUser nomUsuari**: denega el permís a un usuari específic.
- **AllowAll**: dona el permís a tots els usuaris.
- **DenyAll**: denega el permís a tots els usuaris.

Vegeu alguns exemples de configuracions, que poden anar dins del fitxer *proftpd.conf*:

```

1 #permet entrar al servidor FTP a l'usuari 1 i 2 i denegar l'accés a la resta
2 <LIMIT LOGIN>
3     AllowUser usuari1
4     AllowUser usuari2
5     DenyAll
6 </LIMIT>
7
8 #modifiquem els permisos del directori /var/ftp/dades
9 <Directory /var/ftp/dades>
10     #es dona permisos de lectura a l'usuari 1 i 2 i es nega la lectura de fitxers
11     a la resta
12     <LIMIT READ>
13         AllowUser usuari1
14         AllowUser usuari2
15         DenyAll
16     </LIMIT>
17     #es dona permisos d'escriptura a l'usuari 1, es nega l'escriptura de fitxers
18     a l'usuari 2 i a la resta d'usuaris
19     <LIMIT WRITE>
20         AllowUser usuari1
21         DenyAll
22     </LIMIT>
23 </Directory>
```

Sempre que vulgueu configurar els permisos d'un directori de ProFTPD necessitareu crear la directiva *<Directory>* i indicar la configuració dels permisos que vulgueu editar.

3.4 Quotes

ProFTPD permet configurar quotes d'espai diferenciant entre quotes generals del servidor, quotes vinculades a usuaris o grups de treball.

Les quotes generals del servidor permeten configurar:

- Restringir la velocitat de pujada i de descàrrega dins del servidor FTP.
- Restringir el màxim d'espai d'emmagatzematge d'un fitxer al servidor FTP.
- Restringir el màxim de la mida del fitxer que podem descarregar del servidor.

Les quotes d'usuari o grups de treball permeten:

- Restringir la velocitat de pujada i de descàrrega de l'usuari o el grup de treball.
- Restringir el màxim d'espai d'emmagatzematge d'un fitxer per part de l'usuari o del grup de treball.
- Restringir el màxim de la mida del fitxer que pot descarregar l'usuari o el grup de treball.
- Restringir d'espai propi per emmagatzemar dades en el directori de configuració de l'usuari o del grup de treball.

3.4.1 Quotes generals del servidor ProFTPD

Per realitzar la configuració i limitar les quotes generals a tots els usuaris, cal afegir les directives.

Per configurar el màxim de fitxer d'emmagatzematge dins del servidor FTP:

```
1 #Unitats amb case-insensitive "Gb" (Gigabytes), "Mb" (Megabytes), "Kb" (
  Kilobytes), o "B" (bytes)
2 MaxStoreFileSize      20 Mb
```

Per configurar el màxim de descàrrega d'un fitxer del servidor FTP:

```
1 #Unitats amb case-insensitive "Gb" (Gigabytes), "Mb" (Megabytes), "Kb" (
  Kilobytes), o "B" (bytes)
2 MaxRetrieveFileSize   20 Mb
```

3.4.2 Quotes d'usuari o grups de treball a ProFTPD

ProFTPD fa servir dos tipus de configuracions de quotes dins de dos fitxers:

- **limit**: per fixar els màxims de descàrrega, pujada, ràtios, etc.
- **tally**: per portar el compte de la quantitat fins al moment.

Per realitzar les quotes d'usuari, cal fer servir les ordres de terminal que implementa ProFTPD per configurar les quotes.

Per crear el fitxer *limit* i *tally*, feu el següent dins del terminal Linux i dins del directori de treball on s'emmagatzemaran els fitxers:

```
1 cd /etc/proftpd
2 mkdir /etc/proftpd/quotes
3 cd /etc/proftpd/quotes
4 ftpquota --create-table --type=limit
5 ftpquota --create-table --type=tally
```

Aquestes ordres donen com a resultat els fitxers següents:

1. *ftpquota.limittab*
2. *ftpquota.tallytab*

Dins del fitxer de *proftpd.conf* afegiu les següents línies dins del servidor principal o, si esteu configurant un servidor virtual:

```
1 <IfModule mod_quotatab_file.c>
2   QuotaEngine on
3   #mostrar a l'usuari les unitats de la quota "b"|"Kb"|"Mb"|"Gb"
4   QuotaDisplayUnits Mb
5   # Permet fer l'ordre quote SITE QUOTA
6   QuotaShowQuotas on
7   QuotaLog /var/log/proftpd/quotatab.log
8   QuotaLimitTable file:/etc/proftpd/quotes/ftpquota.limittab
9   QuotaTallyTable file:/etc/proftpd/quotes/ftpquota.tallytab
10 </IfModule>
```

La instrucció `<IfModule mod_quotatab_file.c>` permet que s'executi el conjunt de configuracions que hi ha dins d'aquesta etiqueta, sempre que el `mod_quotatab_file.c` estigui habilitat.

Detall de les directives:

- ***QuotaEngine***: permet iniciar el sistema de quotes amb el valor *on*.
- ***QuotaDisplayUnits unitat***: permet mostrar l'estat de la quota en la unitat d'emmagatzematge que especifiquem, "*b*"|"Kb"|"Mb"|"Gb".
- ***QuotaShowQuotas***: habilitem l'ordre *SITE QUOTA* dins d'FTP. Permet mostrar l'estat de la quota i habilitar amb el valor *on*.
- ***QuotaLog***: permet configurar el directori de registres de Log.
- ***QuotaLimitTable* i *QuotaTallyTable***: especifica on es troben els fitxers de quota creats amb les ordres *ftpquota*.

Una vegada creats els fitxers de quotes i afegida la configuració bàsica al fitxer *proftpd.conf*, executeu les ordres per crear la quota a un usuari específic.

Executeu l'ordre següent a tants usuaris o grups com s'hagin de configurar:

```
1 ftpquota --add-record --type=limit --name=nomUsuari --quota-type=user --units=B
   --bytes-download=15728640
```

Per a més detall de l'ordre *ftpquota* i les opcions de quotes adreceu-vos a les configuracions de quotes a ProFTPD dels annexos.

El detall de l'ordre:

- **-add-record**: afegeix un registre al fitxer de quotes que s'especificarà.
- **-type**: especifica a quin fitxer anirà el registre, fitxer *limit* o *tally*.
- **-name**: especifica el nom de l'usuari o grup a qui és aplicable la quota.
- **-quota-type**: especifica si la quota és per usuari o grup, *user* o *group*.
- **-units**: unitats que es faran servir per calcular les unitats d'emmagatzematge i processament de dades, "B" o "byte", "Kb" o "kilo", "Mb" o "mega", i "Gb" o "giga". Per defecte, fa servir "byte" si no es fa servir la directiva *units*.
- **-bytes-download**: especifica el nombre màxim de bytes que s'han de descarregar de la quota en unitats d'emmagatzematge bytes.

Un cop finalitzada la configuració de quotes, en podeu veure l'estat tant per a les quotes límit com per a les quotes *tally*, amb:

```
1 ftpquota --show-records --type=limit
2 ftpquota --show-records --type=tally
```

Per eliminar les quotes dels fitxers *limit* o *tally* feu:

```
1 ftpquota --delete-record --type=limit --name=usuari --quota-type=user
```

El detall de l'ordre:

- **-delete-record**: elimina el registre del fitxer *limit* o *tally*.
- **-type**: especifica a quin fitxer eliminarà el registre, fitxer *limit* o *tally*.
- **-name**: especifica el nom de l'usuari o grup a qui és aplicable la quota.
- **-quota-type**: especifica si la quota és per usuari o grup, *user* o *group*.

Per actualitzar les quotes del fitxer *limit* o *tally* feu:

```
1 ftpquota --update-record --type=limit --name=usuari --quota-type=user --bytes-
  download=100 --files-download=1 --units=B
```

El detall de l'ordre:

- **-update-record**: actualitza el registre del fitxer *limit* o *tally* amb les configuracions noves.
- **-type**: especifica a quin fitxer actualitzarà el registre, fitxer *limit* o *tally*.
- **-name**: especifica el nom de l'usuari o grup a qui és aplicable la quota.
- **-quota-type**: especifica si la quota és per usuari o grup, *user* o *group*.
- **-bytes-download**: especifica el nombre màxim de bytes que s'han de descarregar de la quota en unitats d'emmagatzematge bytes.

- ***-files-download***: especifica el nombre de fitxers que es poden descarregar del servidor.
- ***-units***: unitats que es faran servir per calcular les unitats d'emmagatzematge i processament de dades (B o byte, Kb o quilo, Mb o mega i Gb o giga. Per defecte fa servir byte si no es fa servir la directiva *units*.

3.5 Tipus d'usuaris i accessos al servei

Hi ha dos tipus d'usuaris dins del servei FTP:

- **Usuari del sistema**: usuari propi del sistema on hi ha el servei FTP i que accedeix al seu directori personal.
- **Usuari anònim**: usuari que no té contrasenya de validació i d'accés públic al servei.

Els permisos es poden configurar per usuari i per grups de treball. Dirigiu-vos a l'apartat "Permisos" per a més detall.

D'aquest tipus d'usuaris podem especificar-ne un tercer que deriva dels usuaris de sistema, anomenats **usuaris virtuals**. La diferència ve donada perquè aquests no són dependents del sistema sinó del servidor FTP directament.

Tots els usuaris que tenen accés al servei FTP, com a administradors del servei, habilitareu permisos per poder manipular els fitxers i directoris del servei. Aquests permisos poden ser lectura, escriptura, llistar, eliminar, etc. Són els que permeten treballar amb els directoris d'accés realitzats a la configuració del servei i els que permet la definició del protocol.

Dins dels tipus d'usuaris diferenciarem també els tipus d'accessos al servei.

Accés anònim.

Els servidors poden oferir servei lliurement a tots els usuaris, accedir sense tenir un identificador d'usuari, llegir i navegar pel contingut dels directoris lliurement, indiferentment de què hi accedeix i del lloc on ho fa.

L'accés anònim és una forma còmoda de permetre que tots els clients tinguin accés a certa informació sense que l'administrador del servei hagi de controlar els comptes d'usuaris.

La informació que es treballa per a l'accés anònim és de caràcter públic i es poden llegir els continguts dels directoris però no eliminar-los ni modificar-los. Normalment, el contingut sol ser programari de domini públic o de lliure distribució, imatges, so, vídeos, etc.

Accés anònim

Exemples de servidors públics amb accés anònim: <ftp://ftp.rediris.es> i <ftp://cdimage.ubuntu.com>.

El requisit per accedir per accés anònim és mitjançant un nom predefinit que existeix en el servei FTP i que ha d'estar configurat prèviament.

Aquest usuari que permet l'accés anònim es diu *anonymous*. Quan es valida la connexió, el nom de l'usuari que posem és *anonymous*, i sense contrasenya (encara que demani una contrasenya, no és necessari escriure res o, si ho demana obligatòriament, podeu posar qualsevol correu electrònic com a contrasenya vàlida).

L'accés anònim és un tipus d'accés que és inviable en el cas del desplegament web, on el control d'accés dels usuaris és important, ja que és de caràcter privat, confidencial i depèn també la nostra aplicació web. Permetre un accés al directori arrel de l'aplicació web amb un accés anònim mitjançant FTP és una falta greu de seguretat i amb conseqüències desastroses.

Accés per usuari identificat

Es dona en els casos de la necessitat de privilegis i la informació amb la qual es treballa és d'indole privada. S'ha d'accedir al servei mitjançant usuaris identificats dins de l'FTP, anomenats comptes.

Els comptes d'usuari poden ser:

- **Usuari de sistema:** usuari definit dins del sistema on s'ofereix el servei.
- **Usuari virtual:** no té una relació directa amb el sistema.

Tots aquests usuaris tindran configurat una sèrie de permisos depenent de la implicació que tinguin els usuaris, per exemple dins del projecte web. Us poden interessar usuaris que només puguin llegir la informació del projecte i d'altres que puguin actualitzar els fitxers, tot això gestionant la jerarquia de l'equip del projecte que està fent l'aplicació web.

3.6 Creació d'usuaris a ProFTPD

Hi ha dos tipus d'usuari que es poden crear per accedir a ProFTPD:

- Usuari de sistema, creat dins de */etc/shadow* (fitxers d'usuaris) i */etc/passwd* (fitxer de configuració de l'usuari) i grups a */etc/group*.
- Usuari virtual, fitxer per a ProFTPD amb usuari no dependent del sistema.

3.6.1 Usuari de sistema a ProFTPD

Per crear usuaris de sistema i que tinguin vinculació amb ProFTPD, procediu a fer les comprovacions i executar les ordres detallades.

Modifiquem el fitxer */etc/shells* amb:

```
1 sudo nano /etc/shells
```

Afegiu-lo al final de tot de la línia */bin/false* com a resultat:

```
1 # /etc/shells: valid login shells
2 /bin/sh
3 /bin/dash
4 /bin/bash
5 /bin/rbash
6 /usr/bin/tmux
7 /usr/bin/screen
8 /bin/false
```

Si deixeu accés als usuaris amb un *shell* vàlid podran accedir al servei FTP dins del directori arrel configurat o al seu directori *home*.

Quan afegiu */bin/false* permetrà que quan es creïn usuaris hi afegiu el *shell* */bin/false*, així evitarem que els usuaris FTP tinguin accés al servei FTP només si l'administrador permet l'accés.

Creeu un usuari amb l'ordre:

```
1 useradd usuari -d directoriTreball -s /bin/false
2 passwd usuari
```

On:

- usuari: nom de l'usuari que creareu dins del sistema.
- directoriTreball: directori de treball de l'usuari, per exemple */home/usuari*.

Eliminació de l'usuari:

```
1 sudo userdel -r nomUsuari
```

Creació del grup de treball:

```
1 sudo groupadd nomGrup
```

Eliminació del grup de treball:

```
1 sudo groupdel nomGrup
```

3.6.2 Usuari virtual a ProFTPD

Per crear usuaris virtuals dins de ProFTPD, executeu l'ordre *ftpasswd*, que crearà un fitxer anomenat *ftpd.passwd*:

```
1 sudo mkdir /etc/proftpd/usuaris
2 cd /etc/proftpd/usuaris
3 sudo ftpasswd --passwd --name=nomUsuari --home=pathTreball --shell=/bin/false
   --uid=500
```

On:

- **-name**: nom de l'usuari que creareu.
- **-home**: directori de treball, per exemple /home/usuari.

Per crear grups de treball virtuals dins de ProFTPD, executeu l'ordre *ftp passwd* que crearà un fitxer anomenat *ftpd.group*:

```
1 cd /etc/proftpd/usuarios
2 sudo ftp passwd --group --name=nomGrup --gid=idGrup --member=nomUsuari
```

On:

- **-name**: nom del grup que creareu.
- **-gid**: id del grup en format numèric enter.

Dins del fitxer de configuració *proftpd.conf*, afegiu les directives que permeten que s'agafin els usuaris i grups virtuals creats.

```
1 RequireValidShell off
2 AuthUserFile /etc/proftpd/ftpd.passwd
3 AuthGroupFile /etc/proftpd/ftpd.group
```

3.7 Modes de connexió del client

El mode de transferència s'estableix a l'inici de les comunicacions FTP i és dependent de com és el sistema de fitxers del servidor.

La majoria dels sistemes fan ús de l'estructura de fitxers binaris, antics sistemes Unix i *mainframe* i poden utilitzar l'estructura de fitxers ASCII. En qualsevol cas, aquest mode es decideix dins del servidor FTP i el client automàticament detectarà quin dels dos modes farà servir.

El protocol FTP es basa en el protocol TCP, amb dos canals de dades amb diferents ports, un per enviar les dades i l'altre per enviar les ordres.

Els ports que es fan servir per defecte són:

- Port número 21, per al canal de control
- Port número 20, per al canal de dades de transmissió

Dins del protocol FTP es defineixen dos modes de connexió que es configuren dins del servei, el mode ftp actiu i el mode ftp passiu.

El 'mainframe'

Ordinador gran i potent utilitzat principalment per les grans companyies per a processament de grans quantitats de dades, com pot ser per al processament de transaccions bancàries en les corporacions bancàries.

3.7.1 Mode FTP actiu

En el mode FTP actiu, el client connecta aleatòriament per un port més gran de 1024 (anomenem-lo N) cap al port 21 d'ordres del servidor.

El client inicia l'escolta al port (N+1) i envia l'ordre FTP al port (N+1) del servidor FTP. El servidor connecta de nou al client pel port de dades especificat per part del client, que és el port 20.

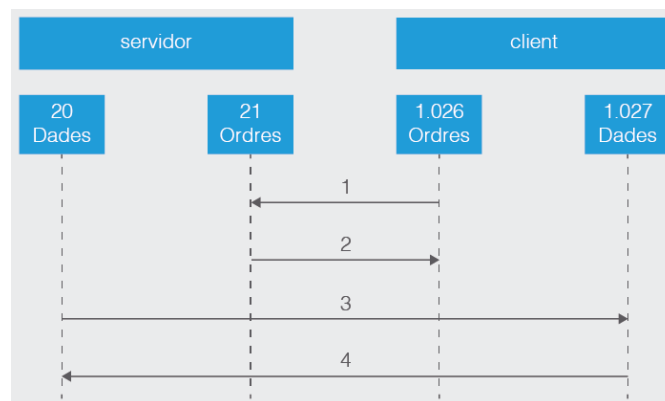
Quan treballa en mode actiu es té en compte el tallafoc del sistema. El tallafoc ha de tenir els ports oberts de treball del servidor i del client, per poder establir les comunicacions.

Ports que s'obriran en mode actiu dins del servidor:

- El client connectarà al port 21 del servidor FTP amb un port més gran de 1024 del client. (Iniciació de la connexió del client)
- El port 21 del servidor FTP connectarà a un port més gran de 1024. (El servidor respon al port de control del client.)
- El port 20 del servidor FTP connectarà a un port més gran de 1024. (El servidor inicia la connexió de dades cap al port de client de dades.)
- El client connectarà amb un port més gran de 1024 cap al port 20 del servidor FTP. (El client envia la confirmació de connexió al port de dades del servidor FTP.)

A la figura 3.4, dins de l'etapa 1, el client connecta amb el port 21 d'ordres mitjançant un port més gran de 1024 aleatori, en aquest cas el 1026. El servidor li envia en el pas 2 el reconeixement al port 1026 del client.

FIGURA 3.4. Gràfic de funcionament del sistema FTP actiu



En el pas 3 el servidor inicia l'obertura del canal de comunicació de dades pel port 20 cap al port del client 1027 (origen d'iniciació del port 1026 + 1). Per finalitzar el pas 4, el client confirma el canal al servidor.

3.7.2 Mode FTP passiu

Per evitar que el servidor iniciï la connexió al client hi ha un altre mètode de connexió anomenat passiu.

En el mètode FTP passiu el client inicia les dues connexions al servidor, resolent el problema de control del tallafoc en el filtratge del port de dades en el servidor cap al client.

El client, a l'iniciar la connexió FTP, agafa un port aleatori més gran de 1024 N i el següent com N + 1.

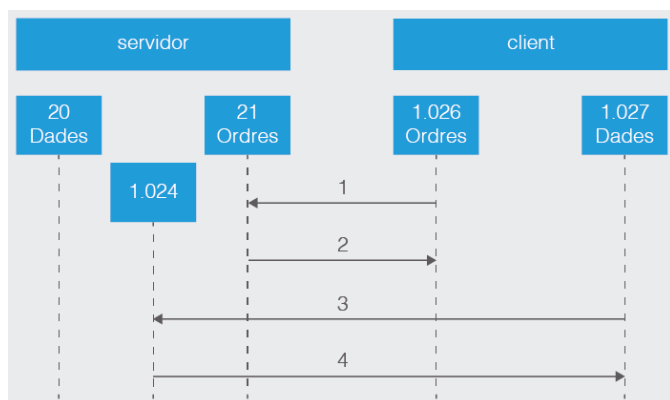
El primer port N fa la connexió pel port 21 del servidor i evita fer la connexió al port de dades 20. El client farà ús d'una ordre PASV. El servidor obre un port aleatori P més gran de 1024 i li retorna al client amb l'ordre PASV. El client inicia el canal de dades del port (N + 1) al port P.

Per controlar el tallafoc en el servidor FTP en mode passiu obrirem els ports següents:

- El client connectarà al port 21 del servidor FTP amb un port més gran de 1024 del client. (Iniciació de la connexió del client.)
- El port 21 del servidor FTP connectarà a un port més gran de 1024 del client. (El servidor respon al port de control del client.)
- Un port més gran de 1024 del client connectarà a un port més gran de 1024 del servidor. (El client inicia el canal de dades a un port aleatori del servidor.)
- Un port més gran de 1024 del servidor connectarà a un port més gran de 1024 del client. (El client inicia el canal de dades a un port aleatori del servidor.)

A la figura 3.5 al pas 1 el client contacta amb el servidor pel port 21 demanant una connexió passiva amb l'ordre PASV.

FIGURA 3.5. Gràfic de funcionament del sistema FTP passiu



El servidor respon en el pas 2 amb un port aleatori, en l'exemple 2024, demanant al client quin port és el que farà servir per obrir el canal de dades. En el pas 3 el client inicia el canal de dades del port de dades del client 1027 al port que ha obert el servidor 2024. En el pas 4 el servidor confirma la connexió.

Amb el mode passiu es resolen molts problemes del client, però s'amplien els problemes del servidor. Un dels principals problemes és l'obertura d'un gran rang de ports en el servidor per poder iniciar canals de dades.

Un dels avantatges actualment és que les implementacions de servidors FTP permeten escollir el rang de ports que es faran servir.

Per realitzar la configuració dins de ProFTPD en el mode passiu, afegiu dins de la configuració del fitxer `/etc/proftpd/proftpd.conf` la directiva:

1	<code>PassivePorts 1024 2000</code>
---	-------------------------------------

El rang de ports de configuració anirà mínim del port número 1024 fins, com a màxim, al 65536.

3.8 Protocol de transferència de fitxers segur

Quan es va redactar el protocol FTP dins la RFC 959 la seguretat no era un tema crític. Amb l'evolució de les xarxes i la transferència massiva de dades dins les xarxes públiques, ha canviat molt respecte a les idees originals en els anys 70 i 80. Aquesta evolució fa que enviar dades sense encriptar sigui molt arriscat i que protocols antics hagin d'evolucionar per garantir que la tramesa de dades sigui segura.

Amb l'evolució de les xarxes, el protocol FTP va fer que se n'originessin noves revisions per pal·liar les deficiències de seguretat i l'any 1997 es va redactar l'actualització del protocol FTP que dona com a resultat l'FTPS.

Els autors de la RFC van llistar l'any 1999 les diferents vulnerabilitats FTP:

- Atacs Spoofing
- Atacs de força bruta
- Atacs rebot (*bounce attacks*)
- Captura de paquets (*sniffing*)
- Robatori de ports (*port stealing*)
- Claus d'usuari i dades no xifrades

L'especificació FTP amb SSL (*secure Sockets Layer*) es troba dins la RFC 2228 i afegeix les extensions de seguretat fent ús d'SSL i dins RFC 4217 es defineix FTP amb TLS (*transport layer security*).

El protocol FTPS fa ús d'SSL, TLS i és una combinació d'algoritmes de xifrat asimètrics (RSA,DSA), algoritmes simètrics (DES/3DES,AES etc.) i un algoritme d'intercanvi de claus amb autenticació de certificats X.509.

Hi ha dos modes de treball amb FTPS:

- El **mode FTPS implícit SSL**. Es requereix una sessió SSL entre el client i el servidor abans que s'intercanviï qualsevol dada. Com el nom diu, l'ús d'SSL és implícit i qualsevol intent de connexió dels clients sense fer ús d'SSL és rebutjat pel servidor. Els ports de treball de l'FTPS implícit són el 990 i el 989.

Actualment es fa servir molt poc FPS implícit a favor de fer ús del segon mètode FTPS explícit SSL.

- El **mode FTPS explícit SSL**. El client i el servidor negocien el nivell de protecció que es farà servir. Aquesta situació és útil per poder treballar amb el mateix port amb sessions encriptades o no encriptades.

En el mode explícit SSL el client inicia una connexió sense encriptar amb el servidor FTP. El client demana una petició al servidor FTP per iniciar una ordre sobre SSL, enviant les ordres AUTH TLS o AUTH SSL.

Una vegada iniciat el canal SSL el client envia les credencials al servidor FTP. Totes les credencials són encriptades i enviades pel canal SSL. De la mateixa manera que s'ha fet la protecció del canal de control, el canal de dades segueix el mateix procediment fent ús de l'ordre PROT. Els ports que fa servir són el 20 i el 21, on s'efectua el xifrat.

Hi ha una altre tipus de servei segur amb FTP anomenat **SFTP**.

SFTP sol ser confós amb el servei FTPS i viceversa, i realment no tenen res a veure mútuament. Excepte per la seguretat en la tramesa de fitxers, el procediment intern és diferent.

SFTP està basat en SSH (*secure shell*), protocol conegut per proveir seguretat als terminals remots.

No fa ús de canals d'ordres i de dades. Els dos canals que fem servir en FTPS s'envien en paquets amb format dins d'un mateix canal, és a dir, el canal de dades i d'ordres és únic.

Totes les dades enviades i rebudes són encriptades mitjançant un algoritme d'encriptació prèviament acordat.

Les sessions estan protegides mitjançant claus públiques i privades, que ofereixen un sistema d'autenticació conegut com a autenticació de clau pública que es pot fer servir com a alternativa o unió dels sistemes d'autenticació tradicionals de noms d'usuari i contrasenyes.

3.8.1 Configuració FTPS a ProFTPD

Dins dels sistemes Debian o basats en Debian ja pot venir instal·lada l'aplicació *openssl*. Reviseu-ne l'existència amb:
`dpkg -l | grep openssl`

Per configurar FTPS dins de ProFTPD, feu les instal·lacions i configuracions necessàries per habilitar-ho.

Procediu a instal·lar el paquet d'eines d'administració i biblioteques relacionades amb encriptació *openssl*.

```
1 sudo apt-get update
2 sudo apt-get install openssl
```

Creeu el directori */etc/proftpd/ssl* i situeu-vos dins:

```
1 sudo mkdir /etc/proftpd/ssl
2 cd /etc/proftpd/ssl
```

Dins del directori */etc/proftpd/ssl* creeu els certificats x.509 per configurar l'FTPS. Executeu:

```
1 sudo openssl req -new -x509 -days 365 -nodes -out /etc/proftpd/ssl/proftpd.cert
   .pem -keyout /etc/proftpd/ssl/proftpd.key.pem
```

Ompliu les dades del certificat amb les dades que us demanarà.

Una vegada finalitzada l'ordre *openssl*, obtindreu dos fitxers:

- Certificat x509, anomenat *proftpd.cert.pem*
- Claus públiques i privades *proftpd.key.pem*

Canvieu els permisos als fitxers de certificats amb permisos de lectura i escriptura per a usuaris.

```
1 sudo chmod 600 /etc/proftpd/ssl/proftpd.*
```

Un cop obtinguts els certificats, editeu el fitxer de configuració */etc/proftpd/tls.conf*:

```
1 sudo nano /etc/proftpd/tls.conf
```

I afegiu-hi:

```
1 <IfModule mod_tls.c>
2   # Fem servir el servei FTPS
3   TLSEngine on
4   # configuració del log de sortida al path corresponent
5   TLSLog /var/log/proftpd/tls.log
6   # protocols que es poden fer servir SSLv23 SSLv3 TLSv1 TLSv1.1 TLSv1.2
7   TLSProtocol TLSv1.2
8   #combinació d'algoritmes per autenticar, xifrar etc.
9   TLSCipherSuite AES128+EECDH:AES128+EDH
10  TLSOptions NoCertRequest AllowClientRenegotiations
11  # path dels certificats i claus generades
```



```
12 TLSRSCertificateFile    /etc/proftpd/ssl/proftpd.cert.pem
13 TLSRSCertificateKeyFile /etc/proftpd/ssl/proftpd.key.pem
14 # demana un certificat al client
15 TLSVerifyClient         off
16 # obliga el client a fer una connexió TLS
17 TLSRequired             on
18 RequireValidShell       no
19 </IfModule>
```

Dins del fitxer de configuració proftpd.conf:

```
1 sudo nano /etc/proftpd/proftpd.conf
```

Afegiu la directiva a dalt del fitxer, que permetrà agafar la configuració per a FTPS, o feu-ne la cerca i traieu el *tag*:

```
1 include /etc/proftpd/tls.conf
```

3.9 Utilització d'eines gràfiques

Les aplicacions gràfiques que implementen el protocol FTP fan ús dels avantatges dels llenguatges de programació mitjançant les llibreries gràfiques del sistema operatiu i les llibreries que implementen el protocol FTP.

Un exemple de llibreria FTP és la implementació `ftplib` a Python, que té les funcionalitats necessàries per realitzar un client FTP. Es podria desenvolupar en poc temps un client gràfic o un terminal alfanumèric que faria les mateixes funcionalitats que qualsevol programari que hi ha al mercat.

3.9.1 Client gràfic Filezilla

Tenim diferents opcions per fer servir clients FTP en mode gràfic. Un dels més coneguts és Filezilla client: filezilla-project.org.

El projecte Filezilla, sota la llicència GNU, posseeix una solució FTP com a client per a sistemes Microsoft, Linux i Mac OS i una solució servidor només implementat per sistemes Microsoft.

Característiques principals del client Filezilla:

- Suport per a FTP, FTP amb SSL/TLS (FTPS) i SSH FTP (conegut com SFTP)
- Traducció a múltiples llenguatges
- Suport per treballar amb fitxers més grans de 4 GB (GigaBytes)
- Interfície amb pestanyes

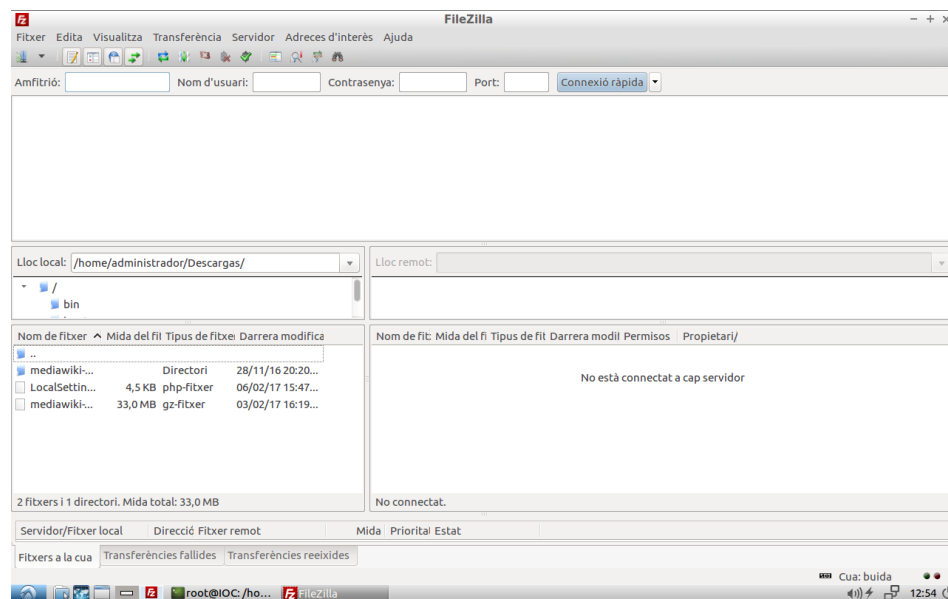
- Cua de transferència i gestor d'administració avançat
- Suport per a *Drag & Drop*
- Configuració de velocitats de transferència
- Navegació de directoris sincronitzada
- Cerca de fitxers remot
- Editor de fitxers remot
- Comparació de directoris

Per instal·lar el client Filezilla en el sistema basat en Debian, executeu dins d'un terminal:

```
1 sudo apt-get update
2 sudo apt-get install filezilla
```

Executeu Filezilla. Podeu veure que l'entorn gràfic és intuïtiu. Aneu a l'opció sota el menú *Fitxer* i sobre *Amfitrió*, hi ha una icona que representa un servidor (vegeu la figura 3.6).

FIGURA 3.6. Finestra inicial de Filezilla



Dins del menú que us mostra permet crear diferents configuracions d'accés a servidors remots FTP.

En l'exemple configureu un lloc anomenat repositori Ubuntu, amb les opcions:

- Amfitrió o adreça del servidor remot *cdimage.ubuntu.com*.
- Protocol FTP, no volem fer servir SFTP.
- Xifratge, si està disponible l'FTP explícit sobre TLS. Si el servidor requereix connexió encriptada es farà servir aquesta. En cas que no tingui xifratge, es farà servir sense xifrar.

- Tipus d'entrada, anònim. Podem demanar usuari i posar *anonymous* com a usuari.

Vegeu els passos a seguir a la figura 3.7 i la figura 3.8.

FIGURA 3.7. Finestra de configuracions de connexions a Filezilla

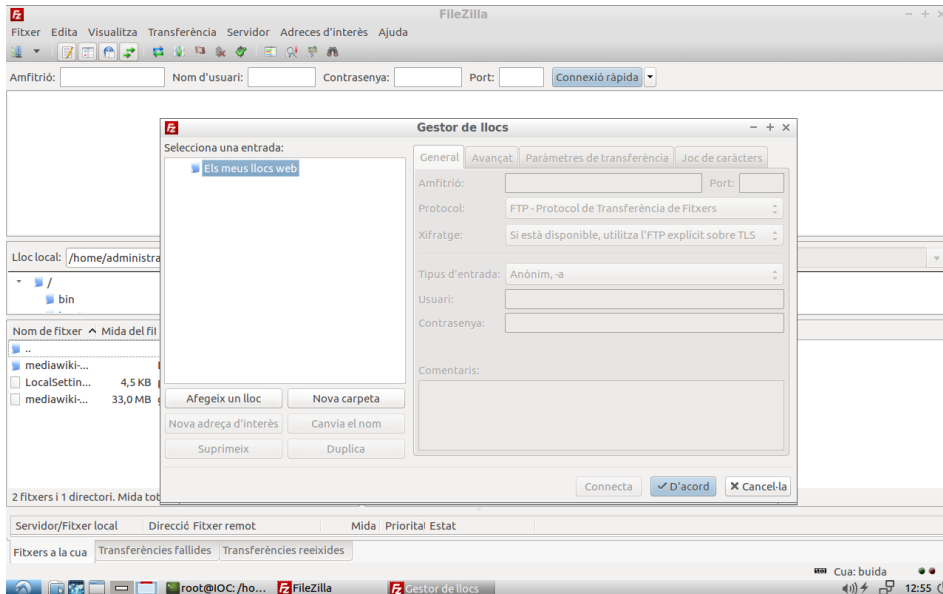
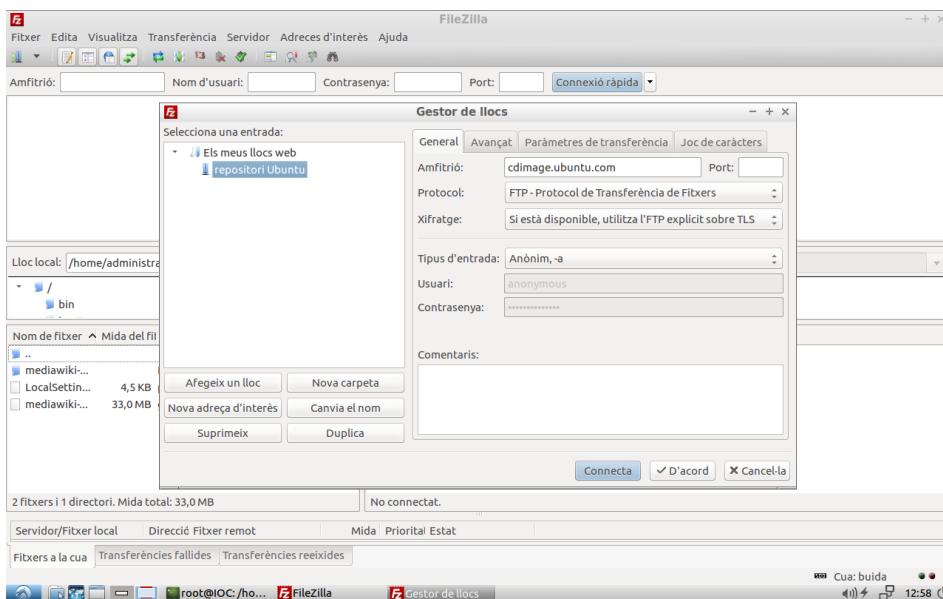


FIGURA 3.8. Exemple de configuració a Filezilla



Connecteu i Filezilla connectarà amb el servidor remot FTP. Vegeu la sortida de missatges que us mostrarà dins la finestra.

Una vegada el client ha connectat amb el servidor remot, fixeu-vos amb la finestra que es divideix en dues parts, l'arbre de directoris de l'equip local i l'arbre de directoris del servidor remot.

Podeu navegar dins del servidor remot fins a trobar el fitxer que us interessa i prémer botó dret per veure el menú d'opcions, tal com podeu veure a la figura 3.9 i la figura 3.10.

El menú d'opcions permet donar les ordres per copiar el contingut remot cap al directori actual de l'arbre de directoris de l'equip local on esteu en aquell moment. Podeu fer el mateix arrossegant els fitxers i directoris d'un llistat a l'altre.

FIGURA 3.9. Finestra de navegació a un servidor FTP remot dins de Filezilla

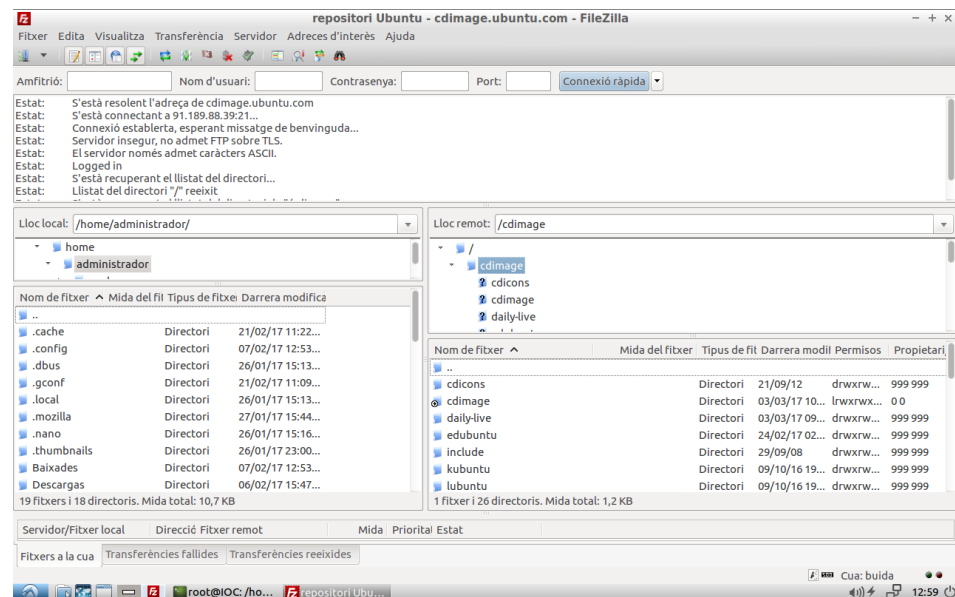
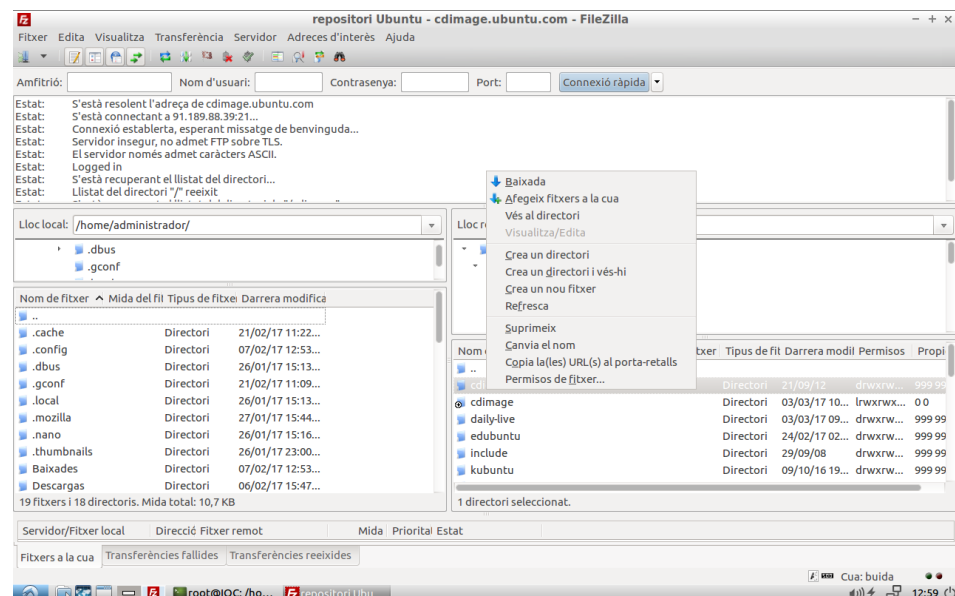


FIGURA 3.10. Ordres FTP dins de Filezilla



3.9.2 Client gràfic dels sistemes operatius

Dins de l'entorn gràfic del sistema operatiu Windows o Linux permeten accedir amb l'explorador de fitxers del sistema a un servidor remot FTP.

Dins del navegador de fitxers dels sistemes Windows fem com a ruta de navegació <ftp://cdimage.ubuntu.com>. Veureu que farà la connexió que us enllaça al servidor FTP remot. Pot ser que ens demani l'usuari i la contrasenya si és

necessari. Vegeu la figura 3.11 i la figura 3.12 d'un exemple d'accés a un servidor remot configurat amb proFTPD mitjançant Windows 10.

FIGURA 3.11. Pantalla d'accés i validació a un servidor FTP

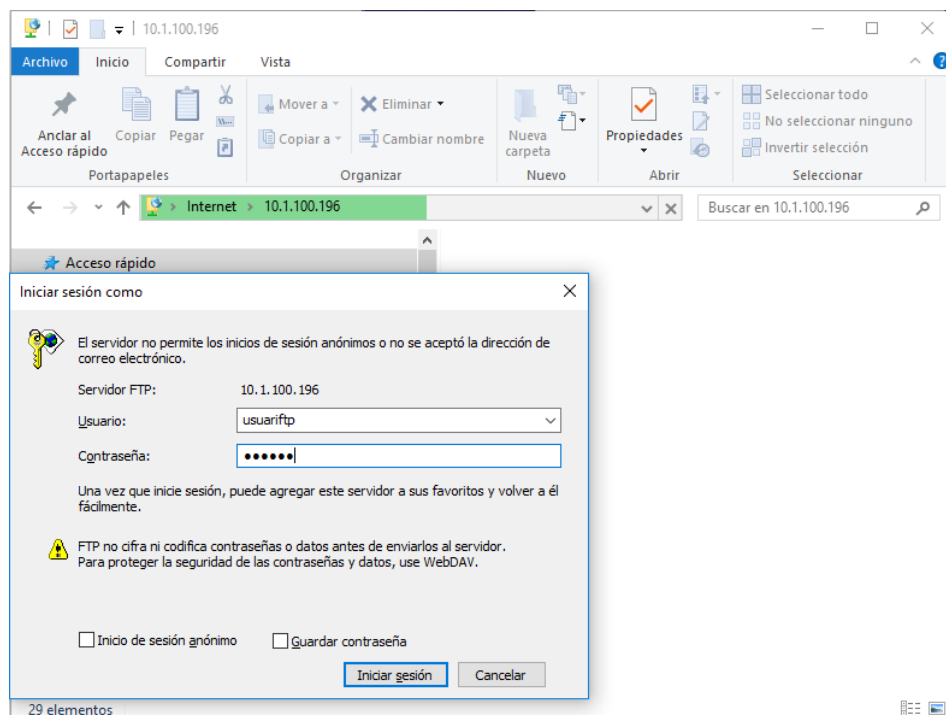
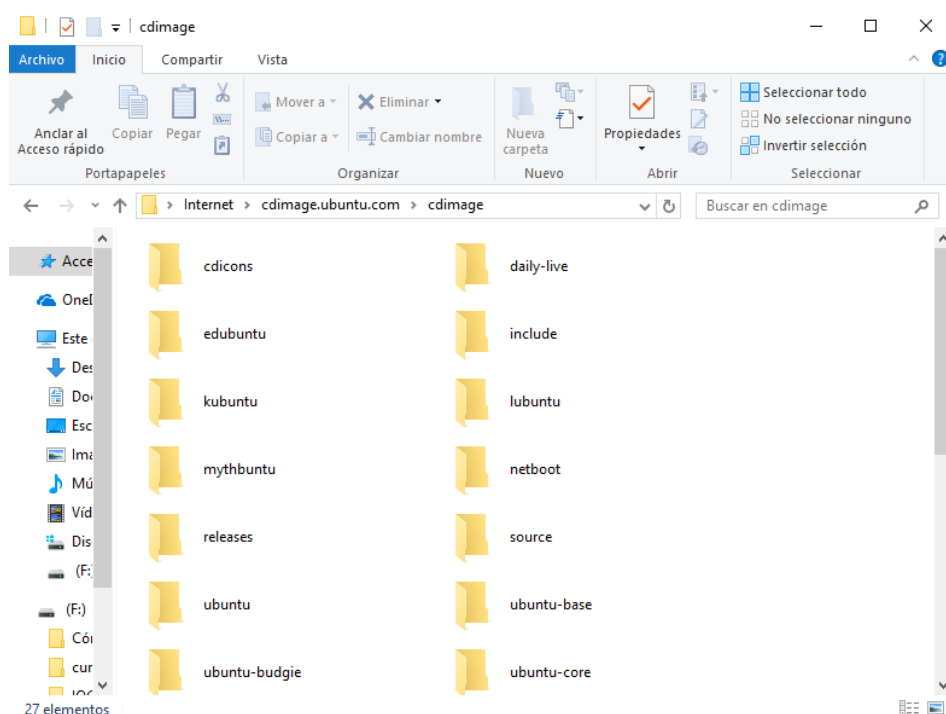


FIGURA 3.12. Pantalla de navegació dels directoris remots



Per poder copiar o escriure fitxers dins d'una connexió FTP treballeu de la mateixa manera que si fos una carpeta local del sistema o en xarxa. Això sí, no podreu modificar els continguts si esteu en una connexió anònima o l'administrador no ho permet.

En el cas del funcionament dins d'un sistema Linux serà exactament igual dins del navegador de fitxers del sistema Linux.

Poseu dins de la barra de navegació <ftp://cdimage.ubuntu.com> i premeu la tecla intro. La connexió es realitzarà i demanarà, si és necessari, l'usuari i la contrasenya. Vegeu la figura 3.13 i la figura 3.14 d'un exemple d'accés a un servidor remot configurat amb proFTPD mitjançant Ubuntu.

FIGURA 3.13. Pantalla d'accés i validació a un servidor FTP

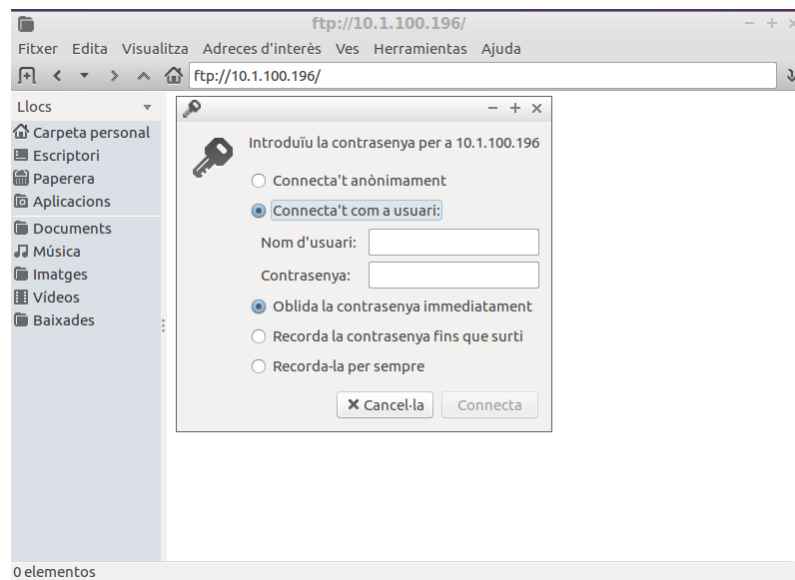
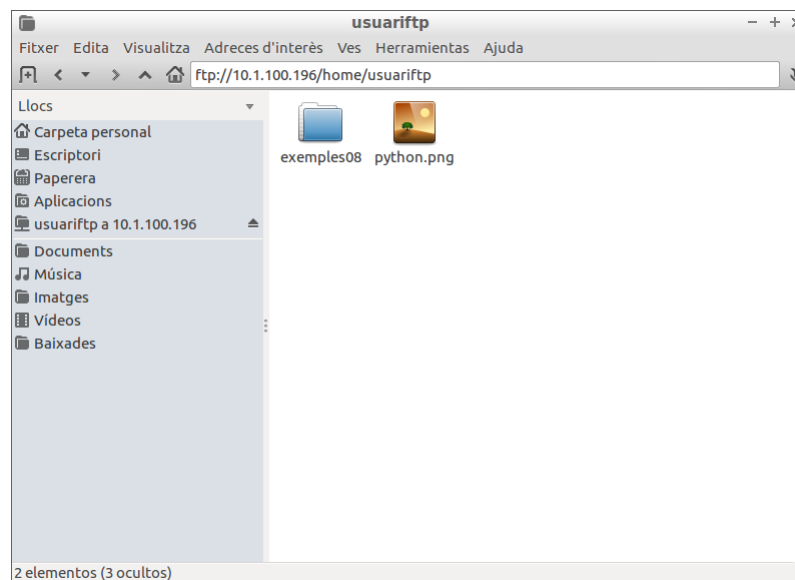


FIGURA 3.14. Pantalla de navegació dels directoris remots



3.9.3 Utilització del servei de transferència de fitxers des del navegador

Un dels grans avantatges que té el protocol FTP es que permet ser implementat fàcilment en diferents tipus de sistemes operatius (escriptori, mòbils, consoles, etc.).

A més, el protocol FTP és incorporat dins d'aplicacions, com una extensió de la funcionalitat principal de l'aplicació. Imagineu aplicacions IDE de desenvolupament o els navegadors web.

Els navegadors web permeten accedir als servidors FTP com un client FTP directe, amb la seva validació d'usuari, llegir el directori remot i accedir a la informació.

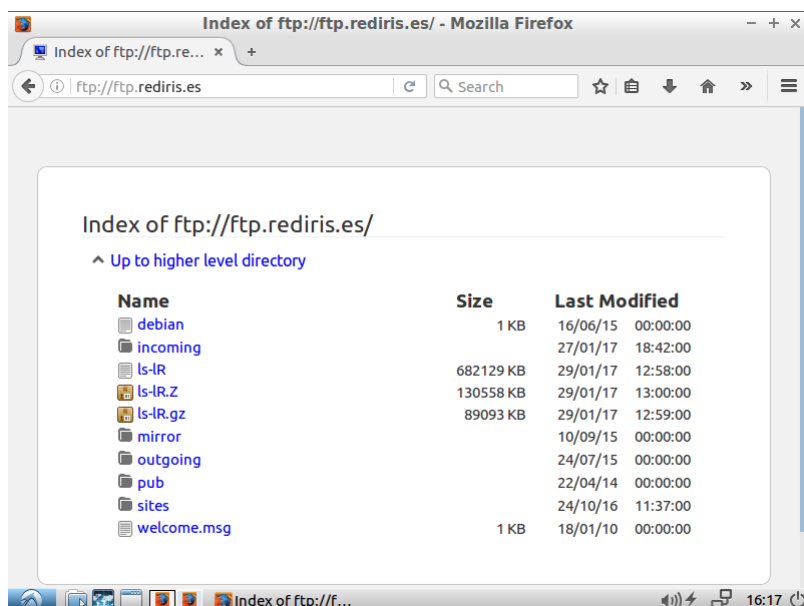
La forma d'accedir a un servidor FTP mitjançant el navegador és escriure dins de la barra de navegació l'esquema:

```
1 ftp://<url>
```

Vegeu un exemple bàsic dins del navegador. Obriu una finestra i copieu a la barra de navegació la següent direcció: <ftp://ftp.rediris.es>.

A la figura 3.15 veureu que dins de la part de visualització web es llista un directori amb el seu contingut. Aquesta llista del directori pertany a l'arrel del servidor remot al qual ens hem connectat. Proveu de navegar i de baixar algun contingut.

FIGURA 3.15. Pantalla del navegador accedint a un servidor anònim



La limitació que ofereix el navegador és que no podem modificar els fitxers. Això vol dir que les accions crear, modificar o esborrar fitxers no es poden realitzar. Per aquesta deficiència, hi ha una altra alternativa, que és la d'ampliar el funcionament del navegador amb extensions del navegador.

Les extensions no són més que programari alternatiu que funcionen juntament amb el navegador per ampliar les seves capacitats. Feu una cerca dins de Google Chrome o Mozilla Firefox de les diferents extensions que es poden afegir al navegador.

3.10 Utilització del servei de transferència de fitxers en el procés de desplegament de l'aplicació web

El procés de desplegament (en anglès *deployment*) consisteix en l'encapsulació del projecte i la distribució final dins d'un entorn de producció on s'executarà.

La distribució final de l'aplicació web a un entorn de producció es pot dur a terme:

- Mitjançant un fitxer d'encapsulació que engloba tot el contingut del projecte i que depèn de la tecnologia que fem servir per desenvolupar l'aplicació web.
- Mitjançant una estructura de directoris. Aquests directoris contenen els fitxers html, php, fulles d'estil i fitxers com imatges, sons, vídeos, etc.

La pràctica habitual és dissenyar l'aplicació web en un disc dur en local amb un servidor web local (anomenat entorn de desenvolupament), per després enviar-ho a un servidor final de producció. A vegades existeix un pas intermedi que consisteix a enviar l'aplicació, en un entorn de proves el més semblant possible a l'entorn de producció.

La transferència dels fitxers de l'aplicació web es farà mitjançant un client FTP cap a un servidor FTP que enllaçarà amb el directori principal del servidor web. Aquesta transferència de fitxers es farà mitjançant un usuari autenticat dins del servidor FTP.

Les dades necessàries per fer la transferència seran:

- URL o adreça IP del servidor.
- Usuari i contrasenya creats dins del servidor FTP i amb permisos d'escriptura als directoris on s'emmagatzemen els fitxers de l'aplicació web.

El client FTP que feu servir per a pujar l'aplicació és indiferent, podeu fer servir clients FTP integrats en els IDE de desenvolupament, clients FTP dins del terminal Unix, Microsoft o clients gràfics.

El desplegament de l'aplicació web, dins del cas d'actualització de les funcionalitats, sempre serà en hores on l'ús del servei és minoritari, per evitar problemes als usuaris. A vegades també l'aplicació web pot tenir programada una opció per aturar el servei i permetre que es puguin actualitzar els continguts del servidor.