

Administració del servei de directori

Andrés Pérez Payeras i Miquel Tarazona Belenguer

Índex

Introducció	5
Resultats d'aprenentatge	7
1 El servei de directori	9
1.1 Servei de directori	9
1.1.1 Clients i servidors de directori	10
1.1.2 Avantatges dels serveis de directori	11
1.2 El protocol LDAP	12
1.2.1 Origen de l'LDAP	12
1.2.2 Característiques de l'LDAP	13
1.2.3 Funcionament de l'LDAP	15
1.2.4 Missatges LDAP	16
1.3 Els models LDAP	17
1.3.1 Model d'informació	17
1.3.2 Model de nomenclatura	22
1.3.3 Nom distintiu (DN)	23
1.3.4 Sufix de directori (directory suffix)	23
1.3.5 Àlies (alias)	23
1.3.6 Referències (referrals)	24
1.3.7 Arrel DSE (rootDSE)	24
1.3.8 Model funcional	25
1.3.9 Model de seguretat	30
1.4 El format d'intercanvi de dades LDIF	35
1.4.1 Format d'un fitxer LDIF	36
2 Instal·lació, configuració i manteniment del servei de directori	39
2.1 Introducció a l'OpenLDAP	39
2.2 Planificació del servei de directori	41
2.2.1 Escenari d'exemple	41
2.3 Instal·lació de l'OpenLDAP	43
2.3.1 Requeriments previs	43
2.3.2 Instal·lació del programari	45
2.3.3 Verificació de la instal·lació	46
2.3.4 Reconfigurar el programari	48
2.3.5 Desinstal·lació del programari	49
2.3.6 Aturada i arrencada del servei	49
2.4 Configuració del servei de directori	49
2.4.1 Configuració dinàmica del servei	50
2.5 Manteniment del directori	52
2.5.1 Utilitats de línia d'ordres	52
2.5.2 Utilitat gràfica phpLDAPadmin	61

3	Integració del servei de directori	69
3.1	Autenticació centralitzada amb un servei de directori	69
3.2	Autenticació a Debian 6 amb l'OpenLDAP	70
3.2.1	Comprovacions preliminars	71
3.2.2	Instal·lació del programari client LDAP	72
3.2.3	Configuració de l'autenticació LDAP	74
3.2.4	Verificació de la configuració	76
3.2.5	Reconfiguració de l'autenticació	77
3.3	Integració de Samba amb l'OpenLDAP	78
3.3.1	Preparació del servidor	78
3.3.2	Afegir samba.schema al directori LDAP	79
3.3.3	Configurar Samba com a client LDAP	81
3.3.4	Comprovació de la configuració	85
3.4	Autenticació a Pure-FTPd amb l'OpenLDAP	85
3.4.1	Comprovació de la connexió FTP	87
3.5	Autenticació a Joomla amb l'OpenLDAP	88
3.5.1	Configuració de Joomla	88

Introducció

Poques vegades els usuaris de sistemes informàtics ens preguntem, quan escrivim un nom d'usuari i una contrasenya, on està emmagatzemada aquesta informació o com s'hi accedeix. Una característica desitjable per a qualsevol sistema és no duplicar informació per tal de facilitar-ne el manteniment. Emmagatzemar la informació de l'organització en un directori i que aquest sigui el punt central on consultar-la sembla una bona opció.

En aquesta unitat s'estudien els conceptes requerits per poder administrar correctament un servei de directori, des de la part teòrica necessària per comprendre el seu funcionament fins a la descripció d'exemples pràctics, similars als que ens podem trobar a la vida real, per mostrar la part més procedimental dels continguts.

En l'apartat "El servei de directori" es tracta la part més teòrica de la unitat. S'hi descriuen tots els conceptes necessaris per entendre els apartats posteriors. Aquest apartat presenta els fonaments dels directoris i el protocol més utilitzat per accedir-hi, l'LDAP.

En l'apartat "Instal·lació, configuració i manteniment del servei de directori" es tracta una implementació específica del protocol LDAP, concretament l'OpenLDAP. Primerament s'expliquen les característiques bàsiques de la implementació triada per veure'n després la instal·lació, configuració i manteniment tot centrant-nos en un supòsit concret per ajudar a assimilar els conceptes que s'expliquen. S'ha triat l'OpenLDAP perquè és una implementació lliure però àmpliament estesa i que disposa de suport comercial.

En l'apartat "Integració del servei de directori" es treballa un dels aspectes més utilitzats i funcionals de l'ús d'un servei de directori com a dipòsit on emmagatzemar la informació: l'autenticació centralitzada. Aquesta autenticació centralitzada facilita enormement la tasca de manteniment d'usuaris i contrasenyes en un sistema informàtic en el qual requereixen autenticació diverses aplicacions. L'apartat cobreix la integració del directori per fer l'autenticació amb un sistema operatiu client, sigui Windows o Linux; amb un servei, prenent com a exemple un servidor FTP; i amb una aplicació web, prenent com a exemple un gestor de contingut.

Els continguts d'aquesta unitat tenen una orientació bàsicament pràctica. Es proposa una formació pràctica i aplicada amb l'objectiu que l'alumnat aprengui i interioritzi els conceptes fent les coses. És molt recomanable llegir els continguts de cada apartat i anar realitzant els exemples proposats, així com els exercicis i les activitats del web de la unitat, per tal de posar en pràctica i comprovar els coneixements adquirits.

Resultats d'aprenentatge

En acabar aquesta unitat, l'alumne:

1. Administra el servei de directori interpretant especificacions i integrant-lo en una xarxa.

- Identifica la funció, els elements i les estructures lògiques del servei de directori.
- Determina i crea l'esquema del servei de directori.
- Realitza la instal·lació del servei de directori al servidor.
- Realitza la configuració i personalització del servei de directori.
- Integra el servei de directori amb altres serveis.
- Aplica filtres de cerca en el servei de directori.
- Utilitza el servei de directori com a mecanisme d'acreditació centralitzada dels usuaris en una xarxa.
- Realitza la configuració del client per a la seva integració en el servei de directori.
- Utilitza eines gràfiques i comandaments per a l'administració del servei de directori.
- Documenta l'estructura i implantació del servei de directori.

1. El servei de directori

Avui dia les persones i empreses confien en els sistemes informàtics connectats en xarxa per utilitzar aplicacions distribuïdes. La informació de la xarxa pot ser recollida en una base de dades especial que es diu *directori*. Un directori s'utilitza habitualment per emmagatzemar la informació dels usuaris, com ara el nom de l'usuari i la contrasenya. També es possible emmagatzemar-hi informació com les dades de contacte dels usuaris (directori d'una empresa) i també s'utilitza sovint per inventariar recursos (inventari de màquines, impressores, servidors i altres recursos de xarxa).

A mesura que el nombre de xarxes i aplicacions creix, el nombre de directoris especialitzats d'informació també augmenta, cosa que dona lloc a "illes" d'informació que són difícils de compartir i gestionar. El Lightweight Directory Access Protocol (LDAP) és un estàndard obert que ha evolucionat per proporcionar un mètode estàndard per accedir a la informació en un directori i actualitzar-la. L'LDAP ha obtingut una gran acceptació com a mètode d'accés a directoris dins de les xarxes corporatives. Està recolzat per un gran nombre de proveïdors de programari i ha estat incorporat a un nombre creixent d'aplicacions.

1.1 Servei de directori

Un directori és un llistat d'informació sobre uns objectes disposats en un ordre que dona detalls sobre cada objecte. Exemples comuns de directoris són una guia telefònica d'una ciutat o un catàleg d'una biblioteca. En una guia telefònica, els objectes de la llista són les persones, els noms estan ordenats alfabèticament i els detalls guardats sobre cada persona són la direcció i el número de telèfon. Els llibres d'un catàleg d'una biblioteca estan ordenats per autor o per títol, i se'n guarda informació com l'ISBN o altres característiques de la publicació.

En termes informàtics, un **directori** és una base de dades especialitzada (també anomenada *repositori de dades*) que emmagatzema informació sobre objectes i que està específicament dissenyada per optimitzar les operacions de consulta. Un **servei de directori** és el programari que emmagatzema, organitza i facilita l'accés a la informació d'un directori.

Per exemple, un directori pot incloure informació sobre les impressores (els objectes), que consistirà en dades com ara la ubicació (una cadena de caràcters), la velocitat en pàgines per minut (numèric), els fluxos d'impressió compatibles (per exemple, PostScript o ASCII) i així successivament.

Els directoris permeten als usuaris o a les aplicacions trobar informació que compleixi unes característiques determinades. Per exemple, un directori es pot utilitzar per buscar una adreça de correu electrònic d'una persona concreta, per trobar una impressora en color propera, per consultar la informació de facturació d'un client específic...

En un directori es pressuposa un nombre molt més alt de lectures que d'escriptures, ja que generalment la informació continguda canvia rarament (per exemple el nombre de telèfon d'una persona). Aquest aspecte és important i el diferencia d'una base de dades de propòsit general, en la qual les operacions són tant de lectura com d'escriptura. En el disseny d'un directori, els esforços d'optimització es concentren en les cerques i les lectures, i no és cap problema que això penalitzi les actualitzacions.

Els directoris poden contenir molts tipus d'informació i estendre la seva utilitat a diverses aplicacions, però complint sempre aquestes característiques:

1. La informació que contenen està basada en objectes (cada objecte és una entrada del directori) i en els atributs d'aquests objectes.
2. Les actualitzacions i modificacions de les dades són simples i no gaire freqüents.
3. Estan optimitzats per donar una resposta ràpida a operacions de cerca i revisió.
4. Poden replicar la informació per augmentar-ne la disponibilitat i la fiabilitat alhora que disminueix el temps de resposta a les consultes.

Alguns exemples d'ús de directori són:

1. Autenticació d'usuaris.
2. Autenticació de màquines.
3. Llibreta d'adreces.
4. Representació de l'estructura de l'organització.
5. Emmagatzematge d'informació telefònica.
6. Cerca d'adreces de correu electrònic.

API i protocol

Una API defineix la interfície de programació que un llenguatge de programació particular utilitza per accedir a un servei. El format i contingut dels missatges intercanviats entre el client i el servidor s'han d'adherir al protocol acordat.

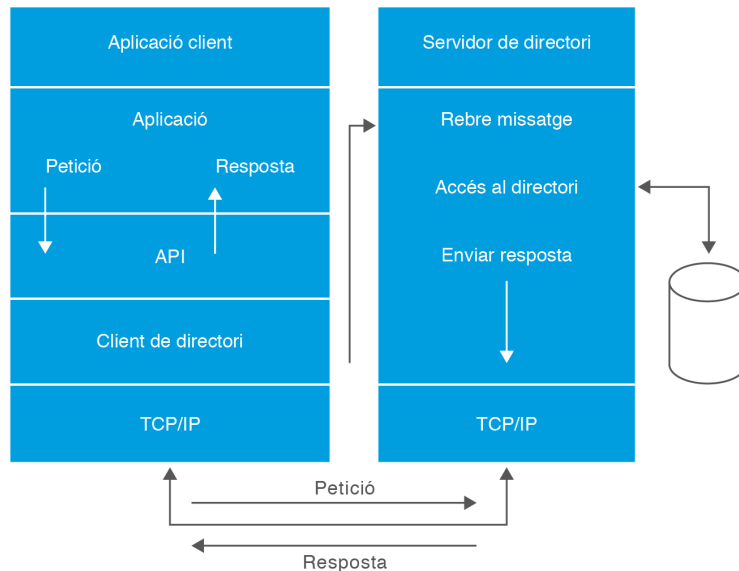
1.1.1 Clients i servidors de directori

Els serveis de directori solen implementar-se seguint el model client-servidor, de manera que una aplicació que vol accedir al directori no accedeix directament a la base de dades, sinó que, tal com mostra la figura 1.1, crida a una funció de l'API (Application Programming Interface) proporcionada pel servei, que envia

un missatge a un procés del servidor. Aquest procés accedeix al directori i retorna el resultat de l'operació.

Algunes vegades, el servidor pot convertir-se en el client d'un altre servidor per aconseguir la informació necessària per processar la petició que se li ha realitzat.

FIGURA 1.1. Arquitectura client-servidor del directori



Seguint aquest model, el client no depèn de l'arquitectura del servidor i el servidor pot implementar el directori de la manera més convenient.

El servidor del directori i el client han d'implementar un protocol comú per comunicar-se i és aquest el que caracteritza les diferents solucions que s'han desenvolupat. El **Lightweight Directory Access Protocol (LDAP)** facilita serveis centralitzats i distribuïts d'informació mitjançant una xarxa TCP/IP i s'ha convertit en l'estàndard de facto d'accés a la informació emmagatzemada en un servidor de directori per a usuaris i aplicacions.

1.1.2 Avantatges dels serveis de directori

Un directori específic d'una aplicació només emmagatzema la informació necessària per a aquella aplicació en particular i no és accessible per altres aplicacions. La mateixa adreça de correu electrònic emmagatzemada per una aplicació de calendari també pot ser emmagatzemada per una aplicació de correu electrònic i per una aplicació que avisa els operadors del sistema de problemes en els equips. Cada aplicació crea i administra el seu propi directori.

En un entorn on només una aplicació ha d'accedir a les dades, potser l'esforç necessari per implantar un servei de directori electrònic estàndard és innecessari, però l'experiència demostra que tard o d'hora diverses aplicacions acaben utilit-

zant aquestes dades, i en el cas de no haver implantat un directori centralitzat ens trobem amb diversos directoris que han d'estar sincronitzats i que accedeixen de maneres diferents a les dades contingudes en directoris creats a mida. Això implica un major esforç per fer el manteniment i un fre al desenvolupament d'aplicacions basades en el directori.

El fet de disposar d'un servei de directori comú, multiplataforma, accessible mitjançant un protocol estàndard i amb una API estàndard, permet als programadors desenvolupar les aplicacions sense haver de crear directoris específics.

Des del punt de vista de l'administració de sistemes, configurar el sistema perquè les aplicacions utilitzin un servei de directori comú, dissenyat de manera adequada, permet controlar més fàcilment els riscos de fallada per inconsistència de dades i concentrar els esforços en millorar l'administració i la tolerància a fallades del servei.

1.2 El protocol LDAP

LDAP són les inicials de Lightweight Directory Access Protocol (protocol lleuger d'accés a directoris). Com el seu nom indica, és un protocol pensat per permetre l'accés a serveis de directori.

L'LDAP funciona amb un esquema client-servidor, en el qual un (o diversos) servidors mantenen la mateixa informació de directori (actualitzada mitjançant rèpliques) i els clients fan consultes a qualsevol d'ells.

A l'LDAP la informació està estructurada en forma d'arbre. Aquesta estructura s'anomena *directori* i cada node s'anomena *entrada*. Per la seva part, cada entrada està formada per un conjunt d'atributs, cadascun dels quals és d'un tipus i conté un o més valors.

1.2.1 Origen de l'LDAP

El 1984, la UIT-T (llavors anomenada CCITT) va decidir desenvolupar una especificació de directori de propòsit general. La necessitat més immediata era proporcionar un directori per a la gestió de l'estàndard X.400 de missatges de correu electrònic. Al mateix temps, l'ISO/IEC JTC1 va iniciar una activitat similar. Les dues organitzacions van acordar fusionar els dos treballs en un de sol per evitar la producció de dues normes diferents per al mateix propòsit. Aquesta col·laboració va donar com a resultat la norma X.500. Aquesta norma és un conjunt d'estàndards de xarxes d'ordinadors per a serveis de directori. La norma defineix un protocol anomenat DAP (Directory Access Protocol o protocol d'accés a directoris).

Organitzacions relacionades amb el protocol LDAP

UIT-T és el Sector de Normalització de la Unió Internacional de Telecomunicacions.

CCITT és l'acrònim de Comitè Consultiu Internacional Telegràfic i Telefònic.

ISO és l'acrònim de la International Organization for Standardization i **IEC** ho és de la International Electrotechnical Commission. **ISO/IEC** treballen conjuntament en la preparació, adopció i utilització de diferents estàndards.

ISO/IEC JTC1 és el comitè tècnic que treballa en l'estandardització dins del camp de les tecnologies de la informació.

El DAP és un protocol molt complex (i d'elevat cost computacional), ja que està definit sobre la pila completa del model OSI (*Open System Interconnection*, en català 'Interconnexió de Sistemes Oberts'). L'LDAP és un protocol més lleuger, gairebé equivalent al DAP, més senzill i eficient, ja que està dissenyat per funcionar directament per TCP/IP.

L'LDAP és un subconjunt del DAP que s'ha desenvolupat per al seu ús amb TCP/IP, de manera que hereta els conceptes fonamentals de la norma X.500, però simplifica la comunicació entre el servidor i el client i no inclou les característiques menys utilitzades del DAP.

L'especificació LDAP es defineix a l'RFC 4510 de l'Internet Engineering Task Force (IETF).

1.2.2 Característiques de l'LDAP

Existeixen altres tecnologies que ofereixen informació a un client, però els serveis de directori que implementen el protocol LDAP ofereixen alguns avantatges significatius:

1. **Consulta ràpida de dades:** l'objectiu d'un servidor de directori és la recuperació ràpida d'informació. L'LDAP ofereix un mètode lleuger de connexió a un dipòsit, lectura de dades i tancament de la connexió.
2. **Solució distribuïda:** com que l'LDAP és un protocol natiu de xarxa d'arquitectura distribuïda, pot distribuir a tota la xarxa informació llesta per ser usada per totes les aplicacions. Fins i tot pot reproduir una part del directori per donar un servei amb certes característiques en un entorn determinat. O moure (delegar) la informació a diversos llocs sense afectar cap accés extern a aquestes dades.
3. **Independència de la plataforma:** com que l'LDAP és un protocol estàndard, que proporciona un mètode d'accés a dades remot i local, és possible intercanviar completament la implementació de l'LDAP sense afectar la forma en que es podrà accedir a les dades. Hi ha una àmplia gamma de implementacions. Per suposat, el client i el servidor es poden executar en sistemes operatius diferents.

4. **Seguretat integrada en el repositori:** la informació d'accés s'emmagatzema en el mateix repositori. Alguns productes poden treballar amb drets i permisos de granularitat fina i també proporcionen la capacitat de controlar els drets d'accés en funció de factors externs, com ara l'adreça IP del client. Com que aquests controls s'executen de manera centralitzada pel servidor, són fàcils de mantenir. Els clients, per tant, no s'han de preocupar d'aquests detalls.
5. **Implementació fàcil del client:** la disponibilitat d'interfícies de programació (API) de l'LDAP per a gairebé qualsevol llenguatge de programació facilita la compatibilitat de l'LDAP amb pràcticament totes les aplicacions.
6. **Gran difusió:** com que l'LDAP és un protocol estàndard, molts proveïdors l'han adoptat al seu programari. Per exemple, IBM Tivoli, Microsoft Active Directory, Novell eDirectory o RedHat Directory Server.
7. **Baix cost:** l'LDAP és un protocol estàndard i no cal pagar llicències pel seu ús ni per disposar de clients o servidors (de codi font obert).

Característiques de l'LDAPv3

L'LDAPv2 es va dissenyar amb l'objectiu de ser implementat amb facilitat. Encara que ja realitza les principals operacions, va ser substituït per l'LDAPv3 perquè tenia algunes limitacions. Actualment només es recomana la compatibilitat amb la versió 3 de l'LDAP.

Algunes de les característiques particulars de l'LDAPv3 són:

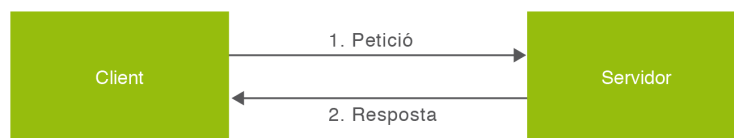
1. La majoria dels elements de dades del protocol es poden codificar com a cadenes normals.
2. Els valors dels atributs i els noms distintius s'han internacionalitzat mitjançant l'ús del conjunt de caràcters ISO 10646 (UTF-8).
3. El protocol pot ser ampliat per permetre noves operacions.
4. El resultat retornat poden ser referències. Això significa que un servidor de directoris que no contingui les dades sol·licitades les pot demanar a un altre servidor de directori o pot informar el client d'on les pot trobar.
5. Els clients poden demanar l'esquema que descriu les estructures de dades disponibles al directori.
6. Hi ha atributs operacionals (amb finalitats administratives) gestionats pel servidor.
7. Disposa de protecció per contrasenya, ús de certificats digitals i accés SSL.
8. Els mecanismes SASL (capa d'autenticació i seguretat) es poden utilitzar amb l'LDAP per proporcionar serveis de seguretat.

1.2.3 Funcionament de l'LDAP

En el model client-servidor de l'LDAP, davant d'una consulta concreta d'un client, el servidor contesta amb la informació sol·licitada. La resposta pot ser, de forma alternativa (o a més de la informació que s'havia sol·licitat), un punter que indica on aconseguir aquesta informació o dades addicionals (habitualment, el punter apunta a un altre servidor de directori).

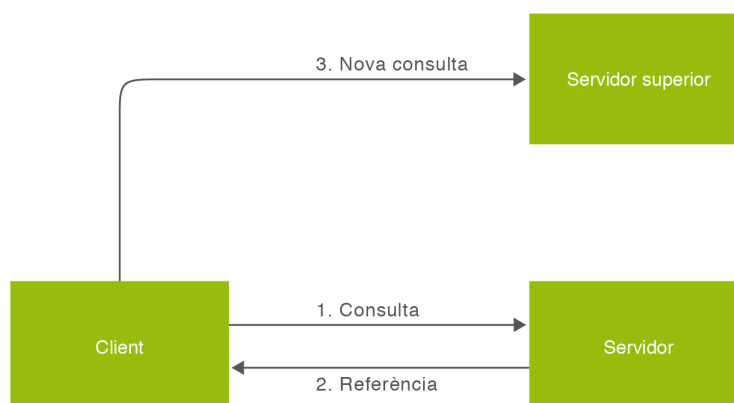
Aquesta configuració client-servidor es pot fer amb un únic servidor, com es pot veure en la figura 1.2. Amb aquest funcionament es proveeix el servei de directori per a una sola organització.

FIGURA 1.2. LLDAP com a servei local



Es pot decidir separar el directori entre diversos servidors per motius organitzatius o per facilitar-ne la gestió. En aquest cas es poden delegar parts del directori a un altre servidor, com s'observa en la figura 1.3. Aquesta configuració també es fa servir si es vol que el directori formi part d'un directori global. El servidor està configurat per tornar referències a altres servidors LDAP en el cas que se li demani informació de la qual no disposa, però que sap on aconseguir.

FIGURA 1.3. LDAP amb delegació

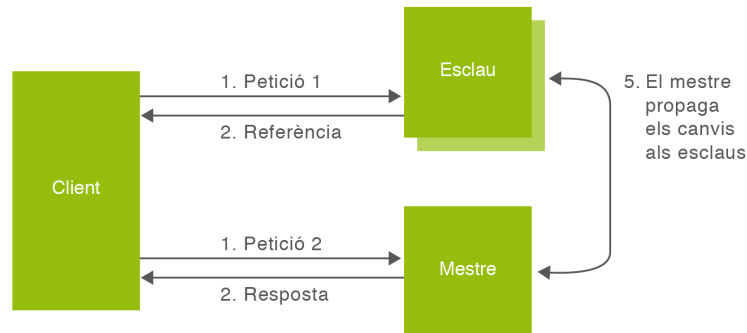


Es pot augmentar la disponibilitat i la fiabilitat del directori fent servir més d'un servidor LDAP per mantenir la informació. D'altra banda, durant un temps limitat pot manca sincronització entre mestre i esclaus.

L'actualització de dades es produeix a nivell d'entrada i no d'atribut, de manera que si es modifica un atribut, el mestre envia als esclaus tota l'entrada, cosa que és problemàtica quan hi ha entrades grans o modificacions freqüents.

A més de la configuració mestre-esclau com la que mostra la figura 1.4, hi ha implementacions amb replicació multimestre (les actualitzacions es repliquen a diversos nodes que poden actuar com a mestre).

FIGURA 1.4. LDAP amb replicació



1.2.4 Missatges LDAP

L'LDAP s'implementa directament a TCP/IP i fa servir cadenes simples per transportar les dades. És un protocol orientat a missatges, és a dir, tan bon punt arriba un missatge al receptor, l'operació de recepció ha acabat, perquè ja s'ha rebut tota la informació referent a aquell missatge.

Tots els missatges d'anada i tornada entre el servidor i el client —les peticions que el client envia al servidor, els resultats que el servidor envia al client i els codis de resultat—viatgen en aquest format.

El procés de negociació és el següent:

1. El client LDAP envia una sol·licitud al servidor LDAP.
2. El servidor executa una o diverses operacions (segons el que s'especifiqui a la sol·licitud).
3. En el cas d'una transmissió correcta, el servidor envia els resultats. En cas d'error s'envia al client el codi d'error corresponent.

Cada missatge LDAP, tant si és una petició com una resposta, conté la identificació del missatge, un codi d'operació i les dades. L'identificador del missatge es necessita per determinar la sol·licitud a la qual correspon una resposta, dada molt important, perquè el mode l'LDAP no requereix que les sol·licituds i respostes es facin de forma síncrona.

1.3 Els models LDAP

L'LDAP és un estàndard i no pas un maquinari o programari que es pot comprar. El que s'instal·la en l'equip client o servidor és la implementació d'aquest protocol; la qüestió de com emmagatzemar o tractar les dades es deixa als proveïdors de l'aplicació de la norma final.

Implementacions del protocol LDAP

Existeixen diverses implementacions del protocol LDAP realitzades per diferents companyies, entre d'altres:

- Active Directori: és la implementació de Microsoft en els seus sistemes operatius Windows Server.
- RedHat Directory Server o 389 Directory Server: una implementació realitzada per RedHat/Fedora.
- ApacheDS: un servei de directori que ofereix l'Apache Software Foundation.
- OpenDS: una implementació Java del protocol LDAP.
- OpenLDAP: una implementació lliure de l'estàndard.

Tot i la llibertat d'implementació, el sistema pot caracteritzar-se segons algun dels quatre models següents:

1. El **model d'informació** descriu l'estructura de la informació emmagatzemada en el directori LDAP.
2. El **model de noms** descriu com s'organitza i identifica la informació en el directori LDAP.
3. El **model funcional** descriu quines operacions poden ser realitzades amb la informació emmagatzemada en el directori LDAP.
4. El **model de seguretat** descriu com es pot protegir la informació continguda en el directori LDAP davant d'intents d'accés no autoritzats.

1.3.1 Model d'informació

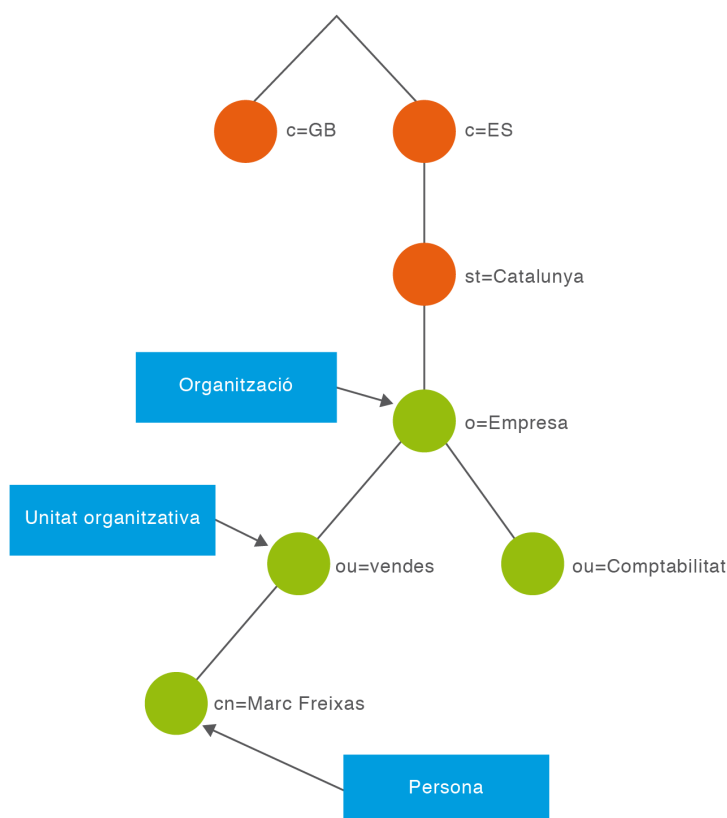
El model d'informació defineix quin tipus d'informació es pot emmagatzemar en el directori i les unitats bàsiques en què l'LDAP estructura la informació.

DIT

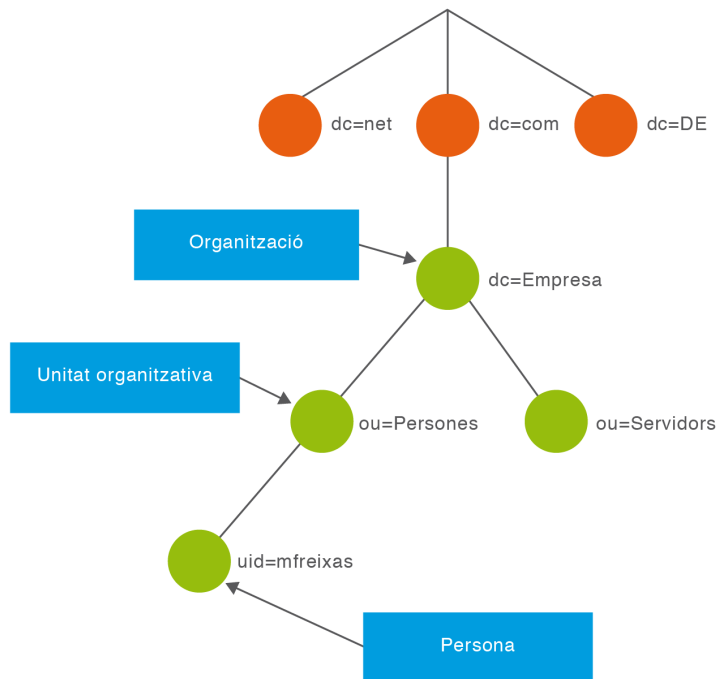
A l'LDAP la informació està estructurada en forma d'arbre. Aquest arbre s'anomena **arbre d'informació del directori** o **DIT** (Directory Information Tree).

Tradicionalment, aquesta estructura ha reflectit límits geogràfics i organitzatius. A la part superior de l'arbre s'han representat els països, i a sota, els estats, les organitzacions, les unitats organitzatives, les persones, els ordinadors, les impressores, els documents o altres conceptes. Aquesta estructura es mostra en la figura 1.5.

FIGURA 1.5. Arbre de directori LDAP amb nomenclatura tradicional



L'arbre del directori també es pot estructurar en funció dels noms de domini d'Internet (DNS). Aquesta pràctica està cada vegada més estesa, ja que permet trobar un servidor LDAP realitzant una consulta DNS. En la figura 1.6 es pot veure un exemple d'arbre de directori amb una organització basada en el DNS. Aquest tipus de representació és el més habitual a dia d'avui a l'hora d'implementar serveis de directori per administrar dominis informàtics.

FIGURA 1.6. Arbre de directori LDAP amb nomenclatura DNS

Entrada (entry)

Cada node de l'arbre s'anomena *entrada*. Una entrada és una col·lecció d'atributs identificada per un **nom distintiu** (*distinguished name* o **DN**). Aquest DN és únic al directori i fa referència a l'entrada de manera unívoca (de la mateixa manera que un identificador caracteritza de manera unívoca un registre en una base de dades relacional).

El DN de cada entrada del directori es representa mitjançant una cadena de caràcters formada per parells de la forma:

```
1 <tipus_tribut>=<valor>[,<tipus_tribut>=<valor>]
```

Aquest nom representa la ruta des de la posició de l'entrada fins a l'arrel de l'arbre (*base*, segons la nomenclatura LDAP). Com que se suposa que un directori es fa servir per emmagatzemar els objectes d'una organització determinada, la base serà la ubicació d'aquesta organització. Així, la base es converteix en el sufix de totes les entrades del directori.

Cada entrada té una entrada pare (*parent*) i pot tenir o no entrades fills (*child*). Cada entrada fill és un germà (*sibling*) de les altres entrades fills del mateix pare.

Exemples de noms distintius

En l'arbre de la figura 5 la base seria "o=Empresa,st=Catalunya,c=ES" i el DN de la persona representada seria "cn=Marc Freixas,ou=Vendes,o=Empresa,st=Catalunya,c=ES".

En l'arbre de la figura 6 la base seria "dc=Empresa,dc=com" i el DN de la persona representada seria "uid=mfreixas,ou=Persones,dc=Empresa,dc=com".

Atributs (attributes)

Les entrades estan formades per un conjunt d'atributs. Cada atribut d'una entrada té un tipus i un valor (SINGLE-VALUE) o més (MULTI-VALUE, que és el comportament predeterminat). El tipus d'atribut té associat els valors permesos i el seu comportament a l'hora de fer comparacions, ordenacions o treballar amb subcadena.

Els tipus són habitualment cadenes de text mnemòniques, com poden ser *cn* per fer referència a *common name* (nom comú), *sn asurname* (cognom) o *mail* a *email address* (adreça de correu electrònic).

La sintaxis dels valors dependrà del tipus de l'atribut, per exemple, l'atribut *cn* podria contenir el valor "Marc Freixas", l'atribut *mail* podria contenir un valor com "<mailto:mfreixas@empresa.com>" i un atribut *jpegPhoto* contindria una imatge en format binari.

Classes d'objecte (objectClass)

Els atributs que conté una entrada estan condicionats per les classes d'objecte a les quals pertany. Una classe d'objecte agrupa conjunts d'atributs que poden ser opcionals (indicat amb MAY) o obligatoris (indicat amb MUST) i la unió de tots els atributs de totes les classes d'objecte de les quals forma part una entrada són els atributs permesos per aquesta entrada.

La classe d'objecte es reconeix amb un identificador (OID) i opcionalment el nom (NAME).

L'atribut *objectClass* pot ser de tres tipus:

1. *objectClass* estructural: s'utilitza per crear entrades (objectes de dades). Cada entrada ha de tenir una classe d'aquest tipus i prou.
2. *objectClass* auxiliar: es pot afegir a qualsevol entrada. Per exemple, *dcObject* (que inclou l'atribut *dc*, *domain component*, per fer referència a noms DNS).
3. *objectClass* abstracta: utilitzada per tractar entitats inexistents.

La classe d'objecte abstracta més comú és top, que constitueix el nivell més alt de les jerarquies d'objecte estructurals. Cal tenir en compte que una entrada només pot tenir una classe d'objecte abstracta.

Herència

Totes les classes d'objectes tenen un origen comú, la classe d'objecte top. També és possible que una classe d'objecte sigui part d'una jerarquia. En aquest cas hereta totes les característiques de les classes d'objecte pare (classes superiors o SUP). En particular cal fer servir tots els atributs obligatoris de les classes superiors, a més dels atributs addicionals necessaris que defineixen la nova classe.

Exemple d'herència a les classes d'objecte

La classe d'objecte `inetOrgPerson` deriva de la classe d'objecte `organizationalPerson`, que al seu torn deriva de `person`, que, a la fi, té el pare top. Per tant, `inetOrgPerson` inclourà els atributs de les classes d'objecte top, `person`, `organizationalPerson` i les que ella mateixa hagi definit.

Si una classe d'objecte és part d'una jerarquia ha de ser del mateix tipus (estructural o auxiliar) que l'*objectClass* superior. L'excepció a aquesta regla és si el superior és un tipus abstracte (per exemple, top).

Regles de coincidència (matching rules)

Les regles de coincidència defineixen els mètodes de comparació disponibles al servidor LDAP.

En general no cal definir-les de manera explícita, sinó que el servidor ja les inclou per defecte.

Les regles de coincidència es fan servir per a cada atribut mitjançant les propietats opcionals EQUALITY, SUBSTR i ORDERING, a les que s'assigna els seus noms auto-descriptius o el seu OID.

Consulteu el document "Sintaxi de l'esquema" de la secció "Annexos" del material web.

Esquemes (schemas)

Els esquemes són unitats d'embalatge: en general totes les classes d'objecte i atributs estan definits dins dels esquemes. S'ha de considerar que, inicialment, el servidor de directori no coneix cap esquema, de manera que cal carregar-hi els arxius d'esquema que contenen la descripció de les classes d'objecte que el directori ofereix.

Els esquemes següents són d'ús habitual:

1. *core.schema*: per a les classes i els atributs bàsics.
2. *cosine.schema*: per a algunes extensions útils com es defineix en l'RFC 1274, com ara la identificació d'usuari, el correu...
3. *inetorgperson.schema*: útil en alguns casos quan es necessiten més atributs, com s'especifica en l'RFC 2798.

Els esquemes són reutilitzables: la creació d'un *objectClass* dins d'un nou esquema pot fer servir atributs d'*objectClass* definits en altres esquemes.

Excepcionalment, hi ha algunes classes d'objecte i atributs definits com a operacionals, que estan implícits en el programari del servidor LDAP i no necessiten un arxiu d'esquema. Aquests afecten totes les entrades d'un servidor i es fan servir a l'entrada "subschema", localitzable pel valor de l'atribut *subschemaSubentry* de qualsevol entrada.

Consulteu el document "Sintaxi de l'esquema" de la secció "Annexos" del material web.

Exemple de creació d'una classe d'objecte dins d'un esquema

A continuació es mostra la definició de la classe d'objecte *person*. Aquesta classe d'objecte està definida en l'esquema *core.schema*; és, per tant, una classe d'objecte bàsica.

```

1  attributetype ( 2.5.4.41 NAME 'name'
2  EQUALITY caseIgnoreMatch
3  SUBSTR caseIgnoreSubstringsMatch
4  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{32768} )
5
6  attributetype (2.5.4.3 NAME ( 'cn' 'commonName' ) SUP name )
7
8  attributetype (2.5.4.4 NAME ( 'sn' 'surname' ) SUP name )
9
10 objectclass ( 2.5.6.6 NAME 'person' SUP top STRUCTURAL
11   MUST ( sn $ cn )
12   MAY ( userPassword $ telephoneNumber $ seeAlso $ description )
13   )
14
15 dn: uid=jperez,ou=people,dc=exemple,dc=edu
16 objectclass: top
17 objectclass: person
18 cn: José Pérez
19 sn: Pérez

```

En fer servir *objectclass: person* per declarar una entrada, automàticament es requereix que els atributs *sn* (cognom) i *cn* (nom comú) continguin un valor, ja que així està establert en la definició de la classe d'objecte. A més, permet l'ús dels atributs *userPassword*, *telephoneNumber*, *seeAlso* i *description*, però no obliga a assignar-hi un valor.

Informació no textual en directoris LDAP

Tradicionalment s'associen els directoris amb informació de tipus text. Cada implementació ofereix un ventall de possibles tipus de dades. En general n'hi ha de molts tipus diferents, incloses les dades binàries. Però s'han desenvolupat opcions en el cas que calgui incloure informació que no s'ajusti als tipus disponibles (per exemple, mitjançant la codificació Base64).

Base64

Base64 és un sistema de numeració posicional que fa servir 64 com a base. És la major potència de dos que pot ser representada usant únicament els caràcters imprimibles de ASCII i això ha propiciat el seu ús per la codificació de diversos tipus d'informació.

1.3.2 Model de nomenclatura

El model de nomenclatura defineix com s'organitza i es referencia la informació. Les entrades de directori es disposen en una estructura arborescent jeràrquica per tal de reproduir fronteres polítiques, geogràfiques, d'organització o altres criteris.

Com cal esperar, igual que els noms de domini (www.google.com) o la ruta completa d'un fitxer (*/etc/passwd*), el nom d'una entrada ha de ser únic en cada servidor LDAP.

A diferència d'altres models, en aquest tots els nodes poden tenir contingut.

1.3.3 Nom distintiu (DN)

La posició d'una entrada en la jerarquia del DIT ve determinada pel seu nom complet (DN). Cada component d'un DN es diu *nom distintiu relatiu* (RDN).

No hi ha normes específiques sobre com construir aquest nom distintiu, l'única cosa important és que aquest nom ha de ser únic dins de l'espai de noms. És a dir, tots els fills d'una entrada han de tenir RDN diferents. Un RDN pot estar compost per un o més parells atribut/valor. És similar als noms de domini, però el separador és una coma (*dn: uid=mfreixas,ou=Personal,dc=empresa,dc=com*). També hi poden haver RDN multivalor (*dn: cn=Marc Freixas+uid=mfreixas,ou=Personal,dc=empresa,dc=com*)

Però en el cas que la entrada serveixi per a la autenticació de l'usuari (operació *bind*), cal indicar el paràmetre "Bind DN" i opcionalment una contrasenya. Com que no es produeix cap operació de cerca, cal que el "Bind DN" coincideixi amb el DN de creació de l'entrada.

Consulteu el document "Sintaxi del DN" de la secció "Annexos" del material web.

1.3.4 Sufix de directori (directory suffix)

Totes les entrades del directori tenen un node comú (arrel) que es diu *sufix del directori* o *base*.

L'LDAP permet construir el sufix del directori amb qualsevol d'aquests tres estils:

1. L'estil de noms X.500 (el nom de l'organització seguit del codi de país):
`o=nom_empresa,c=es.`
2. L'estil de DNS. La mateixa entrada ara queda: `o=nom_empresa.es.`
3. L'estil de component de domini utilitza l'atribut *dc* per separar el nom de domini en components *dc*: `dc=nom_empresa,dc=es.`

1.3.5 Àlies (alias)

Hi ha situacions en què és útil estalviar-se duplicar la inserció d'una entrada real en el DIT. Es pot crear una entrada àlies, amb un comportament similar al dels enllaços simbòlics (accessos directes) en un sistema de fitxers.

Un àlies pot apuntar fins i tot a un servidor de directori. Com que un àlies erroni pot provocar un desastre, no totes les implementacions de directoris permeten l'ús d'àlies. Si la implementació no permet àlies, pot utilitzar referències.

El procediment de creació és el mateix que s'usa per afegir una nova entrada: amb

un DN i classe d'objecte àlies, i només hi ha un atribut necessari, *aliasedObjectName*, que indica on és l'entrada real.

Exemple de creació d'un àlies

```
1 dn: cn=barcelona,dc=empresa,dc=com
2 objectClass: top
3 objectClass: alias
4 objectClass: extensibleObject
5 cn: barcelona
6 aliasedObjectName: cn=bcn,dc=empresa,dc=com
```

1.3.6 Referències (referrals)

Pot arribar un punt en què ja no és pràctic o eficient mantenir l'arbre del directori en un únic servidor. Per motius de rendiment pot interessar fer particions (*partitioning*), ja que tenir més servidors que ofereixen serveis de directori permet distribuir les peticions dels clients entre aquests servidors. O pot ser interessant per raons administratives, per permetre diferents polítiques per a diferents parts de l'arbre de directoris.

El procediment de creació és el mateix que s'usa per afegir una nova entrada: amb un DN i classe d'objecte referral, i només hi ha un atribut necessari, *ref*, que indica on resideix l'entrada real amb una URL LDAP.

Exemple on es referencia un subarbre del directori

Un servidor gestiona *dc=empresa,dc=com*, però un altre servidor, anomenat *srvB*, conté la informació de *dc=bcn,dc=empresa,dc=com*:

```
1 dn: DC=bcn,DC=exemple,DC=edu
2 objectClass: referral
3 objectClass: extensibleObject
4 dc: bcn
5 ref: [[ldap://srvB/DC=bcn,DC=exemple,DC=edu|ldap://srvB/dc=bcn,dc=
    =exemple,dc=edu]]
```

1.3.7 Arrel DSE (rootDSE)

La informació operacional del servidor és en una entrada anomenada *arrel DSE*. DSE vol dir “entrada DSA específica”. DSA vol dir “agent de servidor de directori” i conté atributs del servidor.

L'entrada “arrel DSE” té un DN de longitud zero i conté:

1. La versió LDAP compatible amb el servidor. Si el client no aconsegueix

recuperar aquesta informació, l'usuari pot concloure que és l'LDAP versió 2, que no és compatible amb aquesta funció.

2. Totes les classes d'objecte i tipus d'atributs reconeguts pel servidor.
3. Els sufixos emmagatzemats al servidor de directori.
4. Les operacions ampliades i els controls compatibles amb el servidor de directori.
5. Informació sobre els mecanismes de suport SASL.
6. Una llista de servidors LDAP per consultar la informació que el servidor original no pot proporcionar.

1.3.8 Model funcional

El model funcional defineix com es recupera i modifica la informació del directori, descrivint les operacions que es poden realitzar en l'accés, el manteniment i la gestió del directori.

Hi ha funcions que permeten buscar, comparar, afegir, modificar i esborrar entrades al directori. Totes són atòmiques, és a dir, l'operació s'executa en la seva totalitat o s'avorta si ocorre un error. Acompanyant al paquet d'instal·lació d'un servidor LDAP, s'acostuma a oferir un conjunt de programes per fer servir des de l'interpret de línia d'ordres que implementen l'accés a aquestes funcions.

A continuació descrivim les operacions agrupades per funcionalitat, sense aprofundir en la gestió d'errors.

Operacions d'autenticació i control

La primera operació és obrir la connexió amb el servidor (*bind*); després el client pot proporcionar al servidor les credencials de l'usuari (per exemple, *uid* i *userPassword*) per provar la seva identitat. Si el servidor accepta aquestes credencials, associa o "uneix" certs drets d'accés a les credencials d'usuari fins que es tanca la connexió (*unbind*) o s'envien noves credencials per obtenir drets d'accés diferents.

Connexió (*bind*)

Permet que el client s'autentiqui al servidor de directori.

Paràmetres:

1. *version*: la versió de l'LDAP que el client vol utilitzar.
2. *name*: nom de l'objecte de directori al qual el client desitja unir-se (un DN).
En cas de referències o àlies no es resoldrà.

3. *authentication*: autenticació d'elecció, que té dos valors possibles:

- (a) *simple*: indica que el missatge viatja en text clar.
- (b) *sasl*: utilitza el mecanisme d'autenticació i seguretat de la capa SASL com es descriu en l'RFC 4752.

Desconnexió (unbind)

Allibera els recursos assignats al client, descarta la informació d'autenticació i tanca la connexió. No cal cap paràmetre ni es retorna cap valor.

Abandonar (abandon)

S'informa al servidor que aturi una operació prèviament sol·licitada. El servidor no envia cap resposta.

Paràmetre:

- 1. *operationID*: identificació de l'operació que s'abandona.

Operacions interrogatives

Les operacions interrogatives són les que s'utilitzen tant per buscar com per llegir les entrades. Són les següents:

- 1. Cercar (*search*)
- 2. Comparar (*compare*)

Cercar (search)

Sol·licita a un servidor que retorni el conjunt d'entrades coincidents amb un criteri de cerca complex.

Paràmetres:

- 1. *baseObject*: un DN base per iniciar la recerca.
- 2. *filter*: un filtre que defineix les condicions que ha de complir la cerca perquè coincideixi amb una entrada determinada.
- 3. *scope*: àmbit d'aplicació: en el nivell del DN (*baseObject*), en un nivell inferior (*singleLevel*) o en el conjunt del subarbre (*wholeSubtree*).
- 4. *Attributes*: quins atributs s'han de retornar. Hi ha dos casos especials:
 - (a) llista buida (sense atributs) o *%x2A*, és a dir, un asterisc, *. Ambdós valors indiquen que es retornin tots els atributs de les entrades coincidents.
 - (b) *%x31.2E.31*, és a dir, *1.1*: si és l'únic OID de la llista, no es retorna cap atribut, però sí els DN. Si no és l'únic s'ignora.

5. *derefAliases*: indica com s'ha de fer la resolució de les referències:
 - (a) *neverDerefAliases*: no resol les referències dels àlies.
 - (b) *derefInSearching*: resol la referència d'àlies en les entrades subordinades a l'objecte base en la cerca, però no per localitzar l'objecte base de la cerca.
 - (c) *derefFindingBaseObj*: resol la referència d'àlies per localitzar l'objecte base de la cerca, però no en les entrades subordinades a l'objecte base en la cerca.
 - (d) *derefAlways*: resol sempre les referències.
6. *sizeLimit*: el nombre màxim d'entrades a retornar. "zero" significa que el client no imposa cap límit de mida. El servidor pot imposar un límit màxim.
7. *timeLimit*: nombre màxim de segons que una consulta pot trigar. "zero" significa que el client no imposa cap límit de temps. El servidor pot imposar un límit màxim.
8. *typesOnly*: un valor booleà. Establert a "true" indica que només es retornen els tipus d'atributs. En cas d'estar establert a "false", retorna els tipus d'atributs i els valors dels atributs.

Comparar (compare)

Permet a un client comparar una afirmació de valor amb els valors d'un atribut en particular en una entrada particular. El resultat és `compareTrue` en cas positiu, `compareFalse` en cas negatiu, o algun codi d'error, si no.

Paràmetres:

1. *entry*: el nom de l'entrada que s'ha de comparar (un DN). En cas de referències o àlies no es resol.
2. *ava*: una afirmació de valor d'un atribut que serà comparada.

Operacions d'actualització

Tot i no ser les més habituals sobre un directori, les operacions d'actualització tenen una gran importància a l'hora de mantenir la correcció de les dades.

Les operacions d'actualització són les següents:

1. Afegir (*add*)
2. Esborrar (*delete*)
3. Modificar (*modify*)
4. Modificar DN (*modify DN*)

Afegir (add)

Demana l'addició d'una entrada.

Paràmetres:

1. *entry*: nom complet de la nova entrada (un DN). En cas de referències o àlies no es resol.
2. *attributes*: una llista de parells nom-valor dels atributs continguts en l'entrada.

Esborrar (delete)

Demana l'eliminació d'una entrada.

Paràmetre:

1. *entry*: nom complet de la entrada a eliminar (un DN). En cas de referències o àlies no es resol.

Modificar (modify)

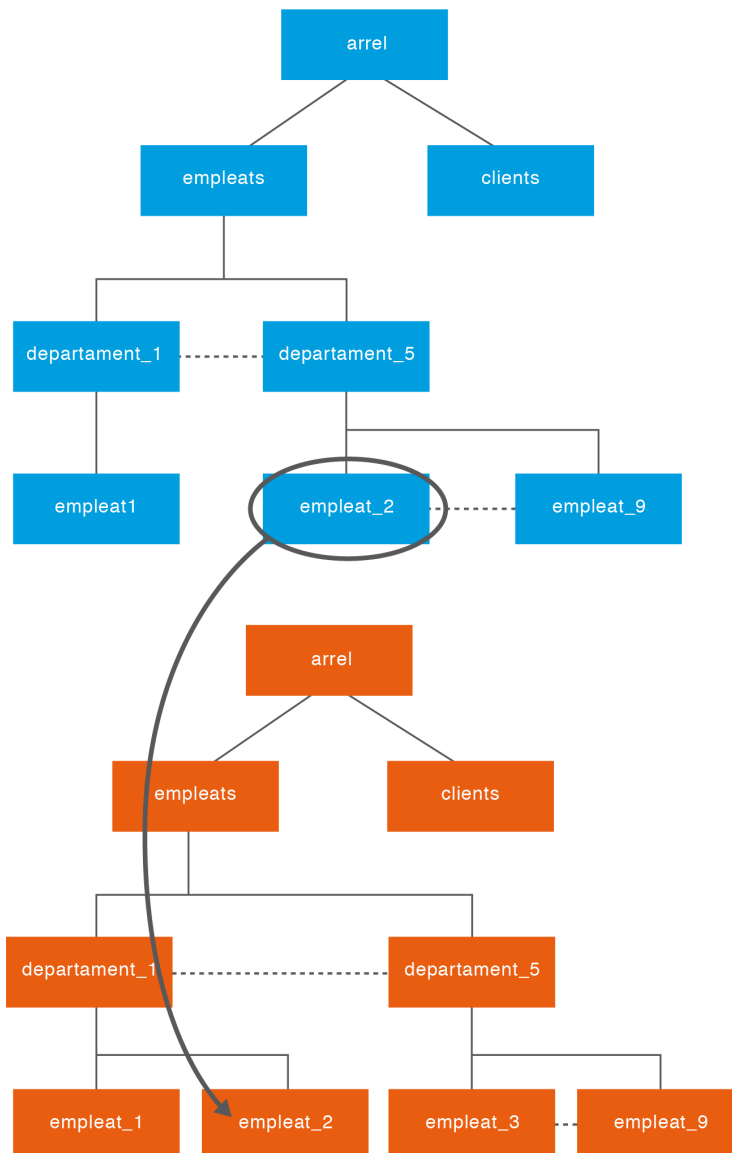
Modifica una entrada.

Paràmetres:

1. *object*: nom complet de la entrada a canviar (un DN). En cas de referències o àlies no es resol.
2. *changes*: una llista de modificacions a realitzar. Les opcions són:
3. *operation*: tipus d'operació que s'executa en aquesta entrada, amb tres valors possibles:
4. *add*: afegeix un nou atribut.
5. *delete*: elimina un atribut.
6. *replace*: modifica un atribut.
7. *modification*: una llista de parelles nom-valor que s'afegeix o modifica.

Modificar DN (modify DN)

Canvia l'RDN d'una entrada o passa un subarbre d'entrades a una nova ubicació al directori (vegeu la figura 1.7).

FIGURA 1.7. Modificació del DN**Paràmetres:**

1. *entry*: nom complet de la entrada a canviar (un DN). En cas de referències o àlies no es resol. En l'exemple és `uid=empleat_2,ou=departament5,ou=empleats,dc=exemple,dc=edu`.
2. *newrdn*: nou nom complet relatiu (un RDN). En l'exemple és `uid=empleat_2`.
3. *deleteoldrdn*: valor booleà que indica si l'RDN vell s'ha de mantenir en el directori.
4. *newSuperior*: és un paràmetre opcional que indica quin serà el superior immediat d'*entry*. En cas de referències o àlies no es resol. En l'exemple és `ou=departament1,ou=empleats,dc=exemple,dc=edu`.

Operacions ampliades

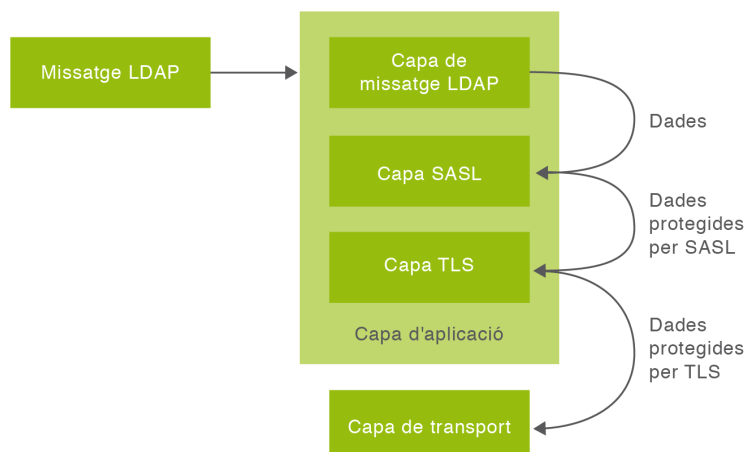
Les operacions ampliades es poden utilitzar per definir les noves operacions que no eren part de l'especificació del protocol original.

L'operació estesa més habitual és StartTLS.

StartTLS

L'operació StartTLS (definida a l'RFC 2830) permet a un client sol·licitar una connexió de tipus Transport Layer Security (el descendent d'SSL) mitjançant la transmissió d'una petició ampliada (ExtendedRequest, LDAPv3). Si el servidor és capaç i està en disposició de negociar TLS, el client ha de començar la negociació TLS (RFC 5246) o tallar la connexió. En la figura 1.8 es mostren les capes de seguretat en una sessió LDAP.

FIGURA 1.8. Capes de seguretat SASL i TLS en una sessió LDAP



1.3.9 Model de seguretat

El model de seguretat mostra com es controla l'accés a la informació continguda en el directori.

Abans que un client pugui accedir a les dades d'un servidor LDAP, s'han de dur a terme dos processos: autenticació i autorització.

El model de seguretat (RFC 4513) descriu aquests processos juntament amb la integritat i confidencialitat en la transmissió de dades o la limitació en l'ús de recursos.

Autenticació

L'autenticació és el procés que assegura que les identitats dels usuaris i màquines (servidors o clients) estan correctament validades.

Normalment, la seva funció es redueix a verificar la identitat de l'usuari abans de permetre-li l'accés al sistema. Aquest control permet definir el nivell d'accés de cada usuari i objecte del directori.

Els serveis de directori s'utilitzen habitualment com a eines d'autenticació que emmagatzemen credencials de manera centralitzada, no només contrasenyes per autenticar els usuaris del mateix directori, sinó també mecanismes per verificar contrasenyes que autèntiquin usuaris d'altres sistemes. Tanmateix, el protocol no imposa que les contrasenyes associades amb un usuari estiguin al servidor, sinó que permet que s'integrin amb serveis d'autenticació externs (per exemple, SASL).

Autenticació anònima

Tots els servidors LDAP han d'acceptar el tipus d'autenticació "sense autenticació en absolut", també anomenat *usuaris anònims*, ja que el servidor no sap qui està demanant una connexió. Aquest tipus d'autenticació té associada una autorització de tipus anònima.

S'aconsegueix escollint la opció d'autenticació simple a *bind*, però sense proporcionar credencials d'usuari al servidor.

Connexió no autenticada

Una connexió no autenticada dóna lloc a una autorització anònima, però la identificació facilita la traçabilitat de la connexió.

S'aconsegueix escollint la opció d'autenticació simple a *bind*, però sense proporcionar cap contrasenya al servidor.

Autenticació bàsica

L'autenticació bàsica també s'utilitza en altres protocols com HTTP. El client simplement envia les credencials d'usuari per la xarxa en format clar.

S'aconsegueix escollint l'opció d'autenticació simple a *bind*, proporcionant nom distintiu i contrasenya al servidor. El servidor busca un atribut anomenat *user-Password* en l'entrada corresponent al nom distintiu i el compara amb l'entrada enviada pel client. Si la contrasenya coincideix, s'estableix la connexió amb el servidor LDAP. Si no, s'envia un missatge d'error i es tanca la connexió.

SASL

És el mètode recomanat pel protocol LDAP. SASL Separa els mecanismes d'autenticació dels protocols de l'aplicació, permetent que qualsevol protocol d'aplicació que faci servir SASL utilitzi qualsevol mecanisme d'autenticació suportat per

SASL (RFC 4422 i 4752). La capa d'autenticació i seguretat (SASL) proporciona serveis d'autenticació a un protocol orientat a la connexió, com el LDAP. Un cop el servidor i el client estan connectats, estableixen un acord sobre el mecanisme de seguretat.

Els clients poden determinar els mecanismes SASL que un servidor admet llegint l'atribut *supportedSASLMechanisms* des de l'arrel DSE.

A més de l'autenticació, ofereix seguretat de dades (integritat i confidencialitat).

Alguns mecanismes d'autenticació definits per SASL són:

- EXTERNAL (per exemple, IPSec o TLS)
- ANONYMOUS
- PLAIN (text clar)
- OTP (One Time Password)
- SKEY
- CRAM-MD5
- DIGEST-MD5
- NTLM
- GSSAPI
- GATEKEEPER

Contrasenyes

El protocol LDAP (RFC 4519) ofereix *userPassword*, un tipus d'atribut multivalor definit per donar suport a l'operació *bind* amb autenticació simple i accessible únicament pel propi usuari. En ser multivalor, per validar l'usuari es prova la contrasenya candidata amb cada valor (útil en períodes de transició). Aquest atribut apareix en classes d'objecte com *organizationalUnit* o *person* i a la seva definició es diu que s'emmagatzema com a text clar.

Algunes implementacions aprofiten el text clar per fer ús de més tipus de mecanismes d'autenticació, fent servir una funció resum (*one-way hash*) i guardant el resultat codificat en Base64.

L'RFC 3112 oficialitza aquesta pràctica i defineix l'*authentication password schema* per guardar valors derivats de les contrasenyes de l'usuari amb un nou tipus d'atribut, *authPassword*, que permet esquemes d'emmagatzemament múltiple i regles de coincidència per validar que una contrasenya en text clar coincideix amb un dels valors de l'atribut. Aquest atribut apareix a classes d'objecte com *posixAccount*, *posixGroup* o *shadowAccount*. Els clients poden determinar els esquemes que un servidor suporta llegint l'atribut *supportedAuthPasswordSchemes* des de l'arrel DSE (per exemple, BASE64, CLEAR, CRYPT, MD5, SHA, SMD5, SSHA, SSHA256, SSHA384, SSHA512 o SASL).

Per exemple, si la contrasenya és abcd123:

```

1 abcd123 xifrat amb SHA = fDYHu0YbzxlE6ehQ0mYPIfS28/E=
2 userPassword: {SHA}fDYHu0YbzxlE6ehQ0mYPIfS28/E=
3
4 {SHA}fDYHu0YbzxlE6ehQ0mYPIfS28/E= codificat en Base64 =
   e1NIQX1mRfLIIdU9ZYnp4bEU2ZWhRT21ZUElmUzI4L0U9
5
6 authPassword:: e1NIQX1mRfLIIdU9ZYnp4bEU2ZWhRT21ZUElmUzI4L0U9

```

La creació d'atributs nous en esquemes personalitzats per guardar contrasenyes no es considera una bona pràctica.

En cap cas no és convenient fer la transferència de la contrasenya ni del *hash*. Per això és important saber que si no s'utilitza SSL/TLS o SASL amb algorisme de negociació segur el client LDAP enviarà les credencials amb text clar i serà el servidor qui calcularà el valor de *hash* i el comparará amb el valor emmagatzemat.

D'altra banda, és important crear una política de contrasenyes, encara que la seva implementació i compliment no sigui fàcil (impliqua que els usuaris canviïn les seves contrasenyes periòdicament, requisits de construcció de les contrasenyes, restringir la reutilització d'una contrasenya vella, impedir els atacs d'endevinació de contrasenyes...).

Internet draft

Internet draft són esborranys vàlids per a un màxim de sis mesos que poden ser actualitzats o reemplaçats per altres documents o esdevenir obsolets en qualsevol moment. No és apropiat l'ús d'*Internet drafts* com a material de referència ni citar-los d'altra manera que com a "treballs en curs".

Consulteu l'adreça web "Model de polítiques de contrasenyes" a la secció "Adreces d'interès" del material web per accedir a la desena versió d'un *Internet draft* que recull un conjunt de regles que controla com s'utilitzen i administren les contrasenyes en un servei de directori basat en l'LDAP.

La classe d'objecte `pwdPolicy` es defineix amb atributs que mantenen la informació de l'estat de la directiva de contrasenyes general per a cada usuari. La seva declaració és:

```

1 ( 1.3.6.1.4.1.42.2.27.8.2.1
2   NAME 'pwdPolicy'
3   SUP top
4   AUXILIARY
5   MUST ( pwdAttribute )
6   MAY ( pwdMinAge $ pwdMaxAge $ pwdInHistory $ pwdCheckQuality $
7     pwdMinLength $ pwdMaxLength $ pwdExpireWarning $
8     pwdGraceAuthNLimit $ pwdGraceExpiry $ pwdLockout $
9     pwdLockoutDuration $ pwdMaxFailure $ pwdFailureCountInterval $
10    pwdMustChange $ pwdAllowUserChange $ pwdSafeModify $
11    pwdMinDelay $ pwdMaxDelay $ pwdMaxIdle ) )

```

pwdAttribute és el nom de l'atribut al qual s'aplica la directiva (per exemple, *userPassword*). Es recomana exigir un nivell de qualitat de les contrasenyes, canvis periòdics i bloqueig (temporal o no) de comptes en superar el límit d'intents.

Autorització i control d'accés

Una política de control d'accés és un conjunt de regles que defineixen la protecció dels recursos en termes de les capacitats de les persones o entitats que accedeixen a aquests recursos.

La sol·licitud pot estar associada a una àmplia varietat de factors de control d'accés (ACFs), per exemple, l'adreça IP d'origen, el nivell de xifrat, el tipus d'operació que se sol·licita, l'hora del dia...

Cada sessió LDAP té associat un estat d'autenticació vinculat a una identitat d'autorització.

L'autenticació anònima i la connexió no autenticada comporten un estat d'autorització anònima, mentre que una autenticació bàsica comporta un estat d'autorització autenticada. SASL permet, opcionalment, que el client comuniqui la identitat d'autorització desitjada.

Cada implementació estableix els seus propis mecanismes d'administració i emmagatzemament. L'RFC 2820 descriu els requeriments de control d'accés, però no en facilita la implementació, cosa que dificulta la interoperabilitat en aquest aspecte.

El control d'accés es gestiona normalment en forma de llistes de control d'accés (ACL). Així, les polítiques de control d'accés sovint s'expressen en termes d'identitats d'autorització: "L'entitat X pot realitzar l'operació Y del recurs Z".

Per exemple:

```
1 access to * by * read
2
3 access to *
4     by self write
5     by anonymous auth
6     by * read
7
8 olcAccess: to *
9     by users read
```

Consulteu el document "Sintaxi de configuració dinàmica del control d'accés a l'OpenLDAP" de la secció "Annexos" del material web.

Privacitat i integritat de les dades

És important assegurar que les dades no es modifiquen ni es donen a conèixer durant la seva transmissió.

L'LDAP amb SSL/TLS

El protocol SSL (Secure Sockets Layer) implementa mecanismes de seguretat basats en criptografia de clau pública a la pila de protocols TCP/IP entre la capa de transport i la capa d'aplicació, és a dir, la capa LDAP.

TLS proporciona un mecanisme de seguretat que permet l'autenticació mútua del servidor i el client, la negociació segura i fiable del protocol d'encryptació i les claus criptogràfiques, tot això abans que el protocol d'aplicació (LDAP) envii o rebí el seu primer *byte* de dades.

L'LDAP li dóna suport mitjançant l'operació ampliada StartTLS i, opcionalment, ofereix autenticació quan es combina amb SASL EXTERNAL.

El protocol TLS 1

Transport Layer Security va néixer a partir de la versió 3 de l'SSL per oferir confidencialitat i integritat a la comunicació entre dues aplicacions. Es descriu als RFC 5246, 5746, 5878 i 6176. Les característiques del TLS, ordenades per prioritats del disseny, són: seguretat criptogràfica, interoperabilitat, extensibilitat i eficiència.

SASL

Aquest tipus de connexió ja s'ha presentat com a opció d'autenticació.

El client sol·licita una connexió segura mitjançant l'operació ampliada *StartTLS*. Si el servidor respon que està llest per negociar amb el protocol *TLS Handshake* de TLS, primer es posen d'acord en la versió del TLS i després en el mecanisme de seguretat que s'utilitzarà per a la comunicació. Ambdues parts decideixen si accepten o no el nivell de seguretat assolit. Si el servidor o el client decideixen que el nivell no és suficient, es tanca la connexió (TLS).

Protecció contra la monopolització

Per tal de garantir la disponibilitat del servei, cal limitar l'ús de recursos per evitar que un únic usuari monopolitzi el sistema.

L'LDAP ofereix límits d'autocontrol en la mida del resultat o temps d'algunes operacions. D'altra banda, l'estàndard recomana definir límits administratius mitjançant els controls de servei per tal de protegir el servidor, però no ho concreta, de manera que cada implementació ha desenvolupat la seva pròpia solució.

1.4 El format d'intercanvi de dades LDIF

L'LDIF (LDAP Data Interchange Format o format d'intercanvi de dades LDAP) és un estàndard de format de text pla utilitzat per representar el contingut d'un directori LDAP i les sol·licituds d'actualització del directori (afegir una entrada, modificar-la, eliminar-la...).

Podem crear fitxers de text pla amb format LDIF per representar la informació que conté un directori i també es poden fer servir per modificar el contingut del directori.

LDIF representa el contingut d'un directori amb una sèrie de registres, cadascun dels quals representa una entrada del directori. De la mateixa manera, una sol·licitud d'actualització del directori (adició, modificació, eliminació) es representa també com un registre (de fet, hi haurà un registre per a cada sol·licitud).

L'ús d'un fitxer de text pla és molt útil. En primer lloc és autoexplicatiu, fàcil de llegir i d'entendre. A més, és molt fàcil de produir o processar utilitzant scripts o algun llenguatge de programació. D'altra banda, com que la compatibilitat entre

les diferents implementacions de servei de directori no està garantida, sempre es poden fer servir els fitxers en aquest format per transportar informació d'un servei de directori a un altre, és a dir, per importar i exportar informació entre servidors LDAP.

El format de fitxer LDIF està definit a l'RFC 2849.

1.4.1 Format d'un fitxer LDIF

Els fitxers LDIF estan formats per un conjunt de línies de diversos tipus. Els tipus de línies que es distingeixen són els següents:

- **Línia de directiva:** una línia que comença amb qualsevol caràcter, excepte espai o coixinet. Aquestes són les línies realment importants, ja que contenen la informació de les entrades i les operacions a realitzar.
- **Línia en blanc:** les línies en blanc s'utilitzen normalment per separar les diferents seqüències d'entrada.
- **Línia de comentari:** una línia que comença amb un coixinet.
- **Línia de continuació:** una línia que segueix una línia de directiva i comença amb un espai. Se suposa que els caràcters següents són part de la línia anterior.
- **Línia de separació:** una línia que comença amb un guió (–). Les línies de separació són utilitzades típicament per acabar les seqüències d'operador.

Els fitxers LDIF són molt sensibles als espais i les línies en blanc. S'ha de ser molt respectuós amb la sintaxi del format.

S'anomena **seqüència** a l'**agrupació de directives** separades per una o més línies en blanc. Existeixen dos tipus de seqüències: d'entrada i d'operador.

Seqüència d'entrada

Les seqüències d'entrada defineixen una entrada del directori. Són un grup de directives que comencen amb una directiva **DN (nom distintiu)**, seguida d'altres directives que descriuen els diferents atributs d'una entrada al DIT. Les seqüències d'entrada sempre comencen amb la directiva DN i acaben amb una línia en blanc.

Exemple de seqüència d'entrada

```
1 dn: ou=Persones, dc=empresa,dc=com
2 ou: Persones
3 objectClass: organizationalUnit
```

L'exemple mostra una entrada del directori, concretament una unitat organitzativa. Inclou **tres directives**: la primera defineix el DN i les altres dues descriuen diferents atributs de l'entrada o objecte. Especifiquen que l'objecte és una unitat organitzativa i donen valor a l'atribut *ou*.

Seqüència d'operador

Una seqüència d'operador és un grup de directives que inclou la directiva **changetype**. Una seqüència d'operador acaba amb una línia de separació o una línia en blanc.

La directiva *changetype* admet com a valors:

- **add**: afegeix entrades.
- **delete**: elimina entrades.
- **modify**: modifica entrades, però cal especificar quin atribut volem modificar i quina és la modificació que cal realitzar. Aquestes modificacions poden ser *add*, *delete* o *replace*. Per fer diverses modificacions en una entrada es fa servir el separador -.

Exemple de seqüència d'operador changetype: modify

```
1 dn: cn=Marc Freixas,ou=Persones,dc=empresa,dc=com
2 changetype: modify
3 add: work-phone
4 work-phone: 931234561
5 work-phone: 931234562
6 -
7 delete: home-fax
8 -
9 replace: home-phone
10 home-phone: 415/697-8899
```

En l'exemple es modifica l'entrada "Marc Freixas". S'hi han afegit dos telèfons de la feina, s'ha esborrat el fax del domicili i s'ha modificat el telèfon particular. Es pot observar com després del DN apareix la directiva *changetype*. Com que hi ha més d'una modificació, cada modificació se separa de la següent mitjançant un guionet (-).

- **modrdn**: modifica el DN relatiu, és a dir, sense canviar la base. Cal fer servir la directiva **newrdn** per definir el nou DN. Es pot decidir si es vol mantenir o eliminar l'antic DN amb la directiva **deleteoldrdn**.

Exemple de directiva modrdn

```
1 dn: cn=Marc Freixas,ou=people,dc=example,dc=com
2 changetype: modrdn
3 newrdn: Marc Freixas Vila
4 # deletes old RDN entry
5 deleteoldrdn: 1
```

En l'exemple s'ha canviat el nom de la persona de "Marc Freixas" a "Marc Freixas Vila" i s'ha esborrat l'anterior DN.

- **moddn**: modifica el DN complet. Això implica moure l'objecte a un altre node del DIT. S'ha d'especificar el nou pare de l'objecte amb la directiva **newsuperior**. Opcionalment, es pot mantenir o esborrar l'entrada anterior amb la directiva **deleteoldrdn**.

Exemple de directiva moddn

```
1 dn: cn=Marc Freixas,ou=Persones,dc=empresa,dc=com
2 changetype: moddn
3 newsuperior: ou=exempleats,dc=empresa,dc=com
4 # Mantenim l'anterior entrada RDN
5 deleteoldrdn: 0
```

En l'exemple, Marc Freixas ha passat a tenir com a nou DN “cn=Marc Freixas,ou=exempleats,dc=empresa,dc=com”.

2. Instal·lació, configuració i manteniment del servei de directori

El programari OpenLDAP és una implementació del protocol LDAP que ens permet instal·lar, configurar i mantenir el servei de directori d'una organització. Aquest programari és compatible amb la majoria de distribucions de Linux i amb altres sistemes operatius, com Android, Mac OS X, Solaris, Microsoft Windows (NT i derivats, és a dir, 2000, XP, Vista, Windows 7...) i z/OS.

2.1 Introducció a l'OpenLDAP

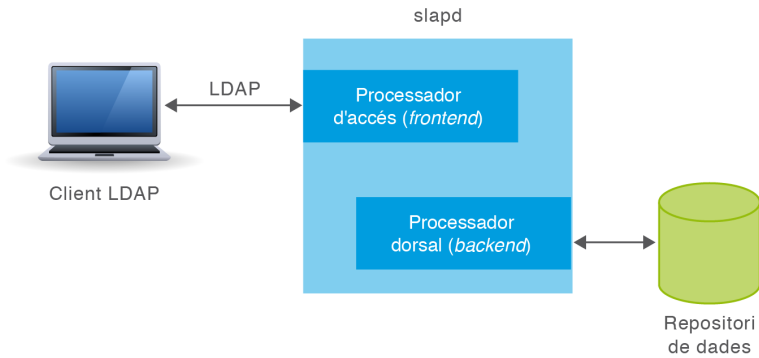
L'OpenLDAP és una implementació lliure, de codi obert, del Lightweight Directory Access Protocol (LDAP) desenvolupat pel projecte OpenLDAP. Disposa d'una llicència anomenada OpenLDAP Public License. El projecte el va iniciar Kurt Zeilenga clonant el codi font de referència de la Universitat de Michigan, que és on es va desenvolupar originalment el protocol LDAP.

Els components bàsics de la implementació OpenLDAP són els següents:

1. El **servidor slapd** (*stand-alone LDAP daemon*): un dimoni que escolta per diferents ports (per defecte pel port 389) peticions de connexió LDAP.
2. Les llibreries de client LDAP: llibreries que implementen el protocol LDAP i que poden ser utilitzades per desenvolupar programari client amb accés al protocol LDAP.
3. Utilitats, eines i diversos clients de mostra.

L'arquitectura del servidor slapd està dividida en una secció de processament frontal (*frontend*), que controla les connexions de xarxa i el processament del protocol, i una secció de processament dorsal o de segon pla (*backend*), que s'encarrega de l'emmagatzematge i la recuperació de les dades en resposta a les peticions que es reben. Aquesta arquitectura es mostra en la figura 2.1.

FIGURA 2.1. Arquitectura OpenLDAP



Vegeu l'apartat "Configuració dinàmica del servei" d'aquesta unitat per a més informació sobre la configuració `cn=config`.

Com que l'arquitectura és modular, hi ha una gran varietat de *backends* disponibles per interactuar amb diferents tecnologies, no només amb bases de dades tradicionals. Entre d'altres, podem emmagatzemar el directori (o també generar de forma automàtica la informació del directori) mitjançant:

1. **back-bdb**: el primer *backend* transaccional per a l'OpenLDAP, construït a partir d'Oracle BerkeleyDB. Es tracta d'una base de dades molt utilitzada a la indústria.
2. **back-hdb**: una variant de back-bdb que és totalment jeràrquica i permet reanomenar subarbres.
3. **back-ldif**: fa servir fitxers LDIF de text. És el *backend* per defecte per a la configuració `cn=config`.
4. **back-ndb**: un *backend* transaccional construït a partir del motor d'emmagatzematge de dades NDB de MySQL. NDB (Network DataBase) permet l'emmagatzematge de dades de forma distribuïda en una xarxa.
5. **back-ldap**: servidor intermediari simple a altres servidors LDAP. És a dir, utilitza altres servidors LDAP per emmagatzemar i recuperar la informació.
6. **back-sql**: estableix connexions a bases de dades SQL fent que puguin ser utilitzades com a magatzem de dades.

A més de la secció de processament frontal i les de processament dorsal o de segon pla, hi poden haver *overlays*. Els *overlays* són components de programari situats en el processador dorsal que n'augmenten les funcionalitats. Així, es pot modificar el comportament del processador sense haver-ne de reescriure el codi o modificar-ne la resposta abans de tornar-la al processador frontal.

Existeix una gran quantitat d'*overlays*. Entre d'altres:

1. **Password Policy**: defineix polítiques de contrasenyes (longitud mínima, caducitat...).
2. **AccessLogging**: manté un registre dels accessos a un *backend* determinat en una base de dades diferent.

3. **Audit Logging:** manté un registre de les modificacions efectuades sobre un *backend* determinat.
4. **Value Sorting:** permet ordenar els valors d'un atribut multivalor.

2.2 Planificació del servei de directori

Abans d'implementar un directori cal fer una planificació anticipada d'alguns aspectes:

1. Definir per a què s'utilitzarà, és a dir, definir els continguts del directori.
2. Decidir quina serà l'organització de les dades. Amb l'LDAP la informació es representa en forma d'arbre, per tant, abans de començar hem de pensar quina forma tindrà aquest arbre d'informació del directori o DIT (Directory Information Tree).
3. Definir aspectes de seguretat de les dades.
4. Definir aspectes de rendiment.

La definició d'aspectes de seguretat i rendiment queden fora de l'abast d'aquesta unitat i només tractarem els dos primers punts.

2.2.1 Escenari d'exemple

Un dels usos més habituals dels directoris és permetre l'autenticació centralitzada dels usuaris d'una xarxa. Veurem un exemple de com fer servir el servidor OpenLDAP per realitzar l'acreditació centralitzada dels usuaris d'una organització.

Suposem que l'organització té les característiques següents:

1. El nom de l'empresa és *Empresa*.
2. L'empresa té comprat un domini Internet anomenat *empresa.com*.
3. L'empresa vol mantenir al directori les dades dels seus empleats, però també les dels clients i proveïdors.
4. L'empresa té dos departaments: administració i departament comercial.

Partint d'aquest supòsit es poden fer diversos dissenys per organitzar les dades del directori.

El primer que cal és establir l'arrel. La manera més habitual de fer-ho és utilitzant el nom del domini per definir la base del DIT. En aquest cas l'arrel

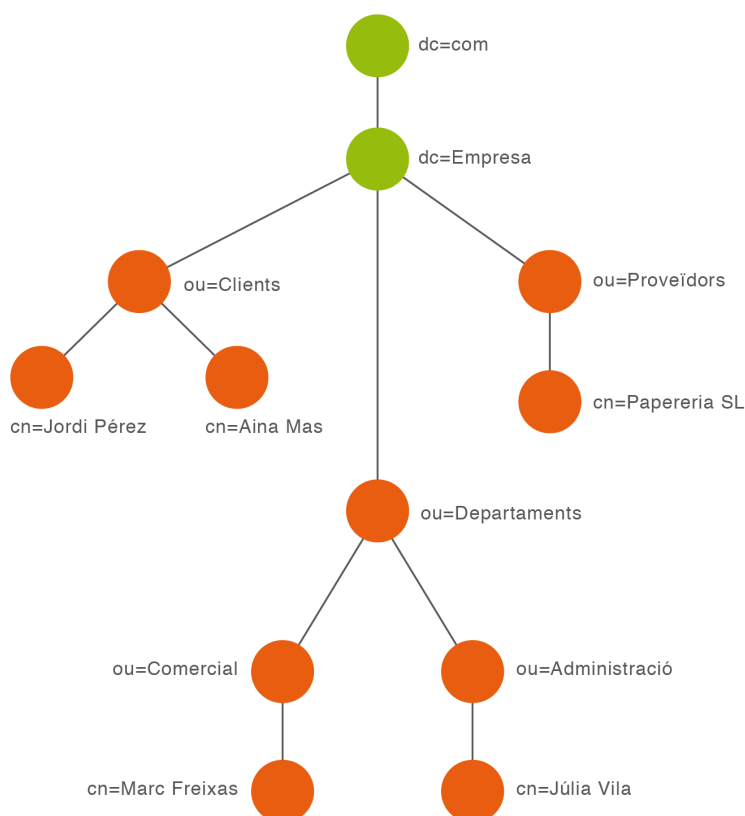
L'RFC 2307bis fa una sèrie de recomanacions de cara a fer servir un servidor LDAP per realitzar una acreditació centralitzada d'usuaris.

Vegeu a la secció d'adreces d'interès del web del mòdul l'enllaç al text complet de l'RFC 2307bis.

serà “dc=empresa,dc=com”. Una vegada definida l’arrel, cal estructurar la resta d’elements. Una classe d’objecte que permet organitzar les entrades de l’arbre per al nostre cas és la classe `organizationalUnit` (unitat organitzativa).

En la figura 2.2 es pot observar una proposta d’organització del directori. Aquesta figura està simplificada, ja que el DIT pot contenir molta més informació i cada entrada també té molts més atributs.

FIGURA 2.2. DIT d'una organització

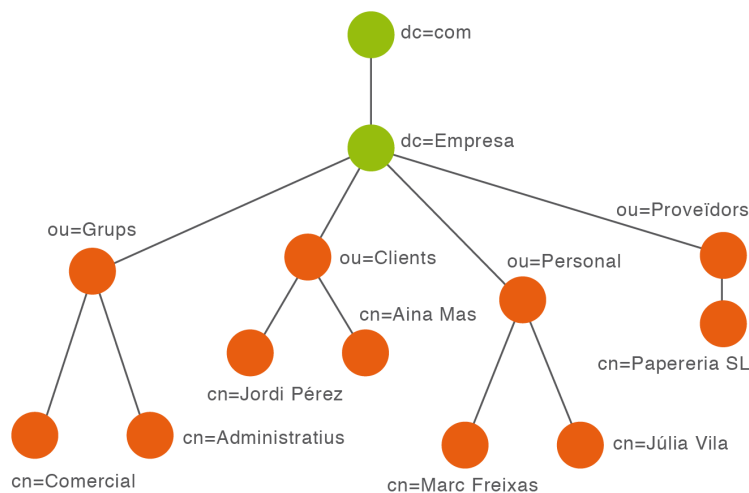


L’organització de la informació de la figura 2.2 té l’inconvenient que no segueix les recomanacions de l’RFC 2307bis. Bàsicament s’ha de tenir en compte que les organitzacions són canviants al llarg del temps, els departaments poden canviar de nom, els empleats poden canviar de departament. . . Per aquesta raó es proposa crear un DIT on hi hagi una unitat organitzativa per englobar els departaments (usualment l’anomenem *Grups*) i una sola unitat organitzativa que contingui tots els usuaris de l’empresa (nosaltres l’hem anomenat *Personal* però sovint la trobarem amb el nom de *People*). Aquesta estructura es mostra en la figura 2.3. A banda de Grups i Personal, crearem una unitat organitzativa addicional per a cada entitat externa a la nostra empresa. En aquest cas en creem una per als clients i una altra per als proveïdors.

L’estructura de la figura 2.3 és molt més adequada per construir un DIT per mantenir la informació dels usuaris i els departaments de l’organització, i també la podrem fer servir per configurar el sistema per fer l’acreditació centralitzada dels usuaris a partir del servei de directori. Per tant, aquesta estructura és la que pren-

drem com a model per implementar el directori amb el programari OpenLDAP i la que prendrem com a referència per entendre i seguir les explicacions d'aquesta unitat.

FIGURA 2.3. DIT d'una organització amb les recomanacions de l'RFC 2307bis



2.3 Instal·lació de l'OpenLDAP

La versió actual de l'OpenLDAP és la 2.4, que compleix l'LDAPv3. En el nostre cas en considerarem la instal·lació per al sistema operatiu Debian (6.0.3-Squeeze).

El programari es pot descarregar des del web del projecte OpenLDAP, però aquest només l'ofereix en forma de codi font. Podem optar per compilar el programari o per usar alguna compilació realitzada per tercers. En el cas de Debian GNU/Linux, el programari és compatible i està disponible als repositoris de la versió estable de Debian, així que se'n simplifica la instal·lació i configuració inicial.

2.3.1 Requeriments previs

Primerament cal disposar d'accés amb drets d'administrador (*root*) a un equip amb sistema operatiu Debian, preferiblement amb connexió a Internet i que no tingui instal·lat prèviament OpenLDAP.

Abans de fer servir apt-get o aptitude, ens hem d'assegurar que el sistema operatiu té el nom de la màquina i l'FQDN correctament configurats. Per comprovar-ho farem servir l'ordre *hostname*. Amb aquesta ordre coneixerem tant el nom de la màquina com l'FQDN. Instal·lem una màquina anomenada **servidor** al domini **empresa.com**. Podem comprovar que és correcte amb les ordres següents:

```

1 root@servidor:~# hostname
2 servidor

```

```
3 root@servidor:~# hostname -f
4 servidor.empresa.com
```

FQDN

FQDN (Fully Qualified Domain Name) és un nom de màquina que inclou a més el nom del domini associat a l'equip, i especifica la situació exacta de la màquina en la jerarquia DNS. Per exemple, si l'ordinador s'anomena *servidor* i el nom de domini és *domini.com*, l'FQDN serà *servidor.domini.com*.

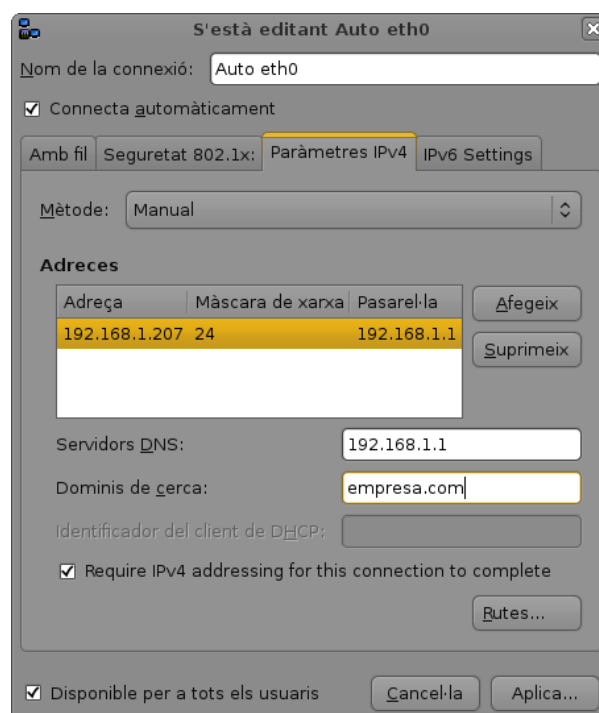
És absolutament necessari tenir correctament configurat el nom de la màquina i del domini abans de fer la instal·lació del servidor OpenLDAP. Per fer-ho a Debian Squeeze podem fer servir els fitxers `/etc/hostname`, on únicament hi ha el nom de la màquina, i `/etc/hosts`, que té un contingut similar a aquest:

```
1 127.0.0.1localhost
2 127.0.1.1servidor.empresa.com servidor
3
4 # The following lines are desirable for IPv6 capable hosts
5 ::1 ip6-localhost ip6-loopback
6 fe00::0 ip6-localnet
7 ff00::0 ip6-mcastprefix
8 ff02::1 ip6-allnodes
9 ff02::2 ip6-allrouters
```

L'altra cosa totalment necessària és configurar correctament la xarxa. No es pot tenir un servidor OpenLDAP amb una adreça IP dinàmica, ja que els clients han de conèixer aquesta adreça, ja sigui per la IP directament o per la consulta a un servidor DNS. Per fer aquesta configuració tenim dues opcions.

Si tenim l'entorn gràfic instal·lat i fem servir NetworkManager, l'haurem de fer servir per configurar la xarxa com mostra la figura 2.4. En aquest exemple estem suposant que el servidor té l'adreça IP fixa 192.168.1.207 i màscara per defecte.

FIGURA 2.4. Configuració TCP en mode gràfic



Si, per contra, l'entorn gràfic no està instal·lat, no disposarem del NetworkManager ni de la seva miniaplicació (*applet*) per a Gnome. Aleshores farem servir el fitxer `/etc/network/interfaces`:

```

1 # This file describes the network interfaces available on your system
2 # and how to activate them. For more information, see interfaces(5).
3
4 # The loopback network interface
5 auto lo
6 iface lo inet loopback
7
8 # The primary network interface
9 auto eth0
10 allow-hotplug eth0
11 iface eth0 inet static
12     address 192.168.1.207
13     netmask 255.255.255.0
14     gateway 192.168.1.1

```

2.3.2 Instal·lació del programari

Als repositoris de Debian es pot trobar el servidor OpenLDAP disponible per ser instal·lat mitjançant `apt-get` o `aptitude`. Com a superusuari executem l'ordre següent:

```

1 root@servidor:~# aptitude install slapd
2 ...

```

Abans de fer qualsevol instal·lació és convenient actualitzar la informació dels paquets dels repositoris mitjançant `apt-get update` o `aptitude update`.

La configuració per defecte es fa mitjançant `debconf`. L'únic que el sistema ens preguntarà és quina contrasenya volem definir per a l'administrador del servidor OpenLDAP.

debconf

`debconf` (Debian Package Configuration System) és un programa que s'utilitza per fer tasques de configuració del sistema. Està desenvolupat principalment per a la distribució de Linux Debian, i està integrat amb el sistema de gestió de paquets `dpkg` de Debian. Quan els paquets estan essent instal·lats, `debconf` pregunta a l'usuari qüestions que determinen el contingut de tot el sistema de fitxers de configuració associats al paquet. Després de la instal·lació del paquet, és possible tornar enrere i canviar la configuració d'un paquet utilitzant `dpkg-reconfigure` o un altre programa com `Synaptic`.

`debconf` tria per defecte les millors opcions per fer la configuració de l'`slapd`. Aquesta configuració està determinada principalment per l'`FQDN` del sistema; és per això que cal realitzar correctament la configuració del nom del sistema.

Si tot ha funcionat correctament, les darreres línies de la instal·lació ens comunicaran que tot ha anat bé i que s'ha iniciat el servidor `slapd`.

```

1 ...
2 S'està configurant slapd (2.4.23-7.2)...
3   Creating new user openldap... done.
4   Creating initial configuration... done.
5   Creating LDAP directory... done.
6 Starting OpenLDAP: slapd.
7 root@servidor:~#

```

Vegeu l'apartat "Verificació de la instal·lació" d'aquesta unitat per aprendre com fer una verificació de la configuració inicial.

Aquestes línies informen de les modificacions realitzades al sistema: s'ha creat un usuari, anomenat *openldap*, que és l'usuari amb el qual s'executarà el servei *slapd*, i s'ha creat la configuració inicial del servei i el directori LDAP.

Podem saber quins executables s'han instal·lat amb el paquet *slapd* mitjançant l'ordre següent:

```
1 root@servidor:~# dpkg -L slapd | grep bin
2 /usr/share/man/man5/slapo-pbind.5.gz
3 /usr/sbin
4 /usr/sbin/slappasswd
5 /usr/sbin/slapindex
6 /usr/sbin/slapd
7 /usr/sbin/slapauth
8 /usr/sbin/slapcat
9 /usr/sbin/slapacl
10 /usr/sbin/slapadd
11 /usr/sbin/slapdn
12 /usr/sbin/slaptest
```

Vegeu l'apartat "Utilitats de línia d'ordres" d'aquesta unitat per conèixer la utilitat dels executables instal·lats amb el paquet *slapd* i el paquet *ldap-utils*.

Les llibreries estan situades al directori */usr/lib/*.

Abans de prosseguir la configuració, és convenient instal·lar el paquet *ldap-utils*, un conjunt d'eines que podem utilitzar per operar com a clients de l'OpenLDAP i poder accedir al directori, afegir dades, modificar-les... Executem l'ordre següent:

```
1 root@servidor:~# aptitude install ldap-utils
```

Podem conèixer els executables instal·lats amb el paquet *ldap-utils* mitjançant l'ordre següent:

```
1 root@servidor:~# dpkg -L ldap-utils | grep bin
2 /usr/bin
3 /usr/bin/ldapdelete
4 /usr/bin/ldappasswd
5 /usr/bin/ldapexop
6 /usr/bin/ldapcompare
7 /usr/bin/ldapmodify
8 /usr/bin/ldapsearch
9 /usr/bin/ldapwhoami
10 /usr/bin/ldapmodrdn
11 /usr/bin/ldapurl
12 /usr/bin/ldapadd
```

2.3.3 Verificació de la instal·lació

Podem comprovar que el servei *slapd* s'està executant amb l'ordre següent:

```
1 root@servidor:~# ps -ef|grep slapd
```

O bé, com és habitual amb tots els serveis, mitjançant l'ordre:

```
1 root@servidor:~# service slapd status
```

També podem veure el port que fa servir (normalment és el 389):

```
1 root@servidor:~# netstat -tunlp | grep slapd
```

A més de comprovar que el dimoni slapd s'està executant i està connectat al port adequat, també hem de comprovar que el servei funciona correctament. Per fer aquesta comprovació es pot usar l'ordre *ldapsearch* (del paquet *ldap-utils*) amb els paràmetres següents:

```
1 # ldapsearch -x -LLL -H ldap:/// -b dc=empresa,dc=com dn
```

L'ordre anterior ens dona un resultat com aquest:

```
1 dn: dc=empresa,dc=com
2
3 dn: cn=admin,dc=empresa,dc=com
```

Vegeu l'apartat "Utilitats de línia d'ordres" d'aquesta unitat per a una explicació més completa de la sintaxi de l'ordre *ldapsearch*.

Aquesta sortida mostra que la instal·lació del paquet slapd ha creat per defecte una configuració bàsica del directori amb un DIT basat en l'FQDN del sistema (en aquest cas l'arrel és "dc=empresa,dc=com") i que ha creat un usuari administrador per al directori (tal com mostra l'entrada "cn=admin,dc=empresa,dc=com") que ens permetrà realitzar operacions de modificació en el directori.

Per defecte, el dimoni slapd també desa tota la informació relacionada amb la seva configuració (opcions globals de configuració, definicions d'esquemes, definicions de *backends* i bases de dades...) en objectes del directori a partir de la base anomenada *cn=config*. Podem comprovar la configuració per defecte del dimoni slapd amb l'ordre:

```
1 # ldapsearch -Q -Y EXTERNAL -LLL -H ldapi:/// -b cn=config dn
```

La sortida de l'ordre és:

```
1 dn: cn=config
2
3 dn: cn=module{0},cn=config
4
5 dn: cn=schema,cn=config
6
7 dn: cn={0}core,cn=schema,cn=config
8
9 dn: cn={1}cosine,cn=schema,cn=config
10
11 dn: cn={2}nis,cn=schema,cn=config
12
13 dn: cn={3}inetorgperson,cn=schema,cn=config
14
15 dn: olcBackend={0}hdb,cn=config
16
17 dn: olcDatabase={-1}frontend,cn=config
18
19 dn: olcDatabase={0}config,cn=config
20
21 dn: olcDatabase={1}hdb,cn=config
```

Vegeu l'apartat "Configuració dinàmica del servei" d'aquesta unitat per a una explicació més detallada de *cn=config*.

2.3.4 Reconfigurar el programari

La instal·lació del paquet `slapd` en un sistema operatiu configurat correctament crea un servidor OpenLDAP amb una configuració per defecte. Ara bé, el nom de la màquina podria no estar correctament configurat, o potser es vol fer alguna variació sobre la configuració estàndard. Si es vol repassar la configuració de l'`slapd` podem fer servir l'ordre:

```
1 # dpkg-reconfigure slapd
```

En aquest cas, `debconf` no només pregunta la contrasenya d'administrador del directori, sinó que fa més preguntes:

1. La primera és si volem ometre la configuració inicial. S'ha de contestar que no i procedir a fer-la.
2. La segona pregunta vol establir la relació entre el nom DNS i el directori que s'està creant. Demana el nom DNS del domini que s'administrarà per construir-ne la base (arrel del DIT). Per al cas de l'exemple s'especificarà "empresa.com". Si el `hostname` està configurat correctament, l'instal·lador pot agafar directament el nom correcte.
3. En tercer lloc demana el nom de l'organització. Pot ser "Empresa Exemple S.L." o qualsevol altre nom que es vulgui donar a l'organització. És habitual escriure el mateix que al punt anterior.
4. Ara ens demana la contrasenya de l'administrador del domini i la confirmació.
5. El protocol LDAP estableix quina informació s'ha d'emmagatzemar, però no defineix com s'ha de fer. L'OpenLDAP permet fer servir dos *backends* per emmagatzemar les dades: BDB (Oracle Berkeley DB) i HDB (Hierarchical Data Base, una variant jeràrquica de BDB), que és la més recomanada.
6. En aquest punt, l'`slapd` ens demana si volem eliminar la base de dades en el cas de purgar el paquet (eliminar el programa i els seus fitxers de configuració). És tasca de l'administrador decidir si es pot arriscar a perdre tota la informació en cas de desinstal·lar el paquet o si prefereix, per contra, mantenir totes les dades.
7. Si s'està fent una reconfiguració, l'instal·lador detectarà que ja hi ha dades d'una instal·lació anterior. No és convenient barrejar dades de dues instal·lacions, per tant és recomanable acceptar moure la base de dades antiga a un altre lloc. És molt **important**, si s'havia fet anteriorment una reconfiguració de la base de dades de l'OpenLDAP i per tant s'havia creat una còpia de seguretat, **moure o eliminar** aquesta **còpia de seguretat** abans de fer la reconfiguració, ja que si no apareixerà l'error següent:

1 Moving old database directory to /var/backups:

2 Backup path /var/backups/unknown-2.4.23-7.2.ldapdb exists. Giving up...

Per finalitzar, no és necessària la compatibilitat amb l'LDAPv2, al menys en la majoria de les situacions que ens podem trobar al llarg de la nostra feina com a administradors de directori.

2.3.5 Desinstal·lació del programari

Podem eliminar el programari OpenLDAP del sistema amb el gestor *aptitude*. Es pot utilitzar l'opció *remove* o l'opció *purge*. La diferència és que amb *remove* només eliminem el programari, mentre que amb *purge* eliminem també els fitxers de configuració i la base de dades del directori. Per exemple:

```
1 # aptitude purge slapd
```

Igualment, si hem instal·lat el paquet *ldap-utils* i el volem eliminar del sistema, podem fer-ho amb l'ordre *aptitude* i les opcions *remove* o *purge*:

```
1 # aptitude remove ldap-utils
```

2.3.6 Aturada i arrencada del servei

Per defecte, el sistema operatiu Debian 6 s'inicia en el nivell d'execució 2. Si comprovem els serveis que Debian inicia en aquest nivell, veurem que el dimoni *slapd* s'activa per defecte en iniciar-se el sistema:

```
1 root@servidor:~# ls /etc/rc2.d/S*slapd
2 /etc/rc2.d/S18slapd
```

Com qualsevol altre servei de Debian, es pot aturar el servei *slapd* mitjançant l'ordre *stop*:

```
1 root@servidor:~# service slapd stop
2 Stopping OpenLDAP: slapd.
```

I es pot tornar a activar manualment amb l'ordre *start*:

```
1 root@servidor:~# service slapd start
2 Starting OpenLDAP: slapd.
```

2.4 Configuració del servei de directori

Si bé la instal·lació del programari OpenLDAP no presenta cap dificultat, la correcta configuració del dimoni *slapd* requereix el coneixement i domini de

Vegeu l'apartat "El format d'intercanvi de dades LDIF" d'aquesta unitat per a més informació sobre el format de fitxer LDIF i la seva funcionalitat.

diversos conceptes, tant de l'estructura i funció dels directoris creats en temps d'instal·lació com del format de fitxer LDIF.

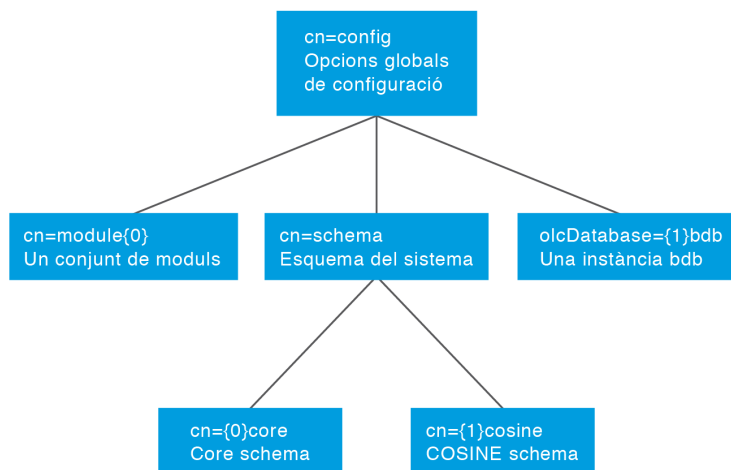
2.4.1 Configuració dinàmica del servei

Tradicionalment, la configuració del servidor OpenLDAP es realitzava mitjançant el fitxer `/etc/slapd.conf`. Es tractava d'una configuració estàtica, de manera que qualsevol canvi en la configuració comportava reiniciar el servei. Si el directori contenia moltes dades, aquesta aturada i posada en marxa necessitava un temps que podia ser inacceptable en determinats entorns.

A partir de la versió 2.3, l'OpenLDAP permet la configuració dinàmica del dimoni `slapd`. Aquesta configuració és coneguda com a *runtime configuration* (RTC) **cn=config**. La configuració de l'`slapd` s'emmagatzema en un directori especial LDAP amb un esquema i un DIT predefinit amb arrel a `cn=config`.

Hi ha entrades d'aquest directori que s'utilitzen per guardar les opcions de configuració global, les definicions d'esquemes, les definicions de *backends* i bases de dades i altres elements. La figura 2.5 mostra un exemple simplificat de configuració del directori.

FIGURA 2.5. DIT de configuració de l'`slapd`



El DIT de configuració de l'`slapd` té una estructura molt específica. La base de l'arbre és `cn=config`, que conté les opcions de configuració globals. La resta de configuracions estan a les entrades filles:

1. Mòduls carregats dinàmicament.
2. Definicions d'esquemes: l'entrada "`cn=schema,cn=config`" conté l'esquema del sistema (tot l'esquema que està programat directament al dimoni

Configuració estàtica o dinàmica

Tot i que en la versió 2.4 de l'OpenLDAP encara es pot fer servir el fitxer de configuració estàtica `/etc/slapd.conf`, en futures versions es retirarà el suport per a aquesta configuració. Debian 6 fa servir per defecte la configuració dinàmica `cn=config`.

slapd). Les entrades filles contenen els esquemes de l'usuari que poden ser afegits per configuració o de manera dinàmica durant l'execució del servei.

3. Configuració específica del *backend*.

4. Configuració específica de la base de dades: com a entrades filles d'aquesta entrada es defineixen els *overlays*.

Aquesta configuració s'emmagatzema en un *backend* LDIF. En el cas de Debian aquesta base de dades està situada al directori `/etc/ldap/slapd.d`. Amb l'ordre *tree* es pot visualitzar l'estructura de directoris i fitxers que conté el directori `/etc/ldap/slapd.d`:

```

1 root@servidor:~# tree /etc/ldap/slapd.d/
2 /etc/ldap/slapd.d/
3 |__ cn=config
4 |   |__ cn=module{0}.ldif
5 |   |__ cn=schema
6 |       |__ cn={0}core.ldif
7 |       |__ cn={1}cosine.ldif
8 |       |__ cn={2}nis.ldif
9 |       |__ cn={3}inetorgperson.ldif
10 |   |__ cn=schema.ldif
11 |   |__ olcBackend={0}hdb.ldif
12 |   |__ olcDatabase={0}config.ldif
13 |   |__ olcDatabase={-1}frontend.ldif
14 |   |__ olcDatabase={1}hdb.ldif
15 |__ cn=config.ldif
16
17 2 directories, 11 files

```

Podeu instal·lar l'ordre *tree* mitjançant l'ordre **`aptitude install tree`**.

Es pot navegar pel directori `slapd.d`, però no és gens convenient fer les modificacions directament sobre els fitxers LDIF de configuració, sinó que s'han de fer servir les eines pròpies de l'OpenLDAP (*ldapmodify*,...). Igualment, per afegir nous esquemes al directori cal fer servir les ordres correctes; no n'hi ha prou de copiar els diferents fitxers de configuració a una carpeta de l'arbre de directoris.

Amb la configuració `cn=config` modifiquem el comportament o la configuració del servidor `slapd` simplement modificant les entrades necessàries al DIT.

Exemple de configuració del nivell de registre (log) del dimoni `slapd`

Per defecte, el dimoni `slapd` té el nivell de registre definit a "none" (ho podeu comprovar al fitxer `/etc/ldap/slapd.d/cn=config.ldif`). Aquesta configuració provoca que pràcticament no quedi registrada cap activitat. La definició del nivell de registre es fa a l'atribut `olcLogLevel` de l'entrada `cn=config` (configuració general). Aquesta entrada és la següent:

```

1 dn: cn=config
2 objectClass: olcGlobal
3 cn: config
4 olcArgsFile: /var/run/slapd/slapd.args
5 olcLogLevel: none
6 olcPidFile: /var/run/slapd/slapd.pid
7 olcToolThreads: 1

```

Per modificar l'entrada, creem un fitxer LDIF (en aquest exemple el creem al nostre directori de treball amb el nom `fitxer.ldif`) amb el contingut següent:

```
1 # Modificació del nivell de registre.
2 dn: cn=config
3 changetype: modify
4 replace: olcLogLevel
5 olcLogLevel: stats
```

El nou nivell de registre el definim amb `stats`. Els diferents nivells de registre estan definits al manual d'administració de l'OpenLDAP. Per realitzar la modificació, farem servir el fitxer LDIF que hem creat i l'ordre `ldapmodify` del paquet `ldap-utils`:

```
1 # ldapmodify -QY EXTERNAL -H ldapi:/// -f ~/fitxer.ldif
2
3 modifying entry "cn=config"
```

A partir d'aquest moment, i sense necessitat de reiniciar el servei, el nivell de registre de l'`slapd` quedarà definit a "stats" i el dimoni enregistrarà els esdeveniments al fitxer definit a l'atribut `olcLogFile` de l'objecte `cn=config`.

2.5 Manteniment del directori

Un directori correctament instal·lat i perfectament configurat no servirà de res si no es fa servir per emmagatzemar informació i consultar-la. Si s'ha fet la instal·lació del programari a partir dels repositoris Debian, el mateix programa d'instal·lació realitza una configuració bàsica, que crea l'arrel del DIT per a l'organització especificada durant la instal·lació i una entrada amb un usuari que permet l'administració del directori.

L'administració del directori es pot fer amb una eina gràfica, ja sigui una aplicació nativa per un determinat sistema operatiu o una aplicació web, que es pot fer servir des de qualsevol navegador. També és possible realitzar-la mitjançant utilitats de línia d'ordres, que es poden fer servir de manera interactiva des d'un terminal, ja sigui local o remot, o que es poden utilitzar en scripts per automatitzar tasques.

2.5.1 Utilitats de línia d'ordres

Les utilitats de línia d'ordres són principalment usades per programadors i administradors de sistemes com a eina de treball, especialment en sistemes operatius basats en Unix, ja sigui localment o amb una connexió remota. El programari `openLDAP` ofereix utilitats per poder administrar el directori des d'una interfície de línia d'ordres. A més de les utilitats proporcionades pel paquet **`slapd`**, el paquet de programari **`ldap-utils`** proporciona una sèrie d'eines per a línia d'ordres que permeten la interacció amb el directori.

Utilitats proporcionades per l'slapd

Les utilitats proporcionades pel paquet `slapd` permeten controlar el funcionament del servidor. Són les següents:

1. ***slappasswd***: es fa servir per generar una contrasenya vàlida per fer servir amb *ldapmodify* per als atributs *userPassword* dels objectes del directori, per a la directiva *rootpw* del fitxer de configuració `slapd.conf` (ja no es fa servir) o per a la directiva de configuració *olcRootPW* de la configuració `slapd-config`.
2. ***slapindex***: reindexa les entrades en la base de dades de l'slapd.
3. ***slapd***: Stand-alone LDAP Daemon, el servidor LDAP.
4. ***slapauth***: serveix per comprovar el funcionament de l'autenticació dels usuaris.
5. ***slapacl***: permet comprovar l'accés a una llista d'atributs, obrint el *backend* de l'slapd-config i mirant les directives *olcAccess*.
6. ***slapadd***: serveix per afegir entrades al directori mitjançant fitxers LDIF.
7. ***slapdn***: serveix per comprovar si una entrada de l'arbre segueix les normes dels esquemes definits.
8. ***slaptest***: comprova que la configuració del fitxer `slapd.conf` sigui correcta, però aquest mètode de configuració ja no es fa servir. L'ordre també es pot utilitzar per convertir fitxers `.schema` en `.ldif`.
9. ***slapcat***: aquesta ordre retorna en format LDIF tot el contingut de la base de dades LDAP. Per exemple:

```
1 root@servidor:~# slapcat
2 dn: dc=empresa,dc=com
3 objectClass: top
4 ...
5 modifyTimestamp: 20120312011231Z
```

Utilitats proporcionades per ldap-utils

A més de les eines pròpies del servidor `slapd`, hi ha una sèrie d'utilitats per accedir com a client al servidor. Per fer-les servir s'ha d'instal·lar el paquet `ldap-utils`. Tot i que estan pensades per fer-se servir des d'un client, la seva instal·lació al servidor en facilitarà l'administració. Les més utilitzades són:

1. ***ldapdelete***: permet eliminar una entrada del directori.
2. ***ldappasswd***: canvia la contrasenya d'una entrada LDAP.
3. ***ldapexop***: permet iniciar operacions ampliades LDAP.

4. *ldapcompare*: permet realitzar comparacions al directori LDAP.
5. *ldapmodify*: modifica entrades al servidor LDAP.
6. *ldapsearch*: és l'eina de cerca per al servidor LDAP.
7. *ldapwhoami*: retorna l'usuari amb el qual s'està treballant. També permet comprovar la connexió d'usuari i el mot de pas.
8. *ldapmodrdn*: utilitat per reanomenar entrades del directori.
9. *ldapurl*: permet compondre o descompondre URIs LDAP.
10. *ldapadd*: afegeix entrades al servidor LDAP.

Exemples d'ús de les utilitats de línia d'ordres

Un cop s'ha configurat correctament un directori OpenLDAP, és el moment d'estructurar i emmagatzemar la informació de l'organització, com ara la informació relativa a usuaris i grups.

Vegeu l'apartat "Escenari d'exemple" d'aquesta unitat per veure l'estructura del DIT a partir del qual crearem els elements.

Creació d'unitats organitzatives, grups i usuaris

Per crear les unitats organitzatives des de la línia d'ordres podem fer servir l'ordre *ldapadd*. Aquesta ordre necessita un fitxer LDIF amb la informació que volem afegir al directori.

Tot i que amb una sola ordre LDIF es podria afegir tota la informació del directori, pot ser interessant a efectes didàctics o de depuració d'errors separar els diversos objectes que es volen inserir al directori en fitxers diferents. D'aquesta manera es pot entendre molt millor l'estructura de cada objecte i, a més, és més fàcil detectar errors de sintaxi o entrades duplicades.

Processament de fitxers LDIF sense aturada

Per evitar que s'aturi l'ordre *ldapadd* en afegir entrades si hi ha algun error al fitxer LDIF i que es continuï afegint la resta d'entrades, cal fer servir el paràmetre *-c*.

Per emmagatzemar objectes al directori cal definir una estructura. La manera habitual d'emmagatzemar els objectes en un directori és mitjançant **unitats organitzatives**. Les més comunes són **People** per als usuaris i **Groups** per als grups. Evidentment, cada situació o problemàtica requereix una solució adaptada a les necessitats particulars. L'administrador sempre té la potestat per decidir l'estructura organitzativa del DIT.

En l'exemple actual simplement s'insereixen dues unitats organitzatives, una per als usuaris anomenada *Personal* i una per als grups anomenada *Grups*.

Organització de l'entorn de treball

Quan es treballa en qualsevol sistema es fan servir fitxers temporals, diferents versions de documents, diferents versions de guions per automatitzar tasques i molts altres tipus de fitxers. Pot ser convenient crear una subcarpeta per emmagatzemar els fitxers LDIF que es van creant o recuperant del directori LDAP dins de la carpeta de treball personal pel simple fet de tenir-los localitzats i mantenir l'entorn mínimament ordenat.

En primer lloc crearem un fitxer LDIF (*ouPersonal.ldif*) amb el contingut següent per inserir-hi la unitat organitzativa *Personal*:

```
1 #Unitat organitzativa Personal
2 dn: ou=Personal,dc=empresa,dc=com
3 objectClass: organizationalUnit
4 ou: Personal
```

L'explicació és la següent:

1. La primera línia és un comentari que defineix el fitxer.
2. La segona línia defineix el DN o nom distintiu de l'objecte. Aquest nom serà el que identificarà l'objecte dins del directori. Per tant, ha de ser únic.
3. La tercera línia indica a quina classe pertany l'objecte. Es pot definir més d'una classe simplement afegint més línies.
4. La quarta línia defineix un atribut de l'objecte. En aquest cas és l'atribut *ou*. Sempre s'han de definir els atributs obligatoris; en cas contrari es produirà un error.

Per afegir l'entrada simplement s'executa l'ordre **ldapadd** assignant-li com a paràmetre el fitxer que s'ha creat. Per exemple:

```
1 root@servidor:~# ldapadd -x -D cn=admin,dc=empresa,dc=com -W -f ouPersonal.ldif
2
3 Enter LDAP Password: *****
4 adding new entry "ou=People,dc=example,dc=com"
```

Aquesta és l'explicació dels paràmetres de l'ordre:

1. **-x**: defineix la forma de connexió al servidor com a simple i no fa servir la connexió SASL per defecte.
2. **-D**: defineix amb quin usuari es realitzarà la connexió. Ha de ser un usuari amb els privilegis suficients per afegir una entrada a la part del directori corresponent. En l'exemple es connecta l'usuari administrador **cn=admin,dc=empresa,dc=com** especificant el seu DN.
3. **-W**: fa que es demani la contrasenya per la línia d'ordres. En cas contrari s'hauria d'especificar la contrasenya a la línia d'ordres mitjançant l'opció **-w** o en un fitxer de contrasenyes mitjançant l'opció **-y**.
4. **-f**: defineix la ruta del fitxer LDIF (en aquest cas **ouPersonal.ldif**) que conté la informació que volem afegir al directori.

Si tot ha funcionat correctament es pot afegir la unitat organitzativa següent, en el nostre cas, la que es farà servir per emmagatzemar els grups. Es crearà un fitxer que es pot anomenar *ouGrups.ldif*.

```
1 #Unitat organitzativa Grups
2 dn: ou=Grups,dc=empresa,dc=com
3 objectClass: organizationalUnit
4 ou: Grups
```

S'afegeix el fitxer al directori mitjançant l'ordre *ldapadd*.

Si es vol afegir més d'un objecte al directori en un sol fitxer LDIF només cal tenir en compte que els diferents objectes han d'estar separats per una línia en blanc. Així, les dues unitats organitzatives que s'han creat es podrien haver definit en un sol fitxer anomenat, per exemple, *ou.ldif*, de la manera següent:

```
1 #Unitat organitzativa Personal
2 dn: ou=Personal,dc=empresa,dc=com
3 objectClass: organizationalUnit
4 ou: Personal
5
6 #Unitat organitzativa Grups
7 dn: ou=Grups,dc=empresa,dc=com
8 objectClass: organizationalUnit
9 ou: Grups
```

Una vegada creades les unitats organitzatives es pot dir que ja està creada l'estructura bàsica del directori. És el moment de situar els grups (departaments) i usuaris (empleats). Sembla lògic començar per crear un grup abans de crear un usuari, ja que habitualment un usuari sempre pertany com a mínim a un grup.

Per crear el primer grup del directori es crea un fitxer que es pot anomenar *grupcomercials.ldif*, en el qual es defineix un grup anomenat *comercials*, que contindrà tots els comercials de l'empresa.

```
1 dn: cn=comercials,ou=Grups,dc=empresa,dc=com
2 objectClass: posixGroup
3 cn: comercials
4 gidNumber: 5001
```

En aquest cas, la classe d'objecte varia, ja que passa a ser un *posixGroup* (ja no és una *organizationalUnit*). En canviar la classe d'objecte, ja no és necessari definir un atribut *ou*, sinó que la nova classe demana els atributs *cn* (*commonName*) i *gidNumber*, que contindrà el *GID* (identificador de grup) amb el qual s'identificarà aquest grup LDAP al sistema.

Una vegada definit el grup, cal definir els usuaris. Es pot fer amb la definició de fitxers LDIF (que poden tenir el format *usuariMarcFreixas.ldif*) amb el contingut següent:

```
1 dn: uid=mfreixas,ou=Personal,dc=empresa,dc=com
2 objectClass: inetOrgPerson
3 objectClass: posixAccount
4 objectClass: shadowAccount
5 uid: mfreixas
6 sn: Freixas
7 givenName: Marc
8 cn: Marc Freixas
9 displayName: Marc Freixas
10 uidNumber: 10001
11 gidNumber: 5001
12 userPassword: marcpassword
13 gecos: Marc Freixas
14 loginShell: /bin/bash
15 homeDirectory: /home/comercials/mfreixas
```

És molt important fer servir un *GID* que no coincideixi amb els *GID* definits al fitxer */etc/group* del sistema client. És per això que habitualment es fan servir nombres elevats.

Cal analitzar aquesta definició, ja que té alguns punts importants:

1. En primer lloc cal destacar que es defineixen tres classes d'objecte. Això vol dir que aquest objecte haurà de contenir els atributs definits a les tres classes.
2. L'atribut *uid* definirà el nom de l'usuari que podrà iniciar sessió a un sistema client.
3. Existeixen molts atributs que donen informació sobre l'usuari: *sn* defineix el cognom, *givenName*, el nom propi, *displayName*, el nom que es mostrarà, i *cn*, el nom complet.
4. Els atributs *uidNumber* i *gidNumber* fan referència als identificadors d'usuari i de grup que aquest usuari tindrà en iniciar sessió des d'un sistema client. En aquest cas es fa que l'usuari "mfreixas" tingui com a grup primari el que hem creat, és a dir que és un comercial.
5. La contrasenya està en text pla en el fitxer LDIF. Sempre queda l'opció de no assignar-li cap contrasenya i després executar l'ordre de canvi de contrasenya de l'objecte.
6. *gecos*, *loginShell* i *homeDirectory* fan referència a la resta d'atributs posix necessaris, com són el nom complet, el *shell* de connexió al sistema i el directori de connexió al sistema o HOME.

Després d'afegir aquest darrer fitxer cal comprovar que tot està correctament situat. Per fer-ho es pot utilitzar l'ordre *slapcat* o realitzar una cerca sobre un objecte concret dels que s'han creat:

```

1 root@servidor:~# ldapsearch -x -LLL -b dc=empresa,dc=com 'uid=mfreixas' cn
2   gidNumber
3 dn: uid=mfreixas,ou=Personal,dc=empresa,dc=com
4 cn: Marc Freixas
5 gidNumber: 5001

```

És molt important fer servir un UID que no coincideixi amb els definits al fitxer */etc/passwd* del sistema client. És per això que habitualment es fan servir nombres elevats.

Els paràmetres que es fan servir són:

1. **-x**: per fer l'autenticació simple en comptes de la SASL per defecte.
2. **-LLL**: per mostrar el resultat de la cerca en format LDIF sense comentaris i sense mostrar la versió del format LDIF.
3. **-b**: és molt important, és la base des d'on comença la cerca, és a dir, l'arrel del DIT.
4. **uid=mfreixas**: és el filtre de cerca. Indica que s'està buscant aquest usuari i no un altre.
5. **cn**: són els atributs concrets que es demanen a la cerca que retorni. Si no es defineix cap atribut es retornaran tots els atributs, que és el comportament per defecte.

És molt important crear els objectes en l'ordre correcte. Per exemple, no podem afegir l'objecte **dn: uid=mfreixas,ou=Personal,dc=empresa,dc=com** si

prèviament no hem afegit l'objecte **dn: ou=Personal,dc=empresa,dc=com**, ja que intentar afegir un objecte a una unitat organitzativa que encara no està creada donaria error.

Realització de cerques en el directori

Un directori està pensat principalment per realitzar operacions de cerca i consulta. Per això l'ordre *ldapsearch* és una de les més potents del conjunt d'utilitats de l'OpenLDAP. L'ajuda de l'ordre en mostra la seva sintaxi particular:

```
1 usage: ldapsearch [options] [filter [attributes...]]
2 where:
3   filter      RFC 4515 compliant LDAP search filter
4   attributes  whitespace-separated list of attribute descriptions
5               which may include:
6     1.1      no attributes
7     *        all user attributes
8     +        all operational attributes
```

Aquesta ordre accepta unes opcions, un filtre de cerca i uns atributs determinats. Les opcions modifiquen el comportament de l'eina i permeten diferents configuracions de la cerca. El filtre de cerca defineix allò que s'està buscant en el directori, i els atributs són els atributs que es retornaran en la cerca. Les principals opcions són les següents:

1. *-A* retorna només el nom dels atributs, no el seu valor.
2. *-b basedn* indica a partir de quina base es realitzarà la cerca.
3. *-c* estableix un mode d'execució continu, no s'atura en els errors.
4. *-L* mostra les respostes en format LDIFv1.
5. *-LL* mostra les respostes en format LDIF sense comentaris.
6. *-LLL* mostra les respostes en format LDIF sense comentaris ni versió.
7. *-P versió* indica la versió del protocol (per defecte és la 3).
8. *-d nivell* estableix el nivell de depuració LDAP a "nivell".
9. *-D binddn* indica el DN amb el qual es connectarà al directori per fer la cerca.
10. *-h host* indica el servidor LDAP a consultar.
11. *-H URI* indica l'adreça del servidor LDAP a consultar en format Uniform Resource Identifier, és a dir, "ldap://servidor/".
12. *-n* mostra el que es faria, però no ho fa. Serveix per comprovar que la sintaxi de l'ordre és correcta.
13. *-p port* és el port on està connectat el servidor LDAP.
14. *-Q* fa servir el mode silenciós SASL.

15. `-v` s'executarà de manera detallada, mostrant els diagnòstics per la sortida estàndard.
16. `-w passwd` contrasenya de connexió per a l'autenticació simple.
17. `-W` demana la contrasenya de connexió.
18. `-x` autenticació simple.
19. `-X authzid`: la identitat d'autorització SASL ("dn:<dn>" o "u:<user>").
20. `-y fitxer` permet obtenir la contrasenya a partir d'un fitxer.

A continuació es mostren alguns exemples d'ús de l'ordre.

Exemple de consulta de tot el directori

```
1 root@servidor:~# ldapsearch -LLL -D "cn=admin,dc=empresa,dc=com"
   -H ldap:/// -W -b dc=empresa,dc=com cn
2 Enter LDAP Password:
3 dn: dc=empresa,dc=com
4
5 dn: cn=admin,dc=empresa,dc=com
6 cn: admin
7
8 dn: ou=Personal,dc=empresa,dc=com
9
10 dn: ou=Grups,dc=empresa,dc=com
11
12 dn: cn=comercials,ou=Grups,dc=empresa,dc=com
13 cn: comercials
14
15 dn: uid=mfreixas,ou=Personal,dc=empresa,dc=com
16 cn: Marc Freixas
17
18 dn: cn=administratiu,ou=Grups,dc=empresa,dc=com
19 cn: administratiu
```

Aquesta consulta té aquests paràmetres:

- **-LLL** per obtenir un text de sortida sense comentaris.
- **-D**, que defineix amb quin objecte es fa la connexió al directori.
- **-H**, `ldap://`, especifica que el servidor LDAP a consultar serà localhost.
- **-W** perquè demani la contrasenya.
- **-b**, que defineix la base de cerca.
- **cn** és l'atribut pel qual s'està preguntant.

Es pot observar que la resposta retorna els diferents objectes del directori i, en aquells objectes que contenen l'atribut `cn`, també aquest atribut. Si l'objecte no conté l'atribut `cn` (com és el cas de les unitats organitzatives Personal i Grups) no pot retornar l'atribut i només mostra el nom de l'objecte.

Es poden realitzar diferents tipus de consulta:

1. Igualtat: (givenname=Marc)
2. Presència: (givenname=*)

3. Subcadena: (givenname=J*)

4. Semblança: (givenname~=Joan)

Exemple de consulta d'un objecte concret (consulta d'igualtat)

```
1 root@servidor:~# ldapsearch -LLL -D "cn=admin,dc=empresa,dc=com"
   -H ldap:/// -W -b 'dc=empresa,dc=com' '(uid=mfreixas)'
2 Enter LDAP Password:
3 version: 1
4
5 dn: uid=mfreixas,ou=Personal,dc=empresa,dc=com
6 cn: Marc Freixas
7 displayName: Marc Freixas
8 gecos: Marc Freixas
9 gidNumber: 5001
10 givenName: Marc
11 homeDirectory: /home/comercials/mfreixas
12 loginShell: /bin/bash
13 objectClass: inetOrgPerson
14 objectClass: posixAccount
15 objectClass: shadowAccount
16 objectClass: top
17 userPassword:: bWFyY3Bhc3N3b3Jk
18 sn: Freixas
19 uidNumber: 10001
20 uid: mfreixas
```

Aquesta consulta té els paràmetres:

- **-LL**, per tenir un text de sortida sense comentaris, però que mostrarà la versió del format LDIF.
- **-D**, que defineix amb quin objecte es fa la connexió al directori.
- **-H**, [ldap:///](#), especifica que el servidor LDAP a consultar serà localhost.
- **-W** perquè demani la contrasenya.
- **-b**, que defineix la base de cerca.
- **"uid=mfreixas"** és el filtre de cerca que es fa servir, en aquest cas d'igualtat.

Es pot observar que la resposta retorna tots els atributs dels objectes que compleixen el filtre (en aquest cas només un objecte el compleix), ja que no es demana cap atribut en particular.

Els filtres de cerca ofereixen una gran versatilitat a l'hora de fer les consultes. Per fer-los servir se segueix una norma especificada a l'RFC 1558.

La forma d'expressar consultes complexes en l'LDAP és mitjançant la notació prefixa, en la qual s'avantposa l'operador lògic. Els operadors vàlids són:

- **!** per a NOT
- **|** per a OR
- **&** per a AND

Exemple d'ús d'operadors lògics

```
1 root@servidor:~# ldapsearch -LLL -D "cn=admin,dc=empresa,dc=com"
   -H ldap:/// -W -b 'dc=empresa,dc=com' '(|(ou=P*)(ou=G*))'
2 Enter LDAP Password:
3 dn: ou=Personal,dc=empresa,dc=com
4 objectClass: organizationalUnit
5 objectClass: top
6 ou: Personal
7
8 dn: ou=Grups,dc=empresa,dc=com
9 objectClass: organizationalUnit
10 objectClass: top
11 ou: Grups
```

Al filtre s'observa l'utilització d'un operador lògic, |, així com dues consultes de subcadena, P* i G*.

Es pot combinar l'ús d'operadors lògics i d'atributs per obtenir únicament els resultats que ens interessin.

Exemple de consulta d'un atribut determinat

```
1 root@servidor:~# ldapsearch -LLL -D "cn=admin,dc=empresa,dc=com"
   -H ldap:/// -W -b 'dc=empresa,dc=com' '(&(cn=Marc*)(sn=F*))'
   uid
2 Enter LDAP Password:
3 dn: uid=mfreixas,ou=Personal,dc=empresa,dc=com
4 uid: mfreixas
```

La consulta demana l'atribut uid per als objectes el cn dels quals comença per "Marc" i l'sn dels quals comença per "F".

2.5.2 Utilitat gràfica phpLDAPadmin

Tot i que les utilitats bàsiques per a la línia d'ordres permeten realitzar totes les operacions necessàries en el servidor OpenLDAP, sempre resulta més còmode fer servir una interfície gràfica per a l'administració del directori. Hi ha diverses opcions d'eines d'administració gràfica, però la utilització d'un entorn web és molt recomanable, ja que permet l'accés al directori des de qualsevol ordinador de la xarxa amb un simple navegador.

Una de les opcions més conegudes i provades com a eina d'administració gràfica basada en web per a l'OpenLDAP és l'aplicació phpLDAPadmin, també coneguda com a **PLA**.

PLA és una eina d'administració per al LDAP que té les característiques següents:

1. Permet administrar entrades en un servidor LDAP, és a dir, permet crear, modificar i esborrar les entrades.
2. És compatible amb els estàndards oberts, de manera que pot treballar amb entrades o registres de qualsevol servidor LDAP que compleixi la normativa.

3. Està pensada per ser usada pels administradors, que ja tenen cert coneixement de l'LDAP.
4. És flexible: pot ser configurada per adaptar-se a l'entorn sense necessitat de canviar el codi.

PLA ha estat dissenyada perquè la facin servir els administradors. Per tant, és útil tant per administrar tota la base de dades LDAP com només una part. Tot i que la resta dels usuaris poden fer servir l'aplicació per, per exemple, editar les seves entrades o registres al directori, l'estructura, la terminologia i el procés a seguir no resulten senzills.

Instal·lació de phpLDAPadmin a Debian

La instal·lació de phpLDAPadmin a Debian és molt senzilla. Com que el paquet es troba als repositoris, només caldrà instal·lar-lo mitjançant l'ordre corresponent:

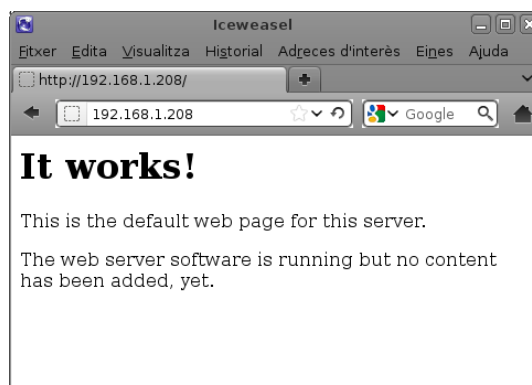
```
1 root@servidor:~# aptitude install phpldapadmin
```

Com que es tracta d'una aplicació per a entorn web, les dependències del paquet fan que també s'instal·len el servidor web Apache i el suport PHP per al servidor. El pas següent és comprovar si tot ha anat correctament i si el servidor Apache està connectat al port esperat, el 80:

```
1 root@servidor:~# nmap -p 80 localhost
2 ...
3 PORT STATE SERVICE
4 80/tcp open  http
5 ...
```

També es pot comprovar que funciona correctament amb una simple consulta des del navegador web, com mostra la figura 2.6.

FIGURA 2.6. Pàgina per defecte del servidor web Apache



Ara cal comprovar que el PHP –que és el llenguatge amb el qual s'ha programat l'aplicació– s'ha instal·lat correctament.

```
1 root@servidor:~# php -v
2 PHP 5.3.3-7+squeezee8 with Suhosin-Patch (cli) (built: Feb 10 2012 14:12:26)
3 ...
```

Evidentment, no ens serveix de gaire tenir l'Apache i el PHP instal·lats si el mòdul d'Apache per fer servir el PHP no està habilitat. Una simple ullada al directori de configuració de mòduls de l'Apache ens permet comprovar-ho:

```
1 root@servidor:~# ls /etc/apache2/mods-enabled/ |grep php
2 php5.conf
3 php5.load
```

En el cas que el mòdul no estigui habilitat, s'ha d'habilitar amb l'ordre d'Apache, en aquest cas *a2enmod*. Quan s'habilita un mòdul al servidor web Apache, cal reiniciar el servidor per tornar a carregar els fitxers de configuració:

```
1 root@servidor:~# a2enmod php5
2 Enabling module php5.
3 Run '/etc/init.d/apache2 restart' to activate new configuration!
4 root@servidor:~# /etc/init.d/apache2 restart
5 Restarting web server: apache2 ... waiting .
```

L'aplicació phpLDAPadmin se situa al directori `/usr/share/phpldapadmin/` i la seva configuració a `/etc/phpldapadmin/`, on el fitxer més important és `/etc/phpldapadmin/apache.conf`, ja que és aquí on hi ha la configuració perquè Apache pugui servir les pàgines de l'aplicació. De fet, aquest fitxer està enllaçat des del directori de configuració d'Apache, que és `/etc/apache2/conf.d/`.

```
1 root@servidor:~# ls -l /etc/apache2/conf.d/php*
2 lrwxrwxrwx 1 root root 29 4 abr 13:16 /etc/apache2/conf.d/phpldapadmin -> /etc/
  phpldapadmin/apache.conf
```

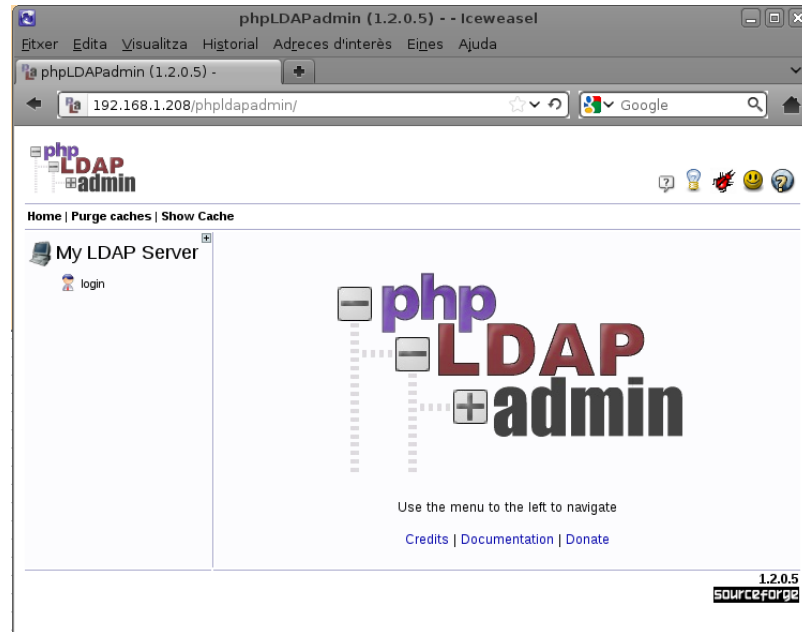
En iniciar el servei Apache, aquest processa tots els fitxers que hi ha al directori `/etc/apache2/conf.d/` i carrega tots els llocs web que serveix.

Al fitxer de configuració de phpLDAPadmin per a Apache `/etc/phpldapadmin/apache.conf` és on es defineix si phpLDAPadmin apareixerà com a àlies (és l'opció per defecte) o si es crearà un VirtualHost.

1. Si es crea un àlies, l'adreça web per accedir a l'aplicació serà <http://servidor.empresa.com/phpldapadmin>, on es pot canviar el nom del servidor per l'adreça IP.
2. Si pel contrari es crea un VirtualHost, l'adreça podria ser de l'estil <http://ldap.empresa.com>.

Configuració de phpLDAPAdmin a Debian

La instal·lació de phpLDAPadmin (PLA) és molt senzilla i la configuració encara ho és més. De fet, el programari ja funciona sense haver de fer cap configuració, simplement amb la instal·lació per defecte. Tan sols s'ha d'accedir a l'adreça per defecte i l'aplicació ja està disponible, com s'aprecia en la figura 2.7.

FIGURA 2.7. phpLDAPadmin en una finestra d'un navegador

Tot i estar disponible des d'un principi, sí que és necessari configurar certs paràmetres perquè funcioni correctament. El fitxer de configuració és `/etc/phpldapadmin/config.php` i cal editar-lo.

Les diferents opcions de configuració estan explicades en el mateix fitxer mitjançant comentaris. La millor opció és cercar dins del fitxer les directives que es volen configurar. Les més interessants són aquestes:

1. `$config->custom->appearance['language'] = 'auto';` Permet configurar l'idioma de l'aplicació. Si està en mode "auto", l'aplicació intentarà determinar l'idioma en funció de l'idioma del sistema. En cas de voler forçar un idioma, els disponibles són *ct, de, en, es, fr, it, nl i ru*.
2. `$servers->setValue('server','base',array('dc=example,dc=com'));`
Aquesta línia defineix el nom de domini que es mostra en la vista d'arbre, a la part esquerra de la pantalla, com es pot observar en la figura 18. Si no està definit, l'aplicació la intenta detectar, però és convenient afegir el domini que ha estat configurat. En l'exemple actual seria: `$servers->setValue('server','base',array('dc=empresa,dc=com'));` Com a orientació, aquesta línia és la 283 del fitxer de configuració.
3. `$servers->setValue('login','bind_id','cn=admin,dc=example,dc=com');`
Aquesta directiva defineix l'usuari mitjançant el qual phpLDAPadmin es connecta al servidor OpenLDAP. Si es defineix, en accedir a l'aplicació, apareix directament, per la qual cosa només cal escriure la contrasenya. Pot ser una bona mesura de seguretat no definir aquesta entrada o definir un usuari fals. La directiva és a la línia 306. Per a l'exemple que s'ha fet servir al llarg de la unitat caldria configurar: `$servers->setValue('login','bind_id','cn=admin,dc=empresa,dc=com');`. Si s'ha definit aquesta entrada, també es pot definir la contrasenya a la directiva següent: `$servers->setValue('login','bind_pass','secret');` on "secret"

s'hauria de canviar pel mot de pas de l'administrador.

4. `$servers->setValue('server','name','Servidor LDAP de l'Empresa');`
En la línia 270 es pot incloure una descripció per identificar el directori al qual es connecta aquesta aplicació.

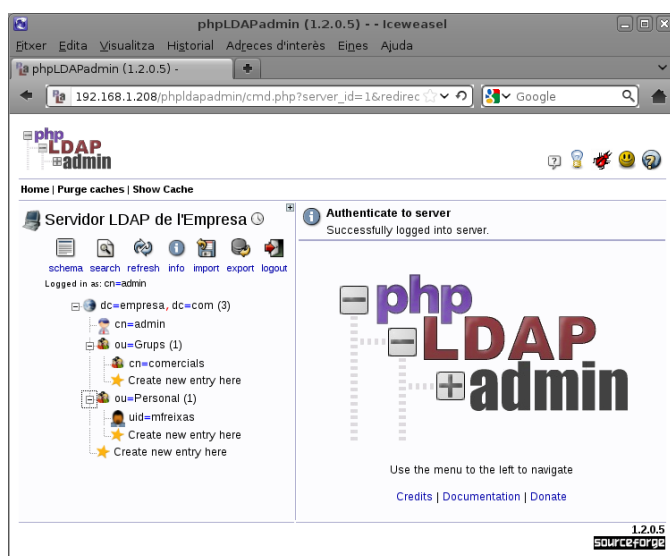
Una vegada ha estat configurada l'aplicació, s'hi pot accedir mitjançant la pantalla de connexió.

FIGURA 2.8. Pantalla de connexió de phpLDAPadmin



En la figura 2.8 es pot observar que al camp *Login DN* ja apareix definit l'usuari mitjançant el qual es farà la connexió al servidor. Tan sols queda escriure la contrasenya per poder-hi accedir. També hi ha una casella que podem activar per connectar-nos de manera anònima al directori, sempre i quant aquest ho permeti (l'OpenLDAP ho permet per defecte), que és útil per fer cerques en el directori.

FIGURA 2.9. Pantalla principal de phpLDAPadmin



Una vegada s'inicia la sessió, la pantalla de l'aplicació queda dividida, com s'aprecia en la figura 2.9. En la part esquerra, a més d'un menú amb algunes

opcions, hi apareix un arbre amb tots els objectes del directori. En la figura també s'aprecia que els objectes que ja hi ha al directori estan disponibles per ser consultats o modificats.

Administració del directori mitjançant phpLDAPAdmin

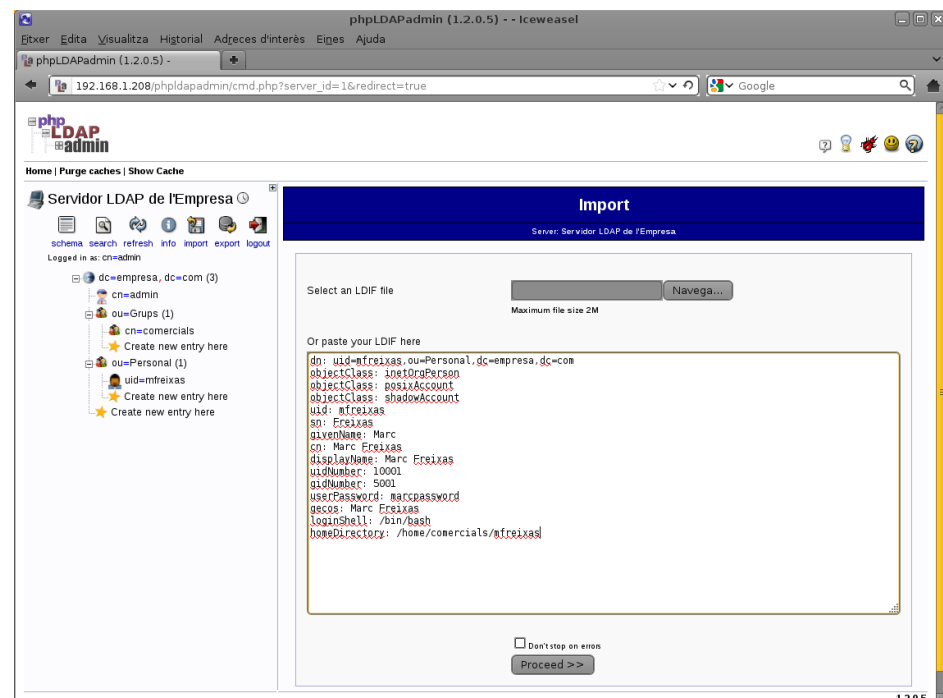
L'ús de **phpLDAPAdmin** facilita molt les tasques de consulta, creació i modificació d'entrades al directori, tot i que es poden seguir fent servir fitxers LDIF per fer aquestes operacions.

Ús de fitxers LDIF

Com que el format de fitxer per defecte per definir entrades i operacions en un servidor LDAP és el LDIF, phpLDAPAdmin permet la importació d'un fitxer o d'un text en aquest format. Per fer-ho, simplement cal fer clic sobre l'opció *import* del menú. Quan se selecciona aquesta opció es pot afegir el contingut LDIF al directori de dues maneres:

1. Seleccionant un fitxer LDIF del disc dur.
2. Escrivint el text amb format LDIF directament en l'espai disponible.

FIGURA 2.10. LDIF a phpLDAPAdmin



En la figura 2.10 es mostra com afegir un fitxer LDIF.

No només es pot importar un fitxer, sinó que també es pot exportar el contingut del directori, d'una part del directori, d'una cerca concreta o una entrada determinada a format LDIF.

FIGURA 2.11. Exportació de dades a phpLDAPadmin

phpLDAPadmin

Home | Purge caches | Show Cache

Export

Export

Server: Servidor LDAP de l'Empresa

Base DN: [browse](#)

Search Scope:

- ☐ Base (base dn only)
- ☐ One (one level beneath base)
- ☒ Sub (entire subtree)

Search Filter:

Show Attributes:

☐ Include system attributes

☐ Save as file

☐ Compress

Export format:

- ☐ CSV
- ☐ DSML
- ☒ LDIF
- ☐ VCARD

Line ends:

- ☐ Macintosh
- ☒ UNIX (Linux, BSD)
- ☐ Windows

[Proceed >>](#)

1.2.0.5
SOURCEFORGE

Quan es fa clic a la icona *export* del menú, apareix la pantalla de la figura 2.11, on es pot definir la cerca a realitzar. Els objectes se cercaran des del punt que es defineix a *Base DN*. Habitualment aquest punt és l'arrel de l'arbre. També es defineix si es buscarà en els fills d'aquest objecte mitjançant la directiva *Search Scope*. Es defineix el filtre de la cerca, que delimita quins objectes es mostren, i els atributs d'aquests objecte que ens interessin. Per acabar, també es pot definir el format d'exportació del fitxer, ja que no està limitat al LDIF.

Modificació d'una entrada del directori

Per modificar una entrada del directori únicament cal fer clic al seu damunt. Quan es fa així, l'aplicació sempre pregunta quina plantilla es vol utilitzar per editar l'entrada. Depenent de la plantilla que se seleccioni es visualitzen, i per tant es poden editar, uns atributs diferents. Per exemple, per a l'entrada "mfreixas", que és un usuari, si es fa servir la plantilla *address book entry* es pot accedir a o modificar la fotografia de l'usuari, però si es fa servir la plantilla *default* no es té accés a aquest atribut (tot i que sí que es podrà accedir a d'altres, com per exemple *homeDirectory*).

A més, una vegada s'està editant una entrada, a la part superior apareixen més opcions. Es permet canviar el nom de l'entrada, copiar o moure l'entrada, canviar la plantilla que s'està fent servir, exportar l'entrada, comparar-la, afegir-li un atribut o esborrar-la.

Creació d'una entrada en el directori

Per crear una entrada només cal fer clic sobre el lloc de l'arbre on es vol crear, on diu *Create new entry here*, o, si l'objecte accepta entrades fill, quan s'està editant l'objecte, fer clic a *Create a child entry*. En tots dos casos, la icona és una estrella groga de cinc puntes. En fer-hi clic apareix un formulari com el de la figura 2.12.

FIGURA 2.12. Creació d'un objecte a phpLDAPadmin

The screenshot shows the phpLDAPadmin interface. On the left is a tree view of the LDAP directory structure. The main area is titled 'Create Object' and shows the 'New User Account (Step 1 of 1)' form. The form includes fields for First name, Last name, Common Name, User ID, Password, UID Number, GID Number, Home directory, and Login shell. The 'Create Object' button is at the bottom.

Home | Purge caches | Show Cache

phpLDAPadmin

Servidor LDAP de l'Empresa

schema search refresh info import export logout

Logged in as: cn=admin

dc=empresa, dc=com (3)

cn=admin

ou=Grups (2)

ou=Personal (1)

uid=mtreixas

Create new entry here

Create new entry here

Create Object

Server: Servidor LDAP de l'Empresa Container: ou=Personal,dc=empresa,dc=com Template: Generic: User Account (posixAccount)

New User Account (Step 1 of 1)

First name alias

Júlia

Last name alias, required

Vila *

Common Name alias, required, rdn

Júlia Vila *

User ID alias, required

jvila *

Password alias, hint

md5 (confirm)

Check password...

UID Number alias, required, hint, ro

10002

GID Number alias, required, hint

Home directory alias, required

/home/users/jvila *

Login shell alias

Create Object

1.2.0.5
sourceforge

Igual que passa quan s'està editant una entrada, en crear-ne una de nova l'aplicació pregunta quina és la plantilla a usar per al procés de creació, i quins seran, per tant, els atributs que contindrà l'entrada. Depenent de quina plantilla i per tant de quina classe d'objecte s'ha triat, uns atributs seran obligatoris, i d'altres, opcionals.

3. Integració del servei de directori

Un servei de directori ofereix accés a un o més directoris. La informació continguda en aquests directoris pot ser consultada amb eines concretes, però també pot ser utilitzada per part d'aplicacions client configurades específicament per a aquest propòsit.

El cas més habitual és fer servir el directori per fer una gestió centralitzada dels usuaris d'una organització i, més concretament, per muntar un sistema d'autenticació centralitzada. El directori contindrà les dades que l'organització hagi decidit, així com aquelles necessàries perquè els sistemes i les aplicacions clients puguin autenticar als usuaris.

3.1 Autenticació centralitzada amb un servei de directori

Poder realitzar consultes en un directori amb una interfície gràfica o amb la línia d'ordres és útil, però el que és realment interessant és que diferents aplicacions puguin fer servir les dades d'un únic directori per tal de no haver de tenir informació duplicada en diferents llocs i possibilitar el manteniment únic i centralitzat d'aquestes dades.

Una de les utilitats més comunes d'un servei de directori és l'autenticació centralitzada dels usuaris d'una xarxa en els diferents serveis (inici de sessió en el sistema, accés a un servei FTP, accés a un gestor de continguts...). La implementació de l'autenticació centralitzada d'usuaris a partir del directori té diversos avantatges:

1. Els noms d'usuaris i contrasenyes són únics en tots els entorns.
2. Se simplifiquen les eines de gestió i administració de comptes d'usuaris (altes i baixes, canvis de contrasenyes...).
3. Permet implementar sistemes d'alta disponibilitat d'una manera més fàcil.
4. Es poden establir polítiques de seguretat coherents i universals.

És evident que en una organització en la qual els usuaris han d'autenticar-se en diferents serveis és convenient mantenir la informació centralitzada i unificada per evitar duplicitats al sistema d'informació de l'empresa i facilitar-ne el manteniment.

Igualment, el directori LDAP pot contenir totes aquelles dades que l'organització cregui necessàries, com ara informació sobre edificis, departaments, ordinadors,

Un sistema d'alta disponibilitat és un sistema dissenyat per assegurar una operació continuada sense errors durant un determinat període de temps.

telèfons, impressores..., a banda de la informació necessària per implementar l'autenticació centralitzada.

3.2 Autenticació a Debian 6 amb l'OpenLDAP

El sistema operatiu Debian 6 fa per defecte l'autenticació bàsica d'usuaris mitjançant uns fitxers determinats: `/etc/passwd`, `/etc/group`, `/etc/shadow` i d'altres. Aquests fitxers són consultats durant el procés d'inici de sessió per autenticar un usuari i obtenir les dades relatives al compte de l'usuari per crear el seu context inicial, com ara el directori de treball. Aquest procés fa servir dues biblioteques: la PAM i l'NSS.

La PAM (Pluggable Authentication Module) és una biblioteca d'autenticació genèrica que qualsevol aplicació pot utilitzar per validar usuaris, utilitzant múltiples esquemes d'autenticació alternatius (fitxers locals, Kerberos, LDAP, dispositius biomètrics...). Aquesta biblioteca és utilitzada pel procés d'inici de sessió (*login*) del sistema operatiu per comprovar si les credencials introduïdes per l'usuari (nom i contrasenya) són correctes.

Mòdul PAM

Als començaments d'Unix, per autenticar un usuari calia que aquest introduís una contrasenya. Llavors el sistema comprovava si la contrasenya coincidia amb la contrasenya xifrada emmagatzemada a `/etc/passwd` (més tard a `/etc/shadow`). La idea era que un usuari era realment aquest usuari només si podia entrar la seva contrasenya secreta correctament.

Amb l'evolució dels sistemes informàtics s'han popularitzat noves formes d'autenticació d'usuaris, entre d'altres: fitxers alternatius i més complicats que l'arxiu `/etc/passwd`, lectors d'empremtes dactilars, reconeixement òptic de cares, certificats USB, claus intel·ligents...

El problema és que cada vegada que es desenvolupa un nou esquema d'autenticació requereix que tots els programes que el volen fer servir (*login*, *ftpd*...) s'hagin de reescriure.

La PAM ofereix una capa intermèdia entre l'usuari i l'aplicació en el moment de l'autenticació. Aquesta capa intermèdia està basada en un sistema modular que li permet oferir diferents funcionalitats.

L'NSS (Name Service Switch) és una interfície genèrica per obtenir els paràmetres d'un compte (com l'identificador d'usuari, l'identificador de grup, el *shell* inicial, el directori de treball...), que és utilitzada pel procés de *login* per crear el procés d'inici de sessió de l'usuari.

Els paràmetres d'un compte d'usuari es poden obtenir tradicionalment de l'arxiu `/etc/passwd`. Aquest fitxer és de text pla i conté una línia per a cada usuari donat d'alta en el sistema. Cada línia està formada per set camps separats per dos punts:

1. El nom d'usuari.
2. *x* si s'usen *shadow passwords* (el més habitual), si no, la contrasenya codificada.

3. L'identificador numèric de l'usuari (UID).
4. L'identificador numèric del grup primari de l'usuari (GID).
5. El nom complet de l'usuari i, opcionalment, informació addicional.
6. El directori d'inici (també anomenat *directori de connexió* o *directori de treball*) de l'usuari.
7. L'interpret d'ordres per defecte de l'usuari.

L'avantatge fonamental d'aquestes dues biblioteques és que es poden reconfigurar dinàmicament mitjançant fitxers, sense necessitat de tornar a compilar les aplicacions que les utilitzen. Per tant, l'únic que es necessita és configurar les dues biblioteques perquè utilitzin el servidor LDAP a més dels fitxers locals (/etc/passwd...) de cada sistema.

3.2.1 Comprovacions preliminars

Per instal·lar el suport LDAP per a l'autenticació d'usuaris caldrà comprovar la correcta instal·lació bàsica del sistema client, és a dir, la configuració IP, el nom de la màquina i la connexió amb el servidor OpenLDAP. Una forma ràpida de fer-ho és, en primer lloc, confirmar que el ping retorna del client al servidor:

```

1 root@client:~# ping servidor.empresa.com
2 PING servidor.empresa.com (192.168.1.208) 56(84) bytes of data:
3 64 bytes from servidor.empresa.com (192.168.1.208): icmp_req=1 ttl=64 time
  =0.638 ms
4 64 bytes from servidor.empresa.com (192.168.1.208): icmp_req=2 ttl=64 time=1.24
  ms
5 64 bytes from servidor.empresa.com (192.168.1.208): icmp_req=3 ttl=64 time
  =0.697 ms
6 ^C
7 — servidor.empresa.com ping statistics —
8 3 packets transmitted, 3 received, 0% packet loss, time 2005ms
9 rtt min/avg/max/mdev = 0.638/0.859/1.244/0.274 ms

```

Una segona comprovació és fer una consulta a la informació del directori des del client al servidor OpenLDAP. Per fer-ho, en primer lloc cal instal·lar en la màquina client les utilitats bàsiques LDAP per poder realitzar la consulta:

```

1 root@client:~# aptitude install ldap-utils

```

Una consulta molt útil sobre el servidor OpenLDAP és la que retorna els identificadors (UID) dels usuaris donats d'alta. A més de comprovar la connexió amb el servidor, també proporciona la llista d'usuaris que haurien de poder iniciar sessió a la màquina client:

```

1 root@client:~# ldapsearch -LLL -x -D 'cn=admin,dc=empresa,dc=com' -W -H ldap://
  servidor/ -b 'dc=empresa,dc=com' objectClass=posixAccount uid
2 Enter LDAP Password:
3 dn: uid=mfreixas,ou=Personal,dc=empresa,dc=com
4 uid: mfreixas

```

Vegeu l'apartat "Instal·lació de l'OpenLDAP" d'aquesta unitat per a més detalls sobre la instal·lació i configuració d'un servidor OpenLDAP.

La consulta anterior dóna com a resultat un únic usuari, que és el que podrà iniciar sessió.

El filtre de cerca que s'ha utilitzat demana els objectes del tipus **posixAccount**, ja que aquest tipus d'objecte defineix una sèrie d'atributs que són necessaris per poder fer servir l'objecte per autenticar usuaris d'un sistema Linux. Els atributs que defineix aquesta classe d'objecte, tant els obligatoris (MUST) com els opcionals (MAY), es mostren en la figura 3.1. La classe d'objecte **posixAccount** proporciona els mateixos atributs que es poden trobar al fitxer `/etc/passwd`.

FIGURA 3.1. Classe d'objecte **posixAccount**

Classe d'objecte posixAccount
MUST cn uid uidNumber gidNumber homeDirectory
MAY userPassword loginShell gecos description

La classe que permet guardar els atributs necessaris per a un usuari d'un sistema operatiu de tipus Unix és **posixAccount**. Per poder fer servir un objecte del directori per fer l'autenticació d'un usuari, aquest ha de ser de la classe **posixAccount**.

Una vegada s'ha comprovat que el sistema client és capaç d'accedir a la informació del directori caldrà configurar el sistema perquè en comptes d'utilitzar els fitxers clàssics per validar els usuaris faci una consulta al servidor LDAP.

3.2.2 Instal·lació del programari client LDAP

El programari necessari per configurar un sistema operatiu Debian GNU/Linux com a client d'un servidor OpenLDAP està inclòs als repositoris de programari oficials. L'ordre d'instal·lació és:

```
1 root@client:~# apt-get install libpam-ldap libnss-ldap nscd
```

Problemes amb el dimoni **nscd**

En algunes ocasions, el dimoni **nscd** pot provocar problemes durant la configuració. Es pot aturar amb l'ordre `/etc/init.d/nscd stop`, configurar el sistema i reiniciar-lo amb `/etc/init.d/nscd start`.

Aquests paquets instal·len el suport LDAP tant per al mòdul PAM com per a l'NSS, i també el dimoni **nscd** (*name service cache daemon*), que fa de cau de noms per al servei NSS per accelerar les consultes.

Després de la instal·lació caldrà contestar algunes preguntes per realitzar la configuració inicial dels paquets.

Configuració inicial de libnss-ldap

Per configurar el paquet libpam-ldap es demanen una sèrie de dades:

1. **URI (Uniform Resource Identifier)** del servidor LDAP. En aquest punt s'ha d'introduir una cadena de la forma `ldap://<hostname_o_ip>:<port>/` com pot ser `"ldap://ldap.empresa.com/"` o `"ldap://192.168.1.208/"`. Opcionalment es pot incloure el port on es connecta el servidor, `ldap://192.168.1.208:389/`, però si no s'especifica, es fa servir el port per defecte.
S'ha de fer servir `ldap://` per una connexió estàndard per TCP, `ldaps:` si la connexió està xifrada per SSL i `ldapi:` si es fa per IPC (fa servir un sòcol Unix). Aquesta última és l'opció per defecte i funciona si el client i el servidor són la mateixa màquina. El més habitual és fer servir l'LDAP per TCP.
2. **DN (nom distintiu)** de la base del DIT. Com que la majoria de llocs fan servir la notació basada en DNS és una bona opció seguir la mateixa norma. En qualsevol cas, la base aquí definida ha de ser la mateixa que es va definir en el servidor LDAP i ha d'estar escrita en la notació correcta: `dc=empresa,dc=com`.
3. **Versió LDAP** que s'utilitza. Excepte en situacions molt excepcionals (per mantenir la compatibilitat amb instal·lacions antigues), es fa servir sempre la versió 3 del protocol.
4. **Compte LDAP per a l'administrador**. Quan es fa la configuració inicial del servidor LDAP es creen dues entrades: la base i l'administrador. Aquest administrador és el que crea l'instal·lador del dimoni slapd per defecte, i és de l'estil `cn=admin,dc=empresa,dc=com`. El DN de l'entrada coincideix amb la base que s'ha definit en el pas anterior de la instal·lació.
5. **Contrasenya LDAP per a l'administrador**. La contrasenya que es va definir en el moment de la instal·lació i configuració del dimoni slapd per a l'administrador.

Una vegada entrades aquestes dades procedim a configurar el paquet libpam-ldap.

Configuració inicial de libpam-ldap

El paquet libpam-ldap realitza dues preguntes durant la configuració bàsica:

1. **Permetre a l'administrador** de l'LDAP actuar com a **administrador local**. Es pot decidir si es vol concedir permisos a l'administrador LDAP per realitzar també canvis en els comptes locals del sistema. A aquesta qüestió es contesta habitualment que **no**.
2. **Requisit d'inici de sessió** per accedir al directori. La configuració per defecte del directori OpenLDAP permet la consulta sense validació (la

validació s'anomena *bind* a l'LDAP) de les dades del directori. **No és necessària cap autenticació**, però si s'ha canviat la configuració per no permetre una consulta anònima, en aquest punt es pot definir un usuari amb el qual poder fer les consultes. L'usuari administrador sempre podrà fer-les.

3.2.3 Configuració de l'autenticació LDAP

Amb el programari instal·lat i configurat, és el moment de configurar el sistema perquè faci servir els nous mòduls d'autenticació.

El primer fitxer que cal modificar és `/etc/ldap/ldap.conf`. Aquest fitxer únicament conté comentaris. Serveix per definir les opcions per defecte per a tots els clients LDAP.

Les directives que cal modificar són:

1. `BASE`, que defineix la base del DIT.
2. `URI`, que defineix l'adreça en la qual es connecta el servidor OpenLDAP.

Exemple de fitxer `ldap.conf`

El fitxer de configuració general LDAP per a Debian serà un fitxer com aquest:

```
1 #
2 # LDAP Defaults
3 #
4
5 # See ldap.conf(5) for details
6 # This file should be world readable but not world writable.
7
8 BASEdc=empresa,dc=com
9 URIldap://192.168.1.208
```

Les diferents opcions de configuració del fitxer es poden consultar en la seva pàgina del manual:

```
1 root@client:/etc# man ldap.conf
```

El fitxer `nsswitch.conf` està situat en el directori `/etc`. Aquest fitxer configura el Name Service Switch. Com que hem instal·lat el suport LDAP per a l'NSS, caldrà configurar l'NSS perquè a més de les bases de dades tradicionals (bàsicament `/etc/passwd`) faci servir l'LDAP per a les seves operacions.

El fitxer per defecte és:

```
1 # Example configuration of GNU Name Service Switch functionality.
2 # If you have the 'glibc-doc-reference' and 'info' packages installed, try:
3 # 'info libc "Name Service Switch"' for information about this file.
4
5 passwd: compat
6 group: compat
7 shadow: compat
8
```

```
9 hosts: files mdns4_minimal [NOTFOUND=return] dns mdns4
10 networks: files
11
12 protocols: db files
13 services: db files
14 ethers: db files
15 rpc: db files
16
17 netgroup: nis
```

Per configurar la nova base de dades caldrà modificar tres línies i informar l'NSS que, a més de les bases de dades tradicionals, també ha de fer servir l'LDAP:

```
1 passwd: compat ldap
2 group: compat ldap
3 shadow: compat ldap
```

Com que s'ha canviat la configuració NSS, és una bona pràctica reiniciar el dimoni `nscd` que fa de memòria cau de noms:

```
1 root@client:/etc# /etc/init.d/nscd restart
2 Restarting Name Service Cache Daemon: nscd.
```

Ara que l'NSS ja sap on obtenir els atributs del usuari, cal configurar l'LDAP perquè realitzi l'autenticació en el servidor LDAP. La llibreria PAM es configura en el directori `/etc/pam.d/`. En aquest directori hi ha diversos fitxers, molts dels quals són específics per a una aplicació determinada que desitja fer servir aquesta llibreria per realitzar l'autenticació dels usuaris.

Els fitxers de configuració de la PAM són:

1. **common-auth**: on es defineix com saber si la contrasenya d'un usuari és correcta.
2. **common-account**: on es defineix com saber si un compte d'usuari és vàlid, comprovant si ha expirat o si s'hi ha d'aplicar alguna restricció de temps.
3. **common-password**: on es defineix com pot un usuari canviar la seva contrasenya.
4. **common-session** i **common-session-noninteractive**: on es defineix la configuració de sessió per a l'usuari amb diverses accions, com pot ser informant si té correu al sistema, creant el seu directori de connexió o mostrant l'últim inici de sessió.

Aquests fitxers ja estan configurats per permetre l'ús de l'LDAP per a l'autenticació dels usuaris i inclouen el mòdul `pam_ldap.so`. Aquest mòdul és el que proporciona autenticació, autorització i canvi de contrasenya fent servir un servidor LDAP.

Caldrà, però, afegir una línia al final del fitxer `/etc/pam.d/common-session`:

```
1 session required pam_mkhomedir.so
```

Aquesta línia té una gran importància i forçarà la creació del directori de treball de l'usuari que inicia sessió en el cas que aquest directori no existeixi.

El directori de treball (en anglès, *home directory*) té molta importància, ja que és el directori de connexió de l'usuari en iniciar una sessió interactiva. Sense aquest directori, l'usuari no pot treballar normalment i alguns sistemes fins i tot li impediran iniciar la sessió.

Com que fem servir una base de dades centralitzada d'usuaris, els usuaris es donen d'alta únicament al servidor LDAP. Cada usuari, per poder iniciar sessió en un ordinador client, hauria de tenir creat un directori de treball. Això implicaria crear aquest directori en totes les màquines client. Evidentment no és la solució desitjable, ja que possiblement aquest usuari no iniciarà sessió en tots els ordinadors de l'organització.

El que fa la solució proposada és una crida al mòdul `pam_mkhomedir.so` de la PAM, que comprovarà si existeix o no el directori definit en l'atribut *homeDirectory* (vegeu la figura 3.1 per recordar els diferents camps obligatoris) i si no existeix el crearà, fent servir el directori `/etc/skel/` com a base.

3.2.4 Verificació de la configuració

La comprovació bàsica del funcionament de la configuració es realitza amb l'ordre *getent*. Aquesta ordre obté entrades d'una base de dades administrativa que s'especifica.

La base de dades pot ser `passwd`, `group`, `hosts`, `services`, `protocols` o `networks`. La que ens interessa és `passwd`, ja que ens proporcionarà les entrades d'aquesta base de dades.

Exemple d'ús de l'ordre *getent*

```
1 root@client:~# getent passwd
2 root:x:0:0:root:/root:/bin/bash
3 daemon:x:1:1:daemon:/usr/sbin:/bin/sh
4 bin:x:2:2:bin:/bin:/bin/sh
5 ...
6 mitabe:x:1000:1000:Miquel Tarazona Belenguer,,,:/home/mitabe:/bin
  /bash
7 sshd:x:109:65534:./var/run/sshd:/usr/sbin/nologin
8 mfreixas:x:10001:5001:Marc Freixas:/home/comercials/mfreixas:/bin
  /bash
9 jvila:*:10002:5002:Júlia Vila:/home/users/jvila:/bin/sh
```

Els dos usuaris ressaltats no són al fitxer `/etc/passwd` sinó que s'obtenen de la consulta al servidor LDAP.

El resultat de l'ordre *getent passwd* és el conjunt d'usuaris que es poden autenticar en el sistema. És la unió del fitxer `/etc/passwd` amb la consulta LDAP realitzada per l'NSS. Les dues últimes entrades són els usuaris creats en el servidor LDAP amb els atributs d'objecte `posixAccount`, que, com s'observa, són totalment compatibles amb els atributs que hi ha en el fitxer `/etc/passwd`.

Es pot obtenir el mateix resultat si es consulta la base de dades dels grups fent

servir el paràmetre *group*.

Exemple de l'ordre `getent group`

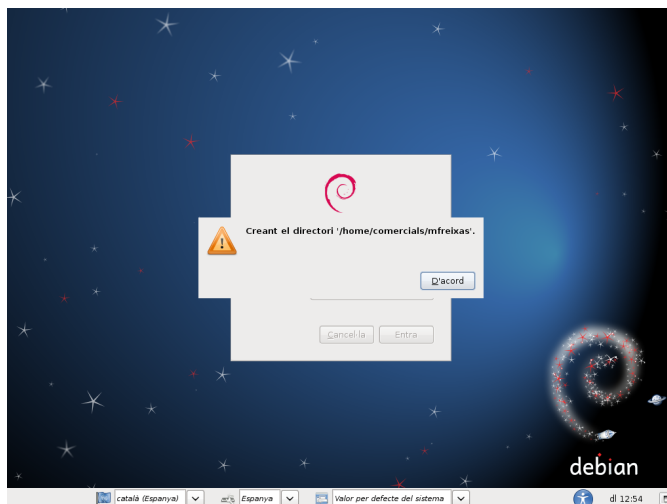
```
1 root@client:~# getent group
2 root:x:0:
3 daemon:x:1:
4 ...
5 mitabe:x:1000:
6 comercials:*:5001:
7 administratius:*:5002:
```

A més dels grups locals apareixen els dos grups creats en el directori.

Per iniciar sessió en el client amb un usuari del directori, simplement s'ha d'introduir el nom d'usuari i la contrasenya. Per a l'usuari és totalment transparent, ja que no sap si la seva informació d'usuari està en el sistema local Linux o en un servidor remot amb LDAP.

Quan un usuari inicia sessió per primera vegada en una màquina configurada en el domini es mostra un missatge advertint de la creació del directori personal, com s'aprecia en la figura 3.2.

FIGURA 3.2. Creació d'un directori personal en un entorn gràfic



3.2.5 Reconfiguració de l'autenticació

Si les coses no han funcionat correctament, sempre es pot tornar a realitzar la configuració de les utilitats PAM i NSS per al LDAP fent servir l'ordre *dpkg-reconfigure*:

```
1 # dpkg-reconfigure libpam-ldap
2 # dpkg-reconfigure libnss-ldap
```

Les preguntes que ens fan són les mateixes que en la instal·lació inicial, i es prenen per defecte els valors que ja s'havien introduït.

3.3 Integració de Samba amb l'OpenLDAP

Samba és un conjunt de programes que implementa en sistemes basats en Unix (GNU/Linux) una sèrie de serveis i protocols que permeten compartir fitxers i impressores entre els equips d'una xarxa local.

La característica principal de Samba és que permet compartir recursos entre màquines Windows i GNU/Linux connectades en xarxa. Possibilita, per tant, compartir recursos en un escenari heterogeni.

Samba permet configurar un servidor GNU/Linux com a controlador primari de domini. D'aquesta manera s'autenticaran i compartiran recursos de màquines tant Windows com GNU/Linux. Si teniu un servidor Samba configurat a la xarxa, per facilitar la gestió del domini podeu centralitzar la gestió dels usuaris i les màquines, emmagatzemant aquesta informació al directori LDAP en comptes de fer servir una base de dades local.

Per realitzar aquesta configuració s'ha de configurar Samba com a client LDAP. Per aconseguir-ho el primer que caldrà fer és afegir l'esquema `samba.schema` al directori LDAP per poder tenir accés a tots els objectes i atributs necessaris. Una vegada el directori posseeix les característiques necessàries, es pot configurar Samba com a client LDAP.

3.3.1 Preparació del servidor

Com que l'autenticació de Samba amb l'LDAP està molt relacionada amb l'autenticació del mateix sistema servidor, cal configurar, en primer lloc, el servidor com a client LDAP, en aquest cas d'ell mateix.

En primer lloc cal instal·lar el programari necessari:

```
1 root@servidor:~# aptitude install libnss-ldap libpam-ldap ldap-utils
```

Quan el programa de configuració pregunta quin serà el servidor LDAP per fer l'autenticació, es pot contestar tant amb l'adreça IP del servidor, com amb "localhost" o "127.0.0.1".

Cal configurar el fitxer `/etc/nsswitch.conf` per habilitar l'autenticació LDAP al sistema.

```
1 root@servidor:~# cat /etc/nsswitch.conf
2 # /etc/nsswitch.conf
3 #
4 # Example configuration of GNU Name Service Switch functionality.
5 # If you have the 'glibc-doc-reference' and 'info' packages installed, try:
6 # 'info libc "Name Service Switch"' for information about this file.
7
8 passwd: compat ldap
9 group: compat ldap
```

Vegeu l'apartat
"Configuració de
l'autenticació LDAP"
d'aquesta unitat per a una
explicació més detallada.

```

10 shadow: compat ldap
11
12 hosts: files dns
13 networks: files
14
15 protocols: db files
16 services: db files
17 ethers: db files
18 rpc: db files
19
20 netgroup: ldap

```

En el fitxer `/etc/pam.d/common-password` s'ha de modificar una línia canviant:

```

1 password [success=1 user_unknown=ignore default=die] pam_ldap.so use_authtok
   try_first_pass

```

per:

```

1 password[success=1 user_unknown=ignore default=die]pam_ldap.so try_first_pass

```

En el fitxer `/etc/pam.d/common-session` s'ha d'afegir al final la línia per habilitar la creació automàtica de directoris:

```

1 session optional pam_mkhomedir.so skel=/etc/skel umask=077

```

En aquest cas, es defineixen, a més de l'ús de la llibreria, dues opcions: el directori a partir del qual es crearà el directori de treball i la màscara de creació del directori.

3.3.2 Afegir `samba.schema` al directori LDAP

La configuració de l'OpenLDAP amb el mètode `cn=config` permet realitzar canvis dinàmics en la configuració del directori. Així es pot afegir el suport Samba sense necessitat de reiniciar el servei; simplement s'ha d'afegir un fitxer LDIF amb la definició de les estructures desitjades. Aquest fitxer LDIF no està disponible als repositoris de Debian i per crear-lo hi ha dues opcions:

1. Navegar per la xarxa i trobar-lo en algun lloc on ja estigui creat i a la nostra disposició.
2. Crear-lo a partir dels antics fitxers de definicions d'esquema `samba.schema`.

Per crear el fitxer `samba.ldif`, en primer lloc cal obtenir el fitxer `samba.schema` per agafar-lo com a base. Aquest fitxer es troba al paquet `samba-doc`:

```

1 root@servidor:~# apt-get install samba-doc

```

Se situa el fitxer en el directori corresponent:

El fitxer LDIF necessari per al suport de Samba està disponible a la secció "Annexos" del material web del mòdul.

```
1 root@servidor:~# cp /usr/share/doc/samba-doc/examples/LDAP/samba.schema.gz /etc/ldap/schema/
```

Es descomprimeix el fitxer `samba.schema.gz`:

```
1 root@servidor:~# gzip -d /etc/ldap/schema/samba.schema.gz
```

Cal crear un arxiu de configuració temporal on es defineixen els diferents fitxers amb format `schema` que es faran servir per generar un directori de configuració des d'on es podran obtenir els fitxers LDIF necessaris. Aquest fitxer pot tenir qualsevol nom, però és molt important l'ordre de les línies *include*, ja que en l'LDAP hi ha herència de classes i d'atributs i les definicions s'han de fer en l'ordre correcte. En cas contrari es pot produir algun error.

```
1 root@servidor:~# cat schema_convert.conf
2 include /etc/ldap/schema/core.schema
3 include /etc/ldap/schema/collective.schema
4 include /etc/ldap/schema/corba.schema
5 include /etc/ldap/schema/cosine.schema
6 include /etc/ldap/schema/duaconf.schema
7 include /etc/ldap/schema/dyngroup.schema
8 include /etc/ldap/schema/inetorgperson.schema
9 include /etc/ldap/schema/java.schema
10 include /etc/ldap/schema/misc.schema
11 include /etc/ldap/schema/nis.schema
12 include /etc/ldap/schema/openldap.schema
13 include /etc/ldap/schema/ppolicy.schema
14 include /etc/ldap/schema/samba.schema
```

Amb el fitxer de configuració personalitzat es fa servir l'ordre *slapcat* per generar un directori temporal de configuració de l'estil `cn=config`, des del qual es poden obtenir els fitxers LDIF necessaris.

```
1 root@servidor:~# mkdir -p ./tmp/ldif_output
2 root@servidor:~# slapcat -f schema_convert.conf -F ./tmp/ldif_output -n0 -s "cn={12}samba,cn=schema,cn=config" > ./tmp/cn=samba.ldif
```

Ara en el directori temporal hi ha un nou fitxer que cal editar per eliminar l'índex de l'entrada i informació extra que no és necessària:

```
1 root@servidor:~# vim ./tmp/cn=samba.ldif
```

En aquest fitxer cal modificar dues línies, la 1 i la 3:

```
1 dn: cn={12}samba,cn=schema,cn=config
2 ...
3 cn: {12}samba
```

Que passen a ser:

```
1 dn: cn=samba,cn=schema,cn=config
2 ...
3 cn: samba
```

El que s'ha fet és eliminar els índexs del fitxer temporal, quan es tornen a afegir al directori tornaran a tenir índex, però possiblement serà diferent (i correcte, que és l'important).

Per acabar, s'eliminen les línies de la 186 al final, que es mostren a continuació:

```
1 structuralObjectClass: olcSchemaConfig
2 entryUUID: 01a09774-3073-1031-9c6c-798a169c02ec
3 creatorsName: cn=config
4 createTimestamp: 20120512114024Z
5 entryCSN: 20120512114024.039233Z#000000#000#000000
6 modifiersName: cn=config
7 modifyTimestamp: 20120512114024Z
```

Cal afegir el fitxer `cn=samba.ldif` al directori una vegada editat.

```
1 root@servidor:~# ldapadd -Y EXTERNAL -H ldapi:/// -f ./tmp/cn=samba.ldif
```

Amb el nou esquema afegit al directori, és convenient configurar l'`slapd` perquè l'indexi a partir dels nous atributs definits en l'esquema de Samba. D'aquesta manera es millorarà el rendiment de bases de dades grans. Aquesta configuració es fa mitjançant un fitxer LDIF en el qual es defineixen els nous índexs:

```
1 root@servidor:~# cat samba_indexes.ldif
2 dn: olcDatabase={1}hdb,cn=config
3 changetype: modify
4 add: olcDbIndex
5 olcDbIndex: uidNumber eq
6 olcDbIndex: gidNumber eq
7 olcDbIndex: loginShell eq
8 olcDbIndex: uid eq,pres,sub
9 olcDbIndex: memberUid eq,pres,sub
10 olcDbIndex: uniqueMember eq,pres
11 olcDbIndex: sambaSID eq
12 olcDbIndex: sambaPrimaryGroupSID eq
13 olcDbIndex: sambaGroupType eq
14 olcDbIndex: sambaSIDList eq
15 olcDbIndex: sambaDomainName eq
16 olcDbIndex: default sub
```

Aquest fitxer configura la base de dades amb els nous índexs. Simplement cal afegir-lo a la configuració de l'`slapd`:

```
1 ldapmodify -Y EXTERNAL -H ldapi:/// -f samba_indexes.ldif
```

No cal reiniciar el dimoni `slapd`; la configuració ja està aplicada i en funcionament.

A partir d'aquest moment, l'OpenLDAP està preparat per tenir un client Samba.

3.3.3 Configurar Samba com a client LDAP

Per configurar Samba com client LDAP hi ha un paquet anomenat *smbldap-tools*, que disposa dels fitxers necessaris per facilitar la configuració que s'ha d'instal·lar.

```
1 root@servidor:~# aptitude install samba smbldap-tools
```

Aquest paquet conté un fitxer amb un exemple de configuració per a Samba: `"/usr/share/doc/smbldap-tools/examples/smb.conf"`. Aquest fitxer es pot fer servir

com a punt de partida de la configuració de Samba si no està correctament configurat. Si ja ho està, seria convenient agafar-lo com a referència per modificar la configuració de Samba.

El fitxer `/etc/samba/smb.conf` permet definir un directori LDAP com a processador dorsal (*backend*) per emmagatzemar els usuaris Samba. Serà en aquest moment quan Samba es convertirà en client LDAP.

Les línies que cal configurar (prenent com a referència el fitxer d'exemple) per definir aquest processador dorsal són les següents:

```
1 ldap admin dn = cn=admin,dc=empresa,dc=com
2 ldap group suffix = ou=Grups
3 ldap idmap suffix = ou=Idmap
4 ldap machine suffix = ou=Computers
5 ldap suffix = dc=empresa,dc=com
6 ldap ssl = no
7 ldap user suffix = ou=Personal
```

Consulteu la secció "Annexos" del material web del mòdul per obtenir el llistat complet del fitxer de configuració `/etc/samba/smb.conf`.

Una vegada configurat correctament Samba, cal reiniciar el servei per carregar el fitxer de configuració i afegir la contrasenya Samba a l'administrador del directori:

```
1 root@servidor:~# service samba restart
2 Stopping Samba daemons: nmbd smbd.
3 Starting Samba daemons: nmbd smbd.
4 root@servidor:~# smbpasswd -W
5 Setting stored password for "cn=admin,dc=empresa,dc=com" in secrets.tdb
6 New SMB password:
7 Retype new SMB password:
```

Per a la correcta integració entre Samba i l'LDAP, les utilitats *smbldap-tools* han d'estar configurades correctament. Aquesta configuració es realitza mitjançant un script que proporcionen els mantenidors del paquet.

L'script llegeix la configuració de Samba i adapta la configuració de les utilitats.

```
1 root@servidor:~# gzip -d /usr/share/doc/smbldap-tools/configure.pl.gz
2 root@servidor:~# perl /usr/share/doc/smbldap-tools/configure.pl
```

La configuració ofereix una sèrie d'opcions en la majoria de les quals cal deixar l'opció per defecte, ja que aquestes opcions s'obtenen a partir de les configuracions actuals del sistema. En qualsevol cas, cada opció està perfectament explicada.

Podem veure un exemple de configuració de les utilitats *smbldap-tools* en el codi que apareix a continuació:

```
1 root@servidor:~# perl /usr/share/doc/smbldap-tools/configure.pl
2 $# is no longer supported at /usr/share/doc/smbldap-tools/configure.pl line
3 314.
4 =====
5 smbldap-tools script configuration
6 =====
7 Before starting, check
8 . if your samba controller is up and running.
9 . if the domain SID is defined (you can get it with the 'net getlocalsid')
10 . you can leave the configuration using the Ctrl-c key combination
11 . empty value can be set with the "." character
12 =====
```

```

13 Looking for configuration files...
14
15 Samba Configuration File Path [/etc/samba/smb.conf] >
16
17 The default directory in which the smbldap configuration files are stored is
  shown.
18 If you need to change this, enter the full directory path, then press enter to
  continue.
19 Smbldap-tools Configuration Directory Path [/etc/smbldap-tools/] >
20 =====
21 Let's start configuring the smbldap-tools scripts ...
22
23 . workgroup name: name of the domain Samba act as a PDC
24 workgroup name [EMPRESA] >
25 . netbios name: netbios name of the samba controller
26 netbios name [SERVIDOR] >
27 . logon drive: local path to which the home directory will be connected (for NT
  Workstations). Ex: 'H:'
28 logon drive [H:] >
29 . logon home: home directory location (for Win95/98 or NT Workstation).
  (use %U as username) Ex: '\\SERVIDOR\%U'
30 logon home (press the "." character if you don't want homeDirectory) [\\
  SERVIDOR\%U] > .
31 . logon path: directory where roaming profiles are stored. Ex: '\\SERVIDOR\
  profiles\%U'
32 logon path (press the "." character if you don't want roaming profile) [\\
  SERVIDOR\profiles\%U] > .
33 . home directory prefix (use %U as username) [/home/%U] >
34 . default users' homeDirectory mode [700] >
35 . default user netlogon script (use %U as username) [logon.bat] >
36 default password validation time (time in days) [45] >
37 . ldap suffix [dc=empresa,dc=com] >
38 . ldap group suffix [ou=Grups] >
39 . ldap user suffix [ou=Personal] >
40 . ldap machine suffix [ou=Computers] >
41 . Idmap suffix [ou=Idmap] >
42 . sambaUnixIdPooldn: object where you want to store the next uidNumber
  and gidNumber available for new users and groups
43 sambaUnixIdPooldn object (relative to ${suffix}) [sambaDomainName=EMPRESA] >
44 . ldap master server: IP address or DNS name of the master (writable) ldap
  server
45 ldap master server [192.168.1.208] >
46 . ldap master port [389] >
47 . ldap master bind dn [cn=admin,dc=empresa,dc=com] >
48 . ldap master bind password [] >
49 . ldap slave server: IP address or DNS name of the slave ldap server: can also
  be the master one
50 ldap slave server [192.168.1.208] >
51 . ldap slave port [389] >
52 . ldap slave bind dn [cn=admin,dc=empresa,dc=com] >
53 . ldap slave bind password [] >
54 . ldap tls support (1/0) [0] >
55 . SID for domain EMPRESA: SID of the domain (can be obtained with 'net
  getlocalsid SERVIDOR')
56 SID for domain EMPRESA [S-1-5-21-972867998-1773855070-1288752139] >
57 . unix password encryption: encryption used for unix passwords
  unix password encryption (CRYPT, MD5, SMD5, SSHA, SHA) [SSHA] > MD5
58 . default user gidNumber [513] >
59 . default computer gidNumber [515] >
60 . default login shell [/bin/bash] >
61 . default skeleton directory [/etc/skel] >
62 . default domain name to append to mail address [] >
63 =====
64 Use of uninitialized value $# in concatenation (.) or string at /usr/share/doc/
  smbldap-tools/configure.pl line 314, <STDIN> line 33.
65 backup old configuration files:
66 /etc/smbldap-tools/smbldap.conf->/etc/smbldap-tools/smbldap.conf.old
67 /etc/smbldap-tools/smbldap_bind.conf->/etc/smbldap-tools/smbldap_bind.conf.old
68 writing new configuration file:
69 /etc/smbldap-tools/smbldap.conf done.

```

```
73 /etc/smbldap-tools/smbldap_bind.conf done.
```

Es defineixen tots els paràmetres. Com que Samba ja estava correctament configurat, en la majoria dels casos caldrà escollir les opcions per defecte. L'únic que s'ha fet en l'exemple és desactivar els perfils mòbils.

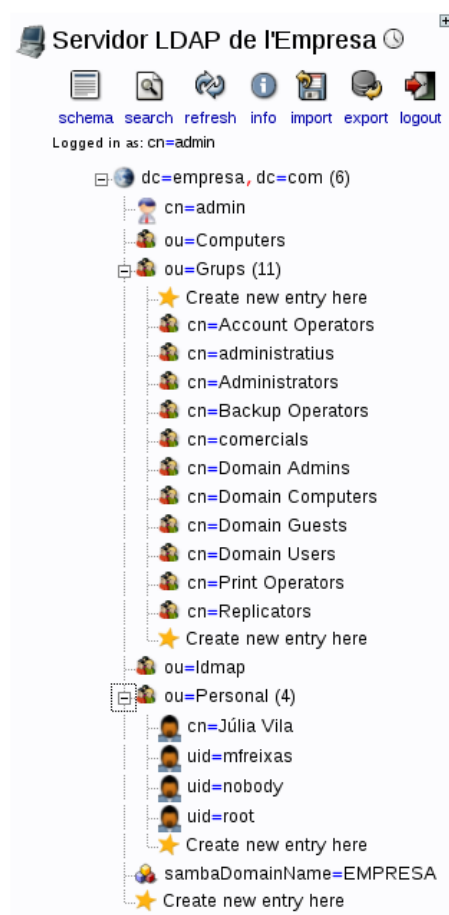
Amb les utilitats correctament configurades, el pas següent és crear en el directori l'estructura necessària d'entrades (unitats organitzatives i grups) per al correcte funcionament de Samba. Aquesta creació es fa amb l'ordre *smbldap-populate*:

```
1 root@servidor:~# smbldap-populate
```

Al final de l'execució d'aquesta ordre se'ns demana la contrasenya de l'usuari *root* per al sistema Unix i per a Samba.

En la figura 3.3 s'observen els objectes que s'han creat en el directori per permetre que Samba realitzi les autenticacions amb la base de dades LDAP.

FIGURA 3.3. DIT resultant de realitzar smbldap-populate



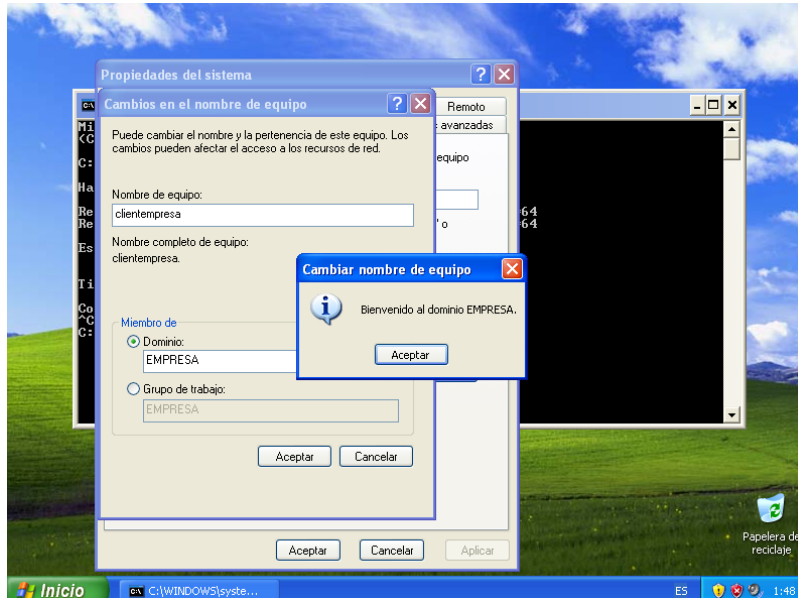
Abans d'afegir les màquines Windows al domini, cal crear els directoris que s'han definit en el fitxer de configuració de Samba.

```
1 root@servidor:/etc/samba# mkdir /srv/samba
2 root@servidor:/etc/samba# mkdir /srv/samba/printers
3 root@servidor:/etc/samba# mkdir /srv/samba/spool
4 root@servidor:/etc/samba# mkdir /srv/samba/profiles
```

```
5 root@servidor:/etc/samba# mkdir /srv/samba/netlogon
```

A partir d'aquest moment es pot afegir una màquina Windows al domini, com mostra la figura 3.4.

FIGURA 3.4. Afegir una màquina Windows a un domini



3.3.4 Comprovació de la configuració

El principal motiu de fer servir Samba és permetre la integració entre totes les màquines de la xarxa que estem administrant, ja que aquest programari facilita la integració de sistemes GNU/Linux, Windows i OS X. És habitual combinar un servidor Samba i una base de dades LDAP per permetre la gestió centralitzada d'usuaris i equips, simulant el comportament d'un domini creat amb Active Directory de Microsoft.

Es pot afegir una màquina Windows al domini que s'ha configurat de manera totalment transparent, de la mateixa manera que s'afegiria a un domini d'Active Directory. A més, tots els usuaris de l'LDAP podran iniciar sessió a la màquina Windows.

3.4 Autenticació a Pure-FTPd amb l'OpenLDAP

Pure-FTPd és un servidor FTP lliure amb llicència BSD que ha estat dissenyat pensant en la seguretat. Ha estat portat a un gran nombre de sistemes operatius com Linux, OpenBSD, NetBSD, FreeBSD, DragonFly BSD, Solaris, Tru64, Darwin, IRIX, HP-UX i fins i tot Android.

Els repositoris de Debian contenen un paquet compilat i preparat per fer servir en

el sistema.

El dimoni Pure-FTPd permet l'autenticació dels usuaris mitjançant LDAP. Per fer aquesta autenticació només caldrà instal·lar el paquet pure-ftpd-ldap:

```
1 root@servidor:/etc/moodle# aptitude install pure-ftpd-ldap
```

Aquest paquet és el dimoni Pure-FTPd compilat amb suport LDAP. Els fitxers de configuració del dimoni estan a /etc/ftpd/.

No es tractarà en aquest apartat la configuració del comportament del dimoni Pure-FTPd més enllà de l'autenticació per al LDAP.

El fitxer on es configura l'autenticació LDAP és /etc/pure-ftpd/db/ldap.conf.

A continuació podem veure un exemple de configuració LDAP per al dimoni pure-ftpd:

```
1 root@servidor:/etc/pure-ftpd/db# cat ldap.conf
2 #####
3 # #
4 # Sample Pure-FTPd LDAP configuration file. #
5 # See README.LDAP for explanations. #
6 # #
7 #####
8
9 # Optional : name of the LDAP server. Default : localhost
10 LDAPServer servidor.empresa.com
11
12 # Optional : server port. Default : 389
13 LDAPPort 389
14
15 # Mandatory : the base DN to search accounts from. No default.
16 LDAPBaseDN dc=empresa,dc=com
17
18 # Optional : who we should bind the server as.
19 # Default : binds anonymously or binds as FTP users
20
21 LDAPBindDN cn=admin,dc=empresa,dc=com
22
23
24 # Password if we don't bind anonymously
25 # This configuration file should be only readable by root
26 LDAPBindPW contrasenya
27
28 # Optional : default UID, when there's no entry in a user object
29 # LDAPDefaultUID 500
30
31 # Optional : default GID, when there's no entry in a user object
32 # LDAPDefaultGID 100
33
34 # Filter to use to find the object that contains user info
35 # \L is replaced by the login the user is trying to log in as
36 # The default filter is (&(objectClass=posixAccount)(uid=\L))
37 # LDAPFilter (&(objectClass=posixAccount)(uid=\L))
38
39 # Attribute to get the home directory
40 # Default is homeDirectory (the standard attribute from posixAccount)
41 # LDAPHomeDir homeDirectory
42
43 # LDAP protocol version to use
44 # Version 3 (default) is mandatory with recent releases of OpenLDAP.
45 # LDAPVersion 3
46
```

```
47 # Optional: use TLS to connect to the LDAP server
48 # LDAPUseTLS True
49
50 # Can be PASSWORD or BIND.
51 # PASSWORD retrieves objects and checks against the userPassword attribute
52 # BIND tries to bind
53 LDAPAuthMethod PASSWORD
```

La majoria d'opcions permeten definir quin atribut de l'objecte servirà per definir les opcions específiques de la connexió. L'explicació de les opcions és la següent:

1. **LDAPServer servidor.empresa.com** és la URI del servidor o l'adreça IP.
2. **LDAPPort 389** és el port on se suposa que està connectat el servidor LDAP.
3. **LDAPBaseDN dc=empresa,dc=com** és la base a partir de la qual se cercaran els objectes. Es pot definir un subarbre, si es vol. Per exemple, es pot definir una unitat organitzativa especial per als usuaris que tindran accés al servei FTP.
4. **LDAPBindDN cn=admin,dc=empresa,dc=com** és l'usuari amb el qual el servidor fa la consulta al directori. També es pot definir una consulta anònima si el directori ho permet.
5. **LDAPBindPW contrasenya** és la contrasenya de l'usuari amb el qual es fa la consulta.
6. **LDAPDefaultUID 500**. En el cas que l'usuari que s'ha validat no tingui definit l'atribut uidNumber es faria servir aquest número.
7. **LDAPDefaultGID 100**. En el cas que l'usuari que s'ha validat no tingui definit l'atribut gidNumber es faria servir aquest número.
8. **LDAPFilter (&(objectClass=posixAccount)(uid=\L))**. Quan es fa una consulta al directori sempre es fa servir un filtre. Aquesta opció permet definir quin serà.
9. **LDAPHomeDir homeDirectory** és l'atribut on està definit el directori de connexió de l'usuari que s'ha autenticat.
10. **LDAPVersion 3** és la versió del protocol. Habitualment és la 3.
11. **LDAPUseTLS True** defineix si es fa servir autenticació segura.
12. **LDAPAuthMethod PASSWORD** permet canviar el mètode d'autenticació, que pot ser comprovant la contrasenya o fent una connexió al directori (BIND).

3.4.1 Comprovació de la connexió FTP

Una simple connexió al servidor FTP permet comprovar el correcte funcionament de l'autenticació mitjançant LDAP.

Exemple de connexió FTP

```
1  usuari@hp:~$ ftp 192.168.1.208
2  Connected to 192.168.1.208.
3  220----- Welcome to Pure-FTPd [privsep] [TLS] -----
4  220-You are user number 1 of 50 allowed.
5  220-Local time is now 13:53. Server port: 21.
6  220-This is a private system - No anonymous login
7  220-IPv6 connections are also welcome on this server.
8  220 You will be disconnected after 15 minutes of inactivity.
9  Name (192.168.1.208:usuari): mfreixas
10 331 User mfreixas OK. Password required
11 Password:
12 230-User mfreixas has group access to: comercials
13 230 OK. Current directory is /home/comercials/mfreixas
14 Remote system type is UNIX.
15 Using binary mode to transfer files.
16 ftp>
```

A l'exemple s'observa la connexió al servidor FTP amb un usuari anomenat mfreixas que està donat d'alta al directori.

3.5 Autenticació a Joomla amb l'OpenLDAP

Més enllà de l'autenticació d'usuaris que inicien sessió en una màquina determinada i fan servir un sistema operatiu, actualment existeixen multitud d'aplicacions web que requereixen l'autenticació de l'usuari. En són exemples Moodle, Joomla, MediaWiki i Zimbra.

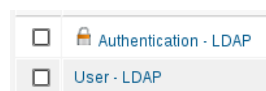
Unes d'aquestes aplicacions són els anomenats *sistemes de gestió de continguts* o CMS (Content Management System), d'entre els quals un dels més utilitzats és Joomla.

3.5.1 Configuració de Joomla

Joomla fa servir per defecte MySQL per gestionar tot el contingut, incloent la base de dades dels usuaris i les seves contrasenyes. Aquest mètode d'autenticació es pot canviar triant l'autenticació LDAP com a opció preferent. De fet, el disseny de Joomla permet fer servir més d'un mètode.

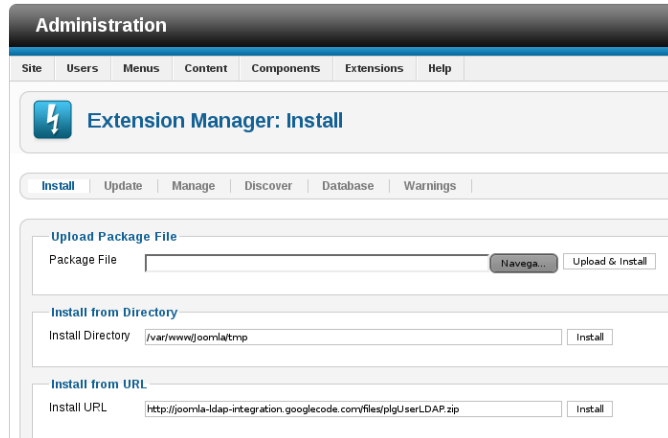
Per fer la configuració cal entrar en la configuració de Joomla, en l'apartat de gestió d'extensions, on s'han d'habilitar els dos connectors (*plug-in*) que es mostren a la figura 3.5.

FIGURA 3.5. Connectors LDAP de Joomla



Authentication - LDAP està inclòs per defecte en la instal·lació de Joomla, però si no es disposa de *User - LDAP* es pot instal·lar fàcilment com es mostra en la figura 3.6. La instal·lació es fa amb la utilitat *extension manager*, introduint la URL directament.

FIGURA 3.6. Instal·lació de User - LDAP a Joomla



Una vegada habilitats, *User - LDAP* no requereix cap configuració, simplement l'habilitació. La configuració d'*Authentication - LDAP* sí que conté els paràmetres necessaris per activar l'autenticació.

Exemple de configuració del connector Authentication - LDAP

L'exemple connecta Joomla al servei LDAP que s'ha fet servir al llarg de la unitat.

En la figura 3.7 es poden observar els paràmetres més importants de connexió de Joomla per al LDAP. Són:

1. **Host**: l'adreça del servidor LDAP.
2. **Port**: el port de connexió per defecte.
3. **LDAP V3**: informa de la versió del protocol.
4. **Negotiate TLS**: especifica si es farà servir una connexió segura o no.
5. **Authorisation Method**: permet definir si es farà una consulta anònima al directori o si, per contra, caldrà connectar-se amb un usuari per fer la consulta.
6. **Connect Username** i **Connect Password**: l'objecte del directori i la contrasenya que es faran servir per realitzar la consulta.
7. **Map**: permet definir els atributs LDAP que es faran coincidir amb els atributs Joomla.

FIGURA 3.7. Configuració d'Authentication - LDAP

The screenshot shows the Joomla! Administration interface for the 'Authentication - LDAP' plugin. The page is titled 'Administration' and 'Plug-in Manager: Authentication - LDAP'. It features a 'Details' section on the left and a 'Basic Options' section on the right. The 'Details' section includes fields for Status (Enabled), Access (Public), Ordering (0. Authentication - LDAP), Plug-in Type (authentication), Plug-in File (ldap), ID (402), and a Description. The 'Basic Options' section includes fields for Host (192.168.1.208), Port (389), LDAP V3 (Yes), Negotiate TLS (No), Follow Referrals (No), Authorisation Method (Bind and Search), Base DN (dc=empresa,dc=com), Search String (uid=[search]), User's DN, Connect Username (cn=admin,dc=empresa,dc=com), Connect Password (masked), Map: Full Name (fullName), Map: email (mail), and Map: User ID (uid). The Joomla! version 2.5.4 is displayed at the bottom.

Amb una correcta configuració es pot accedir a Joomla amb l'usuari donat d'alta al LDAP, tal com s'aprecia en la figura 3.8 i figura 3.9.

FIGURA 3.8. Formulari d'inici de sessió de Joomla

The screenshot shows the Joomla! Login Form. It includes a 'User Name' field with the value 'mfreixas', a 'Password' field with masked characters, and a 'Remember Me' checkbox. There is a 'Log in' button and links for 'Forgot your password?', 'Forgot your username?', and 'Create an account'. The Joomla! logo and 'SUPPORT Joomla!™' text are visible at the bottom.

FIGURA 3.9. Sessió iniciada a Joomla

The screenshot shows the Joomla! user interface after a successful login. It displays the 'Login Form' header, a greeting 'Hi mfreixas,', and a 'Log out' button. The Joomla! logo and 'SUPPORT Joomla!™' text are visible at the bottom, along with the text 'Contribute!'.

Joomla permet anar una mica més enllà en la integració amb l'LDAP, ja que té

definit el seu propi esquema. Cal afegir aquest esquema al directori si volem permetre la total integració de Joomla i l'LDAP i no només l'acreditació dels usuaris.

Per convertir el fitxer esquema al format LDIF cal crear un fitxer de configuració temporal anomenat *schema_convert.conf* amb el contingut següent:

```
1 root@servidor:~# cat schema_convert.conf
2 include /etc/ldap/schema/core.schema
3 include /etc/ldap/schema/cosine.schema
4 include /etc/ldap/schema/inetorgperson.schema
5 include /etc/ldap/schema/nis.schema
6 include /etc/ldap/schema/joomla.schema
```

A partir d'aquest fitxer de configuració es pot generar, amb l'ordre *slapcat*, el fitxer LDIF amb la informació de l'esquema.

```
1 root@servidor:~# slapcat -f schema_convert.conf -F ./tmp/ldif_output/ -n0 -s "
   cn={4}joomla,cn=schema,cn=config" > ./tmp/cn=joomla.ldif
```

Abans d'afegir-lo al directori, cal modificar-lo per eliminar els índexs, ja que el dimoni *slapd* serà l'encarregat d'afegir aquests índexs. Les primeres línies queden així:

```
1 dn: cn=joomla,cn=schema,cn=config
2 ...
3 cn: joomla
```

També cal esborrar les últimes línies, ja que la informació que contenen no és necessària i produiria un error si intentéssim afegir el fitxer al directori. Cal eliminar les línies finals:

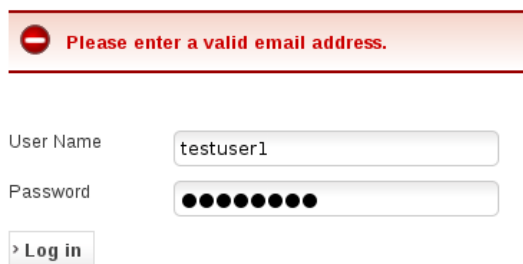
```
1 structuralObjectClass: olcSchemaConfig
2 ...
3 modifyTimestamp: 20120513173054Z
```

L'últim pas és afegir el fitxer al directori. Es fa amb l'ordre *ldapadd*:

```
1 root@servidor:~# ldapadd -Y EXTERNAL -H ldapi:/// -f ./tmp/cn=joomla.ldif
2 SASL/EXTERNAL authentication started
3 SASL username: gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth
4 SASL SSF: 0
5 adding new entry "cn=joomla,cn=schema,cn=config"
```

Amb l'extensió de l'esquema, Joomla ja pot treballar amb els objectes de l'LDAP com si fossin usuaris i grups tradicionals.

Quan es dona d'alta un usuari al LDAP que es vol fer servir amb Joomla, és molt important que aquest usuari tingui al menys una adreça de correu electrònic definida. L'atribut és *mail*. Si no és així, tot i estar correctament configurada la integració, l'usuari no podrà iniciar sessió, com s'aprecia en la figura [3.10](#).

FIGURA 3.10. Usuari sense correu electrònic a Joomla

The image shows a Joomla! login interface. At the top, there is a red error message box with a minus icon and the text "Please enter a valid email address." Below this, the login form consists of two input fields: "User Name" containing the text "testuser1" and "Password" which is masked with ten black dots. At the bottom left of the form is a button labeled "Log in" with a right-pointing arrow.

— Please enter a valid email address.

User Name testuser1

Password ●●●●●●●●●●

Log in