IV.Instalación Servicios de red

IV.1 Instalación del servidor DNS BIND

Para una instalación del servidor DNS BIND en Debian realiza el siguiente procedimiento como usuario **root**, teniendo en cuenta que el servidor está identificado como sigue:

- Hostname: **debian-servidor-fp.**
- IP: **192.168.200.250.**

1. Actualiza los repositorios del sistema operativo

```
root@debian-servidor-fp:~# apt-get update
```

NOTA: es necesario para el buen funcionamiento del comando que tengas configurado correctamente la conexión a Internet.

2. Actualiza el sistema operativo.

```
root@debian-servidor-fp:~# apt-get upgrade
```

3. Instala los paquetes necesarios para el funcionamiento de BIND (bind9).

```
root@debian-servidor-fp:~# apt-get install bind9 bind9utils
```

NOTA: La instalación crea el usuario **bind** que ejecuta el servicio dns denominado **named**.

4. Verifica que el servidor bind9 está activo.

```
root@debian-servidor-fp:~# service bind9 status
bind9 is running.
root@debian-servidor-fp:~# /etc/init.d/bind9 status
bind9 is running.
```

5. Verifica en qué puertos TCP y UDP está activo el servidor **bind9**, para ello comprueba el servicio **named**:

```
root@debian-servidor-fp:~# netstat -natp | grep named
tcp 0 0 192.168.200.250:53 0.0.0.0:* LISTEN 1442/named
tcp 0 0 127.0.0.1:53 0.0.0.0:* LISTEN 1442/named
tcp 0 0 127.0.0.1:953 0.0.0.0:* LISTEN 1442/named
tcp6 0 0 :::53 :::* LISTEN 1442/named
tcp6 0 0 ::1:953 :::* LISTEN 1442/named
root@debian-servidor-fp:~# netstat -naup | grep named
udp 0 0 192.168.200.250:53 0.0.0.0:* 1442/named
```

```
udp  0  0 127.0.0.1:53  0.0.0.0:*  1442/named
udp6 0  0 :::53         :::*       1442/named
```

IV.1.1 Archivos de configuración del servidor DNS

Tras la instalación del servidor DNS BIND (`bind9`) existe la ruta `/etc/bind`, la cual contiene sus ficheros de configuración. Una estructura tipo de `/etc/bind` que puedes encontrar al instalar bind sería similar a la que se muestra en la siguiente imagen:

```
root@debian-servidor-fp:~# tree /etc/bind
/etc/bind
├── bind.keys
├── db.0
├── db.127
├── db.255
├── db.empty
├── db.local
├── db.root
├── named.conf
├── named.conf.default-zones
├── named.conf.local
├── named.conf.options
├── rndc.key
└── zones.rfc1918

0 directories, 13 files
root@debian-servidor-fp:~#
```

El servidor DNS BIND (`bind9`) posee por defecto en su instalación el fichero /etc/bind/named.conf , que contiene la configuración principal, de la que beben todos los demás ficheros de configuración. En su contenido puedes ver las siguientes líneas, que añaden la configuración de determinados ficheros a la configuración principal, dedicados a particularizar la misma:

```
include "/etc/bind/named.conf.options";
include "/etc/bind/named.conf.local";
include "/etc/bind/named.conf.default-zones";
```

Donde,

`/etc/bind/named.conf.options`: hace referencia al archivo de configuración que posee opciones genéricas.

`/etc/bind/named.conf.local`: hace referencia al archivo de configuración para opciones particulares.

`/etc/bind/named.conf.default-zones`: hace referencia al archivo de configuración de zonas.

Dentro de cada uno de estos archivos encontrarás partes de código agrupadas entre llaves que finalizan con el carácter punto y coma (`;`), conocidos como declaraciones, las cuales indicarán secciones de ejecución. Cualquier código en un archivo de configuración que comience con los caracteres doble barra (`//`), almohadilla (`#`) o

aparezca encerrado entre barra asterisco (`/*`) y asterisco barra (`*/`) son considerados comentarios y por lo tanto no se ejecuta.

Puedes modificar los ficheros de configuración a tu antojo. Así, puedes crear incluso nuevos ficheros de configuración que sean llamados desde otros mediante la directiva `include`.

Puedes realizar una verificación de los ficheros de configuración y de zona por posibles fallos mediante los comandos "`named-checkconf`" y "`named-checkzone`" respectivamente. Estos comandos suelen ejecutarse con la siguiente sintaxis:

```
named-checkconf [-p] {filename}
```

donde,

`named-checkconf` → comprueba la sintaxis, pero no la semántica de un fichero de configuración `named`. El fichero se analiza y comprueba por errores de sintaxis, junto con todos los archivos incluidos en él. Si no se especifica ningún fichero, por defecto se comprueba `/etc/named.conf`. `-p` → imprime la salida de `named.conf` y los ficheros incluidos en forma canónica si no fueron detectados errores.

filename → El nombre del archivo de configuración que desea comprobar. Si no se especifica, por defecto es `/etc/named.conf`.

```
named-checkzone {zonename} {filename}
```

donde,

`named-checkzone` → comprueba la sintaxis y la integridad de un archivo de zona. Realiza las mismas comprobaciones que `named` hace al cargar una zona. Esto hace que sea útil para comprobar los archivos de zona antes de configurarlos en un servidor de nombres.

`zonename` → El nombre de dominio de la zona que se comprueba.

`filename` → El nombre del archivo de zona.

Ejemplos de ejecución:

1. Verificar archivo de configuración

   ```
   /etc/bind/named.conf:
   root@debian-servidor-fp:/etc/bind# named-checkconf -p
   /etc/bind/named.conf
   ```

2. Verificar el dominio de zona ejemplo.com en el archivo de zona

   ```
   /var/lib/bind/master/db.ejemplo.com.hosts
   ```

```
root@debian-servidor-fp:/etc/bind# named-checkzone ejemplo.com
/var/lib/bind/master/db.ejemplo.com.hosts
```

IV.1.2 Arranque y parada del servidor DNS

En un sistema operativo Debian 6.0 (Squeeze) puedes comprobar el estado del servicio bind mediante el comando service o mediante el comando **`/etc/init.d/bind`**:

- **Comando service:**
  1. Comprobar las opciones del comando:

```
root@debian-servidor-fp:~# service bind9
Usage: /etc/init.d/bind9 {start|stop|reload|restart|force-
reload|status}.
```

Donde,

`start` → opción que permite arrancar el servicio.

`stop` → opción que permite apagar el servicio.

`reload` → opción que permite recargar la configuración del servicio sin tener que reiniciarlo.

`restart` → opción que permite reiniciar el servicio.

`force-reload` → opción que permite forzar la recarga de configuración del servicio.

`status` → opción que permite comprobar si el servicio está activo o inactivo.

2. Arrancar el servidor DNS:

```
root@debian-servidor-fp:~# service bind9 start
Starting domain name service...: bind9.
```

3. Parar el servidor DNS:

```
root@debian-servidor-fp:~# service bind9 stop
Stopping domain name service...: bind9 waiting for pid 1989
to die.
```

4. Comprobar el estado activo/inactivo del servicio:

```
root@debian-servidor-fp:~# service bind9 status
could not access PID file for bind9 ... failed!
```

Como puedes comprobar la salida del comando determina que el servicio está inactivo, entonces lo arrancamos:

```
root@debian-servidor-fp:~# service bind9 start
Starting domain name service...: bind9.
```

Se vuelve a lanzar el comando para comprobar de nuevo el estado:

```
root@debian-servidor-fp:~# service bind9 status
bind9 is running.
```

Ahora puedes comprobar que la salida del comando determina que el servicio está activo.

- **Comando /etc/init.d/bind9:**
  1. Comprobar las opciones del comando:

```
root@debian-servidor-fp:~# /etc/init.d/bind9
Usage: /etc/init.d/bind9 {start|stop|reload|restart|force-
reload|status}.
```

  2. Arrancar el servidor DNS:

```
root@debian-servidor-fp:~# /etc/init.d/bind9 start
Starting domain name service...: bind9.
```

  3. Parar el servidor DNS:

```
root@debian-servidor-fp:~# /etc/init.d/bind9 stop
Stopping domain name service...: bind9 waiting for pid 2061
to die.
```

  4. Comprobar el estado activo/inactivo del servicio:

```
root@debian-servidor-fp:~# /etc/init.d/bind9 status
could not access PID file for bind9 ... failed!
```

Como puedes comprobar la salida del comando determina que el servicio está inactivo, entonces lo arrancamos:

```
root@debian-servidor-fp:~# /etc/init.d/bind9 start
Starting domain name service...: bind9.
```

Se vuelve a lanzar el comando para comprobar de nuevo el estado:

```
root@debian-servidor-fp:~# /etc/init.d/bind9 status
bind9 is running.
```

Ahora puedes comprobar que la salida del comando determina que el servicio está activo.

IV.1.3 Configuración como caché DNS

Todos los servidores DNS son servidores caché, pero no por ello deben ser maestro o esclavo. Así, existe la posibilidad que un servidor DNS funcione solamente como servidor caché, sin que sea maestro o esclavo.

En GNU/Linux Debian la configuración de un servidor DNS BIND (`bind9`) como caché viene establecida en el archivo `/etc/bind/named.conf.options`, donde se indica: el directorio de caché y los servidores DNS a reenviar las peticiones que no se pueden resolver de forma local mediante la caché: los servidores `forwarders`, para que luego estas consultas se vayan guardando en la caché.

El directorio de caché, `/var/cache/bind`, está configurado y habilitado por defecto tras la instalación y los servidores DNS a reenviar las peticiones que no se pueden resolver de forma local mediante la caché, los servidores `forwarders`, aparecen en una sección del mismo nombre y que por defecto está comentada, esto es, deshabilitada.

Para activar la caché debes realizar el siguiente procedimiento:

1. Verifica que el contenido del fichero `/etc/bind/named.conf.options`, tras la instalación, es el siguiente:

```
options {
 directory "/var/cache/bind";

 // If there is a firewall between you and nameservers you want
 // to talk to, you may need to fix the firewall to allow multiple
 // ports to talk. See http://www.kb.cert.org/vuls/id/800113

 // If your ISP provided one or more IP addresses for stable
 // nameservers, you probably want to use them as forwarders.
 // Uncomment the following block, and insert the addresses replacing
 // the all-0's placeholder.

 //forwarders {
 0.0.0.0;
 };

 auth-nxdomain no; # conform to RFC1035
 listen-on-v6 { any; };
};
```

2. Modificas el fichero /etc/resolv.conf para que solamente tenga activa la siguiente línea:

   ```
   nameserver 127.0.0.1
   ```

   De tal forma que ahora el servidor DNS activo solamente es el local, que tienes configurado como caché.

3. Una vez efectuados los cambios recargas el servidor con el comando: `service bind9 reload` ó `/etc/init.d/bind9 reload`.

El borrado de la caché DNS la puedes realizar en el cliente DNS y en el propio servidor DNS. Así, para un sistema operativo GNU/Linux podrías realizar los siguientes comandos según el caso:

1. Borrado de la caché del cliente DNS:

```
# /etc/init.d/nscd restart
```

2. Borrado de la caché del servidor DNS BIND:

```
# /usr/sbin/rndc flush
```

IV.1.4 Configuración como DNS maestro

En GNU/Linux Debian puedes configurar un servidor DNS BIND como maestro modificando el archivo `/etc/bind/named.conf.local` realizando el siguiente procedimiento:

1. Configuras el fichero `/etc/bind/named.conf.local` para indicar: qué zonas son servidas por el servidor, qué zonas son servidas como master y el fichero donde se guarda el contenido de la zona. Por ejemplo:

```
//zonas creadas tipo master
zone "ejemplo.com" {
 type master;
 file "/var/lib/bind/master/db.ejemplo.com.hosts";
};
```

   En este ejemplo, el servidor sirve el dominio "ejemplo.com" como master, y la zona se guarda en el fichero `/var/lib/bind/master/db.ejemplo.com.hosts`.

   Habrá una entrada de este tipo por cada zona servida.

   Normalmente los ficheros de zona están situados en la ruta `/var/lib/bind`. Entonces, para una mayor comprensión y entendimiento, y para facilidad de uso en posteriores momentos, estaría bien que crearas los directorios master y slave dentro de esa ruta. Así, los ficheros con zonas maestras se pueden encontrar en `/var/lib/bind/master/db.*.hosts` y los ficheros con zonas esclavas se pueden encontrar en `/var/lib/bind/slave/db.*.hosts`.

2. Configuras el fichero `/var/lib/bind/master/db.ejemplo.com.hosts` para agregar los registros RR a la zona, por ejemplo:

```
;
; BIND Database file for ejemplo.com zone
```

```
;

@ IN SOA ejemplo.com. hostmaster.ejemplo.com. (
 2011091601 ; serial number
 3600 ; refresh
 600 ; retry
 1209600 ; expire
 3600 ) ; default TTL
;
 IN NS ns.ejemplo.com.
 IN MX 10 mail.ejemplo.com.
 IN TXT ( "v=spf1 mx ~all" )
;

localhost A 127.0.0.1
ns A 192.168.200.250
mail A 192.168.200.251
www A 192.168.200.252
```

3. Recargas el servidor con el comando: service bind9 reload ó /etc/init.d/bind9 reload.
4. Realizas la siguiente consulta: `dig ejemplo.com` obteniendo una salida similar a la siguiente:

```
; <<>> DiG 9.7.3 <<>> ejemplo.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 25588
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1,
ADDITIONAL: 0

;; QUESTION SECTION:
;ejemplo.com. IN A

;; AUTHORITY SECTION:
ejemplo.com. 3600 IN SOA ejemplo.com. hostmaster.ejemplo.com.
2011091601 3600 600 1209600 3600

;; Query time: 3 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Tue Sep 27 11:48:36 2011
;; MSG SIZE rcvd: 76
```

IV.1.5 Configuración como DNS esclavo

En GNU/Linux Debian puedes configurar un servidor DNS BIND como esclavo modificando el archivo `/etc/bind/named.conf.local` realizando el siguiente procedimiento:

1. Configuras el fichero `/etc/bind/named.conf.local` del servidor esclavo para indicar: qué zonas son servidas por el servidor, qué zonas son servidas como slave, la IP del servidor master -de donde se transferirá la zona cuando se reciba una notificación de cambio, o se supere el TTL de la zona- y el fichero donde se guarda el contenido de la zona. Por ejemplo:

```
//zonas creadas tipo esclavo
zone "ejemplo.com" {
 type slave;
 masters {
 192.168.200.250;
 };
 file "/var/lib/bind/slave/db.ejemplo.com.hosts";
};
```

En este ejemplo, el servidor sirve el dominio "`ejemplo.com`" como slave, y la zona se guarda en el fichero `/var/lib/bind/slave/db.ejemplo.com.hosts`.
Habrá una entrada de este tipo por cada zona servida.

Normalmente los ficheros de zona están situados en la ruta `/var/lib/bind`. Entonces, para una mayor comprensión y entendimiento, y para facilidad de uso en posteriores momentos, estaría bien que crearas los directorios `master` y `slave` dentro de esa ruta. Así, los ficheros con zonas maestras se pueden encontrar en `/var/lib/bind/master/db.*.hosts` y los ficheros con zonas esclavas se pueden encontrar en `/var/lib/bind/slave/db.*.hosts`.

2. En el servidor maestro configuras la sección correspondiente al servidor master en el fichero `/etc/bind/named.conf.local`:
    a. Para indicar qué servidores tienen permitido la transferencia de los ficheros de zona, mediante la directiva `allow-transfer`. Por ejemplo:

```
allow-transfer{192.168.200.100;192.168.210.100;10.10.42.41;10.10.42.42;};
```

    En este listado deberán estar incluidos todos los servidores slave que tengan configurado a éste como servidor master, y adicionalmente alguna IP que debiera tenerlo permitido por alguna razón.

    b. Mediante la directiva notify-yes se consigue enviar automáticamente una notificación de cambio de zona del maestro, cuando ésta se produce, a los servidores DNS especificados en la zona mediante el registro de recurso NS.
    Adicionalmente, se puede enviar una notificación de cambio de zona a servidores esclavos que no aparecen en la misma, mediante la directiva `also-notify`:

```
also-notify {192.168.200.100;10.10.42.41;};
```

    Por ejemplo, una zona tipo master con las directivas anteriores podría ser la siguiente:

```
//zonas creadas tipo master
     zone "ejemplo.com" {
          type master;
          file "/var/lib/bind/master/db.ejemplo.com.hosts";
```

```
        allow-  transfer{192.168.200.100;192.168.210.100;10.10.42.41;10.10.42.42;};
         notify-yes;
        also-notify {192.168.200.100;10.10.42.41;};
    };
```

Mediante la directiva `also-notify` se mantienen los servidores DNS sincronizados. Así, el servidor DNS esclavo podrá satisfacer las peticiones DNS al igual que lo haría el maestro. Esto implica que se garantiza la disponibilidad del servicio DNS puesto que, aunque el servidor maestro deje de funcionar, el servidor esclavo podrá seguir ofreciendo el servicio. Además, en caso de recibir múltiples conexiones concurrentes, siendo, por tanto, el número de peticiones muy elevado, la carga se distribuye entre los servidores.

IV.2 Instalación de OpenLDAP

El proceso de instalación de OpenLDAP en un sistema basado en Debian es sencillo, no tanto, como verás, será la configuración.

Para una instalación de OpenLDAP en Debian realiza el siguiente procedimiento como usuario **root**, teniendo en cuenta que el servidor está identificado como sigue:

- Hostname: **debian-servidor-fp**
- IP: **192.168.200.250**

1. Actualiza los repositorios del sistema operativo.

```
root@debian-servidor-fp:~# apt-get update
```

2. Actualiza el sistema operativo.

```
root@debian-servidor-fp:~# apt-get upgrade
```

3. Instala los paquetes necesarios para el funcionamiento de OpenLDAP. La instalación te pedirá una contraseña, como puedes ver:

```
root@debian-servidor-fp:~# apt-get install slapd ldap-utils
Contraseña del administrador: admin
Verificación de la contraseña: admin
```

4. Verifica que el servidor OpenLDAP está activo, por defecto, en el puerto TCP 389.

```
root@debian-servidor-fp:~# netstat -natp | grep 389
tcp 0 0 0.0.0.0:389 0.0.0.0:* LISTEN 1955/slapd
tcp6 0 0 :::389 :::* LISTEN 1955/slapd
```

IV.2.1 Configuración de OpenLDAP

Una de las principales novedades de la versión 2.4 de OpenLDAP es que se incluye toda la configuración del servidor `slapd` en un directorio de base `cn=config`, (`/etc/ldap/slapd.d/cn=config`), en lugar del habitual fichero `/etc/ldap/slapd.conf`. Esto tiene la ventaja de que las modificaciones de configuración se pueden hacer sin tener que reiniciar el servicio.

Dentro del directorio `/etc/ldap/slapd.d/cn=config`, en una instalación limpia, puedes observar el objeto cn=schema, donde se encuentran los cuatro esquemas instalados por defecto: `core`, `cosine`, `nis` e `inetorgperson`.

Puedes encontrar más esquemas dentro de `/etc/ldap/schema`. Para añadir un esquema nuevo al directorio hay que subir un fichero `ldif` con el nuevo esquema al `dn`: `cn=schema,cn=config`.

La tabla siguiente ofrece un resumen de las clases de objetos utilizadas en el ejemplo de `core.schema` e `inetorgperson.schema` junto con los atributos obligatorios y los valores adecuados de atributo.

Clases de objetos y atributos de uso extendido

Clase de objeto	Significado	Entrada de ejemplo	Atributo obligatorio
`dcObject`	domainComponent (partes del nombre del dominio).	ejemplo.	`dc`
`organizationalUnit`	organizationalUnit (unidad organizativa).	People.	`ou`

IV.2.1.1 Arranque y parada del servidor LDAP

En un sistema operativo Debian 6.0 (Squeeze) puedes comprobar el estado del servicio OpenLDAP mediante el comando service o mediante el comando **`/etc/init.d/slapd`**:

- **Comando `service`**:
  1. Comprobar las opciones del comando:

```
root@debian-servidor-fp:~# service slapd
Usage: /etc/init.d/slapd {start|stop|reload|restart|force-
reload|status}
```

Donde:

`start` → opción que permite arrancar el servicio.

`stop` → opción que permite apagar el servicio.

`reload` → opción que permite recargar la configuración del servicio sin tener que reiniciarlo.

`restart` → opción que permite reiniciar el servicio.

`force-reload` → opción que permite forzar la recarga de configuración del servicio.

`status` → opción que permite comprobar si el servicio está activo o inactivo.

2. Arrancar el servidor OpenLDAP:

```
root@debian-servidor-fp:~# service slapd start
Starting OpenLDAP: slapd.
```

3. Parar el servidor OpenLDAP:

```
root@debian-servidor-fp:~# service bind9 stop
Stopping OpenLDAP: slapd.
```

4. Comprobar el estado activo/inactivo del servicio:

```
root@debian-servidor-fp:~# service slapd status
could not access PID file for slapd ... failed!
```

Como puedes comprobar la salida del comando determina que el servicio está inactivo, entonces lo arrancamos:

```
root@debian-servidor-fp:~# service slapd start
Starting OpenLDAP: slapd.
```

Se vuelve a lanzar el comando para comprobar de nuevo el estado:

```
root@debian-servidor-fp:~# service slapd status
slapd is running.
```

Ahora puedes comprobar que la salida del comando determina que el servicio está activo.

- **Comando /etc/init.d/slapd:**
  1. Comprobar las opciones del comando:

```
root@debian-servidor-fp:~# /etc/init.d/slapd
```

```
Usage: /etc/init.d/slapd {start|stop|reload|restart|force-
reload|status}.
```

2. Arrancar el servidor OpenLDAP:

```
root@debian-servidor-fp:~# /etc/init.d/slapd start
Starting OpenLDAP: slapd.
```

3. Parar el servidor OpenLDAP:

```
root@debian-servidor-fp:~# /etc/init.d/slapd stop
Stopping OpenLDAP: slapd.
```

4. Comprobar el estado activo/inactivo del servicio:

```
root@debian-servidor-fp:~# /etc/init.d/slapd status
could not access PID file for slapd ... failed!
```

Como puedes comprobar la salida del comando determina que el servicio está inactivo, entonces lo arrancamos:

```
root@debian-servidor-fp:~# /etc/init.d/slapd start
Starting OpenLDAP: slapd.
```

Se vuelve a lanzar el comando para comprobar de nuevo el estado:

```
root@debian-servidor-fp:~# /etc/init.d/slapd status
slapd is running.
```

Ahora puedes comprobar que la salida del comando determina que el servicio está activo.

El comando `slaptest` permite verificar la configuración del servidor OpenLDAP.

IV.2.1.2 Administrando un servidor LDAP

OpenLDAP ofrece una serie de comandos para la administración de datos en el directorio LDAP, contenidos en el paquete `ldap-utils`. Los cuatro comandos más importantes para añadir, modificar, buscar y eliminar son explicados a continuación.

1. Añadir entradas: comando `ldapadd`.

   a. Crea la estructura básica del dominio LDAP mediante la ejecución de un fichero estructura_basica.ldif.

```
root@debian-servidor-fp:~# ldapadd -x -D cn=admin,dc=proyecto-
empresa,dc=local -w admin -f estructura_basica.ldif
```

```
adding new entry "dc=proyecto-empresa,dc=local"
adding new entry "ou=usuarios,dc=proyecto-empresa,dc=local"
adding new entry "ou=grupos,dc=proyecto-empresa,dc=local"
```

b. Añade un usuario a LDAP de nombre 'pruebas' y contraseña '123456 ' mediante el archivo usuario.ldif.

```
root@debian-servidor-fp:~# ldapadd -x -D cn=admin,dc=proyecto-
empresa,dc=local -w admin -f usuario.ldif
adding new entry "uid=upruebas,ou=usuarios,dc=proyecto-
empresa,dc=local"
```

2. Modificar entradas: comando `ldapmodify`.

a. Modificar la contraseña del usuario anterior 'pruebas' mediante la ejecución del archivo cambiar_usuario.ldif.

```
root@debian-servidor-fp:~# ldapmodify -x -D
cn=admin,dc=proyecto-empresa,dc=local -w admin -f
cambiar_usuario.ldif
modifying entry "uid=upruebas,ou=usuarios,dc=proyecto-
empresa,dc=local"
```

3. Buscar entradas: comando ldapsearch.

a. Buscar todos los usuarios cuyo nombre contenga los caracteres 'pru':

```
root@debian-servidor-fp:~# ldapsearch -x -b dc=proyecto-
empresa,dc=local "(cn=*pru*)"
```

b. Buscar todos los usuarios cuyo nombre contenga los caracteres 'pru' y cuyo correo electrónico contengan los caracteres 'daw05':

```
root@debian-servidor-fp:~# ldapsearch -x -b dc=proyecto-
empresa,dc=local "(&(cn=*pru*)(mail=*05*))"
```

4. Eliminar entradas: comando `ldapdelete`.

a. Eliminar el usuario upruebas:

```
root@debian-servidor-fp:~# ldapdelete -x -D
cn=admin,dc=proyecto-empresa,dc=local -w admin
uid=upruebas,ou=usuarios,dc=proyecto-empresa,dc=local
```

Los comandos anteriores poseen la opción `-h` con la cual se puede indicar el host (nombre de dominio o IP) que identifica al servidor LDAP. Por ejemplo: `ldapsearch -h 192.168.200.250 -x -b dc=proyecto-empresa,dc=local "(objectclass=*)"`

conectaría con el servidor LDAP en la IP `192.168.200.250` para buscar el DIT del dominio `proyecto-empresa.local`.

Existe un paquete de nombre ldapscripts que contiene una serie de scripts para administrar de forma sencilla los usuarios y grupos almacenados en el servidor LDAP. Puedes encontrar plantillas de ejemplo, formato LDIF, situadas en `/usr/share/doc/ldapscripts/examples/` cuando se instala el paquete `ldapscripts`.

IV.2.1.3 Configuración de los clientes. Instalación de librerías de autenticación

Como ya hemos comentado anteriormente, una de las utilidades más importantes de un servidor LDAP es la de servidor de autentificación. Autentificarse suele ser lo común y necesario para entrar en un sistema GNU/Linux. También para acceder a algunos servicios como un servidor FTP o a páginas privadas en un servidor web.

A continuación, verás las modificaciones que hay que realizar en un sistema GNU/Linux Debian para que autentifique a los usuarios en un servidor LDAP, esto es, verás los pasos a seguir para configurar un equipo como cliente LDAP. Así, el equipo en lugar de utilizar los clásicos archivos /etc/passwd, /etc/group y /etc/shadow, tomará los usuarios y grupos del servidor LDAP, autentificando los usuarios que inicien sesión validándose contra el servidor LDAP.

Esta configuración debe ser replicada en todos los clientes LDAP pertenecientes al dominio, incluido el propio servidor LDAP, si se quiere que los clientes accedan al mismo.

Para ello realiza el siguiente procedimiento:

1. Instala y configura los paquetes

```
root@debian-servidor-fp:~# apt-get install libnss-ldap libpam-ldap
nscd
URI del servidor de LDAP: ldap://192.168.200.250
El nombre distintivo (DN) de la base de búsquedas: dc=proyecto-
empresa,dc=local
Versión de LDAP a utilizar: 3
Cuenta LDAP para root: cn=admin,dc=proyecto-empresa,dc=local
Contraseña para la cuenta LDAP de root: admin
nsswitch.conf no se gestiona automáticamente
Debe modificar su fichero <</etc/nsswitch.conf>> … Aceptar
¿Desea permitir que la cuenta del administrador de LDAP se comporte
como el administrador local? Sí
¿Hacer falta un usuario para acceder a la base de datos de LDAP? No
Cuenta del administrador de LDAP: cn=admin,dc=proyecto-
empresa,dc=local
Contraseña del administrador de LDAP: admin
```

Toda esta configuración se ha guardado en el fichero `/etc/libnss-ldap.conf`

2. Modifica en el archivo `/etc/nsswitch.conf`:

```
/etc/nsswitch.conf::
passwd: files ldap
group: files ldap
shadow: files ldap
```

3. Reinicia el servicio nscd para que se activen los cambios efectuados en el paso anterior, esto es, para que el sistema operativo recoja los usuarios en primer lugar de los ficheros locales de usuarios y grupos y a continuación del servidor LDAP

```
root@debian-servidor-fp:~# service nscd restart
Restarting Name Service Cache Daemon: nscd.
```

4. Revisa mediante el comando pam-auth-update que los servicios: Unix authentication y LDAP Authentication, que el sistema operativo usa para autenticar los usuarios, están activados.

```
root@debian-servidor-fp:~# pam-auth-update
Perfiles PAM a habilitar:
[*] Unix authentication
[*] LDAP Authentication
```

5. Por último, prueba que la configuración del cliente es correcta:
   a. Mediante el comando `getent passwd`, que proporciona todos los usuarios del sistema operativo, en este caso los de Unix authentication y LDAP Authentication.

```
root@debian-servidor-fp:~# getent passwd | grep uprueba
upruebas:*:10001:10001:Pruebas DAW05:/home/upruebas:/bin/bash
upruebas2:*:10002:10001:upruebas2:/home/upruebas2:/bin/bash
```

   b. Iniciar sesión en una consola de texto en el equipo cliente con un usuario del LDAP. En este caso, con el usuario `upruebas` o el usuario `upruebas2`.

IV.2.1.4 Probar la autenticación con pamtest

Ahora que la autentificación de usuarios por LDAP está activada en el sistema operativo, es recomendable que efectúes algunas pruebas con la nueva configuración para comprobar si todo funciona correctamente.

El comando pamtest puede ayudarte a realizar estas pruebas. La instalación del mismo se efectúa realizando el siguiente comando:

```
root@debian-servidor-fp:~# apt-get install libpam-dotfile
```

El comando `pamtest` acepta dos parámetros: el primero es el nombre del servicio al cual se va a conectar para realizar la autentificación y el segundo es el nombre del usuario que se va a autentificar sobre dicho servicio. Veamos unos ejemplos:

1. Intentar autentificar al usuario `upruebas2` en el servicio passwd mediante una clave correcta:

```
root@debian-servidor-fp:~# pamtest passwd upruebas2
Trying to authenticate <upruebas2> for service <passwd>.
Password:
Authentication successful.
```

2. Intentar autentificar al usuario `upruebas2` en el servicio `passwd` mediante una clave incorrecta:

```
root@debian-servidor-fp:~# pamtest passwd upruebas2
Trying to authenticate <upruebas2> for service <passwd>.
Password:
Failed to authenticate: Authentication failure
```

3. Intentar autentificar al usuario `upruebas2` en el servicio `ssh` mediante una clave correcta:

```
root@debian-servidor-fp:~# pamtest ssh upruebas2
Trying to authenticate <upruebas2> for service <ssh>.
Password:
Authentication successful.
```

4. Intentar autentificar al usuario `upruebas2` en el servicio `ssh` mediante una clave incorrecta:

```
root@debian-servidor-fp:~# pamtest ssh upruebas2
Trying to authenticate <upruebas2> for service <ssh>.
Password:
Failed to authenticate: Authentication failure
```

5. Intentar autentificar al usuario `upruebas2` en el servicio `ftp` mediante una clave correcta:

```
root@debian-servidor-fp:~# pamtest ftp upruebas2
Trying to authenticate <upruebas2> for service <ftp>.
Password:
Authentication successful.
```

6. Intentar autentificar al usuario `upruebas2` en el servicio `ftp` mediante una clave incorrecta:

```
root@debian-servidor-fp:~# pamtest ftp upruebas2
Trying to authenticate <upruebas2> for service <ftp>.
Password:
Failed to authenticate: Authentication failure
```

Una vez se ha llegado a este punto, el sistema ya está preparado para autentificar a los usuarios a través de LDAP.