

Introducción a LDAP (Memorias de investigación)

Salvador Rebollo Benítez - Memoria de investigación en formación DUAL en DEKRA

Índice

1. Introducción a LDAP ¿Qué es LDAP y para que se utiliza?	2
Estructura orientada a objetos en LDAP	3
2. Diferencias entre LDAP y Active Directory	4
Qué es Active Directory	4
Diferencia entre LDAP y Active Directory	4
3. ¿Cómo instalar y configurar un servidor LDAP?	5
¿Qué es OpenLDAP?	5
Funcionamiento	5
¿Cómo instalar y configurar un servidor LDAP?	6
3.1 Empezamos - Instalar y configurar OpenLDAP en el servidor Ubuntu	7
3.2 Realizar la configuración básica de OpenLDAP	9
3.3 Configurar la autenticación para los clientes	14
3.4 Configurar el demonio (daemon) SLAPD	16
3.5 Crear la estructura del directorio (Ejemplo)	21
3.6 Añadir un usuario y un grupo	23
Añadir un usuario	23
Añadir un grupo	24
3.7 Comprobar que todo es correcto	26
4. Ejemplos de aplicación que se integran con LDAP y como lo hacen	27
Software Cliente	27
Software Servidor	28
5. ¿Cómo representar un organigrama en LDAP? Ejemplos	29
Ficheros LDIF	29
Para añadir algún objeto al directorio	29
Ejemplo de fichero LDIF	30
6. Componentes y funcionalidades que ofrece Talend para obtener datos de LDAP/Active Directory	31
Configurando METADATA LDAP en Talend	31
Haciendo un JOB de prueba	33
Componentes de Talend en Exchange	34
Resumen de otros componentes que podemos encontrar en Talend Exchange	34
7. Librerías de Java para integrar con LDAP	35
8. Preguntas y Respuestas	36

1. Introducción a LDAP ¿Qué es LDAP y para que se utiliza?

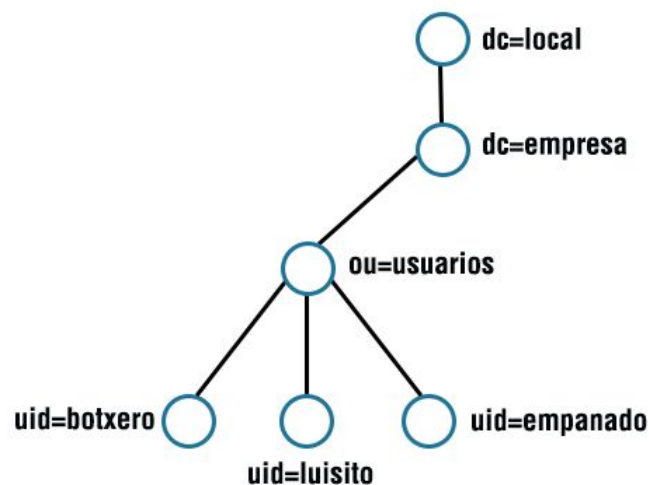
LDAP son las siglas de **L**ightweight **D**irectory **A**ccess **P**rotocol. Es un protocolo a nivel de aplicación que permite realizar consultas sobre un servicio de directorio para poder buscar información.

Un **servicio de directorio** es como una base de datos en la que organizar y almacenar información con objetos de distintas clases. Esta estructura organizada de forma jerárquica de los objetos en el directorio, se consigue con la implementación de LDAP.

LDAP está concebido y altamente optimizado para **lecturas**, como un servicio de consultas, vamos a consultar muchas veces y **realizar pocas modificaciones** sobre nuestro directorio. A pesar de tener características **comunes con las bases de datos** tienen muchas **diferencias** significativas, ya que LDAP está diseñado para sobretodo lecturas, **no soporta** transacciones ni bloqueos, así que por ejemplo para un histórico de logs del sistema, que suelen tener muchas escrituras y pocas lecturas este no sería el mejor protocolo para almacenar, para eso están las bases de datos.

Con LDAP se define la manera de cómo acceder a ese directorio, es decir, **está optimizado para la realización de operaciones de lectura** sobre el directorio como puede ser por ejemplo, la de validar el acceso autenticado a un usuario almacenado en el directorio.

Un **servicio de directorio** ejecuta el modelo **cliente-servidor**, por lo que si un equipo cliente desea acceder al directorio no accede directamente a la base de datos, contacta con un proceso en el servidor. El proceso consulta al directorio y devuelve el resultado de la operación al cliente.



En esta imagen se ve cómo es la **estructura de un directorio LDAP**. Cada círculo que se ve en la imagen representa a un objeto y de esta manera se va creando un árbol.

Estructura orientada a objetos en LDAP

- **Clases:** en las clases se **definen los objetos y sus características**. Por ejemplo el tipo de objeto que se va a definir y los atributos que va a contener en función del tipo de objeto. En el **esquema** se define cada clase con los atributos que serán obligatorios y opcionales para cada entrada creada.
- **Objetos:** los objetos son entradas en el directorio. **Los objetos son instancias** creadas a partir de **una** determinada **clase** o de varias, en función de los atributos necesarios para un objeto. Todo el directorio va a estar compuesto por objetos (usuarios, grupos, unidades organizativas,...).
- **Atributos:** los atributos son los **campos asociados a cada objeto** creado y definen las características del mismo. Cada atributo tiene un valor (información). Por ejemplo, el nombre del usuario, apellidos, número de teléfono, etc.
- **DN:** el campo DN es el **nombre distinguido** (Distinguished Name) para **identificar de forma única** a un determinado objeto en el directorio. Es decir, **cada entrada definida es única en todo el directorio**. Siguiendo la imagen de ejemplo directorio de arriba, el DN de empanado sería:

dn: uid=empanado,ou=usuarios,dc=empresa,dc=local

Como podemos ver el DN de ese objeto (de tipo usuario) va a ser único en todo el directorio y **le identificará unívocamente**. Por poner un ejemplo más sencillo de entender, es como los ficheros en nuestro ordenador. Si tenemos un fichero de texto en **C:\Documentos\sacamantecas.txt** está será la ruta con la que el sistema identifique al fichero y lo hará único. Puede haber más ficheros con ese nombre en todo el equipo, pero nunca en esa misma carpeta.

Si tomamos como ejemplo la explicación de la anterior ruta, se puede entender mejor qué es el RDN (Nombre Distinguido Relativo). Se puede decir que **el DN está formado por la concatenación** de la ruta hasta llegar al objeto y cada parte que forma esa ruta es el **RDN**.

Ejemplo

RDN: ou=usuarios

DN (ruta completa): uid=empanado,ou=usuarios,dc=empresa,dc=local

2. Diferencias entre LDAP y Active Directory.

Antes de matizar las diferencias voy a hacer una pequeña introducción a:

Qué es Active Directory:

De forma sencilla se puede decir que es un servicio establecido en uno o varios servidores en donde se crean objetos tales como usuarios, equipos o grupos, con el objetivo de administrar los inicios de sesión en los equipos conectados a la red, así como también la administración de políticas en toda la red.

Su estructura jerárquica es lo que debemos construir y permite mantener una serie de objetos relacionados con componentes de una red, como usuarios, grupos de usuarios, permisos y asignación de recursos y políticas de acceso.

Active Directory permite a los administradores establecer políticas a nivel de empresa, desplegar programas en muchos ordenadores y aplicar actualizaciones críticas a una organización entera. Un Active Directory almacena información de una organización en una base de datos central, organizada y accesible. Pueden encontrarse desde directorios con cientos de objetos para una red pequeña hasta directorios con millones de objetos.

Diferencia entre LDAP y Active Directory:

De forma clara y concisa podría decirse que LDAP es como el **protocolo principal** en el que se basa Active Directory junto con algunos más como **DNS** (apuntar los dominios al servidor correspondiente y para traducir a la dirección real IP), **DHCP** (asignación dinámica de una dirección IP) o **Kerberos** (sólida autenticación de usuario y también integridad y privacidad). No se pueden comparar en sí mismos sino que Active Directory está basado en LDAP junto con más tecnologías para dar su servicio mientras que LDAP **puede tener un uso concreto en más de un ámbito**, Active Directory podría decirse que es el LDAP de Microsoft.

En la sección [4. Ejemplos de aplicación que se integran con LDAP y como lo hacen](#) de este documento hay un listado de programas de software que pueden comunicarse y/o hospedar servicios de directorio a través de LDAP como lo puede hacer Active Directory.

3. ¿Cómo instalar y configurar un servidor LDAP?

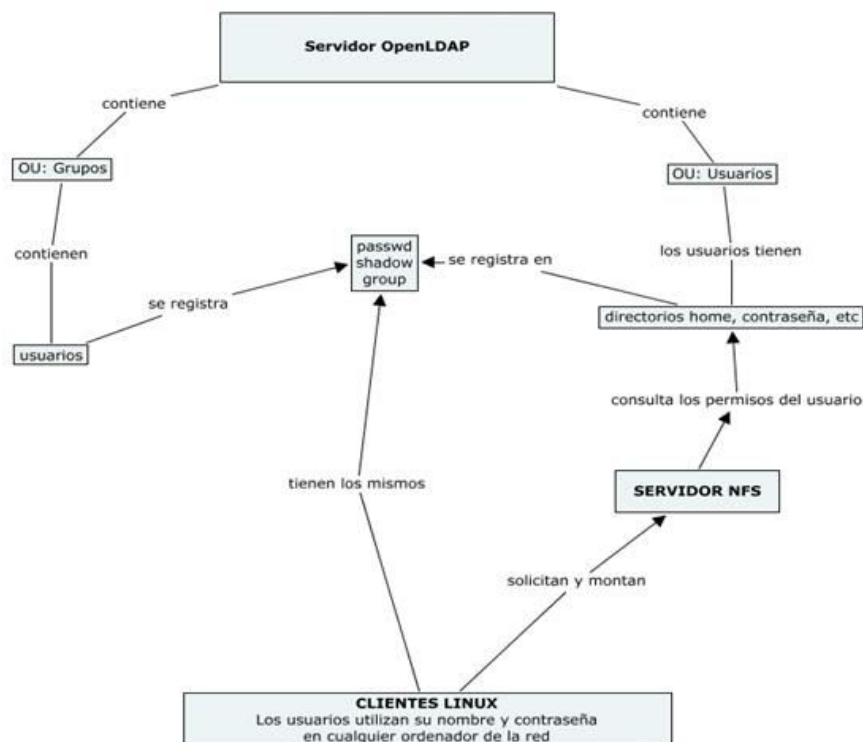
Por lo que he estado investigando, hay diferentes alternativas a la instalación de un servidor LDAP, la más famosa que he encontrado es **OpenLDAP**. Para instalar un servidor LDAP con OpenLDAP necesitamos tener Ubuntu, Debian o alguna distribución de Linux similar, siendo clientes sí que podemos optar por utilizar Windows u otro S.O.

¿Qué es OpenLDAP?

OpenLDAP es una herramienta de código abierto para crear directorios distribuidos de direcciones para autenticar usuarios. Se utiliza para administrar información relacionada desde una ubicación centralizada mediante el uso de una jerarquía de archivos y directorios. Es "ligero" en comparación con el X.500 y acabó sustituyéndolo. Está diseñado para funcionar también en equipos más pequeños como los de escritorio.

Funcionamiento

Un cliente LDAP se une (login) a un servidor LDAP. Se envía una consulta para solicitar información o presente información para actualizarse, todos los cambios efectuados se ven reflejados en un **log** el cual podemos consultar de forma recomendada usando el comando **grep** en el archivo [/var/log/slapd-log](#) en él se registra toda la información que el servicio LDAP tiene que registrar (El nivel de registro se define en el archivo slapd.conf o al inicio del servidor en la línea de comandos). Los permisos de acceso se revisan por el servidor y cuando está autorizado, el servidor responde con respuesta o tal vez con una referencia a otro servidor LDAP donde el cliente puede tener la consulta.



En OpenLDAP los permisos de acceso al directorio de direcciones se basan en dos categorías de funciones en **slapd**, lista de control de acceso y funciones de autorización. En Linux / Unix, los permisos de acceso a los sistemas de archivos se basan en permisos de directorio / archivo.

OpenLDAP implementaciones de LDAP se basa en un estándar. Las especificaciones que rigen la implementación de LDAP están en IETF RFC.

¿Qué es SLAPD?

SLAPD (Standalone LDAP Daemon) es un programa multiplataforma, que se ejecuta en segundo plano, atendiendo las solicitudes de autenticación LDAP que se reciban

en el servidor. Es uno de los componentes principales de OpenLDAP, el cual es el **daemon** o demonio el cual es un tipo especial de proceso informático no interactivo, es decir, que se ejecuta en segundo plano en vez de ser controlado directamente por el usuario.

¿Cómo instalar y configurar un servidor LDAP?

Para ello me he valido de la implementación con **OpenLDAP** y desplegado el servidor en una máquina virtual Ubuntu, se podría resumir en:

1. **Instalar** OpenLDAP en el servidor.
2. Realizar la **configuración básica** de OpenLDAP.
3. Configurar la **autenticación** para los clientes.
4. Configurar el demonio (**daemon**) **SLAPD**.

Y una vez instalado y configurado el servidor...

5. Crear la estructura del directorio.

Siguiendo algunas instrucciones vistas por internet procedemos a la explicación en la siguiente página.

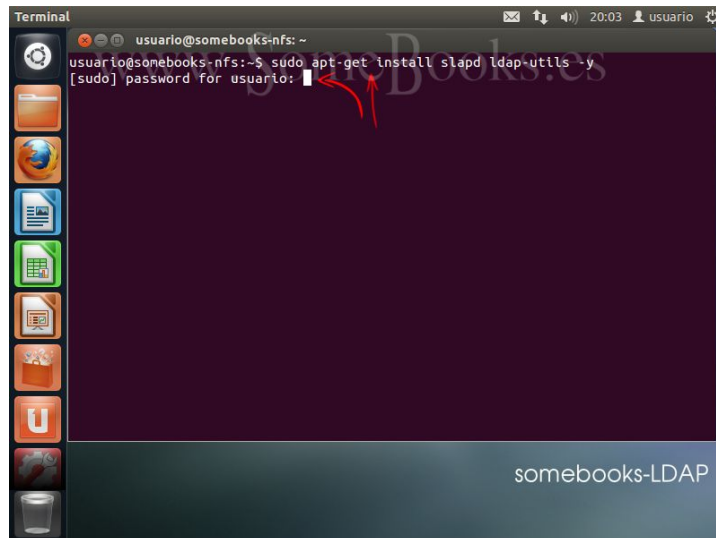
3.1 Empezamos - Instalar y configurar OpenLDAP en el servidor Ubuntu

El proceso de instalación es realmente sencillo. Básicamente consiste en instalar el paquete slapd, aunque nosotros también instalaremos el paquete que contiene las utilidades de administración de LDAP: ldap-utils.

Como ambos paquetes se encuentran en los repositorios oficiales de Ubuntu, sólo tenemos que escribir en la terminal la siguiente orden:

```
sudo apt-get install slapd ldap-utils
```

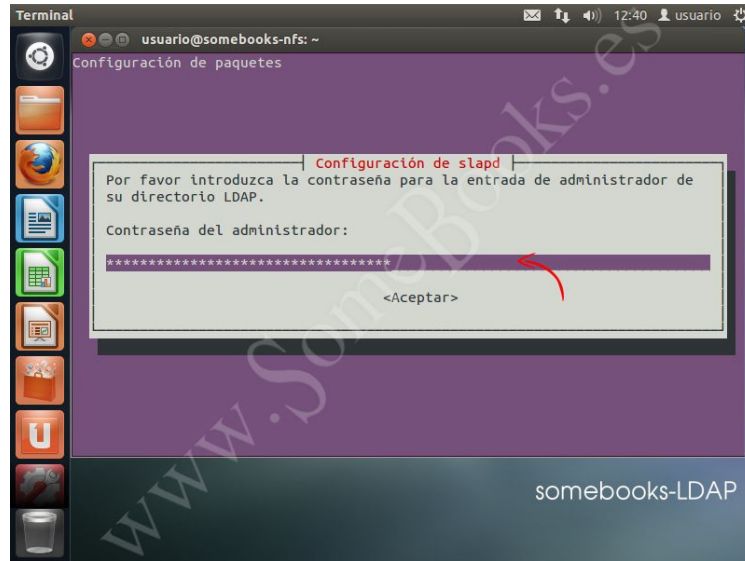
Como de costumbre, el sistema nos solicita la contraseña de administración.



Durante la instalación, aparece en la consola un mensaje que nos solicita la contraseña de administración para LDAP. Como siempre, deberá ser una contraseña segura.

Si consideras que la contraseña local cumple los requisitos, no hay ningún inconveniente para volver a usarla, aunque serás tú quien deba evaluar este aspecto en función de los requisitos de seguridad de tu entorno.

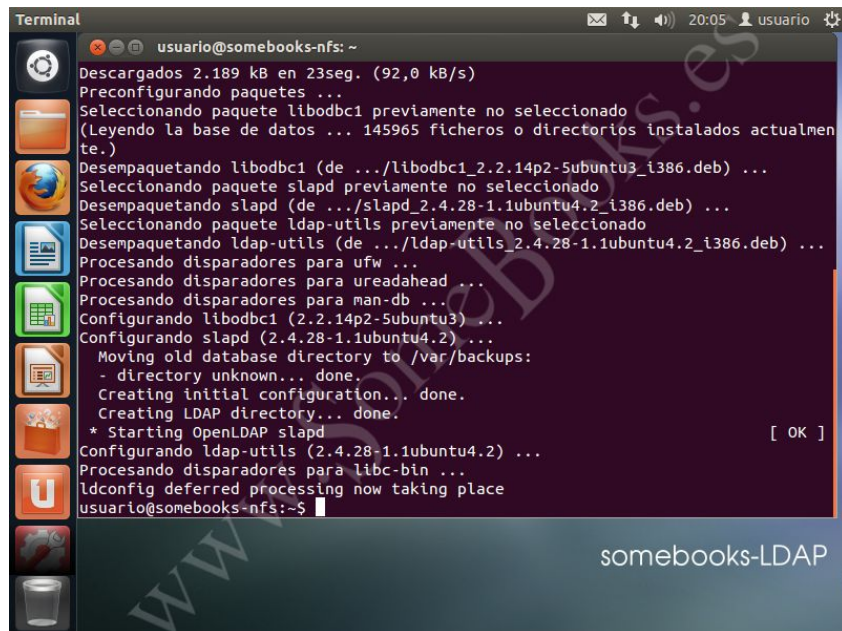
Cuando terminemos de escribir, pulsaremos la tecla Intro.



Como suele ocurrir cuando escribimos una contraseña, para evitar que hayamos cometido algún error tipográfico que después nos impida entrar, el sistema nos pide que volvamos a escribirla. Al hacerlo, volvemos al aspecto normal de la terminal y comprobaremos que la instalación sigue su curso.

NOTA: De forma predeterminada, slapd se configura con las mínimas opciones necesarias para que el demonio funcione de forma correcta.

Poco después, el proceso de instalación habrá terminado.



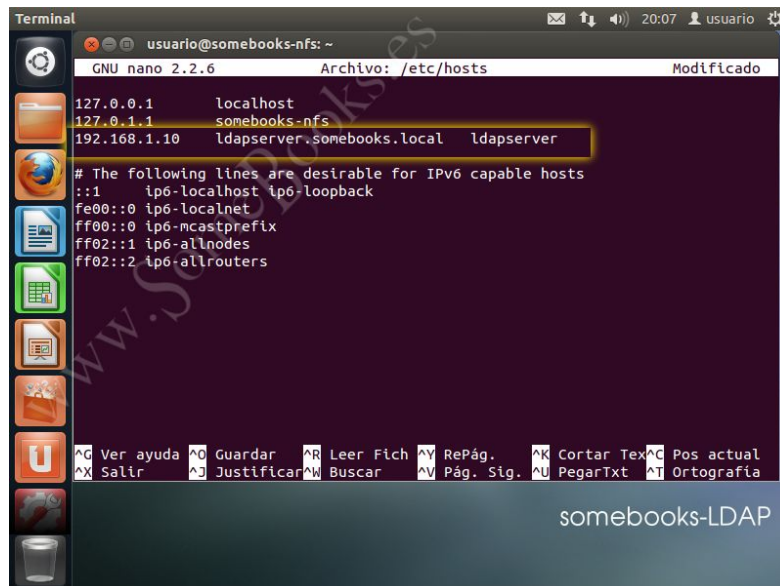
3.2 Realizar la configuración básica de OpenLDAP

Comenzaremos por modificar el contenido del archivo **/etc/hosts**. El objetivo es que, cuando hagamos referencia a los nombres `ldapserv` o `ldapserv.somebooks.local` (nombre de ejemplo), nuestro sistema entienda que nos estamos refiriendo al servidor.

Abrimos el archivo `/etc/hosts` usando el editor nano.

```
sudo nano /etc/hosts
```

Dentro del archivo, añadimos una nueva línea que relacione la dirección IP estática del servidor con los nombres lógicos que tenemos previsto utilizar. Cuando terminemos guardamos los cambios y cerramos.



```
Terminal
usuario@somebooks-nfs: ~
GNU nano 2.2.6 Archivo: /etc/hosts Modificado
127.0.0.1 localhost
127.0.1.1 somebooks-nfs
192.168.1.10 ldapserv.somebooks.local ldapserv
# The following lines are desirable for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
Ver ayuda Guardar Leer Fich RePág. Cortar Tex Pos actual
Salir Justificar Buscar Pág. Sig. PegarTxt Ortografia
somebooks-LDAP
```

A continuación, instalaremos la librería NSS para LDAP. Esta librería ofrece una interfaz para acceder y configurar distintas bases de datos utilizadas para almacenar cuentas de usuario (entre otras, `/etc/passwd`, `/etc/group`, `/etc/hosts`, LDAP, etc.).

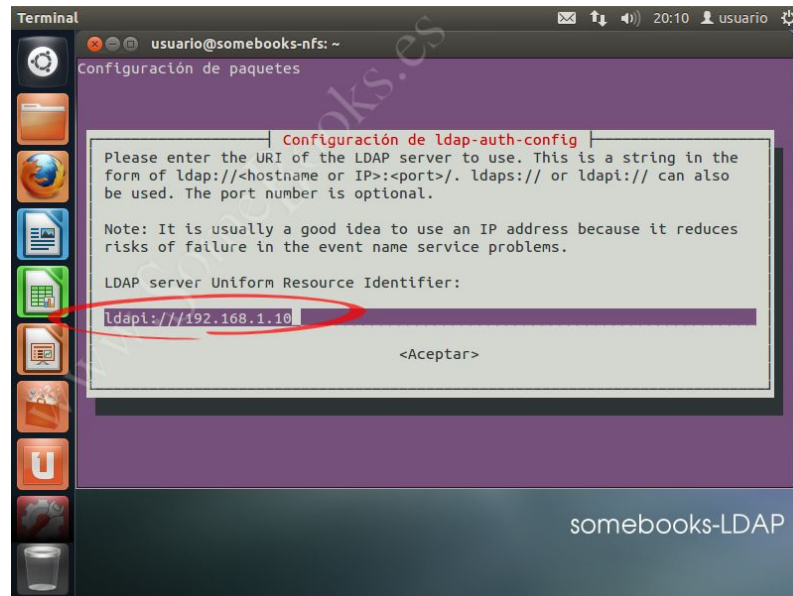
Conseguirlo es tan sencillo como instalar el paquete `libnss-ldap`.

```
sudo apt-get install libnss-ldap -y
```

Una de las dependencias del paquete `libnss-ldap` es el paquete de configuración de la autenticación de LDAP (`ldap-auth-config`). Durante su instalación se iniciará un asistente que nos irá solicitando la información que necesita para su correcta configuración.

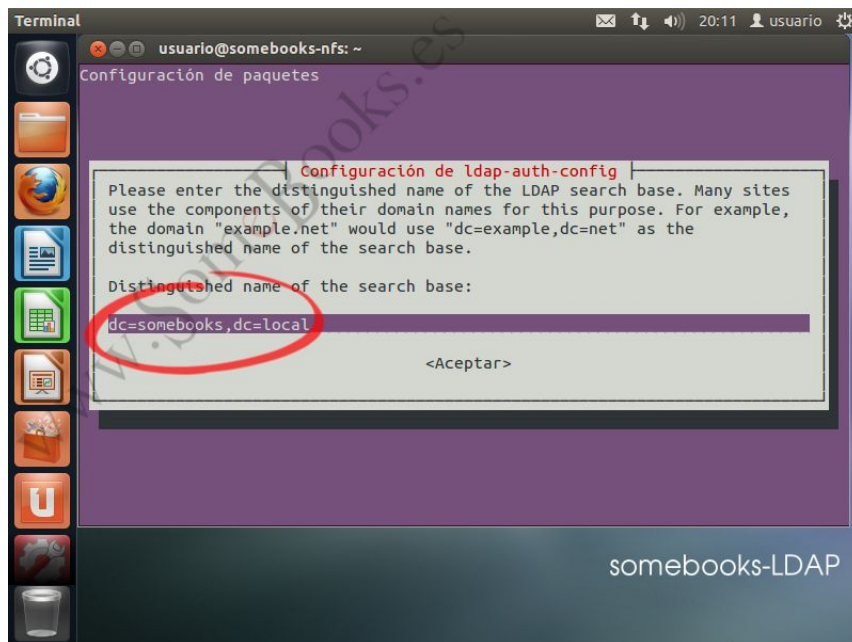
En el primer paso, nos solicita la dirección URI del servidor LDAP. Es importante dejar el principio tal y como lo encontramos (**ldapi:///**).

Cuando completemos la dirección IP, pulsaremos la tecla Intro. También podemos usar la tecla <Tabulador> para desplazarnos hasta la palabra <Aceptar>.



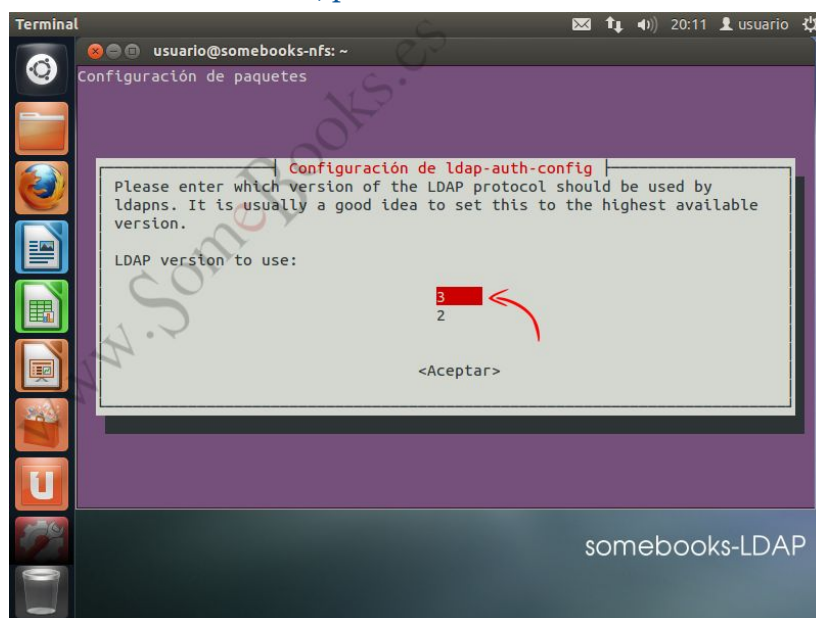
A continuación escribiremos el nombre global único (Distinguished Name – DN) siguiendo las indicaciones que vimos al principio de este capítulo (**dc=somebooks,dc=local**).

Cuando acabemos, volvemos a pulsar la tecla Intro.



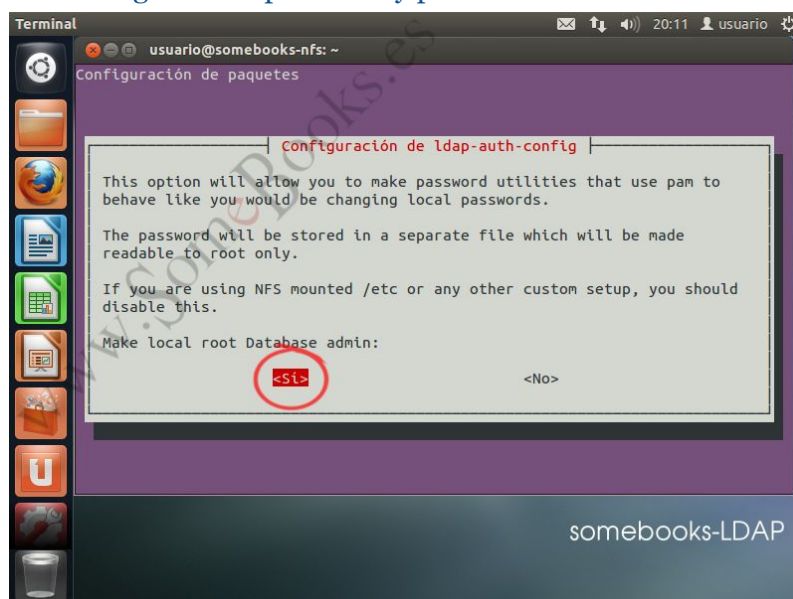
En el siguiente paso, indicaremos la versión del protocolo LDAP que vamos a utilizar. Salvo que dispongamos en nuestra red de clientes muy antiguos, lo normal será elegir el valor más alto.

Cuando acabemos, pulsamos de nuevo la tecla *Intro*

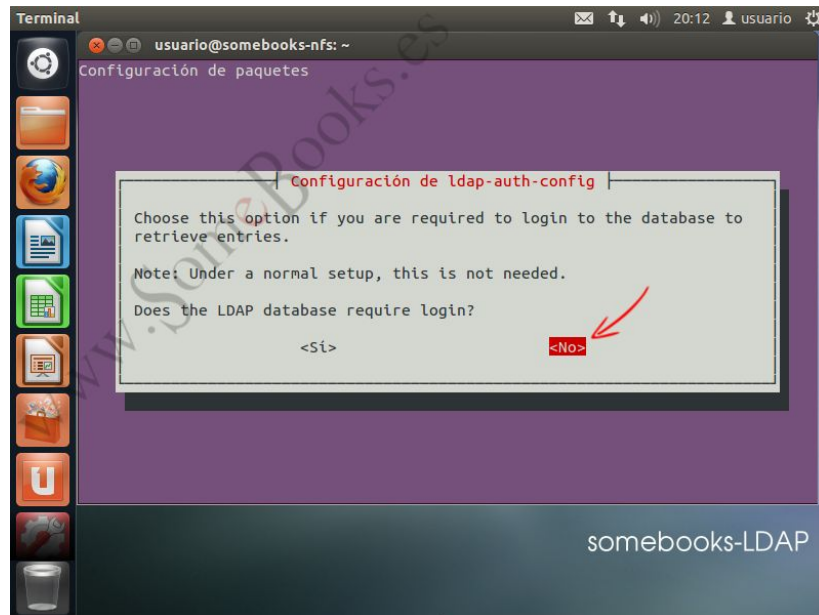


A continuación, indicaremos si las utilidades que utilicen PAM deberán comportarse del mismo modo que cuando cambiamos contraseñas locales. Esto hará que las contraseñas se guarden en un archivo independiente que sólo podrá ser leído por el superusuario.

Elegimos la opción Yes y pulsamos la tecla *Intro*.

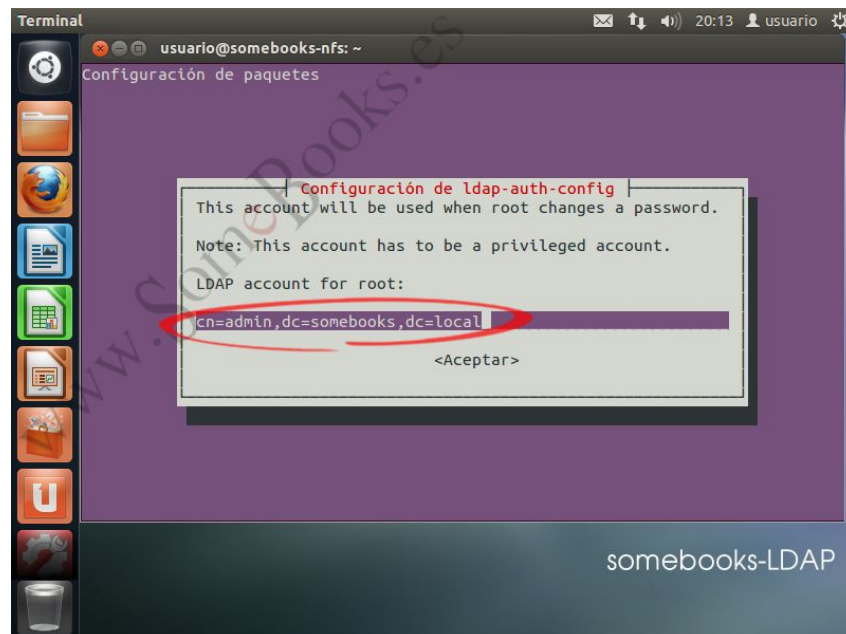


A continuación, el sistema nos pregunta si queremos que sea necesario identificarse para realizar consultas en la base de datos de LDAP. Elegimos la opción No y volvemos a pulsar la tecla *Intro*.



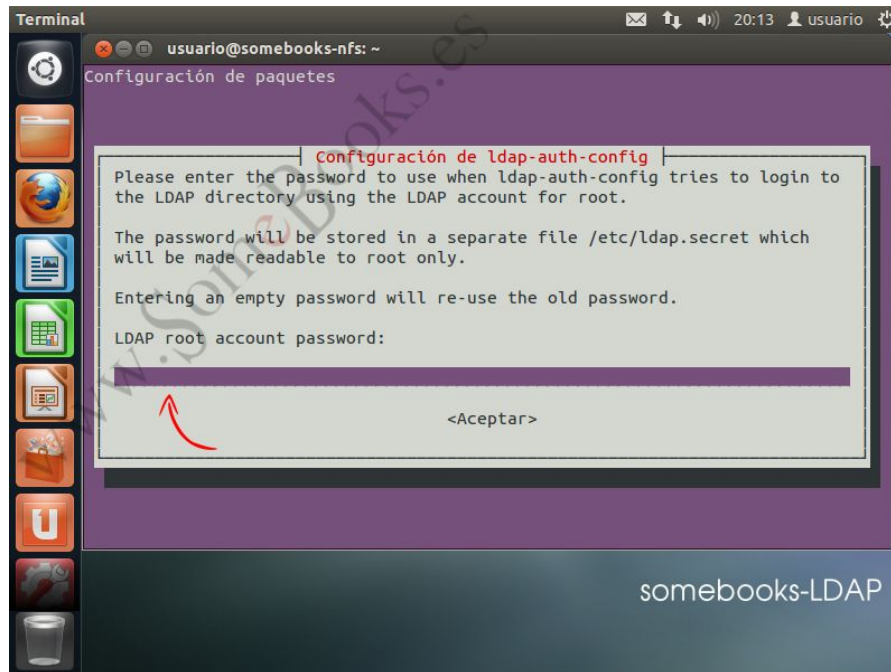
Ya sólo nos queda indicar el nombre de la cuenta LDAP que tendrá privilegios para realizar cambios en las contraseñas. Como antes, deberemos escribir un nombre global único (Distinguished Name – DN) siguiendo las indicaciones que vimos al principio de este capítulo (**cn=admin,dc=somebooks,dc=local**).

Después de escribir el nombre adecuado, pulsaremos la tecla *Intro*.

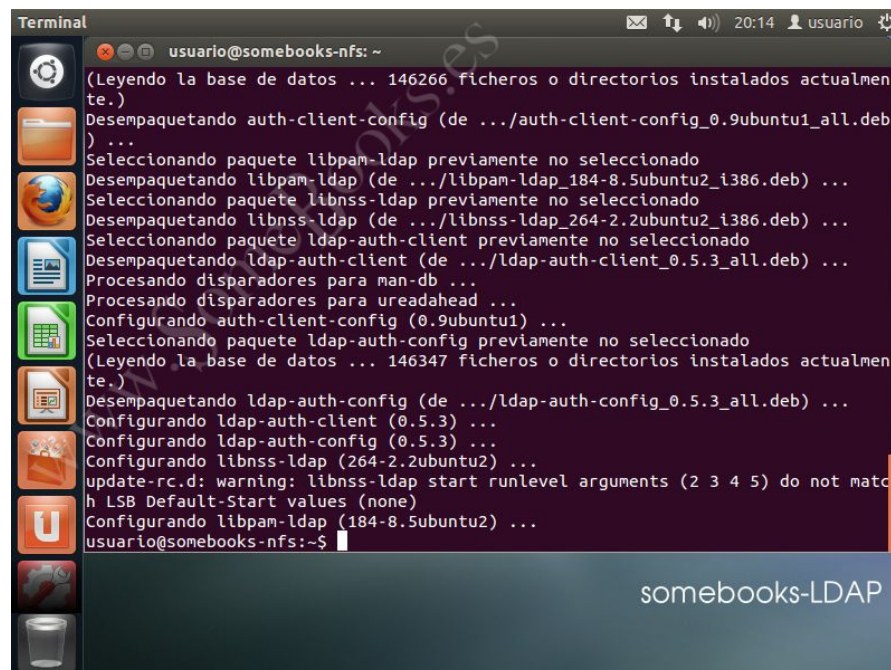


En el último paso, el asistente nos solicita la contraseña que usará la cuenta anterior (como siempre, habrá que escribirla por duplicado para evitar errores tipográficos). Deberá coincidir con la que escribimos en el apartado Instalar OpenLDAP en el servidor.

Cuando terminemos de escribir la contraseña, pulsaremos la tecla Intro.



De vuelta en la pantalla de la terminal, podremos comprobar que no ha habido errores durante el proceso. Con esto habremos terminado la configuración básica de LDAP.



Si más adelante observamos algún error o necesitamos efectuar alguna modificación, sólo tenemos que ejecutar el siguiente comando:

```
sudo dpkg-reconfigure ldap-auth-config
```


3.3 Configurar la autenticación para los clientes

Lo primero que haremos será utilizar `auth-client-config`, un script que nos ayuda a modificar los archivos de configuración de PAM y NSS. Para conseguirlo, ejecutamos el siguiente comando en la terminal:

```
sudo auth-client-config -t nss -p lac_ldap
```

Como puede verse, en nuestro caso hemos utilizado dos atributos:

- **-t nss**, con el que le indicamos que los archivos que vamos a modificar son los correspondientes a NSS
- **-p lac_ldap**, con el que indicamos que los datos para la configuración debe tomarlos del archivo `lac_ldap`. Este archivo se habrá generado durante la ejecución de `ldap-auth-config` en el punto anterior.

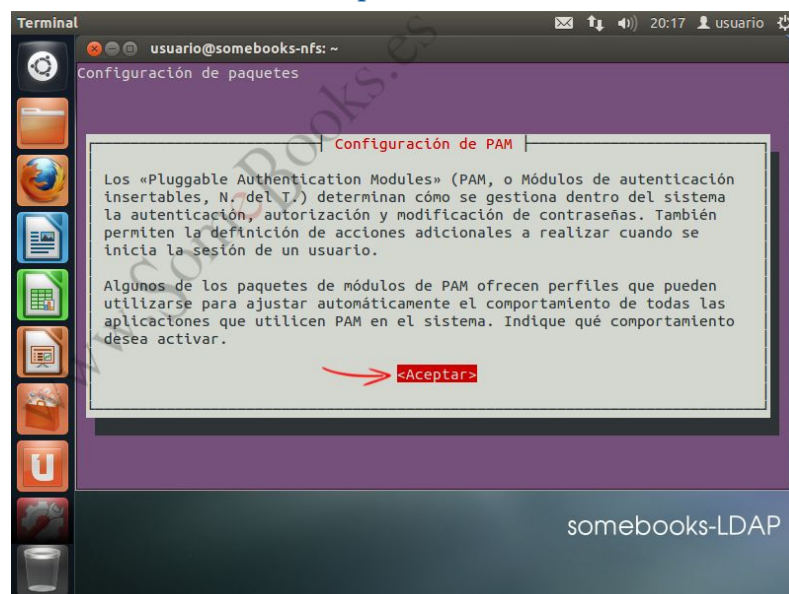
La ejecución de este comando no debe ofrecer ningún tipo de error. De lo contrario, deberíamos repasar la configuración ejecutando de nuevo `ldap-auth-config`.

A continuación, deberemos actualizar la configuración de las políticas de autenticación predeterminadas de PAM, lo que conseguimos ejecutando el siguiente comando:

```
sudo pam-auth-update
```

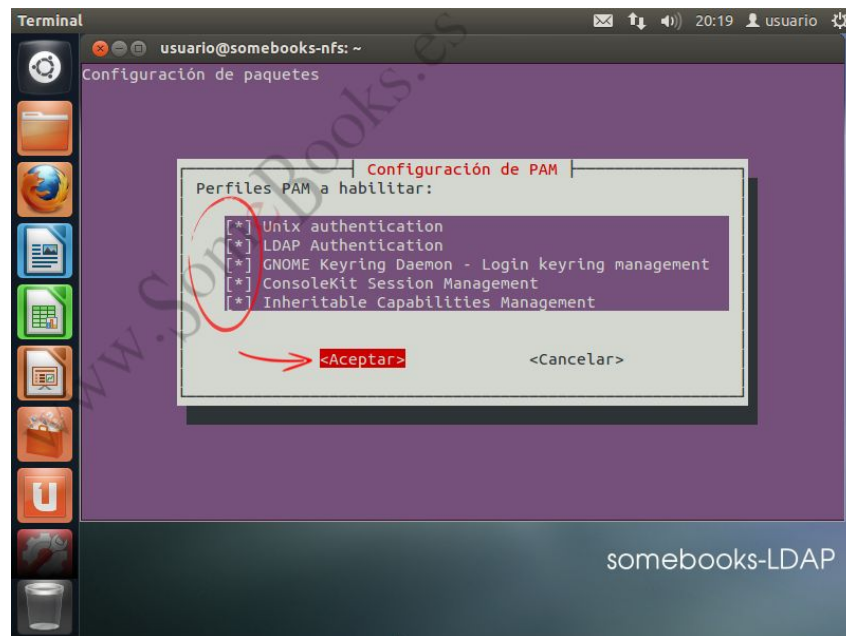
Al ejecutarlo, aparecerá un asistente, que nos muestra una primera pantalla informativa sobre la función de los módulos PAM.

Para continuar, pulsamos la tecla **Intro**.



En la siguiente pantalla, elegiremos cuáles de los módulos disponibles queremos habilitar. De forma predeterminada aparecen todos marcados...

... Por lo tanto, nos limitamos a volver a pulsar la tecla Intro para volver a la terminal



Una vez acabada la configuración automática, podremos hacer algunos cambios complementarios editando el archivo `/etc/ldap.conf`. Para lograrlo, recurriremos, como siempre, al editor nano:

```
sudo nano /etc/ldap.conf
```

Una vez que se abra el editor, podremos ajustar algunos de los valores del documento, pero, sobre todo, comprobaremos que son correctos los siguientes datos:

```
host 192.168.1.10 base
dc=somebooks,dc=local
uri ldapi://192.168.1.10/
rootbinddn cn=admin,dc=somebooks,dc=local
ldap_version 3
bind_policy soft
```

Sólo nos quedará pulsar **Ctrl + x** para salir y asegurarnos de guardar los cambios en el archivo.

Con esto habremos terminado la configuración del servidor LDAP. Ahora está listo para autenticar usuarios. Sin embargo, aún nos queda establecer el comportamiento del demonio (daemon) SLAPD.

3.4 Configurar el demonio (daemon) SLAPD

SLAPD (Standalone LDAP Daemon) es un programa multiplataforma, que se ejecuta en segundo plano, atendiendo las solicitudes de autenticación LDAP que se reciban en el servidor.

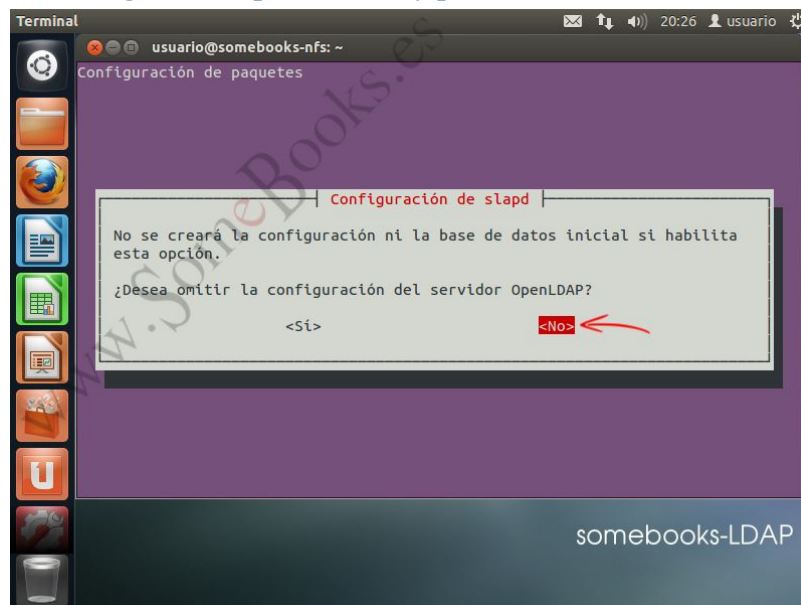
Como hemos dicho más arriba, el último paso en la configuración del servidor LDAP será establecer algunos parámetros en la configuración de este demonio. Para conseguirlo, ejecutaremos el siguiente comando, como es habitual, con privilegios de superusuario:

```
sudo dpkg-reconfigure slapd
```

Aparecerá un asistente cuyo cometido es evitar que tengamos que cambiar a mano el archivo `slapd.conf`.

La primera pantalla que se muestra, actúa como medida de seguridad, para asegurarse de que no hacemos cambios por error. Hay que tener cuidado porque la pregunta se hace al revés, es decir, nos pregunta si queremos omitir la configuración del servidor (imagino que el objetivo será impedir que elijamos Sí sin pensar lo que hacemos). En este caso, lógicamente, deberemos elegir la opción No.

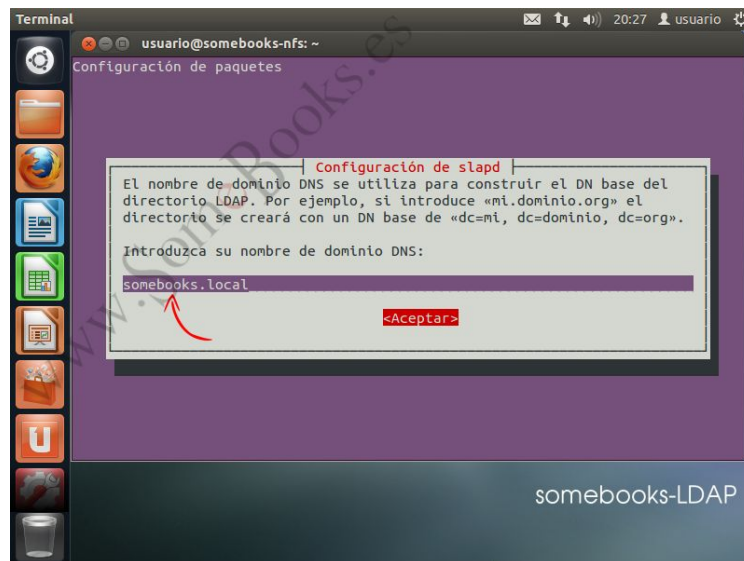
Elegimos la opción <No> y pulsamos la tecla *Intro*.



Recuerda que puedes usar la tecla <tabulador> para cambiar de opción.

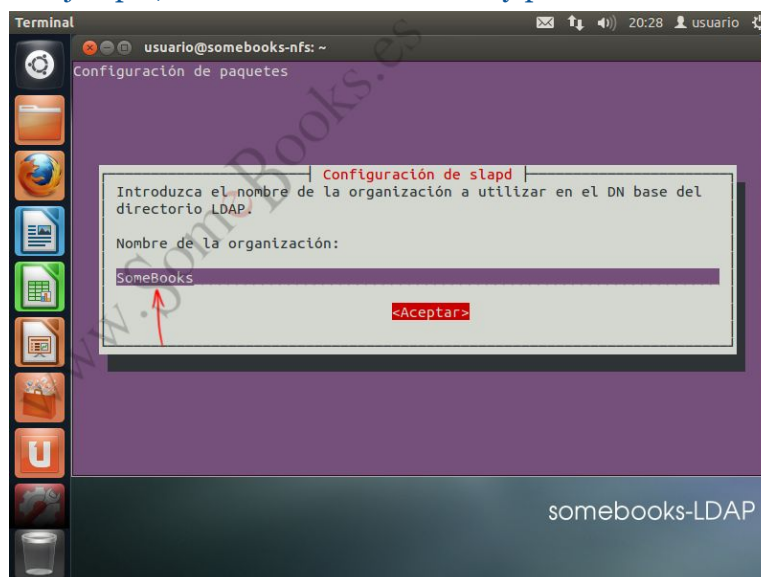
A continuación, deberemos escribir el nombre DNS que utilizamos para crear el DN base (Distinguished Name) del directorio LDAP.

En nuestro caso, escribiremos `somebooks.local` y pulsaremos la tecla Intro.



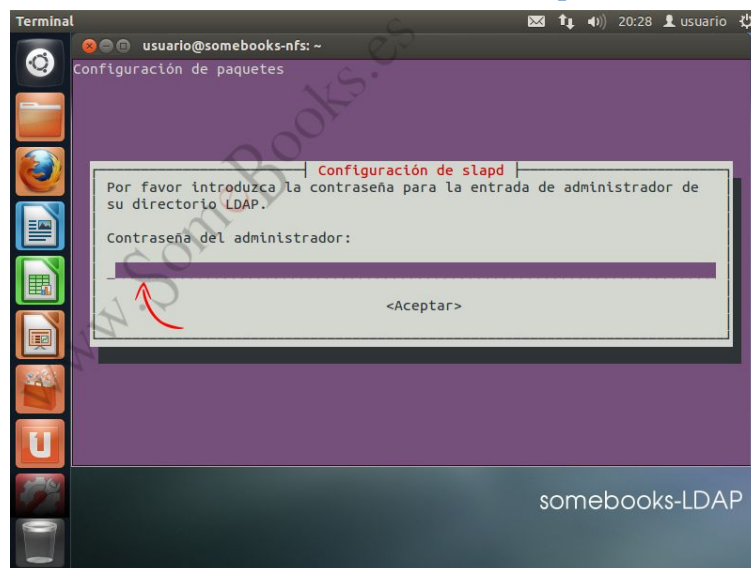
Después, escribiremos el nombre de la entidad en la que estamos instalando el directorio LDAP.

Para este ejemplo, escribiremos `SomeBooks` y pulsaremos la tecla Intro.



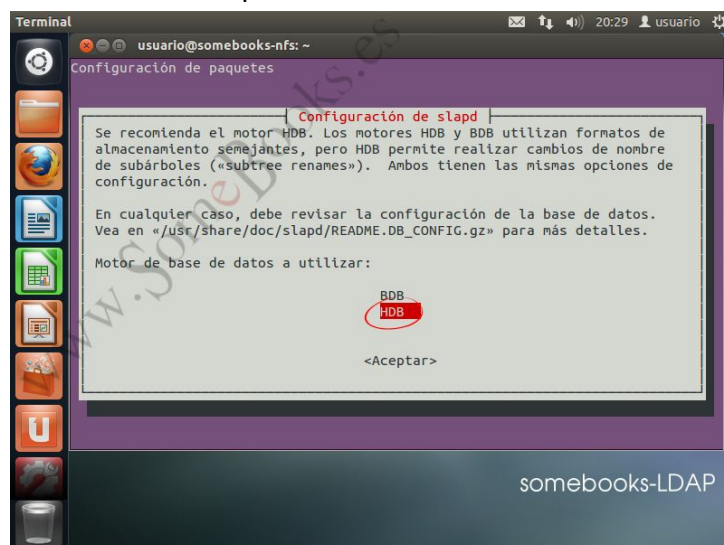
En el siguiente paso, deberemos escribir la contraseña de administración del directorio.

La contraseña debe coincidir con la que escribimos en el apartado [Instalar OpenLDAP en el servidor](#), como es habitual deberemos escribirla dos veces para evitar errores tipográficos.



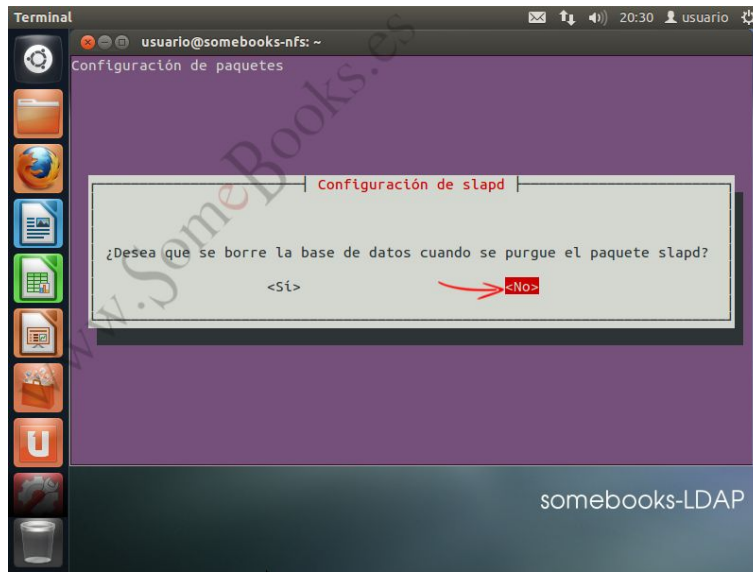
A continuación, elegiremos el motor de la base de datos que usaremos para el directorio. Se recomienda HDB porque nos permitirá, en el futuro, cambiar los nombres de los subárboles si fuese necesario.

Si HDB no aparece elegida de forma predeterminada, usaremos la tecla <tabulador> para desplazarnos. Cuando sea correcto pulsamos la tecla *Intro*:



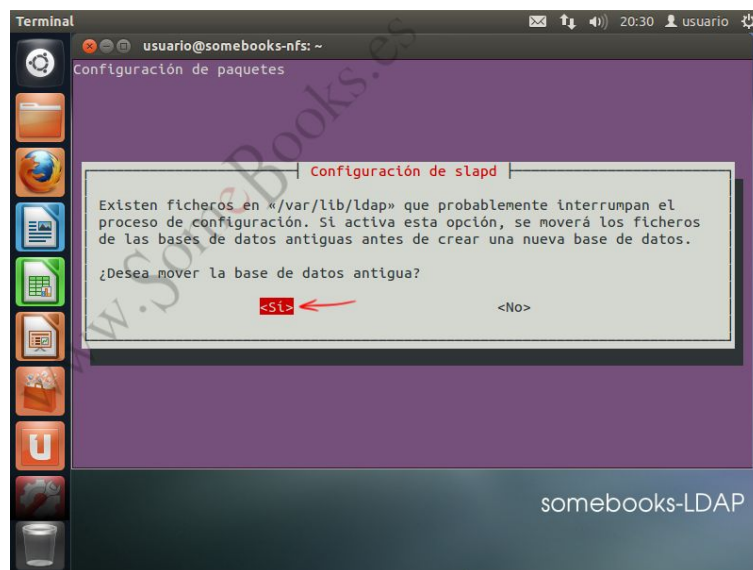
Lo siguiente que nos pregunta el asistente es si queremos que se borre la base de datos anterior del directorio cuando terminemos la configuración de slapd.

Igual que antes, usamos la tecla <tabulador> para elegir No y pulsamos Intro.

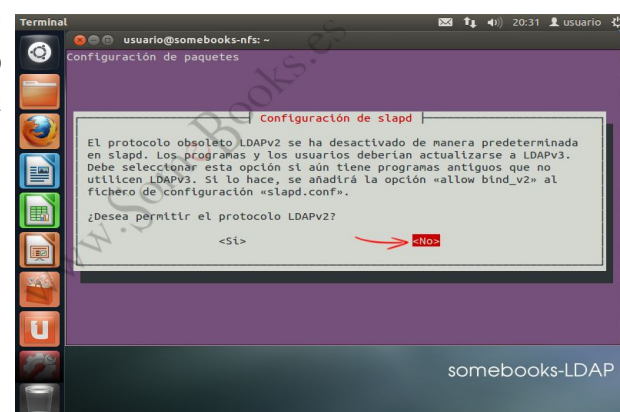


A continuación, como hemos decidido no borrar la base de datos antigua, el asistente nos pregunta si queremos cambiarla de sitio.

Para evitar confusiones entre las dos bases de datos (nueva y antigua), elegiremos la opción Sí y pulsaremos Intro.

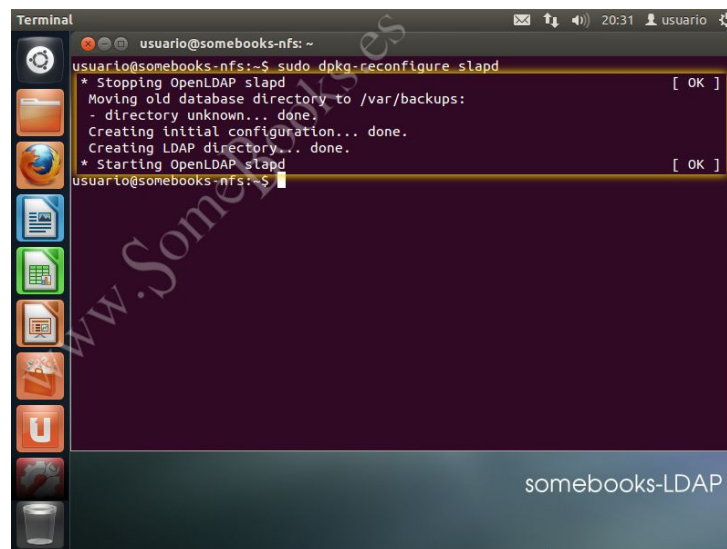


En algunas redes, con clientes muy antiguos, puede ser necesario mantener la versión 2 del protocolo LDAP. Por ese motivo, antes de terminar, el asistente nos pregunta queremos permitir el protocolo LDAPv2. En la mayoría de los casos, la respuesta será **No**.



Después de este último paso, se cierra el asistente y volvemos a la consola.

Ahora podemos ver en la pantalla que la base de datos antigua se ha guardado en /var/backups y que el resto de la configuración se ha realizado con éxito.

A terminal window titled 'Terminal' showing the execution of the command 'sudo dpkg-reconfigure slapd'. The output indicates that the OpenLDAP slapd service is being stopped, the old database directory is moved to /var/backups, the initial configuration is created, the LDAP directory is created, and the service is started successfully. The prompt 'usuario@somebooks-nfs:~\$' is visible at the bottom of the terminal. The desktop background is dark purple with a vertical dock on the left containing various application icons. A watermark 'www.SomeBooks.es' is visible across the terminal window.

```
usuario@somebooks-nfs:~$ sudo dpkg-reconfigure slapd
* Stopping OpenLDAP slapd [ OK ]
Moving old database directory to /var/backups:
- directory unknown... done.
Creating initial configuration... done.
Creating LDAP directory... done.
* Starting OpenLDAP slapd [ OK ]
usuario@somebooks-nfs:~$
```

Con esto habremos terminado la configuración del servidor LDAP. Ahora está listo para autenticar usuarios.

3.5 Crear la estructura del directorio (Ejemplo)

Una vez configurado el servidor, deberemos configurar la estructura básica del directorio. Es decir, crearemos la estructura jerárquica del árbol (DIT – Directory Information Tree).

Para este documento vamos a desarrollar una estructura de directorios de **EJEMPLO orientado a usuarios y grupos**.

Una de las formas más sencillas de añadir información al directorio es utilizar archivos LDIF (LDAP Data Interchange Format). En realidad, se trata de archivos en texto plano, pero con un formato particular que debemos conocer poder construirlos correctamente

El formato básico de una entrada es así:

```
# comentario
dn: <nombre global único>
<atributo>: <valor>
<atributo>: <valor>
...
```

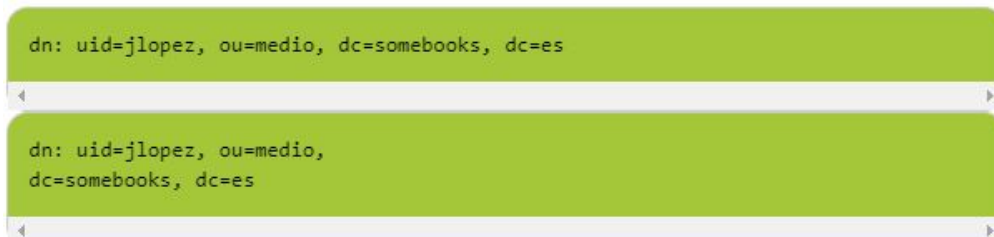
Las líneas que comienzan con un carácter # son comentarios.

<atributo> puede ser un tipo de atributo como cn o objectClass, o puede incluir opciones como cn;lang_en_US o userCertificate;binary.

Entre dos entradas consecutivas debe existir siempre una línea en blanco.

Si una línea es demasiado larga, podemos repartir su contenido entre varias, siempre que las líneas de continuación comiencen con un carácter de tabulación o un espacio en blanco.

Por ejemplo, las siguientes líneas son equivalentes:



```
dn: uid=jlopez, ou=medio, dc=somebooks, dc=es
```



```
dn: uid=jlopez, ou=medio,
    dc=somebooks, dc=es
```

Con esta información en mente, crearemos un archivo que contenga los tipos de objeto básicos del directorio. Primero creamos un archivo de texto con nano con extensión “.ldif”, por ejemplo llamado “base.ldif”.

Una vez abierto el editor, escribiremos un contenido como este:

```
dn: ou=usuarios,dc=somebooks,dc=local
objectClass: organizationalUnit
ou: usuarios

dn: ou=grupos,dc=somebooks,dc=local
objectClass: organizationalUnit
ou: grupos
```

Lógicamente, en cada lugar donde aparecen los valores `dc=somebooks,dc=local` deberemos sustituirlos por los valores correctos en cada implementación. Cuando hayamos terminado de escribirlo, sólo nos quedará pulsar **Ctrl + x** para salir y asegurarnos de guardar los cambios en el archivo.

A continuación, deberemos añadir la información a la base de datos OpenLDAP. Como sabemos, esto se hace con el comando `ldapadd`:

```
sudo ldapadd -x -D cn=admin,dc=somebooks,dc=local -W -f base.ldif
```

Para ejecutar el comando, deberemos escribir la contraseña de administración de LDAP. Después, podremos comprobar que los nuevos objetos se han añadido correctamente.

3.6 Añadir un usuario y un grupo

El método para añadir nuevos usuarios y grupos al árbol es muy similar a lo visto en el punto anterior, ya que consiste en crear un nuevo archivo `ldif` y, a continuación, integrarlo en la base de datos con `ldapadd`.

Añadir un usuario

Para ello creamos un archivo de texto con `nano` con extensión `.ldif`, con el nombre que queramos por ejemplo `usuario.ldif`.

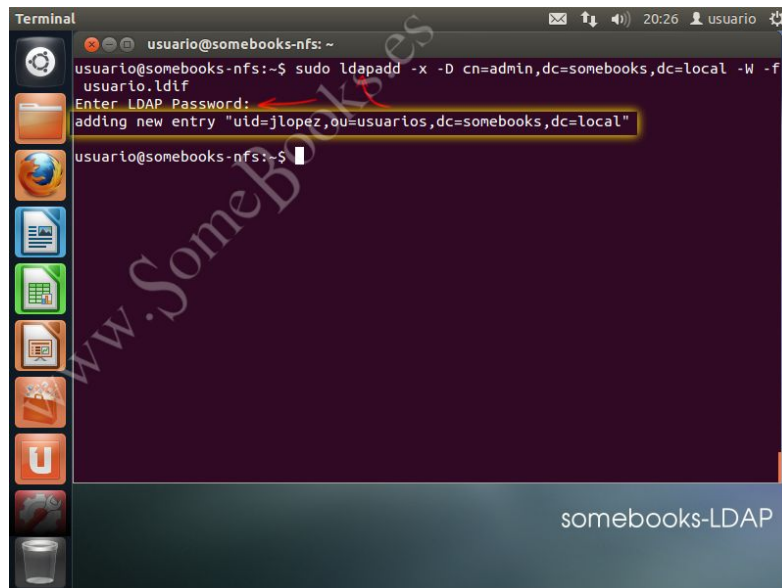
En el área de trabajo del editor, escribiremos un contenido como este, cuando hayamos terminado de escribirlo, sólo nos quedará pulsar **Ctrl + x** para salir y asegurarnos de guardar los cambios en el archivo:

```
dn: uid=jlopez,ou=usuarios,dc=somebooks,dc=local
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: shadowAccount
uid: jlopez
sn: Lopez
givenName: Juan
cn: Juan Lopez
displayName: Juan Lopez
uidNumber: 1000
gidNumber: 10000
userPassword: mi_password
gecos: Juan Lopez
loginShell: /bin/bash
homeDirectory: /home/jlopez
shadowExpire: -1
shadowFlag: 0
shadowWarning: 7
shadowMin: 8
shadowMax: 999999
shadowLastChange: 10877
mail: juan.lopez@somebooks.com
postalCode: 29000
o: somebooks
initials: JL
```

Con esto ya estamos listos para cargar el nuevo usuario en el directorio. Sólo tenemos que escribir el siguiente comando:

```
sudo ldapadd -x -D cn=admin,dc=somebooks,dc=local -W -f usuario.ldif
```

Después de escribir la contraseña de administración de LDAP, podremos comprobar que el usuario se ha añadido correctamente.



Añadir un grupo

Para añadir el grupo, repetimos de nuevo el proceso anterior, creamos un archivo de texto con nano con extensión “.ldif”, con el nombre que queramos por ejemplo “grupo.ldif”.

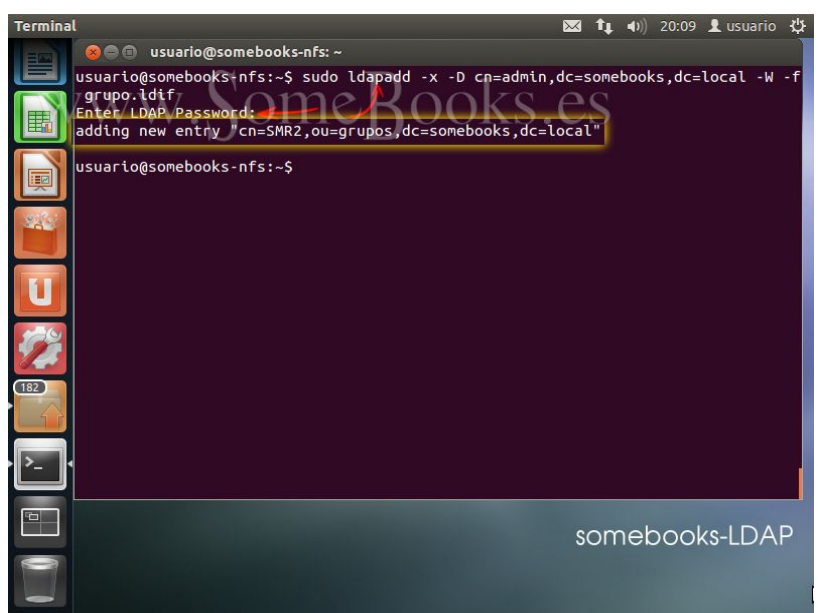
Una vez abierto el editor, escribiremos este contenido, cuando esté listo, pulsamos **Ctrl + x** para salir y nos aseguramos de guardar los cambios en el archivo:

```
dn: cn=SMR2,ou=grupos,dc=somebooks,dc=local
objectClass: posixGroup
cn: SMR2
gidNumber: 10000
```

De vuelta en la terminal, usamos de nuevo el comando ldapadd:

```
sudo ldapadd -x -D cn=admin,dc=somebooks,dc=local -W -f grupo.ldif
```

Después de escribir la contraseña de administración de LDAP, podremos comprobar que el grupo se ha añadido correctamente.



Con esto, ya tendremos en la base de datos un nuevo usuario y un nuevo grupo.

Nota Importante:

Cuando añadas nuevos usuarios, recuerda que los valores para los atributos **uidNumber** y **homeDirectory** deben ser diferentes para cada usuario. También habrá que sustituir el texto **mi_password** por la contraseña adecuada para el usuario.

Lo mismo ocurre con el atributo **gidNumber** de los grupos.

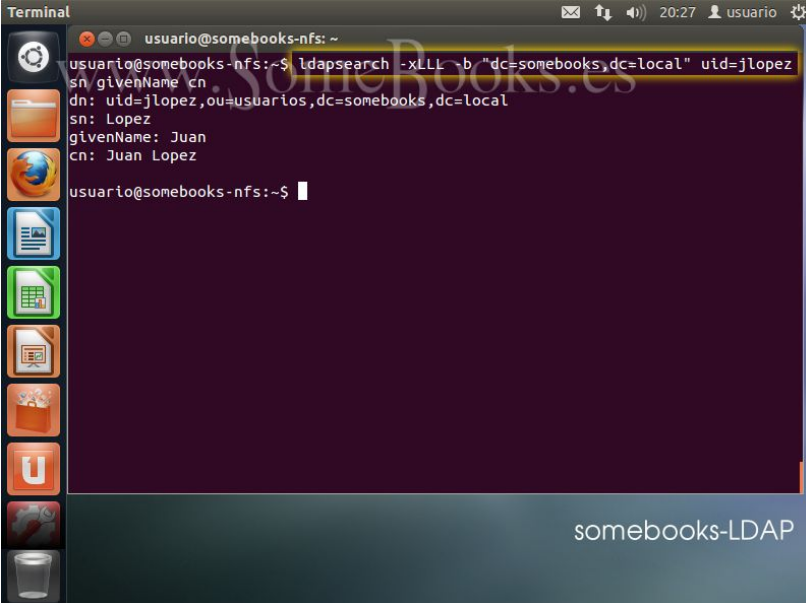
Además, los valores de los campos **uidNumber** y **gidNumber** no deben coincidir con el UID y GID de ningún usuario y grupo local.

3.7 Comprobar que todo es correcto

Ahora podemos comprobar que el contenido anterior se ha añadido correctamente. Para lograrlo podemos utilizar, por ejemplo, el comando **ldapsearch**, que nos permite hacer una búsqueda en el directorio:

```
ldapsearch -xLLL -b "dc=somebooks,dc=local" uid=jlopez sn givenName cn
```

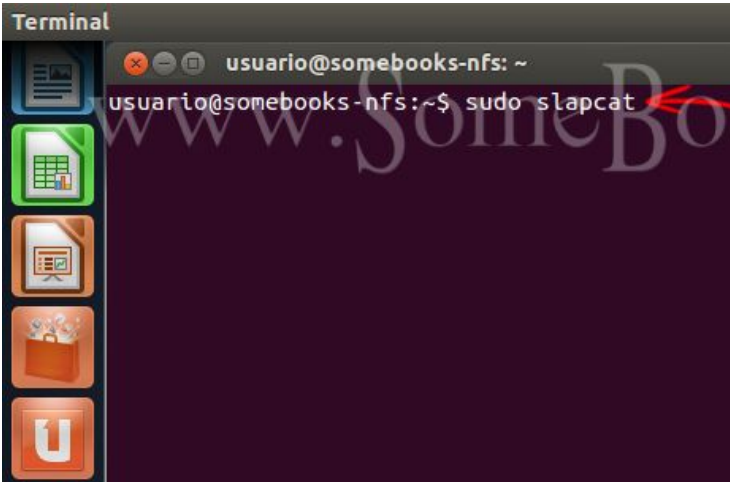
En este ejemplo buscamos un usuario con uid=jlopez y pedimos que nos muestre el contenido de los atributos sn, givenName y cn. Y este será el resultado de la consulta:



```
usuario@somebooks-nfs: ~  
usuario@somebooks-nfs:~$ ldapsearch -xLLL -b "dc=somebooks,dc=local" uid=jlopez  
sn givenName cn  
dn: uid=jlopez,ou=usuarios,dc=somebooks,dc=local  
sn: Lopez  
givenName: Juan  
cn: Juan Lopez  
usuario@somebooks-nfs:~$
```

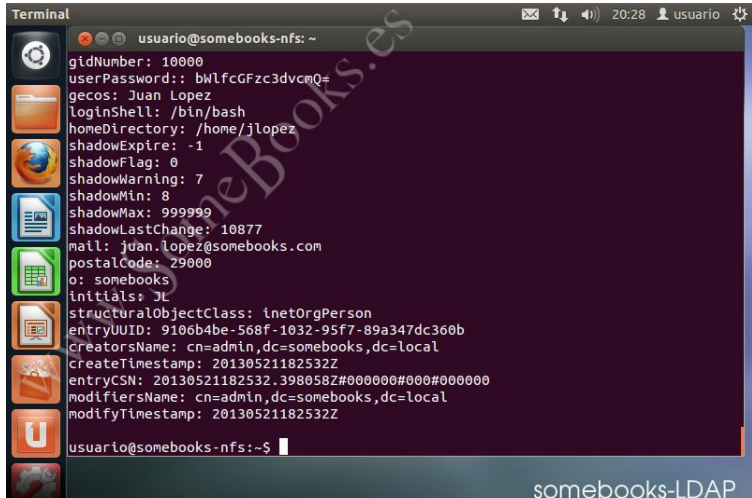
Otra opción interesante para comprobar el contenido del directorio es utilizar el comando **slapcat**. Su cometido es mostrar el contenido completo del directorio LDAP. Además, esta información se obtiene en formato LDIF, lo que nos permitirá volcarla a un fichero y exportar la base de datos de un modo muy sencillo. En nuestro caso, nos limitaremos a obtener la salida en la pantalla:

1. Ejecutamos el comando slapcat con privilegios de superusuario.



```
usuario@somebooks-nfs: ~  
usuario@somebooks-nfs:~$ sudo slapcat
```

2. Puedes utilizar la barra de desplazamiento para ver el contenido completo del archivo.



```
usuario@somebooks-nfs: ~  
gidNumber: 10000  
userPassword:: bWlfcGFzc3dvcmQ=  
gecos: Juan Lopez  
loginShell: /bin/bash  
homeDirectory: /home/jlopez  
shadowExpire: -1  
shadowFlag: 0  
shadowWarning: 7  
shadowMin: 8  
shadowMax: 999999  
shadowLastChange: 10877  
mail: juan.lopez@somebooks.com  
postalCode: 29000  
o: somebooks  
initials: JL  
structuralObjectClass: inetOrgPerson  
entryUUID: 9106b4be-568f-1032-95f7-89a347dc360b  
CreatorsName: cn=admin,dc=somebooks,dc=local  
CreateTimestamp: 20130521182532Z  
entryCSN: 20130521182532.398058Z#000000#000#000000  
modifiersName: cn=admin,dc=somebooks,dc=local  
modifyTimestamp: 20130521182532Z  
usuario@somebooks-nfs:~$
```

4. Ejemplos de aplicación que se integran con LDAP y como lo hacen.

Las siguientes páginas son un listado de programas de software que pueden comunicarse y/o hospedar servicios de directorio a través de LDAP como lo puede hacer Active Directory.

Software Cliente

Cross-platform

- Admin4^[1] - an open source LDAP browser and directory client for [Linux](#), [OS X](#), and [Microsoft Windows](#), implemented in [Python](#).
- [Apache Directory Server/Studio](#) - an LDAP browser and directory client for [Linux](#), [OS X](#), and [Microsoft Windows](#), and as a plug-in for the [Eclipse](#) development environment.
- [FusionDirectory](#),^[2] a web application under license [GNU General Public License](#) developed in [PHP](#) for managing LDAP directory and associated services.
- [JXplorer](#) - a [Java](#)-based browser that runs in any operating environment.
- [JXWorkBench](#) ^[3] - a [Java](#)-based plugin to [JXplorer](#) that includes LDAP reporting using the [JasperReports](#) reporting engine.
- [LDAP Account Manager](#) - a [PHP](#) based webfrontend for managing various account types in an LDAP directory.
- [phpLDAPadmin](#) - a web-based LDAP administration tool for creating and editing LDAP entries in any LDAP server.
- [LDAP User Manager](#) - A simple [PHP](#) interface to add LDAP users and groups. Also has a self-service password change feature. Designed to be run as a [Docker](#) container.
- [SLAMD](#) - an open source load generation software suite, for testing multiple application protocols, including LDAP. Also contains tools for creating test data and test scripts.^[†]
- [RoundCube](#) - an open source and free [PHP](#) [IMAP](#) client with support with LDAP based address books.
- [GOsa²](#) - provides a powerful framework for managing accounts and systems in LDAP databases^[citation needed]
- [web2ldap](#),^[4] a web application under license [GNU General Public License v2](#) developed in [Python](#) for managing LDAP directories.
- [OpenDJ](#) - a [Java](#)-based LDAP server and directory client that runs in any operating environment, under license [CDDL](#)

Linux/UNIX

- [389 Directory Server](#) - A free server implementation by [Red Hat](#) (previously named as [Fedora Directory Server](#))
- [Evolution](#) - the contacts part of [GNOME](#)'s PIM can query LDAP servers.
- [KAddressBook](#) - the address book application for [KDE](#), capable of querying LDAP servers.
- [OpenLDAP](#) - a free, open source implementation.
- [OpenDJ](#) - a free, open source implementation.

Mac OS X

- [Contacts](#) - an LDAP-aware [address book](#) application built into [Mac OS X](#).
- [Directory Utility](#) - a utility for configuring access to several types of [directory servers](#), including LDAP; built into [Mac OS X](#).
- [Workgroup Manager](#) - a utility for configuring access to several types of [directory servers](#), including LDAP; built into [Mac OS X Server](#) and one of [Apple's Server Admin Tools](#).
- [OpenDJ](#) - a free, open source implementation.
- [Slapd](#) - from the Univ of Michigan

Microsoft Windows

- [Active Directory Explorer](#) - a [freeware](#) LDAP client tool from [Microsoft](#)^[5]
- [LDAP Admin](#) - a free, open source LDAP directory browser and editor
- [OpenDJ](#) - a free, open source implementation.

Software Servidor:

	Developer	Software license ^[a]	Comments
389 Directory Server (formerly Fedora Directory Server)	Red Hat	GPL linking exception ^[6] with exception to allow linking to non-GPL ^[7]	
Active Directory	Microsoft	Proprietary	
Authorized Entities Directory (<i>Æ-DIR</i>) [↗]	Michael Ströder	Apache License 2.0	based on OpenLDAP with additional tools
Apache Directory Server	Apache Software Foundation	Apache License 2.0	
Apple Open Directory - A fork of the OpenLDAP project	Apple Inc.	Proprietary ^[8]	
BEJY LDAP Server , ^[9] a Java LDAP Server .	Stefan "Bebbo" Franke	GPL	
CA Directory	CA Technologies	Proprietary	
Critical Path Directory Server	Critical Path	Proprietary	Now owned by Synchronoss Technologies.
Directory services - A fork of the OpenDJ project	ForgeRock	Proprietary	
DirX Directory	Atos (ex-Siemens)	^[citation needed]	
FreeIPA	Red Hat (using 389 Directory Server)	GPL	
IBM Tivoli Directory Server	IBM	Proprietary	
Idapjs , ^[10] implementation of LDAP in JavaScript on Node.js.	Mark Cavage ^[11]	MIT License	
Mandriva Directory Server , now part of Mandriva Management Console	Mandriva development team	GPL	
Nexor Directory	^[citation needed]	^[citation needed]	
NetIQ eDirectory	NetIQ	Proprietary	Successor to eDirectory and NDS
OpenBSD Idapd ^[12]	Martin Hedenfalk, OpenBSD	ISC	
OpenDJ	Open Identity Platform Community	CDDL	A fork of the OpenDS project developed by ForgeRock, until 2016 ^[13] , now maintained by OpenDJ [↗] Community
OpenDS	Sun Microsystems	CDDL	CDDL-licensed product no longer maintained, now Oracle Unified Directory.
OpenLDAP	Kurt Zeilenga and others (based on Slapd)	OpenLDAP Public License [↗]	
Oracle Directory Server Enterprise Edition ^[14]	Oracle, based on Sun DSEE	Proprietary	
Oracle Internet Directory	Oracle	Proprietary	
Oracle Unified Directory	Oracle, based on OpenDS	Proprietary	
PingDirectory (formerly UnboundID Directory Server ^[15])	Ping Identity	Proprietary	Based on OpenDS. UnboundID was purchased by Ping
RadiantOne	Radiant Logic	^[citation needed]	
Red Hat Directory Server	Red Hat	GPL plus exception	Commercial version of 389 Directory Server
ReOpenLDAP ^[16]	Peter-Service R&D	AGPL and OpenLDAP Public License [↗]	fork of OpenLDAP with improved stability for highload and multi-master clustering
Samba4 - Active Directory compatible Domain Controller	Samba Team	GPLv3	
Slapd - Standalone LDAP Daemon	University of Michigan	Free ^[citation needed]	superseded by OpenLDAP ^[17]
Sun Java System Directory Server	Sun Microsystems	^[citation needed]	no longer maintained

5. ¿Cómo representar un organigrama en LDAP? Ejemplos de organigramas en LDAP.

LDAP permite crear una estructura de directorios mediante línea de comandos, sin embargo la forma más cómoda de hacerlo es mediante ficheros **LDIF**.

Ficheros LDIF

Son ficheros estándar con formato de intercambio de datos sencillo que representan contenido de directorios LDAP (protocolo ligero de acceso a directorios) y solicitudes de actualización.

Mediante la confección de estos ficheros podemos definir una estructura e incluso crear los objetos de esa estructura.

Para añadir algún objeto al directorio:

Para ello redactamos la información del objeto (por ejemplo un usuario o grupo) en un archivo en formato **.ldif**, luego escribimos el comando:

```
sudo ldapadd -x -D cn=admin,dc=ceviche-test,dc=local -W -f grupo.ldif
```

Lo marcado en negrita:

1. **DN** del que va a realizar esta acción en este caso admin.
2. **Fichero** en formato .ldif que hemos redactado previamente.

Ejemplo de fichero LDIF

Este fichero podría importarse dentro de LDAP.

The diagram illustrates a directory structure defined in an LDIF file. It consists of four entries, each with a callout explaining its purpose:

- Entry 1:** `dn: dc=ceviche-test,dc=local`. This entry represents the root directory. Callout: "Directorio raiz donde está todo lo que se creará a posteriori".
- Entry 2:** `dn: cn=admin,dc=ceviche-test,dc=local`. This entry defines the administrator. Callout: "Definición del admin".
- Entry 3:** `dn: ou=usuarios,dc=ceviche-test,dc=local`. This entry represents a group named "usuarios". Callout: "Grupo 'usuarios'".
- Entry 4:** `dn: uid=srebollo,ou=usuarios,dc=ceviche-test,dc=local`. This entry represents a user named "srebollo" within the "usuarios" group. Callout: "Usuario 'srebollo' dentro del grupo 'usuarios'".

The LDIF entries are as follows:

```
dn: dc=ceviche-test,dc=local
objectClass: top
objectClass: dcObject
objectClass: organization
o: Ceviche
dc: ceviche-test
structuralObjectClass: organization
entryUUID: 692af656-9312-1038-8b51-8f49014798ac
creatorsName: cn=admin,dc=ceviche-test,dc=local
createTimestamp: 20181213110305Z
entryCSN: 20181213110305.391436Z#000000#000#000000
modifiersName: cn=admin,dc=ceviche-test,dc=local
modifyTimestamp: 20181213110305Z

dn: cn=admin,dc=ceviche-test,dc=local
objectClass: simpleSecurityObject
objectClass: organizationalRole
cn: admin
description: LDAP administrator
userPassword:: e1NTSEF9OUphS3lkVlYxV2kzZSs0NnRDQ01qZzJKc0E4YWhpUzI=
structuralObjectClass: organizationalRole
entryUUID: 692bd882-9312-1038-8b52-8f49014798ac
creatorsName: cn=admin,dc=ceviche-test,dc=local
createTimestamp: 20181213110305Z
entryCSN: 20181213110305.397225Z#000000#000#000000
modifiersName: cn=admin,dc=ceviche-test,dc=local
modifyTimestamp: 20181213110305Z

dn: ou=usuarios,dc=ceviche-test,dc=local
objectClass: organizationalUnit
ou: usuarios
structuralObjectClass: organizationalUnit
entryUUID: dif7ed1e-9313-1038-92da-190c3232bcc7
creatorsName: cn=admin,dc=ceviche-test,dc=local
createTimestamp: 20181213111310Z
entryCSN: 20181213111310.714453Z#000000#000#000000
modifiersName: cn=admin,dc=ceviche-test,dc=local
modifyTimestamp: 20181213111310Z

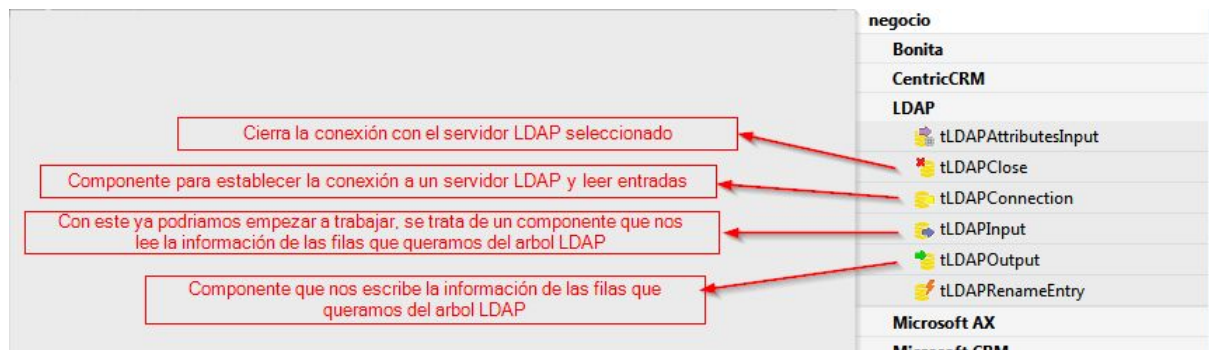
dn: uid=srebollo,ou=usuarios,dc=ceviche-test,dc=local
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: shadowAccount
uid: srebollo
sn: Rebollo
givenName: Salvador
```

Si nos instalamos una interfaz web podremos administrar la estructura de forma visual, para ello tenemos **phpLDAPadmin**, el cuál podemos instalar siguiendo esta guía que lo explica bastante bien:

<http://somebooks.es/12-8-usar-una-interfaz-web-para-gestionar-usuarios-y-grupos-en-el-servidor-openldap/>

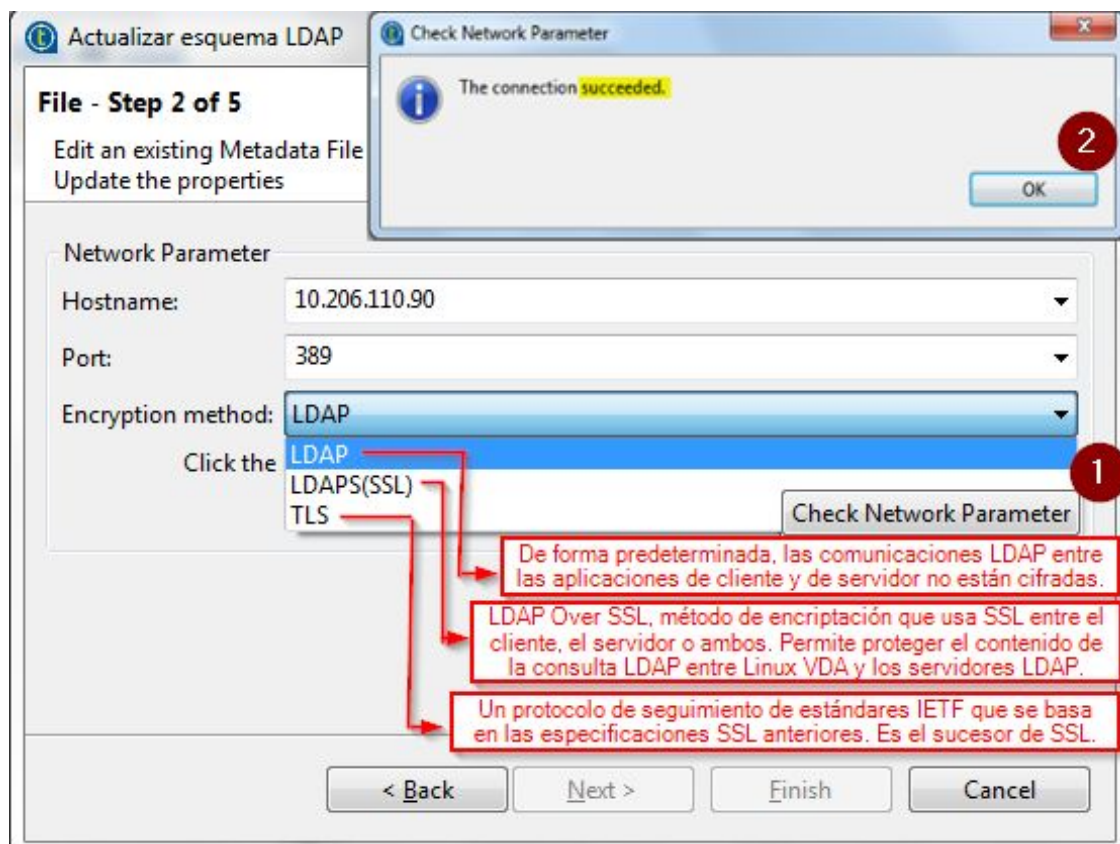
6. Componentes y funcionalidades que ofrece Talend para obtener datos de LDAP/Active Directory (ampliar con Exchange de Talend)

Talend 7.0.1 en su paleta viene de forma nativa con componentes LDAP preinstalados, los podemos encontrar en la sección **negocio** de la paleta, aquí vemos una breve descripción de ellos:



Configurando METADATA LDAP en Talend

En Talend podemos **preconfigurar** una conexión con cualquier servidor para luego no tener que volver a configurarla a posteriori con el resto de componentes que funcionen con el mismo servidor, esta es una de las funciones que brinda el **metadata**, aquí muestro algunas capturas en la que enseño como realizar la conexión con el servidor LDAP que he creado en Ubuntu:



Actualizar esquema LDAP

File - Step 3 of 5
Edit an existing Metadata File on repository
Update the properties

Método de autenticación
Simple Authentication

Authentication Parameter
Bind DN or user: cn=admin,dc=ceviche-test,dc=local
Bind password: *****
☒ Guardar contraseña

Base DN
☐ Get base DN's from Root DSE
Base DN: dc=ceviche-test,dc=local

Aliases Dereferencing
☐ Finding
☐ Searching
☐ Never
☒ Always

Referrals Handling
☒ Ignore
☐ Follow

Limits
Count Limit: 100
Time Limit: 0

Buttons:
1 Checa Autenticación
2 Fetch Base DN's
3 Finish
< Back Next > Cancel

Exportar como contexto **Revertir Contexto**

Annotations:
Elegir entre login anónimo o autenticación simple
DN de usuario con permisos de Administración y su password
Directorio raíz (root) del que partir para la conexión y su acceso
Numero de solicitudes/intentos en la conexión
Puede configurar una entrada de hoja para que apunte a otro objeto en el espacio de nombres. Llamada entrada de alias, contiene el DN del objeto al que apunta. Cuando busca un objeto utilizando el alias, el alias es desreferenciado, de modo que lo que se devuelve es el objeto señalado por el DN del alias. Esta opción te permite elegir cuando hacer esa referencia alias/DN.
Las referencias permiten que un árbol de directorios se particione y distribuya entre múltiples servidores LDAP, lo que significa que los servidores LDAP pueden no almacenar el DIT completo mientras aún son capaces de contener referencias a otros servidores LDAP que ofrecen información solicitada. Con esta opción elegimos como manipular estas conexiones.

Haciendo un JOB de prueba

Vamos a simular un escenario sencillo para verificar que realmente estamos extrayendo los datos del servidor LDAP, para ello vamos a poner sobre el espacio de trabajo un componente **tLDAPInput** conectado a un **tLogRow** para que este último haga de salida y nos muestre por pantalla algunas filas del servidor:

Configuramos el tLDAPInput usando el metadatos que hemos creado para hacer la conexión

Campos de los cuadrado marcados similares a los del metadatos anterior: host, puerto, usuario administrador, directorio raíz...

Para filtrar la consulta por alguna denominación concreta

Separador de campo

Paginación

Finaliza el proceso si llega un error al job


Ejecutamos el job y vemos el resultado:

Starting job Prueba1 at 09:06 19/12/2018.
[statistics] connecting to socket on port 3770
[statistics] connected

cn	dc	description	displayName	gecos
null	cevice-test	null	null	null
admin	null	LDAP administrator	null	null
null	null	null	null	null
Salvador Rebollo	null	null	Salvador Rebollo	Salvador Rebollo
grupoUno	null	null	null	null
grupoNueve	null	null	null	null
Maria Padilla	null	null	Maria Padilla	Maria Padilla
ppruebas	null	null	null	null
grupoTres	null	Grupo número tres para pruebas	null	null

Componentes de Talend en Exchange

Procedemos a realizar una **búsqueda sobre LDAP** en el repositorio de componentes Talend **Exchange**, y apreciamos algunos componentes que pueden ser útiles para la administración LDAP como puede ser este por ejemplo:



bcLDAPInputExt
bcourtine

Extension of Talend tLDAPInput (3.2.3). This extension add the following features : - in LDAPS and TLS mode, a password can be provided to open the certificate file - in LDAPS and TLS mode, certificate







v1.0 ☆☆☆☆☆ (0) ↓ 343

Free

Este un componente que hace de “**extensión**” del componente nativo tLDAPInput (en su versión 3.2.3). Esta extensión agrega las siguientes características:

- En modo LDAPS y TLS, se puede proporcionar una contraseña para abrir el archivo de certificado.
- En los modos LDAPS y TLS, la validación del certificado se puede desactivar (se aceptan todos los certificados).

Resumen de otros componentes que podemos encontrar en Talend Exchange

	tLDAPHelper yassine12you function Helper to extract id from dn	v1.0 ☆☆☆☆☆ (0) ↓ 62	Free	Función helper para extraer id de dn
	tLDAPPreparAuthAD sborion Transform Data for a integration in a Microsoft Active Directory. You can : - Encode password for a user you create or modify (you must use the component tLDAPAdOutput which can use byte[]) -	v1 ☆☆☆☆☆ (0) ↓ 67	Free	Transforma datos para una integración en un Active Directory. Usted puede : - Codificar la contraseña para un usuario que cree o modifique (debe usar el componente tLDAPAdOutput que puede usar el byte []) - Fechas de transformación - Cacul y agregar userAccountControl
	tLDAPAdOutput sborion It's just une little modification of standard tLDAPOutput Component for create/modify password in a Active Directory. The difference is you can use a column of type byte[]. I have create another	v1 ☆☆☆☆☆ (0) ↓ 53	Free	Es solo una pequeña modificación del componente tLDAPOutput Component para crear/modificar la contraseña en un Active Directory. La diferencia es que puede usar una columna de tipo byte [].
	tLDAPInputScopeLevel mat2121 This component allows to set the scope level parameter for a LDAP search. (One-level, subtree or object scope)	v0.1 ☆☆☆☆☆ (0) ↓ 99	Free	Este componente permite establecer el parámetro de nivel de alcance para una búsqueda LDAP. (Alcance de un nivel, subárbol u objeto)
	bcLDAPAttributesInput bcourtine This component is an extension of bcLDAPInputExt : As the bcLDAPInputExt component, this component will do a research in a LDAP server. But in addition to attribute values (defined in the	v1.1 ☆☆☆☆☆ (0) ↓ 174	Free	Además de los valores de atributo (definidos en el esquema del componente), para cada objeto LDAP, brinda información de tesis como: objectclass, mandatoryattributes, optionalattributes, objectattributes
	bcLDAPOutputExt bcourtine Extension of Talend tLDAPOutput (3.2.3). This extension add the following features : - in TLS mode, a password can be provided to open the certificate file - in LDAPS and TLS mode, certificate validation	v1.0 ☆☆☆☆☆ (0) ↓ 266	Free	Esta extensión agrega las siguientes características: - En modo TLS, se puede proporcionar una contraseña para abrir el archivo de certificado. - En los modos LDAPS y TLS, la validación del certificado se puede desactivar (se aceptan todos los certificados).

7. Librerías de Java para integrar con LDAP

- **UnboundID LDAP SDK for Java**

El UnboundID LDAP SDK para Java es una librería de Java rápida, potente, fácil de usar y completamente gratuita para comunicarse con los servidores de directorio LDAP. Ofrece un mejor rendimiento, una mayor facilidad de uso y más funciones que otras API LDAP basadas en Java. Se está desarrollando y mejorando activamente, y hay soporte comercial disponible.

<https://ldap.com/unboundid-ldap-sdk-for-java/>

- **Ldaptive**

Ldaptive es una API Java simple y extensible para interactuar con servidores LDAP. Fue diseñado para proporcionar una integración LDAP fácil para los desarrolladores de aplicaciones.

<http://www.ldaptive.org/>

- **JLDAP**

Las bibliotecas de clases LDAP para Java (JLDAP) le permiten escribir aplicaciones para acceder, administrar, actualizar y buscar información almacenada en directorios accesibles mediante LDAPv3. JLDAP fue desarrollado por Novell. Las últimas fuentes están disponibles en el repositorio de CVS de OpenLDAP.

<http://www.openldap.org/jldap/>

8. Preguntas y Respuestas

- **Cada nivel será un objeto distinto, por lo que para el cliente los objetos de la arquitectura y los niveles sería diferente ¿Es así?**

Cada objeto del árbol tendrá un identificador único al que referirse llamado DN, para el cliente los objetos de la arquitectura son accesibles según sus privilegios y los niveles son también estructuras de objeto las cuales para el cliente pueden ser accesibles o no.

- **Puedo sacar todos los DN para cada uno de los objetos teléfono o para todos los objetos que tengan un atributo teléfono.**

A los servidores LDAP es posible realizarle consultas con esas características. Valiéndonos del comando **ldapsearch** o mediante Java usando alguna librería óptima para tratar con LDAP o alguna otra aplicación externa que trabaje con este protocolo.

- **¿Existen ejemplo online de servidores LDAP que se pueden usar para hacer pruebas?**

Existen, buscando por internet he encontrado uno cuyo objetivo es brindarnos las credenciales para un servidor de prueba LDAP en línea que es posible usar para probar aplicaciones que requieren autenticación basada en LDAP. Su objetivo es eliminar la necesidad de descargar, instalar y configurar un servidor LDAP para realizar pruebas. Si todo lo que se necesita es probar la conectividad y la autenticación con algunas identidades, es una buena opción.

LDAP Server Informacion (read-only access):

Server: ldap.forumsys.com

Puerto: 389

Bind DN: cn=read-only-admin,dc=example,dc=com

Bind Password: password

Todos los usuarios tienen su nombre de contraseña

También puede enlazar con Usuarios individuales (uid) o los dos Grupos (ou) que incluyen:

ou=mathematicians,dc=example,dc=com

- riemann
- gauss
- euler
- euclid

ou=scientists,dc=example,dc=com

- einstein
- newton
- galieleo
- tesla