



# **Curso de LDAP en GNU/Linux (60 horas)**



## **Teoría, Guía de prácticas y ejercicios**






# Creative Commons

## Reconocimiento-No comercial-Compartir bajo la misma licencia 3.0

### Usted es libre de:

-  copiar, distribuir y reproducir públicamente la obra
-  hacer obras derivadas

### Bajo las siguientes condiciones:

-  **Reconocimiento.** Debe reconocer los créditos de la obra de la manera especificada por el autor o el licenciante (pero no de una manera que sugiera que tiene su apoyo o apoyan el uso que hace de su obra).
-  **No comercial.** No puede utilizar esta obra para fines comerciales.
-  **Compartir bajo la misma licencia.** Si altera o transforma esta obra, o genera una obra derivada, sólo puede distribuir la obra generada bajo una licencia idéntica a ésta.
- Al reutilizar o distribuir la obra, tiene que dejar bien claro los términos de la licencia de esta obra.
- alguna de estas condiciones puede no aplicarse si se obtiene el permiso del titular de los derechos de autor
- Nada en esta licencia menoscaba o restringe los derechos morales del autor.

### Los derechos derivados de usos legítimos u otras limitaciones reconocidas por ley no se ven afectados por lo anterior.

Esto es un resumen fácilmente legible del texto legal de versión original en  
Idioma Inglés (la licencia completa)  
<http://creativecommons.org/licenses/by-nc-sa/3.0/ec/legalcode>



# Índice de contenido

- Objetivos del curso.....5
- Requisitos.....5
- Introducción.....6
- ¿Que es LDAP?.....7
- Razones para utilizar LDAP.....7
- ¿Qué es un servicio de directorio?.....8
- ¿Qué tipo de información se puede almacenar en un directorio?.....10
- ¿Cómo se almacena la información en LDAP?.....10
- ¿Cómo se referencia la información en LDAP?.....12
- ¿Cómo se accede a la información en LDAP?.....12
- ¿Cómo lo accede?.....13
- ¿Cómo protege la información de accesos no autorizados?.....13
- ¿Cómo trabaja LDAP?.....13
- X.500.....14
- Diferencias entre LDAP v2 y v3.....15
- ¿Qué es SLAPD?.....16
  - La versión 3 de LDAP.....16
- SASL.....19
- SLDAP (Servidor LDAP) en debian.....20
  - Pasos para la instalación de SLAPD.....20
  - Reconfiguración del paquete slapd.....20
  - Pruebas de la instalación.....25
- Administración de usuarios.....27
  - Creación de los archivos ldif.....28
  - Agregar los contenidos de los ldif.....28
  - Agregar usuarios.....31
  - Eliminar un usuario.....33
  - Agregar atributos.....35
  - Modificar un atributo.....37
  - Para eliminar un atributo.....39
  - Manejo de contraseña.....40
- Administración de grupos.....42
- Autenticación de Clientes en ldap.....43
  - Listas de acceso en LDAP.....44
  - Accesos utilizando DN.....48





Conexiones Seguras.....	49
Múltiples Directorios Réplicas y Cache.....	52
Configuración de servidores esclavos LDAP .....	54
ProxyLdap.....	55
Configuración de un servidor ProxyLdap.....	55



## Objetivos del curso

- Instalar, configurar y administrar de los servicios basados en LDAP.
- Aprender las estrategias para la puesta en operación del servicio LDAP en el proceso de migración a software libre de las instituciones del estado.

## Requisitos

- Requiere conocimientos básicos de soporte en GNU/Linux.
- Conocimiento en la configuración de los servicios de red y la seguridad a un nivel básico.
- Auto-motivación a la investigación y estudio en el campo del software libre GNU/Linux.



## **Introducción**

Esta es una guía práctica para la implementación de LDAP. Comprende desde la instalación hasta la puesta a punto del servicio. Como por ejemplo, la configuración del servicio de modo seguro con autenticación de los clientes ante el servidor, como se accede a los datos, la administración, tareas como añadir o eliminar usuarios y grupos, agregar, eliminar y cambiar atributos, entre otros. Se espera que sea de gran utilidad a la hora de implementar el servicio en su sitio de trabajo.



## ¿Que es LDAP?

La iniciales **LDAP** en inglés significa **Lightweight Directory Access Protocol (LDAP)**; traducido al español su significado es: **Protocolo Ligero para Acceder al Servicio de Directorio**, ésta implementación se basa en el estándar X.500, el cual es un conjunto de estándares de redes de computadoras de la ITU-T sobre el servicio de directorios. **LDAP** se ejecuta sobre **TCP/IP** o sobre otros servicios de transferencia orientado a conexión; que permite el acceso a la data de un directorio ordenado y distribuido para buscar información.

Habitualmente se almacena información de los usuarios que conforman una red de computadores, como por ejemplo el nombre de usuario, contraseña, directorio hogar, etc. Es posible almacenar otro tipo de información tal como, bebida preferida, número de teléfono celular, fecha de cumpleaños, etc.

En conclusión, **LDAP** es un protocolo de acceso unificado a un conjunto de información sobre los usuarios de una red de computadores.

## Razones para utilizar LDAP

Al utilizar **LDAP** se puede consolidar información para toda una organización dentro de un repositorio central. Por ejemplo, en vez de administrar listas de usuarios para cada grupo dentro de una organización, puede usar **LDAP** como directorio central, accesible desde cualquier parte de la red. Puesto que **LDAP** soporta la Capa de conexión segura (**SSL**) y la Seguridad de la capa de transporte (**TLS**), los datos confidenciales se pueden proteger de los curiosos. **LDAP** también soporta un número de bases de datos "**back-end**" en las que se almacena la información. Esto permite que los administradores tengan la flexibilidad para desplegar la base de datos más indicada, para el tipo de información. **LDAP** tiene una interfaz de programación de aplicaciones (**API**)



bien definida, existe un número de aplicaciones acreditadas para **LDAP**, éstas están aumentando en cantidad y calidad, las hay en distintos lenguajes de programación, tales como C, C++, Java, Perl, PHP, entre otros.

## ¿Qué es un servicio de directorio?

Un servicio de directorio (**SD**) es una aplicación o un conjunto de aplicaciones que almacena y organiza la información de los usuarios de una red de computadores, permitiendo a los administradores gestionar el acceso de usuarios a los recursos sobre dicha red. Además, los servicios de directorio actúan como una capa de abstracción entre los usuarios y los recursos compartidos.

Los directorios tienden a contener información descriptiva basada en atributos y tienen capacidades de filtrado muy sofisticada. Los directorios generalmente no soportan transacciones complicadas ni esquemas de vuelta atrás (**Roll Back**) como los que se encuentran en los sistemas de bases de datos diseñados para manejar grandes y complejos volúmenes de actualizaciones. Las actualizaciones de los directorios son normalmente cambios simples.

Un servicio de directorio no debería confundirse con el repositorio de directorio, que es la base de datos, esta es la que contiene la información sobre los objetos nombrados, gestionado por el servicio de directorio. El servicio de directorio proporciona la interfaz de acceso a los datos que se contienen en unos o más espacios de nombre de directorio. La interfaz del servicio de directorio es la encargada de gestionar la autenticación de los accesos al servicio de forma segura, actuando como autoridad central para el acceso a los recursos de sistema que manejan los datos del directorio.





Como base de datos, un servicio de directorio está altamente optimizado para lecturas y proporciona alternativas avanzadas de búsqueda en los diferentes atributos que se puedan asociar a los objetos de un directorio. Los datos que se almacenan en el directorio son definidos por un esquema extensible y modificable. Los servicios de directorio utilizan un modelo distribuido para almacenar su información y esa información generalmente está replicada entre los servidores que forman el directorio.

Los directorios están afinados para dar una rápida respuesta a grandes volúmenes de búsquedas. Estos tienen la capacidad de replicar la información para incrementar la disponibilidad y la fiabilidad, al tiempo que reducen los tiempos de respuesta. Cuando la información de un directorio se replica, se pueden producir inconsistencias temporales entre las réplicas mientras esta se está sincronizando.

Hay muchas formas diferentes de proveer un servicio de directorio. Diferentes métodos permiten almacenar distintos tipos de información en el directorio, tener distintos requisitos sobre cómo la información ha de ser referenciada, consultada y actualizada, cómo es protegida de los accesos no autorizados, etc. Algunos servicios de directorio son locales, es decir, proveen el servicio a un contexto restringido (como por ejemplo, el servicio "*finger*" en una única máquina). Otros servicios son globales y proveen servicio a un contexto mucho más amplio (como por ejemplo, Internet). Los servicios globales normalmente son distribuidos, esto significa que los datos están repartidos a lo largo de distintos equipos, los cuales cooperan para dar el servicio de directorio. Típicamente, un servicio global define un espacio de nombres uniforme que da la misma visión de los datos, independientemente de donde se esté, en relación a los propios datos. El servicio **DNS (Domain Name System)** es un ejemplo de un sistema de directorio globalmente distribuido.



## ¿Qué tipo de información se puede almacenar en un directorio?

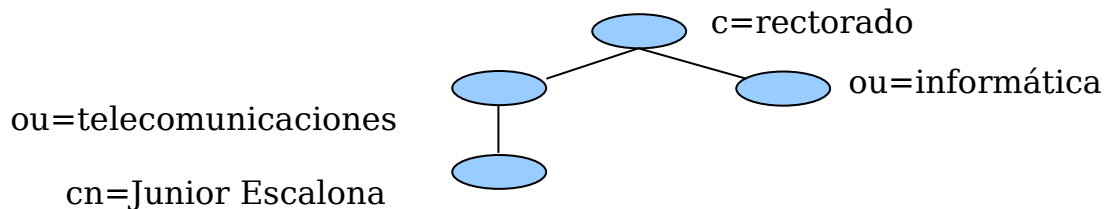
En principio en un servicio de directorio se puede almacenar cualquier tipo de información. Como por ejemplo, nombre, dirección de habitación, nombre de la mascota, música preferida, bebida favorita, etc. Sin embargo, la información que se almacena es aquella que permita organizar de manera jerárquica todos los usuarios de la red. Estructurar la información de los usuarios de la red es de utilidad a la hora de restringir el acceso a los servicios y recursos de la red; Permitiendo gestionar con mayor facilidad la red.

## ¿Cómo se almacena la información en LDAP?

La información es ordenada en el modelo de **LDAP** en entradas. Una entrada es una colección de atributos que tienen un único **Nombre Global Distinguido (DN)**. El **DN** se utiliza para referirse a una entrada sin ambigüedades. Cada atributo de una entrada posee un tipo y uno o más valores. Los tipos son normalmente palabras nemotécnicas, como **"cn"** para **common name**, o **"mail"** para una dirección de correo. La sintaxis de los atributos depende del tipo de atributo. Por ejemplo, un atributo **cn** puede contener el valor **"Luis Márquez"**. Un atributo **email** puede contener un valor **"marquezl@ucla.edu.ve"**.

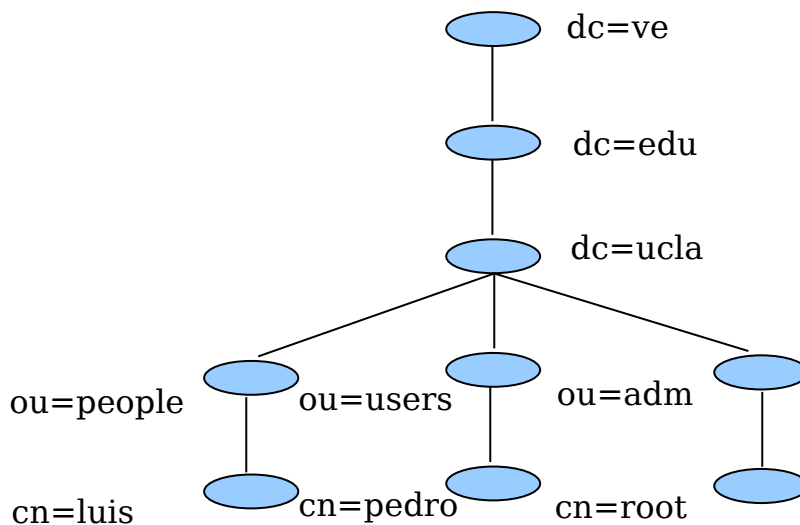
Estas entradas están organizadas en una estructura jerárquica en forma de árbol invertido, de la misma manera como se estructura el sistema de archivos de **UNIX**. Tradicionalmente esta estructura reflejaba los límites geográficos y/o organizacionales. Las entradas que representan países aparecen en la parte superior del árbol. Debajo de ellos, están las entradas que representan los

estados y las organizaciones nacionales. Bajo estas, pueden estar las entradas que representan las unidades organizacionales, empleados, impresoras, documentos o todo aquello que pueda imaginarse. La siguiente figura muestra un árbol de directorio **LDAP** haciendo uso del nombramiento tradicional.



**Figura N° 1** Árbol de directorio LDAP (nombramiento tradicional)

El árbol también se puede organizar basándose en los nombres de dominio de Internet. Este tipo de nombramiento se está volviendo muy popular y en los actuales momentos es el más utilizado, ya que permite localizar un servicio de directorio haciendo uso de los DNS. La siguiente figura muestra un árbol de directorio que hace uso de los nombres basados en dominios.



**Figura N° 2** Árbol de directorio LDAP (nombramiento de Internet)



Un ejemplo del DN sería:

**dn: cn=Luis Márquez, ou=people, dc=ucla, dc=edu, dc=ve**

Observe que el **dn** se construye de abajo hacia arriba. Al igual que se construyen los nombres en **DNS**.

Además, LDAP permite controlar qué atributos son requeridos y permitidos en una entrada gracias al uso del atributo denominado **objectClass**. El valor del atributo **objectClass** determina qué reglas de diseño (**schema rules**) ha de seguir la entrada.

### ¿Cómo se referencia la información en LDAP?

Una entrada es referenciada por su nombre distinguido, que es construido por el nombre de la propia entrada llamado **Nombre Relativo Distinguido (RDN)** y la concatenación de los nombres de las entradas que le anteceden. Por ejemplo, la entrada para **luis** en el ejemplo del nombramiento de Internet anterior tiene el siguiente RDN: **uid=luis** y su DN sería: **uid=luis,ou=people,dc=ucla,dc=edu,dc=ve**. De esta manera se puede acceder a toda la información que se almacenada en el directorio **LDAP**.

### ¿Cómo se accede a la información en LDAP?

**LDAP** define operaciones para interrogar y actualizar el directorio. Provee operaciones para añadir, modificar y eliminar entradas del mismo. La mayor parte del tiempo, **LDAP** se utiliza para buscar información almacenada en el directorio. Las operaciones de búsqueda de **LDAP** permiten encontrar entradas



que concuerdan con algún criterio especificado dado por un filtro de búsqueda.

La información puede ser solicitada desde cada entrada que concuerda con dicho criterio.

### ¿Cómo lo accede?

Por ejemplo, imagínese que quiere buscar en el subárbol del directorio que está por debajo de ***dc=ucla,dc=edu,dc=ve*** a personas con el nombre ***Luis Márquez***, obteniendo la dirección de correo electrónico de cada entrada que concuerde. ***LDAP*** permite hacer esto fácilmente. O tal vez prefiera buscar las organizaciones que posean la cadena ***ucla*** en su nombre o posean número de ***fax***. ***LDAP*** es muy flexible y permite hacer esto y mucho más.

### ¿Cómo protege la información de accesos no autorizados?

Algunos servicios de directorio no proveen protección, permitiendo a cualquier persona acceder a la información. ***LDAP*** provee un mecanismo de autenticación para los clientes, o la confirmación de identidad en un servidor de directorio, facilitando el camino para un control de acceso que proteja la información que el servidor posee. ***LDAP*** también soporta los servicios de privacidad e integridad.

### ¿Cómo trabaja LDAP?

El servicio de directorio de ***LDAP*** está basado en el modelo ***cliente/servidor***. Uno o más servidores ***LDAP*** contienen los datos que conforman la información del árbol del directorio (***DIT***). El cliente se conecta a los servidores y les formula preguntas. Los servidores responden con una



respuesta o con un puntero donde el cliente puede obtener información adicional (normalmente otro servidor **LDAP**). No importa a que servidor **LDAP** se conecte un cliente, este siempre obtendrá la misma visión del directorio; un nombre presentado por un servidor **LDAP** referencia la misma entrada que cualquier otro servidor **LDAP**. Esta es una característica muy importante del servicio global de directorio, como **LDAP**.

## X.500

**X.500** es un conjunto de estándares de redes de computadores de la **ITU** (Unión Internacional de Telecomunicaciones) sobre servicios de directorio, entendidos estos como bases de datos de direcciones electrónicas (o de otros tipos). El estándar se desarrolló conjuntamente con la **ISO** como parte del modelo de interconexión de sistemas abiertos, para usarlo como soporte del correo electrónico **X.400**.

Los protocolos definidos por **X.500** incluyen:

- Protocolo de acceso al directorio (**DAP**)
- Protocolo de sistema de directorio
- Protocolo de ocultación de información de directorio
- Protocolo de gestión de enlaces operativos de directorio.

Dentro de la serie X.500, la especificación que ha resultado ser la más difundida no trata de protocolos de directorio, sino de certificados de clave pública X.509.

El protocolo **LDAP** fue creado como una versión liviana de **X.500** y terminó por reemplazarlo. Por esta razón algunos de los conceptos y estándares que utiliza **LDAP** provienen de la serie de protocolos **X.500**.



Técnicamente, **LDAP** es un protocolo de acceso a directorio para el servicio de directorio **X.500**, del servicio de directorio de **OSI**. Inicialmente, los cliente **LDAP** accedían a través de puertas de enlace al servicio de directorio **X.500**. Esta puerta de enlace ejecutaba **LDAP** entre el cliente y la puerta de enlace, y el Protocolo **X.500** de Acceso al Directorio (**DAP**) entre la puerta de enlace y el servidor **X.500**. **DAP** es un protocolo extremadamente pesado que opera sobre una pila protocolar **OSI** completa y requiere una cantidad significativa de recursos computacionales. **LDAP** está diseñado para operar sobre **TCP/IP** proporcionando una funcionalidad similar a la de **DAP**, pero con un costo muchísimo menor.

Aunque **LDAP** se utiliza todavía para acceder al servicio de directorio **X.500** a través de puertas de enlace, hoy en día es más común implementar **LDAP** directamente en los servidores **X.500**.

El demonio autónomo de **LDAP**, o **SLAPD**, puede ser visto como un servidor de directorio **X.500** ligero. Es decir, no implementa el **DAP X.500**, sino un subconjunto de modelos de **X.500**.

Es posible replicar datos desde un servidor de directorio **LDAP** hacia un servidor **DAP X.500**. Esta operación requiere una puerta de enlace **LDAP/DAP**. **OpenLDAP** no suministra dicha puerta de enlace, pero el demonio de replicación que posee puede ser usado para la replicación, como si de una puerta de enlace se tratase.

### Diferencias entre LDAP v2 y v3

**LDAPv3** fue desarrollado en los años 90 para reemplazar a **LDAPv2**.



**LDAPv3** incorpora las siguientes características a **LDAP**:

- Autenticación fuerte haciendo uso de **SASL (Simple Authentication and Security Layer)**
- Protección de integridad y confidencialidad haciendo uso de **TLS (SSL), Transport Layer Security (Secure Sockets Layer)**
- Internacionalización gracias al uso de Unicode
- Remisiones y continuaciones
- Descubrimiento de esquemas
- Extensibilidad (controles, operaciones extendidas y más)

Como **LDAPv2** difiere significativamente de **LDAPv3**, la interacción entre ambas versiones puede ser un poco problemática. Es recomendable no utilizar la versión de **LDAPv2**, por lo que en la implementación de **OpenLDAP** viene deshabilitado por omisión.

### ¿Qué es SLAPD?

**SLAPD** es un servidor de directorio **LDAP**. Es una de las tantas implementaciones de **LDAP** en Software Libre. Esta versión es la muy popular y se puede decir que es la más implementada; vale la pena mencionar que existen varias implementaciones de **LDAP** en software propietario, como por ejemplo implementaciones de SUN, IBM, etc.

### La versión 3 de LDAP

- Soporta **LDAP** sobre **IPv4, IPv6** y **Unix IPC**
- Tiene soporte de autenticación fuerte gracias al uso de **SASL**. La implementación **SASL** de **SLAPD** hace uso del software **Cyrus SASL**, el





cual soporta un gran número de mecanismos de autenticación, como: **DIGEST-MD5**, **EXTERNAL**, y **GSSAPI**.

- Provee protecciones de privacidad e integridad gracias al uso de **TLS** o **SSL**. La implementación **TLS** de **SLAPD** hace uso del software **OpenSSL**
- Puede ser configurado para restringir el acceso a la capa de **socket** basándose en la información topológica de la red. Esta característica hace uso de los **TCP wrappers** (*Herramienta simple que sirve para monitorear y controlar el tráfico que llega por la red*)
- Provee facilidades de control de acceso muy potentes, permitiéndole controlar el acceso a la información de su(s) base(s) de datos. Puede controlar el acceso a las entradas basándose en la información de autorización de **LDAP**, en la dirección IP, en los nombres de dominio y otros criterios. **SLAPD** soporta tanto el control de acceso a la información dinámico como estático.
- Soporta Unicode y etiquetas de lenguaje.
- Viene con una serie de **backends** para diferentes bases de datos. Estos incluyen **DBD**, un **backend** de una base de datos transaccional de alto rendimiento; **LDBM**, un **backend** ligero basado en **DBM**; **SHELL**, una interface para scripts de **shell**; y **PASSWD**, un **backend** simple para el archivo **passwd**. El **backend BDB** hace uso de **Sleepcat Berkeley DB**. **LDBM** utiliza cualquiera de las siguientes: **Berkeley DB o GDBM**
- Se puede configurar para servir a múltiples bases de datos al mismo tiempo. Esto significa que un único servidor **SLAPD** puede responder a peticiones de diferentes porciones lógicas del árbol de **LDAP**, haciendo uso del mismo o distintos **backends** de bases de datos.



- Si necesita más personalización, **SLAPD** le permite escribir sus propios módulos fácilmente. **SLAPD** consiste en dos partes diferentes: un **frontend** que maneja las comunicaciones protocolares con los clientes **LDAP**; y módulos que manejan tareas específicas como las operaciones con las bases de datos. Debido a que estas dos piezas se comunican a través de una **API** bien definida, puede escribir sus propios módulos, que extenderán **SLAPD** de múltiples maneras. También existen numerosos módulos programables de bases de datos. Estos permiten a **SLAPD** acceder a fuentes de datos externos haciendo uso de lenguajes de programación populares (**Perl, shell, SQL** y **TCL**)
- Hace uso de hilos para obtener alto rendimiento. Un proceso único multihilo maneja todas las peticiones entrantes haciendo uso de una *piscina* de hilos. Esto reduce la carga del sistema a la vez que provee alto rendimiento.
- Se puede configurar para que mantenga copias de la información del directorio. Este esquema de replicación, **un único maestro/múltiples esclavos**, es vital en ambientes con un volumen alto de peticiones, donde un único servidor **SLAPD** no podría proveer la disponibilidad ni la confiabilidad necesarias. **SLAPD** incluye también un soporte experimental para la replicación de **múltiples maestros**. **SLAPD** soporta dos métodos de replicación: **Sync LDAP** y **SLURP (servidor de replicación LDAP)**.
- Puede ser configurado como un servicio proxy de caché LDAP.
- Es altamente configurable a través de un único archivo de configuración, que permite modificar todo aquello que se necesite cambiar. Las opciones por omisión son razonables, lo que facilita mucho el trabajo.



## SASL

“**Simple Authentication and Security Layer**” (**capa de seguridad y autenticación simple**). Es un framework para manejar la autenticación y autorización en protocolos de internet. Este separa los mecanismos de autenticación de los protocolos de la aplicación.

Como **SASL** sólo se maneja la autenticación se requieren otros mecanismos como por ejemplo **TLS** para cifrar el contenido que se transfiere.

Los protocolos definen su representación de intercambios SASL con un *perfil*. Un protocolo tiene un **nombre de servicio** como "**LDAP**" en un registro compartido con GSSAPI (Generic Security Services Application Programming Interface) y KERBEROS.

Entre los protocolos que ahora mismo usan SASL se incluyen IMAP, LDAP, POP3, SMTP y XMPP.



## SLDAP (Servidor LDAP) en debian

Antes de comenzar a describir el proceso de instalación del LDAP se debe tener en cuenta lo siguiente:

1. Al finalizar la instalación y configuración **LDAP** estará en la capacidad de autenticar usuarios ante el servicio de directorio. La versión que se utilizará es la 2.3.30-5+etch1; ésta es la última versión estable para el momento de la realización de este material.
2. Se utilizará el dominio ucla.edu.ve a lo largo de todo el material.

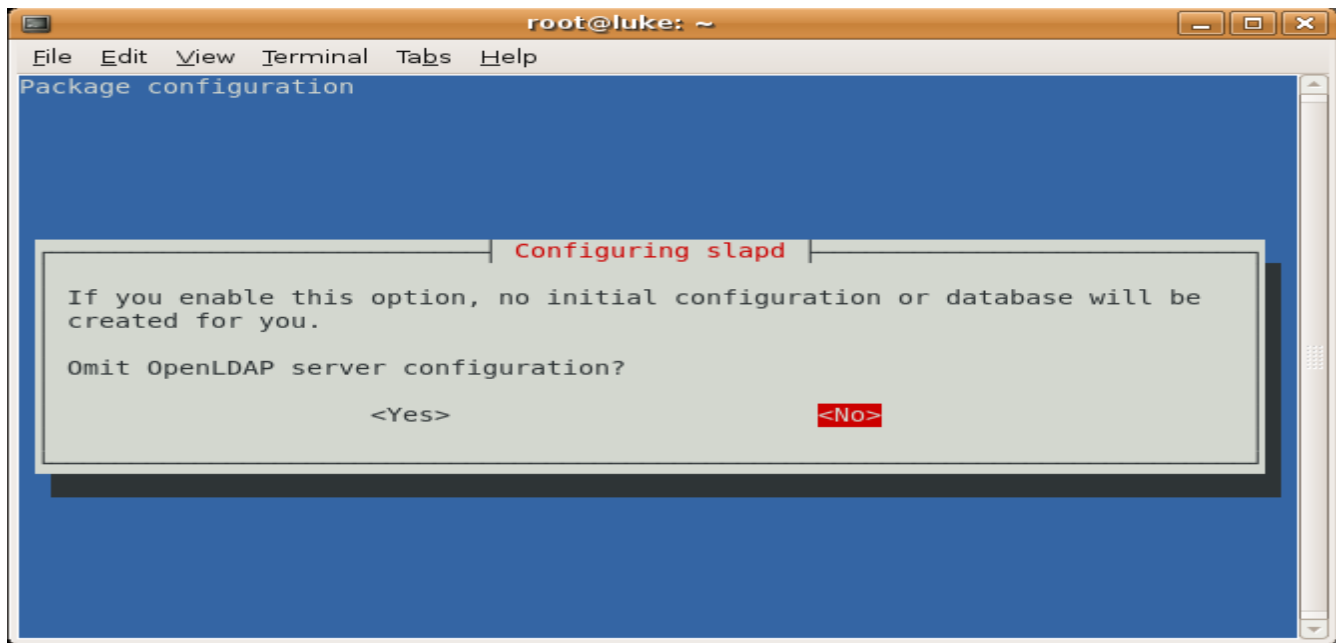
### Pasos para la instalación de **SLAPD**

- Ejecute el siguiente comando:
  - ***#aptitude install slapd ldap-utils***
- Confirme la ejecución del comando.
- Se debe introducir la clave del administrador del **LDAP**.
- Se confirma la clave.

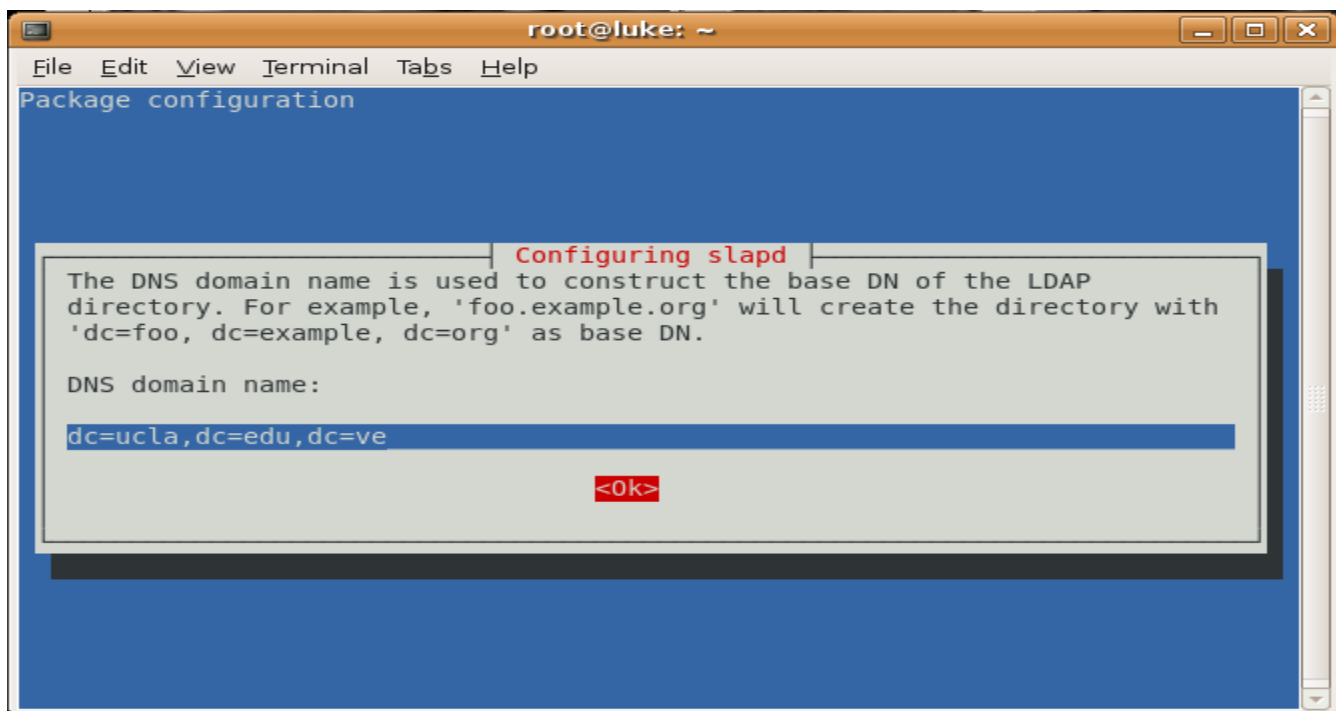
### Reconfiguración del paquete slapd

- Ejecutamos el siguiente comando
  - ***#dpkg-reconfigure slapd***

El objetivo es reconfigurar el slapd. No podemos omitir esta sección

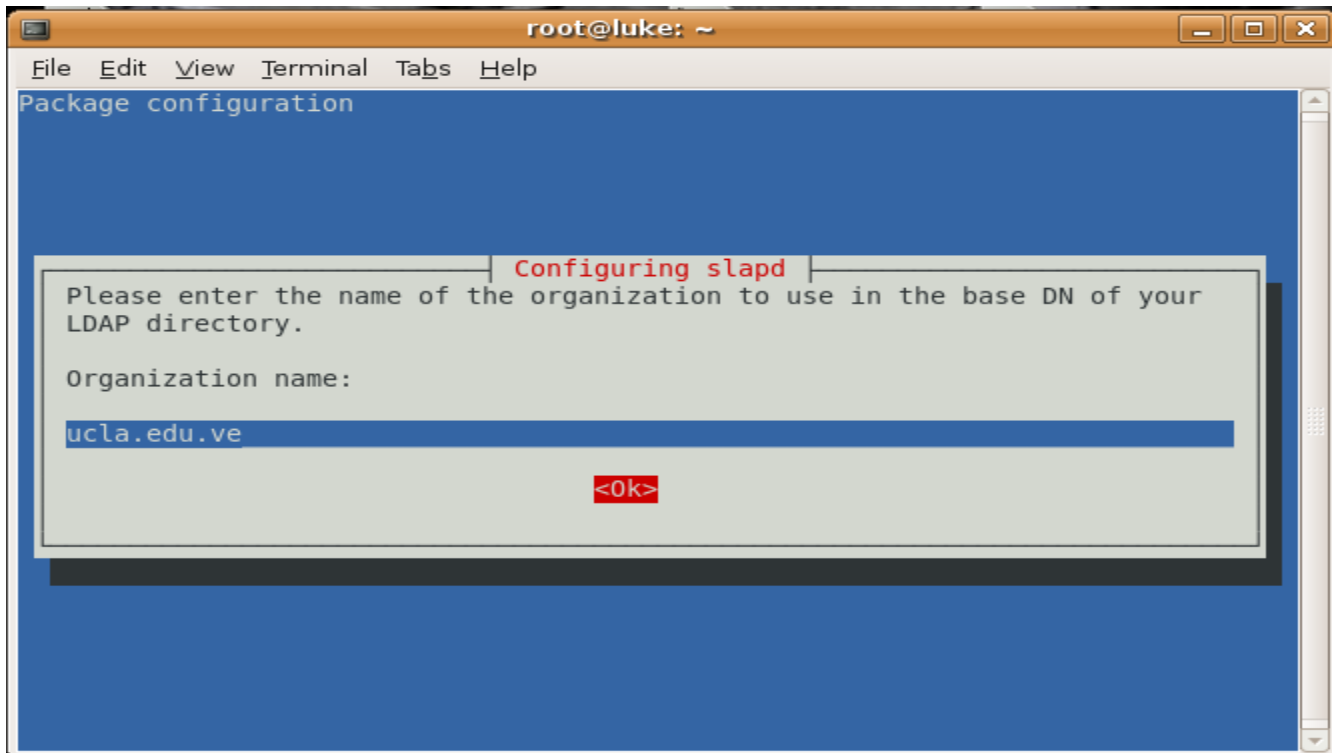


- Configuramos el dominio

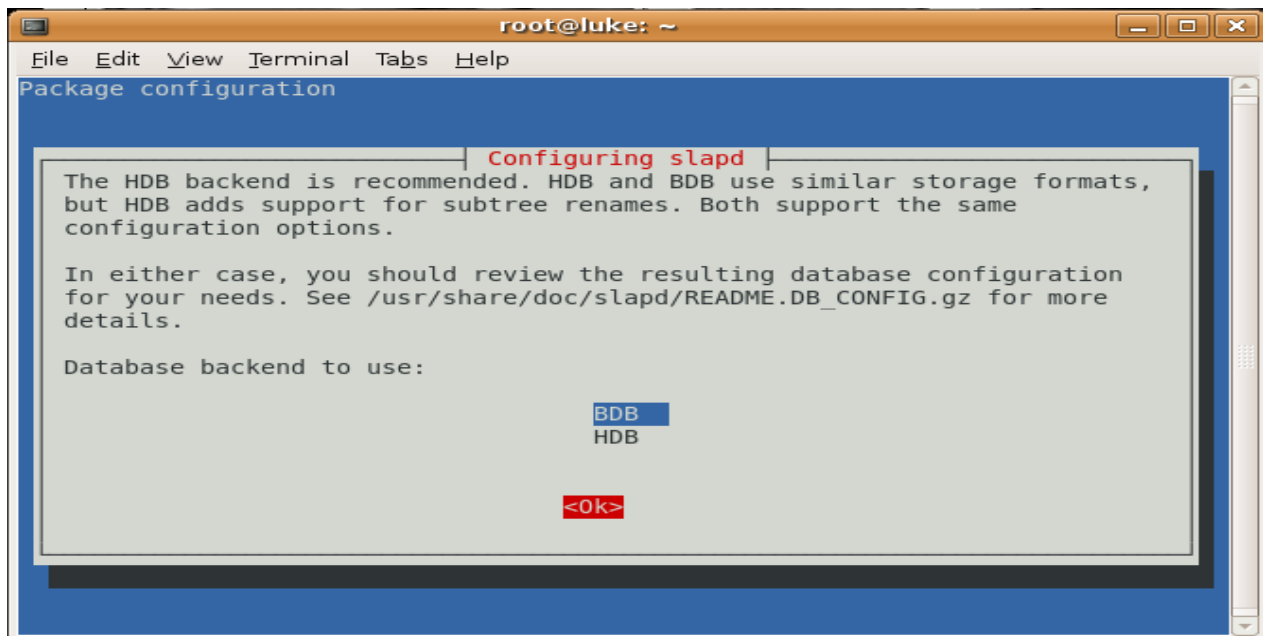




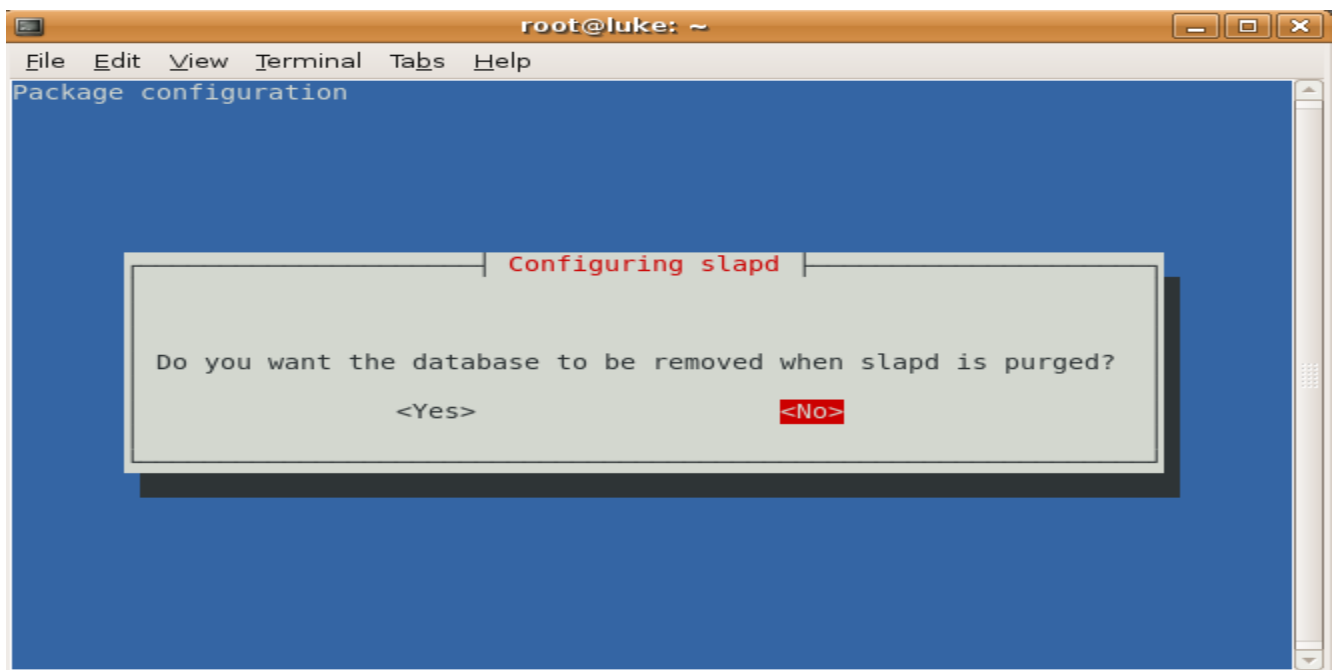
- Configuramos el nombre de la organización

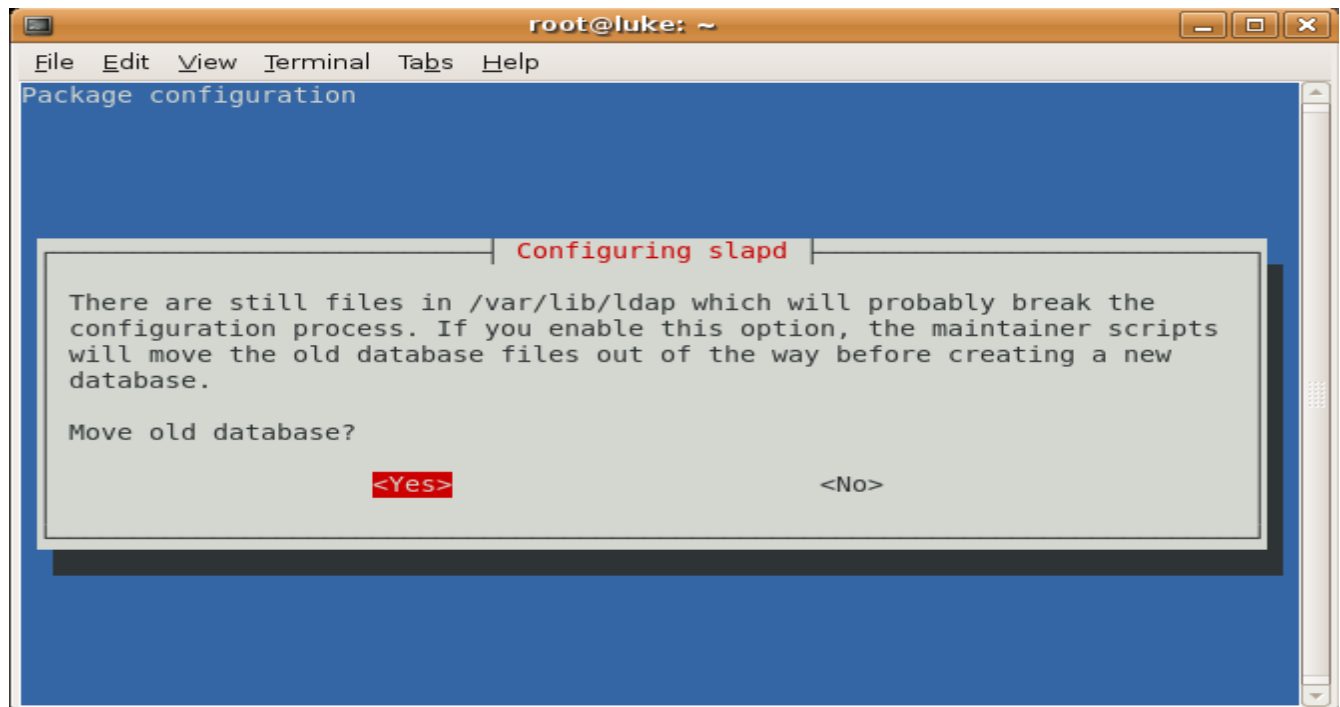


- Configuramos el password del administrador en **LDAP**.
- Confirmamos el password del administrador.
- Elegimos la base de datos.



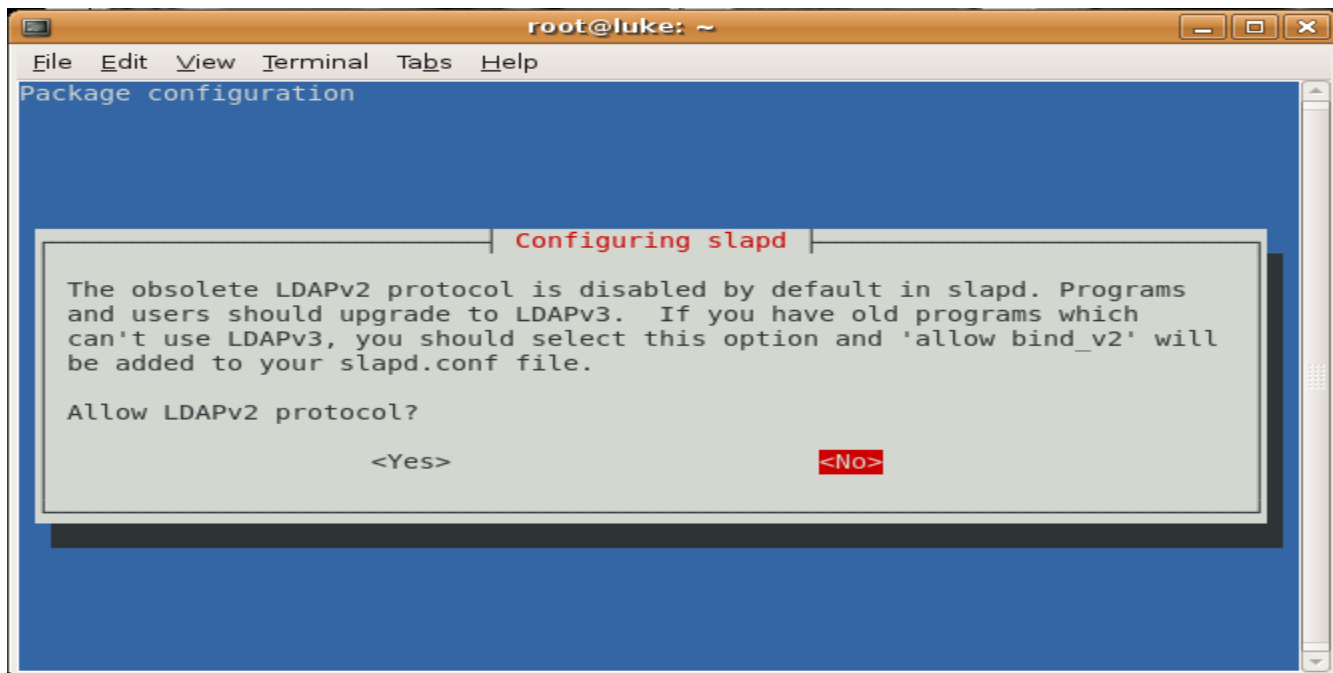
- Es recomendable seleccionar que **no** se remueva la base de datos cuando se desinstale el **SLAPD**





- Como estamos reconfigurando el **SLAPD** haremos una copia de la base de datos anterior.
- Tal y como recomienda la documentación, no daremos soporte para la versión 2 de **LDAP**





Y con esto terminamos la instalación del **LDAP**

### Pruebas de la instalación

- El servidor está escuchando por el puerto correspondiente:
  - ***netstat -tpua | more***
- El resultado del comando anterior debe ser el siguiente:



```
root@luke: ~
File Edit View Terminal Tabs Help
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
PID/Program name
tcp        0      0 *:ldap                  *:*                     LISTEN
10297/slapd
tcp        0      0 localhost:mysql        *:*                     LISTEN
5439/mysql
tcp        0      0 *:www                   *:*                     LISTEN
6306/apache2
tcp        0      0 localhost:ipp          *:*                     LISTEN
5570/cupsd
tcp        0      0 localhost:postgresql  *:*                     LISTEN
5520/postgres
tcp        0      0 localhost:smtp         *:*                     LISTEN
5825/exim4
tcp        0      0 luke.ucla.edu.ve:41132 64-215-156-82.eosin:ww ESTABLISHED
7165/firefox
tcp        1      0 luke.ucla.edu.ve:46961 yw-in-f104.google.c:ww CLOSE_WAIT
7216/npviewer.bin
tcp        1      0 luke.ucla.edu.ve:41076 fh-in-f19.google.co:ww CLOSE_WAIT
7216/npviewer.bin
tcp        0      0 luke.ucla.edu.ve:49760 www.03.01.ash1.face:ww ESTABLISHED
7165/firefox
--More--
```

- Haremos una conexión al **LDAP**

```
root@luke: ~
File Edit View Terminal Tabs Help
root@luke:~# ldapsearch -x -b "dc=ucla,dc=edu,dc=ve"
```



- El resultado del comando anterior debe ser algo como

```
root@luke: ~  
File Edit View Terminal Tabs Help  
#  
# ucla.edu.ve  
dn: dc=ucla,dc=edu,dc=ve  
objectClass: top  
objectClass: dcObject  
objectClass: organization  
o: ucla.edu.ve  
dc: ucla  
  
# admin, ucla.edu.ve  
dn: cn=admin,dc=ucla,dc=edu,dc=ve  
objectClass: simpleSecurityObject  
objectClass: organizationalRole  
cn: admin  
description: LDAP administrator  
  
# search result  
search: 2  
result: 0 Success  
  
# numResponses: 3  
# numEntries: 2  
root@luke:~#
```

## Administración de usuarios

En este apartado se mostrará como agregar un usuario al **LDAP** utilizando para ello los **ldif** (**LDAP Data Interchange Format**). A continuación se mostrará cual es la estructura de los **ldif**.

ldif para la creación de una unidad organizacional "people". El nombre del archivo es people.ldif

**dn: ou=people,dc=ucla,dc=edu,dc=ve**

**ou: people**

**objectclass: organizationalUnit**



## Creación de los archivos ldif

- Ldif para la creación de una unidad organizacional "group".

El nombre del archivo es group.ldif

dn: ou=group,dc=ucla,dc=edu,dc=ve

ou: group

objectclass: organizationalUnit

- Ldif para la creación de un grupo "users". El nombre del archivo es users.ldif

dn: cn=users,ou=group,dc=ucla,dc=edu,dc=ve

objectclass: posixGroup

objectclass: top

cn: users

userPassword: {crypt}\*

gidNumber: 100

## Agregar los contenidos de los ldif

Para agregar los contenidos de los ldif al LDAP se ejecutan los siguientes comandos.

```
#dapadd -x -W -D "cn=admin,dc=ucla,dc=edu,dc=ve" -f people.ldif
```

```
#ldapadd -x -W -D "cn=admin,dc=ucla,dc=edu,dc=ve" -f group.ldif
```

```
#ldapadd -x -W -D "cn=admin,dc=ucla,dc=edu,dc=ve" -f users.ldif
```



Al ejecutar estos comandos te pedirá el password del administrador **LDAP** y el resultado de los mismos se muestra en la siguiente imagen

```
root@luke: ~
File Edit View Terminal Tabs Help
Password:
root@luke:~# vi people.ldif
root@luke:~# vi group.ldif
root@luke:~# vi users.ldif
root@luke:~# ldapadd -x -W -D "cn=admin,dc=ucla,dc=edu,dc=ve" -f people.ldif
Enter LDAP Password:
adding new entry "ou=people,dc=ucla,dc=edu,dc=ve"

root@luke:~# ldapadd -x -W -D "cn=admin,dc=ucla,dc=edu,dc=ve" -f group.ldif
Enter LDAP Password:
adding new entry "ou=group,dc=ucla,dc=edu,dc=ve"

root@luke:~# ldapadd -x -W -D "cn=admin,dc=ucla,dc=edu,dc=ve" -f users.ldif
Enter LDAP Password:
adding new entry "cn=users,ou=group,dc=ucla,dc=edu,dc=ve"

root@luke:~#
root@luke:~#
root@luke:~#
root@luke:~#
root@luke:~#
root@luke:~#
root@luke:~#
root@luke:~#
```



Para revisar el resultado de la inserción ejecutamos el siguiente comando

- **#ldapsearch -x -b "dc=ucla,dc=edu,dc=ve"**

```
root@luke: ~
File Edit View Terminal Tabs Help
# people, ucla.edu.ve
dn: ou=people,dc=ucla,dc=edu,dc=ve
ou: people
objectClass: organizationalUnit

# group, ucla.edu.ve
dn: ou=group,dc=ucla,dc=edu,dc=ve
ou: group
objectClass: organizationalUnit

# users, group, ucla.edu.ve
dn: cn=users,ou=group,dc=ucla,dc=edu,dc=ve
objectClass: posixGroup
objectClass: top
cn: users
gidNumber: 100

# search result
search: 2
result: 0 Success

# numResponses: 6
# numEntries: 5
root@luke:~#
```

Se puede confirmar en la salida del comando anterior que la inserción de los anteriores **ldif** están presentes en la estructura del árbol del **LDAP**



## Agregar usuarios

Se debe crear un **ldif** como se muestra a continuación

- Nombre del archivo **ldif** marquezl.ldif

dn: uid=marquezl,ou=people,dc=ucla,dc=edu,dc=ve

uid: marquezl

cn: Luis Márquez

objectClass: account

objectClass: posixAccount

objectClass: top

objectClass: shadowAccount

userPassword: {crypt}\$1\$HnC/X/r4\$VknfcQlq24qGgdnDVhDIp1

shadowLastChange: 14001

shadowMax: 99999

shadowWarning: 7

loginShell: /bin/bash

uidNumber: 1001

gidNumber: 100

homeDirectory: /home/marquezl

gecos: Luis Márquez



Para agregar el nuevo usuario al nuevo árbol de **LDAP** se debe ejecutar el siguiente comando

```
#ldapadd -x -W -D "cn=admin,dc=ucla,dc=edu,dc=ve" -f marquezl.ldif
```

```
root@luke: ~  
File Edit View Terminal Tabs Help  
# group, ucla.edu.ve  
dn: ou=group,dc=ucla,dc=edu,dc=ve  
ou: group  
objectClass: organizationalUnit  
  
# users, group, ucla.edu.ve  
dn: cn=users,ou=group,dc=ucla,dc=edu,dc=ve  
objectClass: posixGroup  
objectClass: top  
cn: users  
gidNumber: 100  
  
# search result  
search: 2  
result: 0 Success  
  
# numResponses: 6  
# numEntries: 5  
root@luke:~# vi marquezl.ldif  
root@luke:~# ldapadd -x -W -D "cn=admin,dc=ucla,dc=edu,dc=ve" -f marquezl.ldif  
Enter LDAP Password:  
adding new entry "uid=marquezl,ou=people,dc=ucla,dc=edu,dc=ve"  
root@luke:~# █
```

Para revisar el resultado de la inserción ejecutamos el siguiente comando

- **#ldapsearch -x -b "dc=ucla,dc=edu,dc=ve"**





```
root@luke: ~
File Edit View Terminal Tabs Help

# marquezl, people, ucla.edu.ve
dn: uid=marquezl,ou=people,dc=ucla,dc=edu,dc=ve
uid: marquezl
cn: THVpcyBNw6FycXVleg==
objectClass: account
objectClass: posixAccount
objectClass: top
objectClass: shadowAccount
shadowMax: 99999
shadowWarning: 7
loginShell: /bin/bash
uidNumber: 1001
gidNumber: 100
homeDirectory: /home/marquezl
gecos: Luis Marquez

# search result
search: 2
result: 0 Success

# numResponses: 7
# numEntries: 6
root@luke:~#
```

## Eliminar un usuario

Para eliminar un usuario del árbol del directorio se debe crear un archivo del tipo **ldif** como se muestra a continuación

```
dn: uid=marquezl,ou=people,dc=ucla,dc=edu,dc=ve
changetype: delete
```

- Ejecutamos el siguiente comando para eliminar un usuario del árbol **LDAP**

```
#ldapmodify -x -D "cn=admin,dc=ucla,dc=edu,dc=ve" -W -f boorrar.ldif
```



```
root@luke: ~  
File Edit View Terminal Tabs Help  
objectClass: account  
objectClass: posixAccount  
objectClass: top  
objectClass: shadowAccount  
shadowMax: 99999  
shadowWarning: 7  
loginShell: /bin/bash  
uidNumber: 1001  
gidNumber: 100  
homeDirectory: /home/marquezl  
gecos: Luis Marquez  
  
# search result  
search: 2  
result: 0 Success  
  
# numResponses: 7  
# numEntries: 6  
root@luke:~# vi borrar.ldif  
root@luke:~# ldapmodify -x -D "cn=admin,dc=ucla,dc=edu,dc=ve" -W -f borrar.ldif  
Enter LDAP Password:  
deleting entry "uid=marquezl,ou=people,dc=ucla,dc=edu,dc=ve"  
  
root@luke:~# █
```

Para revisar el resultado de la eliminación ejecutamos el siguiente comando

- ***#ldapsearch -x -b "dc=ucla,dc=edu,dc=ve"***



```
root@luke: ~
File Edit View Terminal Tabs Help
# people, ucla.edu.ve
dn: ou=people,dc=ucla,dc=edu,dc=ve
ou: people
objectClass: organizationalUnit

# group, ucla.edu.ve
dn: ou=group,dc=ucla,dc=edu,dc=ve
ou: group
objectClass: organizationalUnit

# users, group, ucla.edu.ve
dn: cn=users,ou=group,dc=ucla,dc=edu,dc=ve
objectClass: posixGroup
objectClass: top
cn: users
gidNumber: 100

# search result
search: 2
result: 0 Success

# numResponses: 6
# numEntries: 5
root@luke:~#
```

## Agregar atributos

- Para agregar un atributo se debe crear un **ldif** como se muestra a continuación. El nombre del **ldif** es add.ldif

```
dn: uid=marquezl,ou=people,dc=ucla,dc=edu,dc=ve
changetype: add
objectclass: top
objectclass: person
objectclass: organizationalPerson
objectclass: inetOrgPerson
cn: Luis Márquez
sn: marquezl
```



telephonenumber: 04127900022

- Se ejecuta la siguiente sintaxis

```
root@luke: ~  
File Edit View Terminal Tabs Help  
root@luke:~# vi add.ldif  
root@luke:~# ldapmodify -x -D "cn=admin,dc=ucla,dc=edu,dc=ve" -W -f add.ldif  
Enter LDAP Password:  
adding new entry "uid=marquezl,ou=people,dc=ucla,dc=edu,dc=ve"  
root@luke:~# █
```

- Para verificar el resultado ejecutamos la siguiente sintaxis



```
root@luke: ~  
File Edit View Terminal Tabs Help  
dn: cn=users,ou=group,dc=ucla,dc=edu,dc=ve  
objectClass: posixGroup  
objectClass: top  
cn: users  
gidNumber: 100  
  
# marquezl, people, ucla.edu.ve  
dn: uid=marquezl,ou=people,dc=ucla,dc=edu,dc=ve  
objectClass: top  
objectClass: person  
objectClass: organizationalPerson  
objectClass: inetOrgPerson  
cn:: THVpcyBNw6FycXVleg==  
sn: marquezl  
telephoneNumber: 04127900022  
uid: marquezl  
  
# search result  
search: 2  
result: 0 Success  
  
# numResponses: 7  
# numEntries: 6  
root@luke:~#
```

## Modificar un atributo

- Para modificar un atributo se debe crear un **ldif** como se muestra a continuación. El nombre ldif es modify.ldif

```
dn: uid=marquezl,ou=people,dc=ucla,dc=edu,dc=ve  
changetype: modify  
replace: telephonenumber  
telephonenumber: 04164716421
```



- Se ejecuta la siguiente sintaxis

```
root@luke: ~  
File Edit View Terminal Tabs Help  
root@luke:~# vi modify.ldif  
root@luke:~# ldapmodify -x -D "cn=admin,dc=ucla,dc=edu,dc=ve" -w -f modify.ldif  
Enter LDAP Password:  
modifying entry "uid=marquezl,ou=people,dc=ucla,dc=edu,dc=ve"  
  
root@luke:~# █
```

- Para verificar el cambio ejecutamos la siguiente sintaxis

```
root@luke: ~  
File Edit View Terminal Tabs Help  
dn: cn=users,ou=group,dc=ucla,dc=edu,dc=ve  
objectClass: posixGroup  
objectClass: top  
cn: users  
gidNumber: 100  
  
# marquezl, people, ucla.edu.ve  
dn: uid=marquezl,ou=people,dc=ucla,dc=edu,dc=ve  
objectClass: top  
objectClass: person  
objectClass: organizationalPerson  
objectClass: inetOrgPerson  
cn:: THVpcyBNw6FycXVleg==  
sn: marquezl  
uid: marquezl  
telephoneNumber: 04164716421  
  
# search result  
search: 2  
result: 0 Success  
  
# numResponses: 7  
# numEntries: 6  
root@luke:~# █
```



## Para eliminar un atributo

- Se debe crear un **ldif** como el que se muestra a continuación, el nombre del archivo **ldif** es del.ldif

```
dn: uid=marquezl,ou=people,dc=ucla,dc=edu,dc=ve
```

```
changetype: modify
```

```
delete: telephonenumber
```

- Se ejecuta la siguiente sintaxis

```
root@luke: ~  
File Edit View Terminal Tabs Help  
root@luke:~# vi del.ldif  
root@luke:~# ldapmodify -x -D "cn=admin,dc=ucla,dc=edu,dc=ve" -w -f del.ldif  
Enter LDAP Password:  
modifying entry "uid=marquezl,ou=people,dc=ucla,dc=edu,dc=ve"  
root@luke:~# █
```



- Para verificar el resultado ejecutamos la siguiente sintaxis

```
root@luke: ~  
File Edit View Terminal Tabs Help  
# users, group, ucla.edu.ve  
dn: cn=users,ou=group,dc=ucla,dc=edu,dc=ve  
objectClass: posixGroup  
objectClass: top  
cn: users  
gidNumber: 100  
  
# marquezl, people, ucla.edu.ve  
dn: uid=marquezl,ou=people,dc=ucla,dc=edu,dc=ve  
objectClass: top  
objectClass: person  
objectClass: organizationalPerson  
objectClass: inetOrgPerson  
cn:: THVpcyBNw6FycXVleg==  
sn: marquezl  
uid: marquezl  
  
# search result  
search: 2  
result: 0 Success  
  
# numResponses: 7  
# numEntries: 6  
root@luke:~#
```

## Manejo de contraseña

- Primero se debe construir un password, para ello utilizaremos el comando **slappasswd**, el resultado lo insertaremos en un archivo del tipo **ldif** para modificar el atributo password.

```
#slappasswd -h {CRYPT}
```

- Ldif

```
dn: uid=marquezl,ou=people,dc=ucla,dc=edu,dc=ve
```

```
changetype: modify
```

```
replace: userPassword
```

```
userPassword: {CRYPT}Mfvpc3Qt50kmQ
```





- La sintaxis se muestra a continuación

```
root@luke: ~  
File Edit View Terminal Tabs Help  
root@luke:~# slappasswd -h {CRYPT}  
New password:  
Re-enter new password:  
{CRYPT}Mfvpc3Qt50kmQ  
root@luke:~# vi cambio_password.ldif  
root@luke:~# ldapmodify -x -D "cn=admin,dc=ucla,dc=edu,dc=ve" -W -f change_password.ldif  
Enter LDAP Password:  
modifying entry "uid=marquezl,ou=people,dc=ucla,dc=edu,dc=ve"  
  
root@luke:~# █
```



## Administración de grupos

En ldap podemos hacer un manejo de los grupos de usuarios, tal cual se puede hacer en el sistema operativo \*nix, así que el mecanismo para su manejo es el mismo que los grupos de sistemas. Debemos crear un grupo y luego asociar usuarios a los mismos.

Crear los grupos se recomienda por razones de estándares. Crear un unidad organizativa OU, y debajo de ella colocar todos los grupos. Esta unidad es llamada “Groups”, así que luego de crearla creamos debajo de ella todo los grupos.

Para crear los grupos debemos asociarle a los mismo la clase de objeto “posixGroup”, el cual me indica que este es un grupo de sistema operativo, la clase de objeto “posixGroup” consta de tres atributos que son los siguientes :

Nombre del Atributo	Requerido	Comentario
cn	Si	Nombre del Grupo
gidNumber	Si	Identificador numérico único del Grupo
memberUid	No	Uid de los usuarios pertenecientes a este grupo.



Para la creación de grupos se puede crear un archivo LDIF, como el siguiente :

```
dn: cn=administradores,ou=groups,dc=universidad,dc=edu,dc=ve
cn: administradores
gidNumber: 10100
memberUid: aperez
memberUid: zrodriguez
objectClass: top
objectClass: posixGroup
```

Aquí se crea un grupo llamado administradores, en el cual el gidNumber es 10100 (debe ser un número distinto para cada grupo), y los miembros de ese grupo son los usuarios que poseen el Uid aperez y zrodriguez.

Para agregar o eliminar usuarios de los grupos, basta con editar el registro del grupo y agregar o eliminar usuarios utilizando el atributo memberUid.

## **Autenticación de Clientes en ldap**

Una vez configurado el servidor de LDAP para almacenar la información del directorio, podemos configurar todos los equipos de nuestra red (servidores y clientes) para realizar la autenticación en el servidor LDAP.

En principio, la información administrativa que tiene sentido centralizar en un servicio LDAP son las cuentas de usuario (incluyendo contraseñas) y cuentas de grupo. En conjunto, la información almacenada en ambos tipos de cuentas permite autenticar a un usuario cuando éste desea iniciar una sesión interactiva en un sistema Linux y, en el caso de que la autenticación sea positiva, crear el contexto de trabajo inicial (es decir, el proceso *shell* inicial) para ese



usuario. Manteniendo ambos tipos de cuentas en el directorio permitiría una gestión completamente centralizada de los usuarios del dominio.

Internamente, este proceso de autenticación y creación del contexto inicial que Linux lleva a cabo cuando un usuario desea iniciar una sesión interactiva utiliza dos bibliotecas distintas:

1. **PAM** (*Pluggable Authentication Module*) es una biblioteca de autenticación genérica que cualquier aplicación puede utilizar para validar usuarios, utilizando por debajo múltiples esquemas de autenticación alternativos (archivos locales, Kerberos, LDAP, etc.). Esta biblioteca es utilizada por el proceso de "login" para averiguar si las credenciales tecleadas por el usuario (nombre y contraseña) son correctas.
2. **NSS** (*Name Service Switch*) presenta una interfaz genérica para averiguar los parámetros de una cuenta (como su UID, GID, *shell* inicial, directorio de conexión, etc.), y es utilizada por el proceso de "login" para crear el proceso de atención inicial del usuario.

La ventaja fundamental de ambas bibliotecas consiste en que pueden reconfigurarse dinámicamente mediante archivos, sin necesidad de recompilar las aplicaciones que las utilizan. Por tanto, lo único que necesitamos es reconfigurar ambas para que utilicen el servidor LDAP además de los archivos locales (/etc/passwd, entre otros.) de cada equipo.

En Debian GNU/Linux la instalación y configuración de los clientes la realizaremos directamente en los archivos de configuración. Cuando el asistente de Debian pregunte la configuración cancelar el mismo. Realizar los pasos siguientes:



1. Instalar el paquete **libnss-ldap**, mediante la ejecución en la consola de **#aptitude install libnss-ldap**, el “Name Service Switch” permite a los sistemas operativos \*nix, el reemplazo de los archivos de configuración de los usuarios (por ejemplo: /etc/passwd, /etc/group), por bases de datos de usuarios centralizadas, en este caso se instala la librería con soporte para LDAP, que será el sistema que utilizaremos.
2. Instalar el paquete **libpam-ldap**, mediante la ejecución en la consola de **#aptitude install libpam-ldap**, este permitirá la autenticación de los usuarios del sistema operativo en base de datos ldap.
3. Instalar el paquete **nscd** , el cual permitirá mantener en el cache del equipo las búsquedas de los registros que realiza el nss, con el fin de evitar tener que realizar esas consultas a los servidores y ahorrar tiempo y tráfico en la red.

#### Configuración de clientes:

1. Editar el archivo **/etc/libnss-ldap.conf**, y se configurará los siguiente :
  1. **base dc=universidad,dc=edu,dc=ve**, esta es la base de búsqueda de nuestro directorio LDAP.
  2. **ldap\_version 3** , con lo cual nuestros clientes utilizarán la versión 3 de LDAP.
  3. **bind\_policy soft**, esto nos permitirá que cuando el servidor LDAP no esté disponible, poder continuar con el proceso de autenticación del sistema utilizando los archivos locales.



4. **ssl start\_tls** , el servidor intentará hacer la conexión utilizando tls.
  5. **tls\_checkpeer no**, esta opción no chequeará el certificado del servidor (útil cuando se utilizan certificados generados localmente).
2. Editar el archivo **/etc/nsswitch.conf**, y se configurará lo siguiente:
    1. **passwd files ldap**, con lo cual la información de los usuarios se buscará primero en el archivo **/etc/passwd** y luego en el servidor LDAP.
    2. **group files ldap** , con lo cual la información de los grupos se buscará primero en el archivo **/etc/group** y luego en el servidor LDAP.
3. Editar el archivo **/etc/pam\_ldap.conf**, y se configurará lo siguiente:
    1. **base dc=universidad,dc=edu,dc=ve**, esta es la base de búsqueda de nuestro directorio LDAP.
    2. **ldap\_versión 3** , con lo cual nuestros clientes utilizarán la versión 3 de LDAP.
    3. **bind\_policy soft**, esto nos permitirá que cuando el servidor LDAP no esté disponible poder continuar con el proceso de autenticación del sistema utilizando los archivos locales.
    4. **ssl start\_tls** , el servidor intentará hacer la conexión utilizando tls.
    5. **tls\_checkpeer no**, esta opción no chequeará el certificado del servidor (útil cuando se utilizan certificados generados localmente).
4. Editar el archivo **/etc/pam.d/common-account**, y configurar lo siguiente:
    1. account sufficient pam\_unix.so nullok\_secure
    2. account sufficient pam\_ldap.so
    3. account required pam\_permit.so
5. Editar el archivo **/etc/pam.d/common-auth**, y configurar lo siguiente:
    1. auth sufficient pam\_ldap.so



2. `auth required pam_unix.so nullok_secure use_first_pass`
6. Editar el archivo **`/etc/pam.d/common-password`**, y configurar lo siguiente:
  1. `password sufficient pam_ldap.so`
  2. `password required pam_unix.so nullok obscure min=4 max=8 md5`
7. Editar el archivo **`/etc/pam.d/common-session`**, y configurar lo siguiente:
  1. `session optional pam_foreground.so`
  2. `session sufficient pam_ldap.so`
  3. `session required pam_unix.so`
  4. `session required pam_mkhomedir.so skel=/etc/skel/`

Probar la configuración mediante los comandos :

**`#getenv passwd o getenv group`**, esto debe mostrar los usuarios que estan en el servidor LDAP.

## Listas de acceso en LDAP

La base de datos LDAP, contiene información sensible, por ejemplo el atributo `userPassword` contiene las contraseñas de los usuarios, pero también existe otro tipo de información como datos personales de las personas que deben ser resguardados.

Para controlar la autorización en los servidores LDAP se utilizan ACLs (Lists de Control de Acceso), cuando un servidor LDAP procesa un requerimiento de un cliente evalúa los permisos de acceso del mismo a la información solicitada. Esta evaluación verifica secuencialmente cada una de las ACLs, ubicadas en los archivos de configuración y aplica las reglas apropiadas al



requerimiento.

La configuración de las ACLs, se pueden realizar de dos maneras:

1. Directamente en el archivo de configuración del servidor LDAP, **/etc/ldap/slapd.conf**, y pueden ser colocadas al principio del archivo con lo cual afectaran a todas las bases de datos que posee el servidor.
2. Dentro de la directiva “**backend**”, con lo cual solo afectará a la base de datos especifica.

Cuando se poseen muchas reglas de acceso, es recomendable colocar en un archivo aparte y utilizar la sintaxis **include /etc/ldap/nombredearchivo**, con lo cual se mantendrá el archivo **slapd.conf** menos complejo.

Las directivas de acceso tienen la siguiente sintaxis :

**access to** [recurso]

**by** [quien] [tipo de privilegio]

**by** [quien] [tipo de privilegio]

Las directivas access pueden tener uno o mas “**by**”, así mismo pueden permitir accesos por DN, atributos, filtros, o una combinación de estos.

## Accesos utilizando DN

Para restringir un acceso a un DN en particular, se debe utilizar una regla como la siguiente:

```
access to dn="uid=pedro,ou=Users,dc=universidad,dc=edu,dc=ve"  
by * none
```

El “**by \* none**”, rechaza los accesos a todos.

Las restricciones a los DN, pueden ser especificadas de la siguiente manera :

- **dn.base** : Restringe el acceso para un DN específico, es la opción por





defecto,

- **dn.exact** y **dn.baselevel** : son sinónimos de **dn.base**.
- **dn.one** : Restringe el acceso a la siguiente entrada que este después del DN especificado.
- **dn.subtree** : Restringe el acceso a todo el árbol debajo del DN especificado.

Las ACLs, también aceptan expresiones regulares lo cual incrementa el nivel de complejidad que se puede utilizar para formular las mismas.

A continuación un ejemplo utilizando expresiones regulares :

```
access to dn.regex="uid=[^,]+,ou=Users,dc=universidad,dc=edu,dc=ve"  
by * none
```

En el ejemplo anterior se restringe el acceso a cualquier DN, con la expresión "uid=cualquier cosa"ou=Users,dc=universidad,dc=edu,dc=ve, donde cualquier cosa debe ser un texto con al menos un carácter y sin comas (,), las expresiones regulares permiten incrementar en gran medida la utilidad de las listas de acceso.

## Conexiones Seguras

Lo primero que se debe evaluar es la seguridad de la red. Los clientes se conectan al servidor LDAP a través de las interfaces de red, y también las respuestas del servidor se transfieren a través de la red.

El protocolo LDAP por defecto recibe y envía los datos en texto plano, lo cual tiene algunas ventajas entre las cuáles tenemos :

- Facilidad de configurar y mantener.
- El servicio funciona más rápido, al no tener que transformar los datos cifrados, lo cual siempre provee de una carga adicional de procesamiento.



Estas ventajas tienen un costo de seguridad, otros dispositivos en la red pueden interceptar los datos y leer todo el contenido de los mismos, mientras más grande es una red esto se convierte en una amenaza mayor.

Para evitar eso los servidores LDAP implementan SSL (Secure Sockets Layer) y TLS (Transport Layer Security), ambos mecanismos son utilizados para cifrar los datos antes de transmitirlos por la red. SSL y TLS son similares y son ampliamente utilizados, la principal diferencia es que TLS es más flexible que SSL.

OpenLDAP provee dos mecanismos para cifrar el tráfico en la red, el primero es escuchar por un puerto específico (puerto 636 por defecto), lo que hace que las comunicaciones en ese puerto sean cifradas, este mecanismo fue introducido en LDAP v2, y se considera un método en desuso. El segundo mecanismo es parte de los estándares de LDAP v3, el cual permite a los clientes conectarse a través de un puerto (389 por defecto), para conexiones cifradas o en texto plano y será el cliente el que seleccionará el tipo de conexión que desea. El uso de certificados permite no solo cifrar la información entre el servidor y los clientes, sino también garantizar que el servidor al cual se conecta el cliente es auténtico.

Actualmente existen autoridades para emitir certificados conocidos como CA (Certification Authority), los cuales a través de un procedimiento de recolección de información y un pago, emiten un certificado que tiene validez por un tiempo específico y los servidores y los clientes reconocen el mismo.

Existe también la posibilidad de crear los certificados para uso de las organizaciones o individuos de manera interna, para esto se debe generar un CA con la cual se firmarán los certificados que se emitirán para los clientes y



servidores. Estos certificados no serán reconocidos como válidos fuera de la organización que los emite, por lo cual solo se recomienda para uso interno.

Para crear una CA, con el fin de firmar nuestros propios certificados, se debe instalar el paquete **openssl**, el cual en Debian GNU/Linux, instala un script que permite la creación de un CA, el mismo queda instalado en la ruta **/usr/lib/ssl/misc/CA.pl**, debe ejecutarse de la siguiente manera :

**#!/usr/lib/ssl/misc/CA.pl -newca**

1. Se mostrará un mensaje con lo siguiente, **CA certificate filename (or enter to create)** , en donde hay que pulsar enter.
2. Luego de generar la clave del certificado preguntará por una contraseña para el mismo, **Enter PEM pass phrase:**
3. Luego solicitará una serie de información sobre la organización, y finalmente terminará de crear el certificado para la CA.
4. Al finalizar tendremos un directorio llamado demoCA, con los certificados.

Luego de esto se tendrá que generar los certificados para el servidor LDAP, lo cual será de la siguiente manera:

1. Ejecutar **/usr/lib/ssl/misc/CA.pl -newreq** desde una consola luego se repetirán los pasos 2 y 3 de la creación de la CA.
2. Al finalizar tendremos dos archivos llamados newkey.pem (Clave Privada) newreq.pem (Certificado).
3. Luego tenemos que firmar los certificados con el CA generado anteriormente, para esto ejecutamos desde la consola **/usr/lib/ssl/misc/CA.pl -signreq**, preguntará la contraseña de la CA y luego tendremos dos archivos, **newkey.pem**, el cual contiene la clave privada y **newcert.pem** el cual contiene el certificado firmado.



4. Se remueve la contraseña que tiene el certificado ejecutando desde la consola el comando **#openssl rsa < newkey.pem > clearkey.pem**
5. Luego de esto copiamos los archivos **clearkey.pem** y **newkey.pem** a un directorio por conveniencia por ejemplo `/etc/ldap/ssl`.
6. Instalar el certificado de la CA, para hacer esto copiamos el archivo **cacert.pem** a `/usr/share/ca-certificates/miCA.crt`, editar el archivo `/etc/ca-certificates.conf`, y colocar al final del archivo **miCA.crt**, y ejecutar desde la consola **#update-ca-certificates**.
7. Agregar en el archivo de configuración de LDAP, para la utilización de los certificados y que el mismo corra en modo SSL/TLS, la configuración quedaría de la siguiente manera :
  1. **TLSCACertificatePath** `/etc/ssl/certs/`
  2. **TLSCertificateFile** `/etc/ldap/clearkey.pem`
  3. **TLSCertificateKeyFile** `/etc/ldap/newkey.pem`

Luego de esto el servidor LDAP estará ejecutándose con soporte SSL/TLS.

## Múltiples Directorios Réplicas y Cache

En entornos de grandes redes y servidores LDAP con grandes base de datos, se requiere mantener más de un servidor LDAP, y esto se realiza mediante el demonio slurpd.

El acrónimo **slurpd** significa: Standalone LDAP Update Replication Daemon y su misión es propagar los cambios de una base de datos slapd hacia otra. Si slapd está configurado para producir logs de replicación, slurpd los lee y envía los cambios a las instancias slapd esclavas a través del protocolo LDAP. slurpd se arranca, normalmente, en el arranque del sistema.



Una vez arrancado, slurpd normalmente hace un fork de si mismo y se independiza de la tty que lo ha llamado, luego lee el log de replicación (dado bien por la directiva relogfile del archivo de configuración de slapd, ó bien por la opción -r de la línea de comandos). Si el archivo log de replicación no existe o está vacío, slurpd se duerme. Después, cada cierto tiempo, se despierta y verifica si hay cambios que propagar.

Cuando slurpd encuentra cambios a propagar hacia las instancias slapd esclavas, bloquea el log de replicación, hace una copia privada del mismo, libera el bloqueo anteriormente puesto y crea un fork de si mismo para réplica de slapd que ha de ser actualizada. Cada proceso hijo se asocia con el demonio slapd esclavo, y envía los cambios.

El funcionamiento es el siguiente :

- El cliente LDAP envía una modificación LDAP al slapd esclavo.
- El slapd esclavo devuelve una remisión hacia el cliente LDAP, referenciándolo hacia el servidor slapd maestro.
- El cliente LDAP envía la operación de modificación LDAP hacia el slapd maestro.
- El slapd maestro realiza la operación de modificación, escribe los cambios en su archivo log de replicación y devuelve un código de éxito hacia el cliente.
- El proceso slurpd verifica que se ha añadido una nueva entrada al archivo log de replicación, lee la entrada del log de replicación y envía el cambio hacia el servidor slapd esclavo vía LDAP.
- El servidor slapd esclavo realiza la operación de modificación y un código de éxito hacia el proceso slurpd.



## Configuración de servidores esclavos LDAP

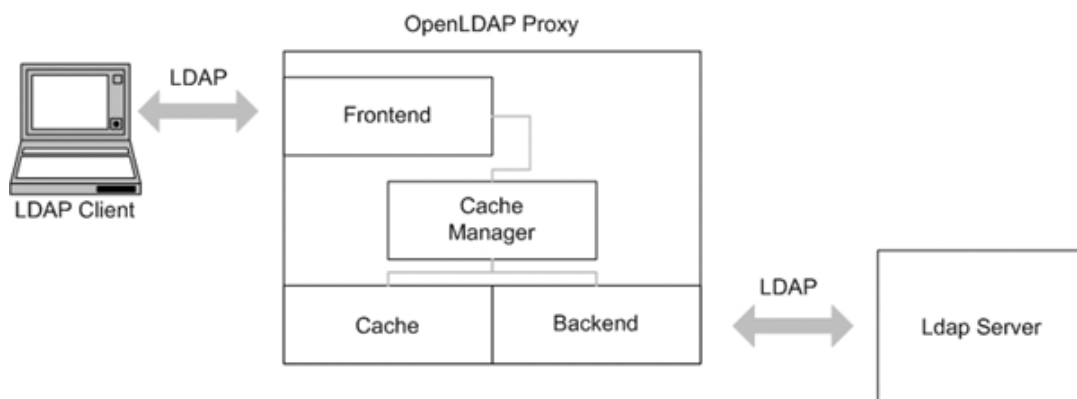
La configuración de servidores ldap esclavos se realiza de la siguiente manera :

1. En el archivo de configuración del servidor maestro se colocará lo siguiente:
  1. **replica uri=ldap://esclavo.universidad.edu.ve:389  
binddn="cn=replicador,dc=universidad,dc=edu,dc=ve"  
bindmethod=simple credentials=replicador**
  2. **repllogfile /var/lib/ldap/repllog**
2. En el archivo de configuración del servidor esclavo, lo siguiente:
  1. **updatedn "cn=replicador,dc=universidad,dc=edu,dc=ve"**
  2. **updateref "<ldap://maestro.universidad.edu.ve>"**

## ProxyLdap

Un servidor proxy ldap actúa como un intermediario entre los clientes y los servidores que poseen los datos, y son de gran utilidad cuando se tienen gran cantidad de servidores y se requiere agrupar en uno solo.

### Configuración de un servidor ProxyLdap



Ejemplo de configuración de servidor proxyldap

Editar el archivo `/etc/ldap/slapd.conf` y colocar lo siguiente:

1. **database meta**, el tipo de base de datos a utilizar.
2. **suffix "dc=reacciun,dc=ve"**, la base del directorio proxy.
3. **uri "<ldap://ldap.universidad.edu.ve/dc=universidad,dc=reacciun,dc=ve>"**
4. **suffixmassage "dc=universidad,dc=reacciun,dc=ve"**  
**"dc=universidad,dc=edu,dc=ve"**