Administración básica del directorio:

Ya tenemos el servidor LDAP funcionando y escuchando en el puerto 389 TCP. Ahora deberíamos generar la estructura de entradas de nuestro directorio y rellenarlas de datos. La forma más básica de añadir información a un directorio cualquiera es utilizar ficheros de texto cuyo contenido está escrito en el formato LDIF (LDAP Data Interchange Format). El formato básico de una entrada es:

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

Entre dos entradas consecutivas debe existir siempre una línea en blanco. Por otro lado, 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.

Una vez creados estos ficheros, para añadirlos al directorio (incluso con el servidor en marcha) podemos utilizar el comando *ldapadd* (disponible al instalar el paquete llamado “ldap-utils” -en Ubuntu-, o "openldap-clients" -en Fedora- aunque en el caso de haber instalado el paquete "389-ds-base" ya se habrá instalado automáticamente como dependencia). La mayoría de veces necesitaremos indicar los siguientes parámetros:

-H ldap://nomDnsServidor:nºpuerto : Indica el servidor LDAP (y, opcionalmente, el número de puerto) contra el cual se van a ejecutar. Si el servidor LDAP estuviera funcionando en la misma máquina donde se ejecuta la comanda *ldapadd* (o cualquier otra de su familia: *ldapsearch, ldapmodify*, etc), normalmente este parámetro se podrá omitir.

> NOTA: Si el comando cliente (cualquiera de los que estudiaremos: *ldapadd, ldapsearch, ldapmodify*, etc) se ejecuta en la misma máquina donde está funcionando el servidor, en vez de conectar con éste mediante TCP (que es lo que pasa cuando se usa la url del tipo ldap://...) automáticamente utiliza sockets internos que emplean el mecanismo IPC, que es un sistema de intercomunicación entre procesos locales más óptimo para estos casos. Aunque ya hemos dicho que no sería necesario, si se quisieraespecificar la url, en este caso entonces se debería escribir así: "ldapi://%2fruta%2fcarpeta%2ffichero.socket" (notar que el protocolo es "ldapi" y no "ldap" y que las "/" se sustituyen por "%2f").

-D cn=admin : Indica el "common name" de la cuenta de usuario (guardada en el propio servidor) con la que nos autenticaremos en el servidor LDAP para realizar la modificación del directorio; esta cuenta ha de tener privilegio para ello, así que usaremos la cuenta "admin" que creamos en la instalación del servidor.
-W : Solicita interactivamente la contraseña de la cuenta anterior. Otra opción sería indicar la contraseña como un parámetro más, así : *-w contraseña* (notar que en este caso la "w" es minúscula).

-f fichero.ldif : Indica el fichero cuyo contenido se desea agregar

1.- a) Crea un fichero llamado “base.ldif” con el contenido mostrado a continuación y seguidamente agrégalo al directorio con el comando: ***ldapadd -D cn=admin -W -f base.ldif*** . Con esto habrás generado dos entradas de tipo “unidad organizativas” que servirán para contener (a modo de “carpetas”) los usuarios y grupos que generaremos a continuación.

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

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

b) Crea un fichero llamado “grupos.ldif” con el siguiente contenido y seguidamente agrégalo al directorio con un comando similar al del apartado anterior:

```
dn: cn=grupoldap,ou=grupos,dc=midominio,dc=local
objectClass: posixGroup
cn: grupoldap
gidNumber: 10000
```

c) Crea un fichero llamado “usuarios.ldif” con el siguiente contenido y seguidamente agrégalo al directorio con un comando similar al del apartado anterior:

> NOTA: Tal como se puede ver, cada objeto “usuario” está formado a partir de la unión de diferentes tipos predefinidos de objeto (posixAccount, shadowAccount, inetOgPerson), donde cada uno aporta un determinado conjunto de atributos: posixAccount incluye la información que encontraríamos en el archivo /etc/passwd clásico, shadowAccount incluye la información que encontraríamos en el archivo /etc/shadow clásico y inetOrgPerson incluye información extra del usuario dentro de la organización (como el correo, cód. Postal...).
>
> NOTA: Hay que tener muy en cuenta que al añadir nuevos usuarios los valores de los atributos uidNumber y homeDirectory (además de userPassword) deben ser diferentes para cada usuario. Lo mismo ocurre con el atributo gidNumber para los grupos. Además, los valores de uidNumber y gidNumber no deben coincidir con el uid y gid de ningún usuario y grupo local de los clientes.

```
dn: uid=usu1ldap,ou=usuarios,dc=midominio,dc=local
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: shadowAccount
uid: usu1ldap
sn: Lopez
givenName: Juan
cn: Juan Lopez
displayName: Juan Lopez
uidNumber: 3000
gidNumber: 10000
userPassword: XXX
gecos: Es muy tonto
loginShell: /bin/bash
homeDirectory: /home/jlopez
shadowExpire: -1
shadowFlag: 0
shadowWarning: 7
shadowMin: 8
shadowMax: 999999
shadowLastChange: 10877
mail: juan.lopez@gmail.com
postalCode: 29000
####################LINEA EN BLANCO#######################
dn: uid=usu2ldap,ou=usuarios,dc=midominio,dc=local
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: shadowAccount
uid: usu2ldap
sn: Perez
givenName: Perico
cn: Pedro Perez
displayName: Pedro Perez
uidNumber: 3001
gidNumber: 10000
userPassword: XXX
gecos: Es un crack
loginShell: /bin/bash
homeDirectory: /home/pperez
shadowExpire: -1
shadowFlag: 0
shadowWarning: 7
shadowMin: 8
shadowMax: 999999
```

shadowLastChange: 10877
mail: pedrito@yahoo.es
postalCode: 29001

Para comprobar si el contenido anterior se ha añadido correctamente, podemos usar el comando *ldapsearch* (también del paquete “ldap-utils”), el cual permite hacer una búsqueda en el directorio. La mayoría de veces necesitaremos indicar, además de los parámetros *-H ldap://nomDnsServidor:nºpuerto, -D cn=admin* y *-W/-w contraseña* ya conocidos, otros como los siguientes:

-LLL : Indica a *ldapsearch* que muestre la respuesta en modo "no verboso" (más fácil de leer)

-b "dc=midominio,dc=local" : Indica el DN "base" a partir del cual se empezará la búsqueda por las entradas inferiores del árbol. El ejemplo anterior buscaría a partir de la entrada raíz "para abajo" por todas las subentradas. Pero no es necesario que la entrada raíz sea el DN "base": si, por ejemplo, escribiéramos *-b "ou=grupos,dc=midominio,dc=local"* entonces solo se buscaría a partir de la organizationalUnit "grupos" "para abajo" por las subentradas que contuviera.

> NOTA: En el servidor 389DS, l'entrada "cn=config" pot fer-se servir com a DN base per obtenir tota la configuració del propi servidor

-s {base | one | sub } : El comportamiento descrito en el párrafo anterior de buscar recursivamente todas las entradas por debajo de una entrada "base" dada (incluyendo ésta) es el comportamiento por defecto, que equivaldría a indicar el parámetro *-s sub* Pero otros valores de este parámetro modifican el "scope" de la búsqueda; por ejemplo, *-s base* solamente muestra información de la entrada marcada como "base" y nada más (es decir, no recorre ninguna rama del árbol); por su parte, *-s one* solamente muestra información de las entradas directamente colgando de la entrada "base" pero nada más (es decir, sólo muestra las entradas "hijas": ni muestra entradas "nietas" ni la propia entrada "base")

"(atributo=valor)" "(atributo=valor)" ... **:** Filtro que permite "quedarse", de todas las entradas recorridas, solo con las que contienen el atributo indicado con el valor indicado. Existen otros filtros más sofisticados que se pueden estudiar en el cuadro inferior. Si aparece este parámetro, ha de ir escrito después de cualquier otro parámetro de los anteriores. Si no aparece, todas las entradas recorridas serán "válidas" para mostrarse.

> NOTA: * is a special value representing “any possible value”. It can be used to build a "presence" filter that requests all objects where the attribute is present and has a valid value, but we do not care what the value is. For instance, by default, *ldapsearch* provides the filter "(objectClass=*)"; because all objects must have an objectClass, this filter is the equivalent to saying “all valid objects”.

atributo atributo ... : De la información encontrada en las entradas recorridas (y ya filtradas, si fuera el caso), se muestra solo aquellos atributos indicados. Si aparece este parámetro, ha de ir escrito después del filtro (si hubiera). Si no aparece -o se escribe "*"-, se mostrarán todos los atributos pertenecientes a las entradas recorridas (y ya filtradas si fuera el caso); en el caso de escribir "+" se mostrarán los metaatributos de la entradas recorridas (como la fecha de creación de la entrada, el usuario que la creó, etc).

> NOTA: To show what basedns are available, you can query the special ‘’ or blank rootDSE (Directory Server Entry) using "namingContexts" as the attribute to look up, like this:
> *ldapsearch -D cn=admin -W -LLL -b "" -s base namingContexts*

2.- Ejecuta el comando ***ldapsearch -D cn=admin -W -LLL -b "dc=midominio,dc=local" uid=usu1ldap sn givenName*** Con este comando de ejemplo estaremos buscando un usuario con uid=usu1ldap y pediremos que nos muestre solo el contenido de los atributos sn y givenName. Comprueba que efectivamente sea así:

> NOTA: Es posible realizar consultas anónimas (es decir, sin necesidad de autenticarse) pero para ello sería necesario configurar el servidor 389DS convenientemente (ver para más información) y sustituir los parámetros *-D* y *-W* de *ldapsearch* por *-x*

*A filter can request objects whose attribute values are greater/less than a value by *"(uid>=test0005)"* or *"(uid<=test0005)"*

A filter can request a partial match of an attribute value on the object by using the '' operator (multiple times if necessary): *"(uid=*005)"* or *"(uid=*st000*)"*. Note you should always have at least 3 characters in your substring filter, else indexes may not operate efficently.

*Filters can be nested with AND (&) or OR (|) conditions. The condition applies to all filters that follow within the same level of brackets, that is: (condition (attribute=value)(attribute=value)). AND requires that for an object to match, all filter elements must match; this is the "intersection" operation and is written like this: *"(&(uid=test0006)(uid=guest0006))"* On the other hand, OR filters will return the aggregate of all filters; this is the union operation so provided an object satisfies one condition of the OR, it will be part of the returned set; it's written like this: *"(|(uid=test0006)(uid=guest0007))"*

*A NOT filter acts to invert the result of the inner set. For example: *"(!(uid=test0010))"* Note you can't list multiple parameters in a "not" condition: to combine NOT's you need to use this in conjunctionwith AND and OR.

*You can nest AND, OR and NOT filters to produce more complex directed queries. For instance this query:
"(&(objectClass=person)(objectClass=posixAccount)(|(uid=test000))(!(uid=test0001)))"* would be equivalent to...:

```
(&
   (objectClass=person)
   (objectClass=posixAccount)
   (|
      (uid=test000*)
   )
   (!(uid=test0001))
)
```

...and it expresses "All person whose name starts with test000* and not test0001". Another example would be this:
"*(|(& (...K1...) (...K2...))(& (...K3...) (...K4...)))*", which means "(K1 AND K2) OR (K3 AND K4)"

Otros comandos importantes del paquete "ldap-util" son *ldapdelete* y *ldapmodify.* Un ejemplo del primero podría ser: ***ldapdelete -D cn=admin -W "uid=usu2ldap,ou=usuarios,dc=midominio,dc=local"***. El segundo tiene tres formas de modificar una entrada: cambiando el valor de un atributo , añadiendo un nuevo atributo o eliminando un atributo existente:

1.-Para cambiar, por ejemplo, el atributo *uidNumber* de un usuario, podríamos ejecutar el comando ***ldapmodify -D cn=admin -W -f fichero.cambios*** donde "fichero.cambios" debería tener un contenido como el siguiente (donde se especifica qué entradas se quieren modificar y de qué manera):

```
dn:uid=usu2ldap,ou=usuarios,dc=midominio,dc=local
changetype:modify
replace:uidNumber
uidNumber:3002
```

NOTA: La información anterior la podríamos haber introducido directamente desde la entrada estándar si no hubiéramos especificado el parámetro -f.

2.-Para añadir un atributo nuevo (en este caso llamado *jpegPhoto*) deberíamos ejecutar el mismo comando *ldapmodify* anterior pero ahora "fichero.cambios" debería tener un contenido como este:

```
dn:uid=usu2ldap,ou=usuarios,dc=midominio,dc=local
changetype: modify
add:jpegPhoto
jpegPhoto:file:///tmp/foto.png
```

3.-Para eliminar un atributo (en este caso llamado *jpegPhoto*) deberíamos ejecutar el mismo comando *ldapmodify* anterior pero ahora "fichero.cambios" debería tener un contenido como este:

```
dn:uid=usu2ldap,ou=usuarios,dc=midominio,dc=local
changetype: modify
delete:jpegPhoto
```

NOTA: Ja que la configuració del servidor 389DS es troba accessible en forma directori, la comanda *ldapmodify* es podria utilitzar per modificar aquesta configuració "en calent". Per exemple, sabent que les directives "nsslapd-ldapilisten" i "nsslapd-ldapifilepath" activen el mecanisme LDAPI i especifiquen la ruta del socket adient, respectivament, per activar aquesta característica (que ja ve de sèrie activada, però és només un exemple), podríem fer com sempre: *ldapmodify -D cn=admin -W -f fichero.cambios* pero ara el contingut del fitxer "fichero.cambios" seria:

```
dn: cn=config
changetype: modify
replace: nsslapd-ldapilisten
nsslapd-ldapilisten: on
-
add: nsslapd-ldapifilepath
nsslapd-ldapifilepath: /var/run/slapd-exemple.socket
```

NOTA: En realitat, per afegir entrades i per eliminar entrades no caldria fer servir les comandes *ldapadd* i *ldapdelete* respectivament, ja que amb *ldapmodify* ja n'hi hauria prou. En el cas de voler afegir una entrada el fitxer "fichero.cambios" hauria de tenir un contingut semblant al següent...:

```
dn: <dn to add>
changetype: add
objectclass: ...
attribute1: ...
attribute2: ...
```

...i en el cas de voler eliminar una entrada, el seu contingut hauria de ser semblant a aquest:

```
dn: <dn to delete>
changetype: delete
```

3.-a) Modifica el atributo “gecos” del usuario Juan López para que muestre la descripción: “Ronca”. Comprueba mediante *ldapsearch* que el cambio lo has realizado correctamente

b) Añade el atributo “jpegPhoto” del usuario Juan López indicando la ruta (ficticia) de su foto identificativa. Comprueba mediante *ldapsearch* que el cambio lo has realizado correctamente

c) Elimina el atributo “jpegPhoto” anterior.Comprueba mediante *ldapsearch* que el cambio lo has realizado correctamente