# 1 Instalación servidor LDAP, ldap-utils: ldapsearch

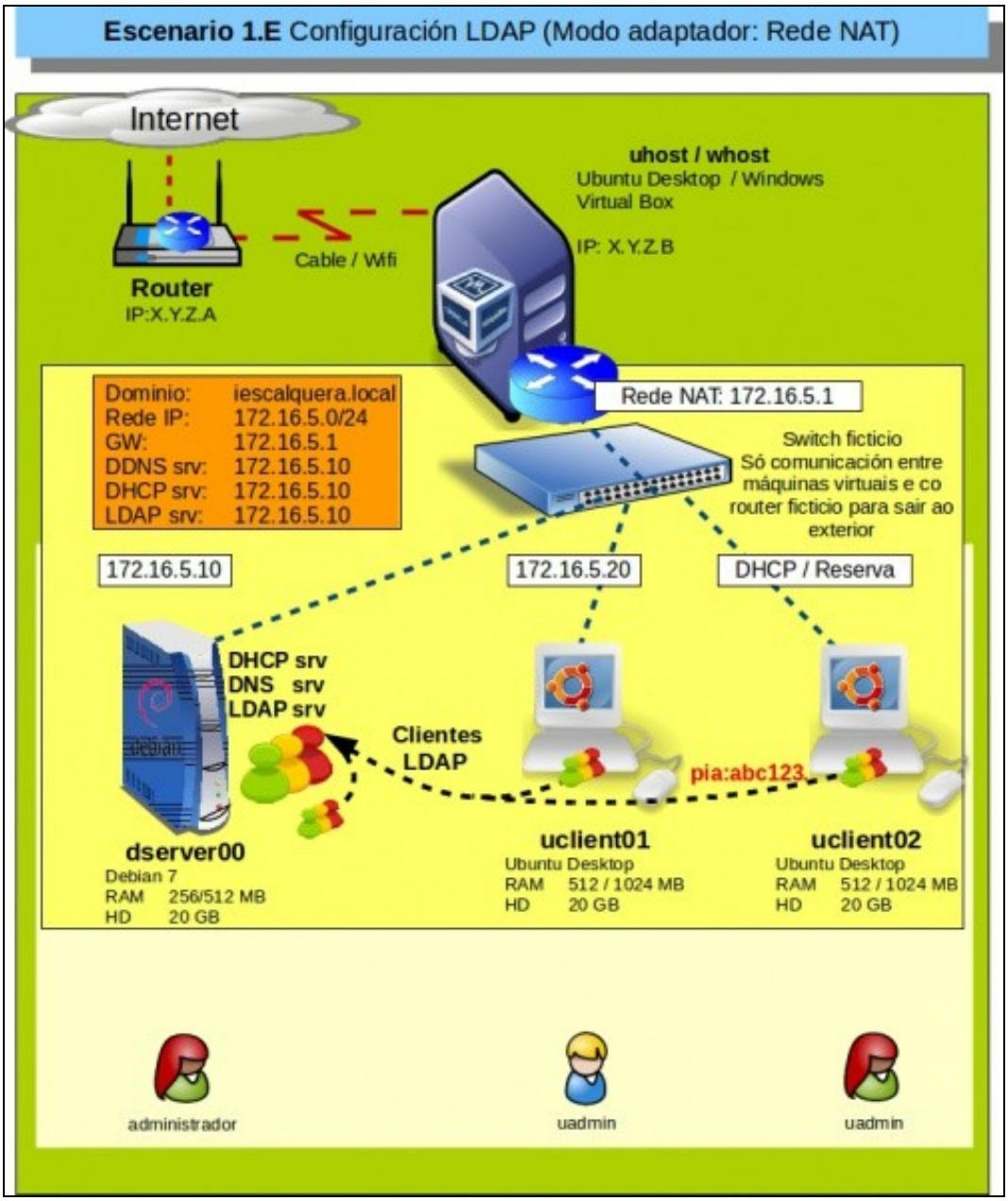
## 1.1 Sumario



## 1.2 Introdución

- Neste apartado aparte de instalar os paquetes necesarios e tamén imos pararnos un pouco en revisar os ficheiros de LDAP para familiarizarnos un pouco máis coa súa estrutura.

- A imaxe amosa o escenario 1.E




- **IMPORTANTE**: para manter a compatibilidade na parte V con SAMBA o nome de dominio non debe exceder os 15 caracteres (Neste caso *iescalquera* ten 11).

### 1.2.1 Instalar os paquetes necesarios

- En primeiro lugar, teremos que instalar no servidor os paquetes necesarios para a execución do servidor LDAP (**slapd (Standalone LDAP Daemon)**) .
- Logo instalaremos as utilidades básicas para manexar a súa información. O comando que usaremos será este:

</source>

### 1.2.2 Agregar os esquemas básicos

- Como xa indicamos na parte introdutoria ao instalar o servizo LDAP en Debian (slapd), este xa instala por defecto os catro esquemas que se indicaron anteriormente: core, COSINE, NIS, inetOrgPerson.
- Pronto o comprobaremos.
- Se non estiveran instalados, ou quixeramos instalar algún outro debemos executar:

```
ldapadd -Y EXTERNAL -H ldapi:/// -f /ruta ao esquema/esquema.ldif
```

- O comando ldapadd instalarémolo máis adiante, este atópase no paquete **ldap-utils** que non instalamos por agora.

- Se observásemos a saída da execución dese comando, veriamos que se produce un erro se intentamos engadir un esquema que xa existe, por exemplo *core*, debido a que xa está engadido:

```
SASL/EXTERNAL authentication started
SASL username: gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth
SASL SSF: 0
adding new entry "cn=core,cn=schema,cn=config"
ldap_add: Other (e.g., implementation specific) error (80)
        additional info: olcAttributeTypes: Duplicate attributeType: "2.5.4.2"
```

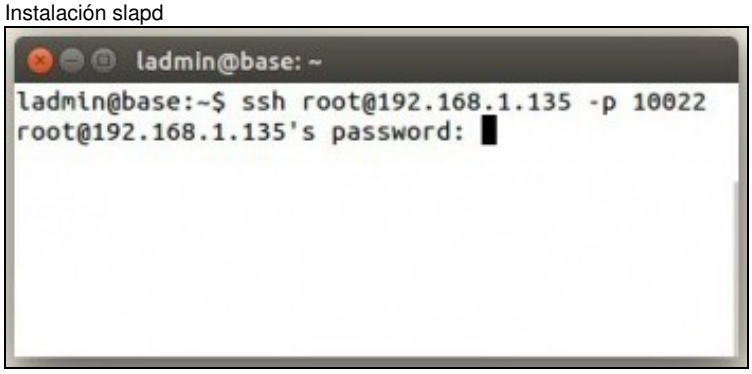
Se se engade un esquema que non existe o resultado sería semellante a, por exemplo o esquema *openldap*:

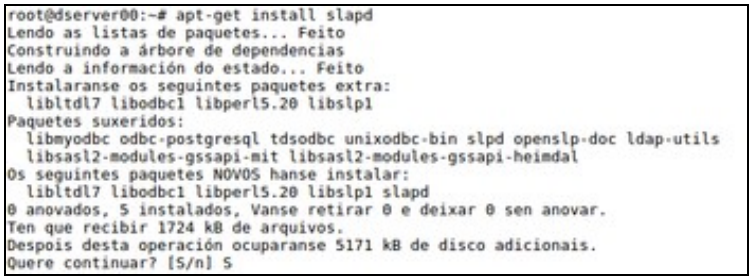
```
SASL/EXTERNAL authentication started
SASL username: gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth
SASL SSF: 0
adding new entry "cn=openldap,cn=schema,cn=config"
```

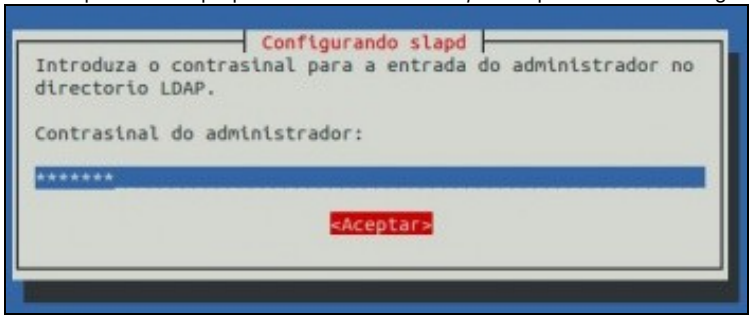
## 1.3 Instalar e configurar slapd

- A continuación imos instalar **slapd**, examinar os ficheiros de ldap e configurar o paquete para o noso dominio **iescalquera.local**.

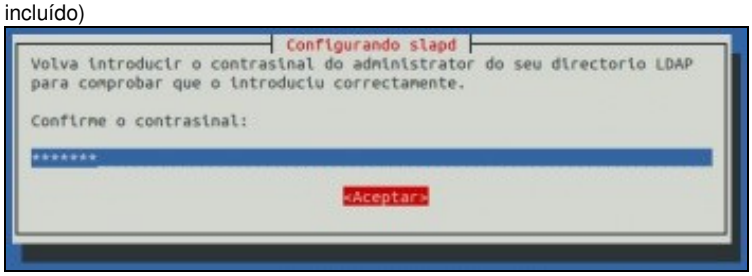
### 1.3.1 Instalación do paquete

- Instalación slapd
- 

Comezamos conectándonos ao servidor **dserver00** dende o exterior.
Lembrar que rediriximos os portos en VirtualBox no escenario 1.A, e estamos conectándonos á IP do host real a un porto que nos redirixe ao servidor **dserver00**
- 

Como xa indicamos executamos **apt-get install slapd**.
Vemos que entre os paquetes suxeridos está *ldap-utils* que instalaremos logo.
- 

Pídenos o contrasinal do usuario **admin**, que é o usuario que vai administrar o directorio de ldap. Introducimos, para variar, **abc123.** (punto incluído)

- 


Repetimos o contrasinal, e veremos que remata a instalación do paquete.

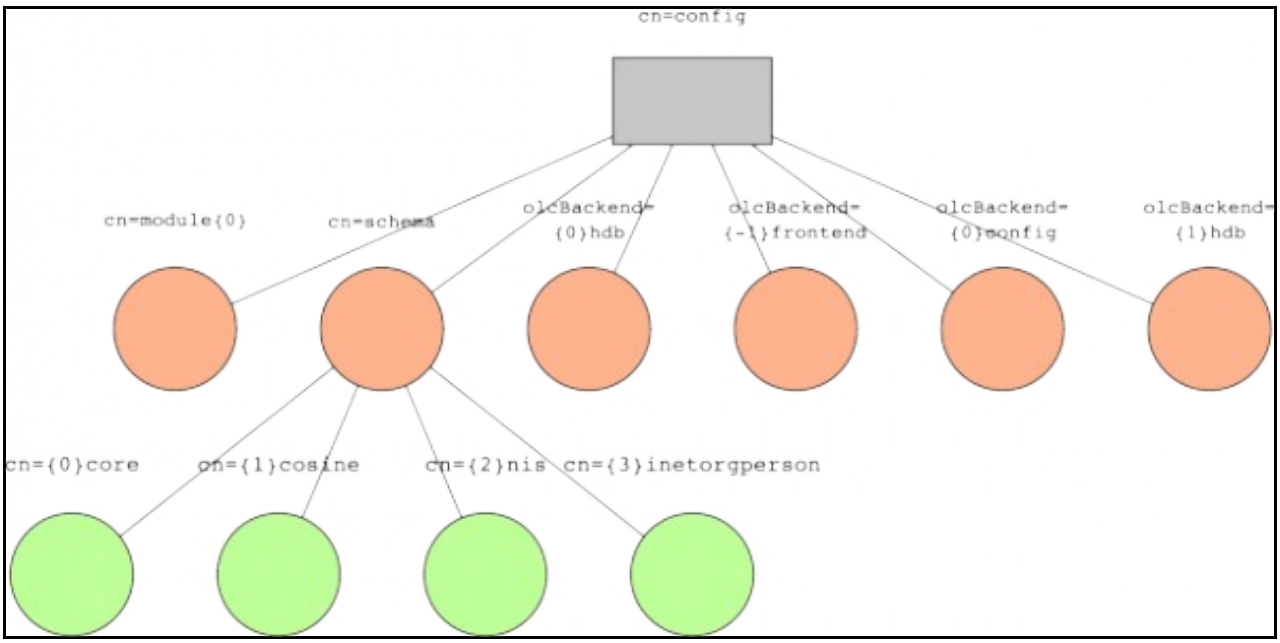
### 1.3.2 Ficheiros de ldap

- Unha vez instalado o paquete imos ver un pouco os ficheiros que que xestiona o servizo slapd. Estes atópanse en:
  - **/etc/ldap/***: ficheiros de configuración do directorio ldap.
  - **/etc/default/sldap**: ficheiro de configuración do servizo slapd.
  - **/var/lib/ldap/***: onde están as bases de datos que usa slapd.
  - **/usr/lib/ldap/***: contén módulos que pode usar slapd.

- Instalamos o paquete **tree** para axudarnos coa estrutura de ficheiros:

```
apt-get install tree
```

- Imos ver por arriba a estrutura da seguinte imaxe




- Instalación slapd

- ```
  root@dserver00:~# tree -d /etc/ldap/
  /etc/ldap/
  ├── sasl2
  ├── schema
  └── slapd.d
      └── cn=config
          └── cn=schema

  5 directories
  root@dserver00:~#
  ```

Executamos **tree -d /etc/ldap** para ver a estrutura de directorios.

- ```
  root@dserver00:~# tree /etc/ldap/
  /etc/ldap/
  ├── ldap.conf
  ├── sasl2
  ├── schema
  │   ├── collective.ldif
  │   ├── collective.schema
  │   ├── corba.ldif
  │   ├── corba.schema
  │   ├── core.ldif
  │   ├── core.schema
  │   ├── cosine.ldif
  │   ├── cosine.schema
  │   ├── duaconf.ldif
  │   ├── duaconf.schema
  │   ├── dyngroup.ldif
  │   ├── dyngroup.schema
  │   ├── inetorgperson.ldif
  │   ├── inetorgperson.schema
  │   ├── java.ldif
  │   ├── java.schema
  │   ├── misc.ldif
  │   ├── misc.schema
  │   ├── nis.ldif
  │   ├── nis.schema
  │   ├── openldap.ldif
  │   ├── openldap.schema
  │   ├── pmi.ldif
  │   ├── pmi.schema
  │   ├── ppolicy.ldif
  │   ├── ppolicy.schema
  │   └── README
  └── slapd.d
      ├── cn=config
      │   ├── cn=module{0}.ldif
      │   ├── cn=schema
      │   │   ├── cn={0}core.ldif
      │   │   ├── cn={1}cosine.ldif
      │   │   ├── cn={2}nis.ldif
      │   │   └── cn={3}inetorgperson.ldif
      │   ├── cn=schema.ldif
      │   ├── olcBackend={0}mdb.ldif
      │   ├── olcDatabase={0}config.ldif
      │   ├── olcDatabase={-1}frontend.ldif
      │   └── olcDatabase={1}mdb.ldif
      └── cn=config.ldif

  5 directories, 39 files
  ```

O mesmo pero sen o parámetro **-d** para ver tamén os ficheiros.

- ```
  root@dserver00:~# ls /etc/ldap
  ldap.conf  sasl2  schema  slapd.d
  root@dserver00:~#
  root@dserver00:~# cat /etc/ldap/ldap.conf
  #
  # LDAP Defaults
  #

  # See ldap.conf(5) for details
  # This file should be world readable but not world writable.

  #BASE   dc=example,dc=con
  #URI    ldap://ldap.example.com ldap://ldap-master.example.com:666

  #SIZELIMIT      12
  #TIMELIMIT      15
  #DEREF          never

  # TLS certificates (needed for GnuTLS)
  TLS_CACERT      /etc/ssl/certs/ca-certificates.crt

  root@dserver00:~#
  ```

En **/etc/ldap/ldap.conf** almacénase a información do cliente ldap para conectarse ao servidor, por agora está sen configurar.

- ```
  root@dserver00:~# ls /etc/ldap/schema/
  collective.ldif    cosine.schema         java.ldif      openldap.schema
  collective.schema  duaconf.ldif          java.schema    pmi.ldif
  corba.ldif         duaconf.schema        misc.ldif      pmi.schema
  corba.schema       dyngroup.ldif         misc.schema    ppolicy.ldif
  core.ldif          dyngroup.schema       nis.ldif       ppolicy.schema
  core.schema        inetorgperson.ldif    nis.schema     README
  cosine.ldif        inetorgperson.schema  openldap.ldif
  root@dserver00:~#
  ```

**/etc/ldap/schema** contén esquemas que poden ser cargados no directorio de ldap. Como xa vimos catro deles xa están cargados. Pódese observar o contido dos ficheiros para familiarizarse con eles.

- ```
  root@dserver00:~# ls /etc/ldap/slapd.d/
  cn=config  cn=config.ldif
  root@dserver00:~#
  root@dserver00:~# cat /etc/ldap/slapd.d/cn\=config.ldif
  # AUTO-GENERATED FILE - DO NOT EDIT!! Use ldapmodify.
  # CRC32 22503900
  dn: cn=config
  objectClass: olcGlobal
  cn: config
  olcArgsFile: /var/run/slapd/slapd.args
  olcLogLevel: none
  olcPidFile: /var/run/slapd/slapd.pid
  olcToolThreads: 1
  structuralObjectClass: olcGlobal
  entryUUID: bd0531fa-6177-1033-84bf-09dd158e721e
  creatorsName: cn=config
  createTimestamp: 20140426101738Z
  entryCSN: 20140426101738.747957Z#000000#000#000000
  modifiersName: cn=config
  modifyTimestamp: 20140426101738Z
  root@dserver00:~#
  ```

En **/etc/slapd/slapd.d** temos o ficheiro da rama de configuración **cn=config**. Para manexarse con estes ficheiros con símbolos como =, {, etc, aconséllase usar a tecla TAB para completar as instrucións (Premer 1 vez, ou 2 veces se hai directorios/ficheiros que teñen o mesmo comezo) ou poñer escribir os nomes completos dos ficheiros/directorios entre " ".
Observar: o dn, e o objectClass. Como vemos calquera rama da árbore DIT (Digital Information Tree, que vimos na teoría) debe ser introducida no directorio LDAP. Con este ficheiro foi co que se cargou a rama **cn=config** por parte do proceso de instalación de slapd.
Dentro do mesmo directorio está o directorio **cn=config**.

- ```
  root@dserver00:~# tree /etc/ldap/slapd.d/
  /etc/ldap/slapd.d/
  ├── cn=config
  │   ├── cn=module{0}.ldif
  │   ├── cn=schema
  │   │   ├── cn={0}core.ldif
  │   │   ├── cn={1}cosine.ldif
  │   │   ├── cn={2}nis.ldif
  │   │   └── cn={3}inetorgperson.ldif
  │   ├── cn=schema.ldif
  │   ├── olcBackend={0}mdb.ldif
  │   ├── olcDatabase={0}config.ldif
  │   ├── olcDatabase={-1}frontend.ldif
  │   └── olcDatabase={1}mdb.ldif
  └── cn=config.ldif

  2 directories, 11 files
  ```

Dentro de **/etc/slapd/slapd.d/cn=config** temos o directorio **cn=schema** que indica cales son esquemas que están cargados na base de datos/directorio ldap.

- ```
  root@dserver00:~# cat /etc/ldap/slapd.d/cn\=config/cn\=module\{0\}.ldif
  # AUTO-GENERATED FILE - DO NOT EDIT!! Use ldapmodify.
  # CRC32 3bd55bce
  dn: cn=module{0}
  objectClass: olcModuleList
  cn: module{0}
  olcModulePath: /usr/lib/ldap
  olcModuleLoad: {0}back_mdb
  structuralObjectClass: olcModuleList
  entryUUID: 18672660-68d8-1036-9d8c-adfdadf9936b
  creatorsName: cn=admin,cn=config
  createTimestamp: 20170107034954Z
  entryCSN: 20170107034954.901326Z#000000#000#000000
  modifiersName: cn=admin,cn=config
  modifyTimestamp: 20170107034954Z
  ```

O ficheiro **/etc/ldap/slapd.d/cn=config/cn=module{0}.ldif** indícalle ao demo slapd cales son os módulos que se deben cargar. Observar como se indica o directorio no que buscar os módulos e que módulos cargar. Neste caso a base de datos.

- ```
  root@dserver00:~# cat /etc/ldap/slapd.d/cn\=config/cn\=schema.ldif
  # AUTO-GENERATED FILE - DO NOT EDIT!! Use ldapmodify.
  # CRC32 7a33e0e2
  dn: cn=schema
  objectClass: olcSchemaConfig
  cn: schema
  structuralObjectClass: olcSchemaConfig
  entryUUID: bd054cf8-6177-1033-84c2-09dd158e721e
  creatorsName: cn=admin,cn=config
  createTimestamp: 20140426101738Z
  entryCSN: 20140426101738.748712Z#000000#000#000000
  modifiersName: cn=admin,cn=config
  modifyTimestamp: 20140426101738Z
  ```

O ficheiro **cn=schema.ldif** contén a definición da entrada **cn=schema** da árbore DIT (mirar gráfico superior). Obviamente este esquema é cargado antes que os catro anteriores (core, cosine, nis, ...) pois os anteriores colgan deste nodo da árbore.

- ```
  root@dserver00:~# cat /etc/ldap/slapd.d/cn\=config/olcDatabase\=\{1\}mdb.ldif
  # AUTO-GENERATED FILE - DO NOT EDIT!! Use ldapmodify.
  # CRC32 61fad8bb
  dn: olcDatabase={1}mdb
  objectClass: olcDatabaseConfig
  objectClass: olcMdbConfig
  olcDatabase: {1}mdb
  olcDbDirectory: /var/lib/ldap
  olcSuffix: dc=nodomain
  olcAccess: {0}to attrs=userPassword,shadowLastChange by self write by anonym
   ous auth by * none
  olcAccess: {1}to dn.base="" by * read
  olcAccess: {2}to * by * read
  olcLastMod: TRUE
  olcRootDN: cn=admin,dc=nodomain
  olcRootPW:: e1NTSEF9eGpHdXE3dFNKSm9XdUtUY0R4TWxYK3lLMm5rRU9yRko=
  olcDbCheckpoint: 512 30
  olcDbIndex: objectClass eq
  olcDbIndex: cn,uid eq
  olcDbIndex: uidNumber,gidNumber eq
  olcDbIndex: member,memberUid eq
  olcDbMaxSize: 1073741824
  ```

Os ficheiros **.../cn=config/olcDatabase*** conteñen directivas sobre as bases de datos. A numeración úsase para distinguir distintas BBDD do mesmo tipo. Neste caso, a que ten o ordinal 1, amosa, entre outras cousas, quen pode acceder á rama **nodomain.com**, tanto en modo escritura como en modo lectura. Tan pronto configuremos slapd veremos como cambia o contido dese ficheiro.

- ```
  root@dserver00:~# cat /etc/default/slapd
  # Default location of the slapd.conf file or slapd.d cn=config directory. If
  # empty, use the compiled-in default (/etc/ldap/slapd.d with a fallback to
  # /etc/ldap/slapd.conf).
  SLAPD_CONF=

  # System account to run the slapd server under. If empty the server
  # will run as root.
  SLAPD_USER="openldap"

  # System group to run the slapd server under. If empty the server will
  # run in the primary group of its user.
  SLAPD_GROUP="openldap"

  # Path to the pid file of the slapd server. If not set the init.d script
  # will try to figure it out from $SLAPD_CONF (/etc/ldap/slapd.conf by
  # default)
  SLAPD_PIDFILE=

  # slapd normally serves ldap only on all TCP-ports 389. slapd can also
  # service requests on TCP-port 636 (ldaps) and requests via unix
  # sockets.
  # Example usage:
  # SLAPD_SERVICES="ldap://127.0.0.1:389/ ldaps:/// ldapi:///"
  SLAPD_SERVICES="ldap:/// ldapi:///"
  ```

Xa noutro directorio, o ficheiro **/etc/default/slapd** permítenos, entre outras cousas indicar cales son os protocolos polos que se pode conectar un cliente ao demo slpad. Neste caso son:

**-ldap**: que usa o protocolo TCP e escoita no porto 389, podemos conectarnos dende calquera sitio coa sintaxe: ldap://ip_do_servidor

**-ldapi**: que se usa para conectarse co cliente dentro do mesmo servidor ao demo slapd usando **Sockets de Unix** (http://es.wikipedia.org/wiki/Socket_Unix) e non a pila de protocolos TCP/IP. Para conectarse só precisamos poñer o nome do protocolo: ldapi://.

**- ldaps**: non está habilitado por defecto e permite realizar conexións cifradas (seguras) entre os clientes e o servidor como veremos no escenario 1.F. Usa tamén o protocolo TCP e escoita no porto 636. Para conectarse: ldaps://ip_servidor:636 (porto opcional).

- ```
  root@dserver00:~# ls -lh /var/lib/ldap/
  total 68K
  -rw------- 1 openldap openldap  64K Xan  7 04:49 data.mdb
  -rw------- 1 openldap openldap 8,0K Xan  7 04:49 lock.mdb
  root@dserver00:~#
  ```

Finalmente en **/var/lib/ldap** almacénase a BD do directorio.

### 1.3.3 Configuración de slapd

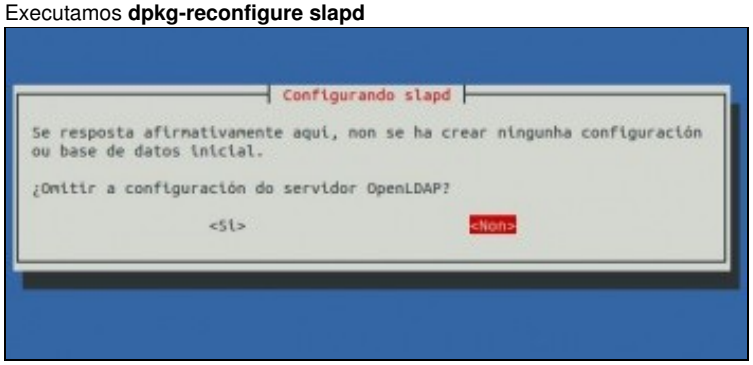
- Trala lectura do punto anterior, que tal se lle damos unha segunda volta? e logo imos configurar por fin o servidor.
- A continuación imos personalizar a configuración do directorio ldap para o noso dominio **iescalquera.local**.
- O que imos facer a continuación con pantallas de configuración vai xerar uns ficheiros ldif, que poderíamos xerar nós á man, e logo cargar.
- Podemos realizar a configuración tantas veces como precisemos, por se nos trabucamos, usando o comando **dpkg-reconfigure** que se usa para a reconfiguración dos paquetes unha vez instalados:

```
dpkg-reconfigure slapd
```

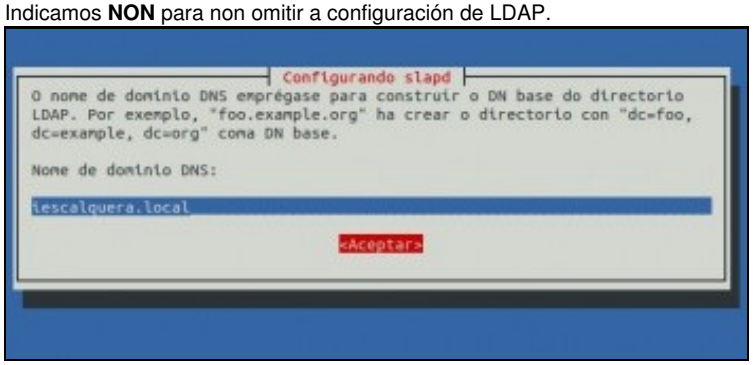
- Configuración de slapd
```

- 

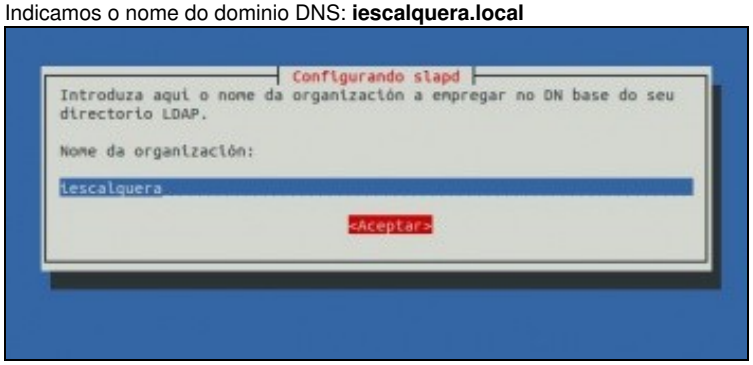

Executamos **dpkg-reconfigure slapd**

- 

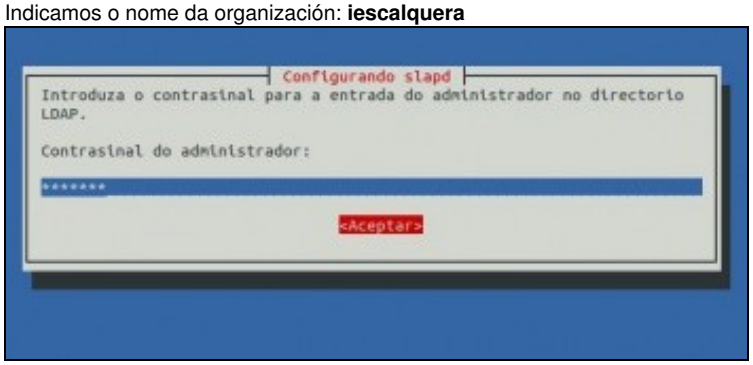

Indicamos **NON** para non omitir a configuración de LDAP.

- 

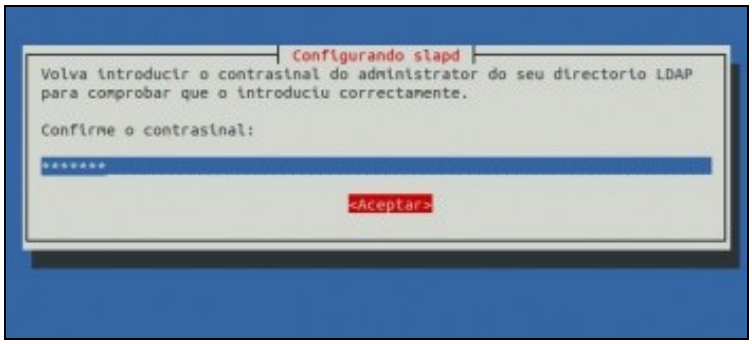

Indicamos o nome do dominio DNS: **iescalquera.local**

- 

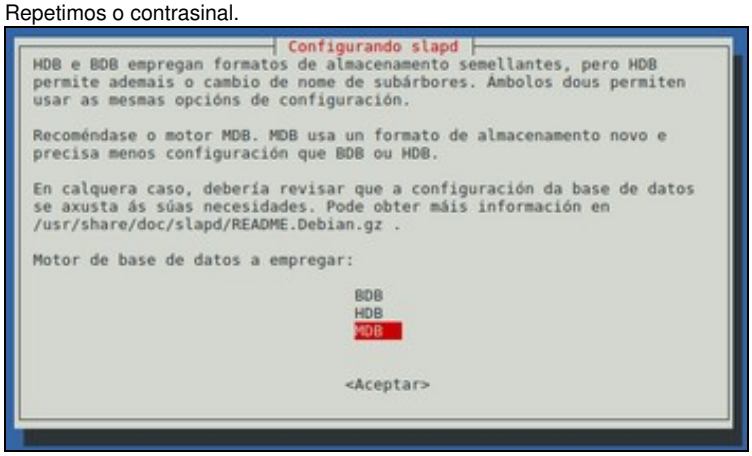

Indicamos o nome da organización: **iescalquera**

- 

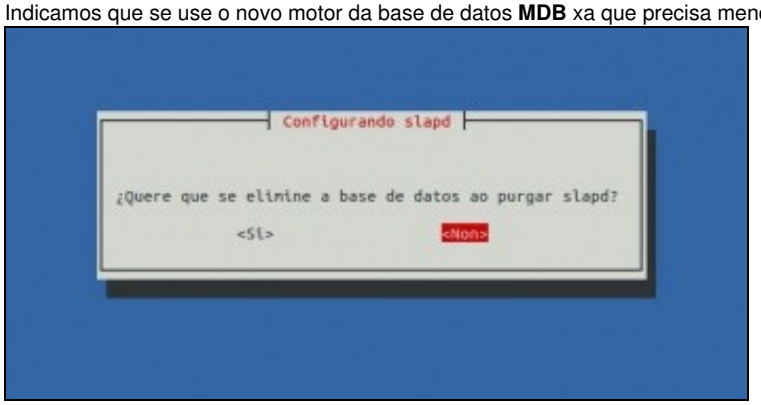

Introducimos a chave para o usuario **admin** que pode administrar a entrada no directorio ldap: **abc123.**

- 

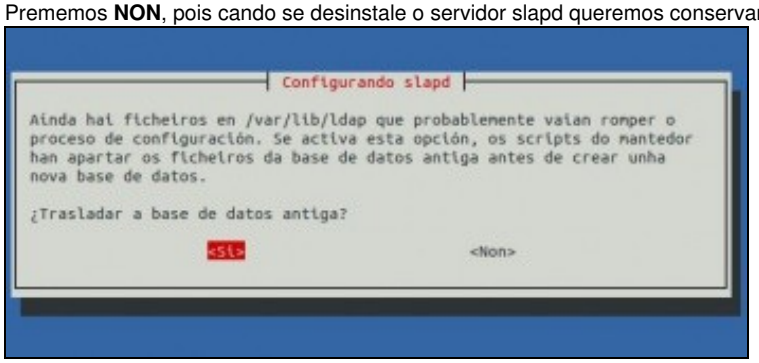

Repetimos o contrasinal.

- 

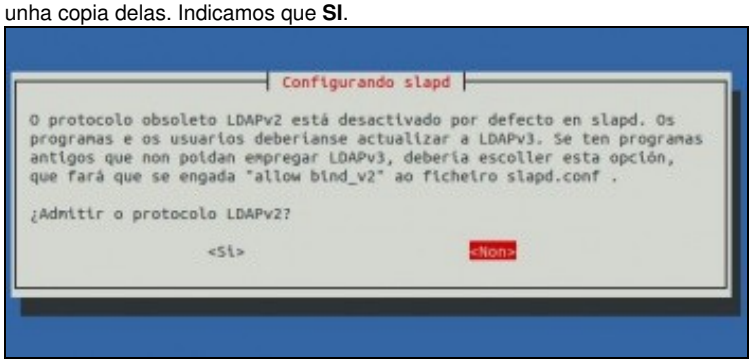

Indicamos que se use o novo motor da base de datos **MDB** xa que precisa menos configuración que o motor HDB.

- 

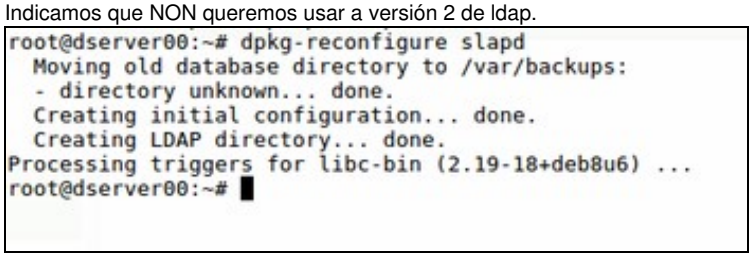

Prememos **NON**, pois cando se desinstale o servidor slapd queremos conservar os ficheiros de configuración.

- 

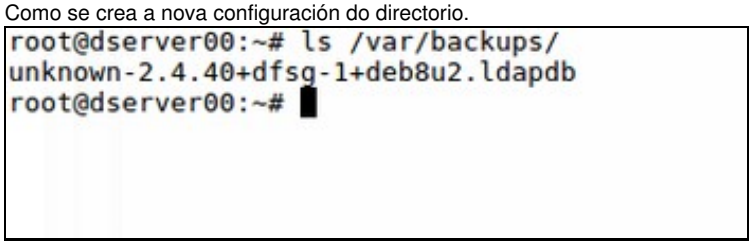

Como vimos antes en **/var/lib/ldap** temos os ficheiros da BD do directorio. As que viñan por defecto. Agora pregúntanos se desexamos facer unha copia delas. Indicamos que **SI**.

- 

Indicamos que NON queremos usar a versión 2 de ldap.

- 


Unha vez rematado o proceso de configuración reiniciase automaticamente o servizo slapd. Observar:
Como se move o a BD antiga a /**var**/**bakcups.**
Como se crea a nova configuración do directorio.

- 

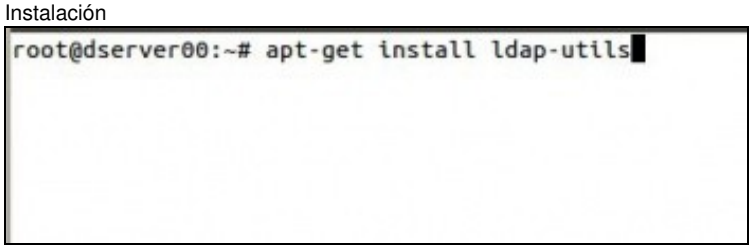

En /**var**/**backups**/**unknown...** está a BD anterior.

- Pero precisamos algo máis para ler e escribir na BD pois agora toca traballar con ficheiros LDIF, para iso instalaremos **ldap-utils**.

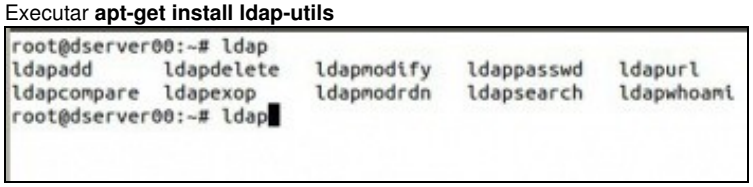
## 1.4 Utilidades ldap: paquete ldap-utils

- A continuación imos instalar o paquete **ldap-utils** que nos vai proporcionar utilidades:
- para consultar o directorio de LDAP
- engadir/modificar/eliminar obxectos
- poder cambiar o contrasinal dun usuario, etc.

### 1.4.1 Instalación de ldap-utils

- Instalación
- 


Executar **apt-get install ldap-utils**

- 


Agora temos un conxunto de comandos que comezan por ldap para: engadir, modificar, borrar elementos na BD de ldap e tamén para consultalos.
Lembrar que se pode escribir parte dun comando ou directorio/ficheiro e premer a tecla TAB (1 vez ou 2 se hai varios comandos/ficheiros que comparte comezo do nome) para completar o seu nome.

### 1.4.2 Consultar a BD: ldapsearch

- A continuación imos testar que todo funciona e para iso usaremos o comando **ldapsearch**.
- No seguinte enlace pódese consultar o manual de ldapsearch: http://linux.die.net/man/1/ldapsearch
- A continuación amosamos algúns dos parámetros que imos usar nesta primeira aproximación ao comando:
    - ♦ **-x**: usar autenticación simple.
    - ♦ **-Y *EXTERNAL***: usa autenticación **SASL** (Simple Authentication and Security Layer - capa de seguridade e autenticación simple)
        - ◊ O mecanismo de SASL que usamos neste caso é **EXTERNAL**, onde a autenticación está implícita no contexto, isto é, o cliente e servidor teñen instalados uns certificados para autenticarse un contra o outro. Neste caso o cliente (ldapsearch)

e o servizo (slapd) están no mesmo equipo en dserver00, co cal eses certificados xa están no equipo.
◊ Nos seguintes enlaces hai máis información sobre este tipo de autenticación:
· http://es.wikipedia.org/wiki/SASL
· http://www.openldap.org/doc/admin21/sasl.html
· http://www.openldap.org/doc/admin21/tls.html

- **-D *dn***: dn para conectarse ao LDAP indicando o nome de usuario co que nos imos conectar.
- **-w *contrasinal***: Indicar o contrasinal para conectarse ao LDAP.
- **-W**: Obrigamos a que o comando pide o contrasinal para conectarse ao LDAP en lugar de recibilo como parámetro.
- **-H** *ldapuri***: Especificar a URI coa que nos imos conectar ao servidor ldap. Por exemplo ldap://localhost ou ldapi://**
- **filtro** indica que nos devolva as entradas que coincidan cos valores dos atributos que indicamos no filtro.
- **-b *base de busca***: para indicar en que obxecto da árbore comezar a buscar
- **-s** *base*|*one*|*sub***: indica se queremos que nos devolva os atributos e valores:**
  ◊ **base**: só do obxecto que estamos consultando
  ◊ **one**: só dos obxectos que están un nivel por debaixo do obxecto consultado.
  ◊ **sub**: do obxecto consultado e de toda a súa subárbore. É o valor por defecto.

- O formato do comando é (non precisamente nesa orde):

```
ldapsearh método_autenticación base ldapuri filtros
```

- Os valores que indicamos para os parámetros poden ir ou non entre comiñas simples ('valores') se vemos que se poden confundir cos parámetros, pero non é obrigatorio.

#### 1.4.2.1 Consultar o dominio creado

- Indicamos como base o tope da xerarquía e como filtros calquera clase de obxecto.
- Observar que o propio comando nas liñas 4-7 amosa que é o que pretendemos consultar.
- O que nos interesa está na liña 12.

```
root@dserver00:~#  ldapsearch -x -b '' -s base objectclass=* namingcontexts
# extended LDIF
#
# LDAPv3
# base <> with scope baseObject
# filter: objectclass=*
# requesting: namingcontexts
#

#
dn:
namingContexts: dc=iescalquera,dc=local

# search result
search: 2
result: 0 Success

# numResponses: 2
# numEntries: 1
```

#### 1.4.2.2 Consultar obxectos da rama dc=iescalquera,dc=local

- Imos agora facer unha serie de consultas na nosa rama de interese. Por agora non temos moito nesa rama, faltan usuarios, grupos, etc, que crearemos no seguinte apartado.

- **Sen autenticación**:
  ♦ Observar as liñas marcadas (En formato PDF non se poden observar as liñas marcadas).

```
root@dserver00:~#  ldapsearch -x -b 'dc=iescalquera,dc=local'
# extended LDIF
```

```
#
# LDAPv3
# base <dc=iescalquera,dc=local> with scope subtree
# filter: (objectclass=*)
# requesting: ALL
#

# iescalquera.local
dn: dc=iescalquera,dc=local
objectClass: top
objectClass: dcObject
objectClass: organization
o: iescalquera
dc: iescalquera

# admin, iescalquera.local
dn: cn=admin,dc=iescalquera,dc=local
objectClass: simpleSecurityObject
objectClass: organizationalRole
cn: admin
description: LDAP administrator

# search result
search: 2
result: 0 Success

# numResponses: 3
# numEntries: 2
```

- **Con autenticación SASL implícita**:
    - precisamos indicar o servidor, neste caso con -H ldapi://
    - Obtemos a mesma información que no caso anterior

```
root@dserver00:~#  ldapsearch -Y EXTERNAL -H ldapi:// -b 'dc=iescalquera,dc=local'
SASL/EXTERNAL authentication started
SASL username: gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth
SASL SSF: 0
# extended LDIF
#
# LDAPv3
# base <dc=iescalquera,dc=local> with scope subtree
# filter: (objectclass=*)
# requesting: ALL
#

# iescalquera.local
dn: dc=iescalquera,dc=local
objectClass: top
objectClass: dcObject
objectClass: organization
o: iescalquera
dc: iescalquera

# admin, iescalquera.local
dn: cn=admin,dc=iescalquera,dc=local
objectClass: simpleSecurityObject
objectClass: organizationalRole
cn: admin
description: LDAP administrator

# search result
search: 2
result: 0 Success

# numResponses: 3
# numEntries: 2
```

- **Indicando o usuario e contrasinal co que nos conectamos**
    - Observar a liña 24, pois como somos o usuario admin ensínanos o hash asociado ao seu contrasinal.
    - Observar como con -D indicamos o usuario e con -w o contrasinal.
    - Nesta ocasión como uri usamos ldap://localhost, pero podería ser ldap://dserver00, ldap://dserver00.iescalquera.local (consultaríase o DNS), etc.

```
root@dserver00:~#  ldapsearch -D cn=admin,dc=iescalquera,dc=local -w abc123. -H ldap://localhost -b 'dc=iescalquera,dc=local'
# extended LDIF
#
# LDAPv3
# base <dc=iescalquera,dc=local> with scope subtree
# filter: (objectclass=*)
# requesting: ALL
#

# iescalquera.local
dn: dc=iescalquera,dc=local
objectClass: top
objectClass: dcObject
objectClass: organization
o: iescalquera
dc: iescalquera

# admin, iescalquera.local
dn: cn=admin,dc=iescalquera,dc=local
objectClass: simpleSecurityObject
objectClass: organizationalRole
cn: admin
description: LDAP administrator
userPassword:: e1NTSEF9Qm5mRG9QTWVSUnUyalNpZUNXMEhWbnhvMlBnUWNrK1o=

# search result
search: 2
result: 0 Success
```

- **Indicando o usuario co que nos conectamos, pero non o contrasinal**
    - Observar a liña 2, como pide o contrasinal de admin.
    - Nesta ocasión como uri usamos ldap://dserver00,

```
root@dserver00:~#  ldapsearch -D cn=admin,dc=iescalquera,dc=local -W -H ldap://dserver00 -b 'dc=iescalquera,dc=local'
Enter LDAP Password:
```

- **Indicando que so queremos consultar os valores do obxecto base, non os fillos**
    - Para iso usamos o parámetro -s base
    - Observar a liña 5

```
root@dserver00:~#  ldapsearch -x -b 'dc=iescalquera,dc=local' -s base
# extended LDIF
#
# LDAPv3
# base <dc=iescalquera,dc=local> with scope baseObject
# filter: (objectclass=*)
# requesting: ALL
#

# iescalquera.local
dn: dc=iescalquera,dc=local
objectClass: top
objectClass: dcObject
objectClass: organization
o: iescalquera
dc: iescalquera

# search result
search: 2
```

```
result: 0 Success

# numResponses: 2
# numEntries: 1
```

- **Indicando que só queremos consultar os valores dos obxectos fillos do base, non os do obxecto base**
    - ♦ Para iso usamos o parámetro: -s one
    - ♦ Lembrar que por defecto o valor do parámetro é: -s sub
    - ♦ Pero só temos un fillo.
    - ♦ Observar a liña 5.

```
root@dserver00:~#  ldapsearch -x -b 'dc=iescalquera,dc=local' -s one
# extended LDIF
#
# LDAPv3
# base <dc=iescalquera,dc=local> with scope oneLevel
# filter: (objectclass=*)
# requesting: ALL
#

# admin, iescalquera.local
dn: cn=admin,dc=iescalquera,dc=local
objectClass: simpleSecurityObject
objectClass: organizationalRole
cn: admin
description: LDAP administrator

# search result
search: 2
result: 0 Success

# numResponses: 2
# numEntries: 1
```

- **Poñemos un filtro no que indicamos que desexamos consultar os obxectos que teñan o atributo objectClass independentemente do valor que teña ese atributo**
    - ♦ Para iso poñemos un filtro cos atributos e os valores buscados.
    - ♦ Obviamente o atributo objectClass teno todo obxecto.
    - ♦ Observar que non é sensible ás maiúsculas.
    - ♦ Observar a liña 6

```
root@dserver00:~#  ldapsearch -x -b 'dc=iescalquera,dc=local' obJecTcLass=*
# extended LDIF
#
# LDAPv3
# base <dc=iescalquera,dc=local> with scope subtree
# filter: obJecTcLass=*
# requesting: ALL
#

# iescalquera.local
dn: dc=iescalquera,dc=local
objectClass: top
objectClass: dcObject
objectClass: organization
o: iescalquera
dc: iescalquera

# admin, iescalquera.local
dn: cn=admin,dc=iescalquera,dc=local
objectClass: simpleSecurityObject
objectClass: organizationalRole
cn: admin
description: LDAP administrator

# search result
```

```
search: 2
result: 0 Success

# numResponses: 3
# numEntries: 2
```

- **Poñemos un filtro no que indicamos que desexamos consultar os obxectos que teñan o atributo objectClass co valor dcObject**
    - ♦ Para iso poñemos un filtro co atributo e o valor desexado.
    - ♦ Agora só obtemos os obxectos que cumpren coa condición.
    - ♦ Observar a liña 6

```
root@dserver00:~#  ldapsearch -x -b 'dc=iescalquera,dc=local' objectclass=dcobject
# extended LDIF
#
# LDAPv3
# base <dc=iescalquera,dc=local> with scope subtree
# filter: objectclass=dcobject
# requesting: ALL
#

# iescalquera.local
dn: dc=iescalquera,dc=local
objectClass: top
objectClass: dcObject
objectClass: organization
o: iescalquera
dc: iescalquera

# search result
search: 2
result: 0 Success

# numResponses: 2
# numEntries: 1
```

- Outros exemplos sobre o mesmo.

- **Poñemos un filtro no que indicamos que desexamos consultar os obxectos que teñan o atributo 'cn' independentemente do valor que teña ese atributo**
    - ♦ Temos un só obxecto que cumpre co filtro.
    - ♦ Observar a liña 6

```
root@dserver00:~#  ldapsearch -x -b 'dc=iescalquera,dc=local' cn=*
# extended LDIF
#
# LDAPv3
# base <dc=iescalquera,dc=local> with scope subtree
# filter: cn=*
# requesting: ALL
#

# admin, iescalquera.local
dn: cn=admin,dc=iescalquera,dc=local
objectClass: simpleSecurityObject
objectClass: organizationalRole
cn: admin
description: LDAP administrator

# search result
search: 2
result: 0 Success

# numResponses: 2
# numEntries: 1
```

- **Poñemos un filtro no que indicamos que desexamos consultar os obxectos que teñan o atributo cn co valor admin**
    - Temos o mesmo obxecto anterior que cumpre co filtro.
    - Observar a liña 6

```
root@dserver00:~#  ldapsearch -x -b 'dc=iescalquera,dc=local' cn=admin
# extended LDIF
#
# LDAPv3
# base <dc=iescalquera,dc=local> with scope subtree
# filter: cn=admin
# requesting: ALL
#

# admin, iescalquera.local
dn: cn=admin,dc=iescalquera,dc=local
objectClass: simpleSecurityObject
objectClass: organizationalRole
cn: admin
description: LDAP administrator

# search result
search: 2
result: 0 Success

# numResponses: 2
# numEntries: 1
```

-- Antonio de Andrés Lema e Carlos Carrión Álvarez