

# **SISTEMA DE CUENTAS CENTRALIZADAS HETEROGÉNEAS**

**José Manuel Ferrete Benítez**



## Índice de contenido

Introducción.....	3
¿Qué es un sistema de cuentas centralizadas?.....	3
¿Qué es un sistema heterogéneo?.....	3
Entonces ¿Cual nos conviene?.....	3
Descripción del proyecto.....	4
Herramientas.....	4
Debian Squeeze 6.0.....	5
Samba 3.5.....	5
OpenLDAP 2.4.....	5
SmbLdapTools.....	5
PAM.....	5
NSS.....	5
Objetivos.....	7
Desarrollo.....	7
Configuración de la red.....	8
Instalación y configuración de OpenLDAP.....	9
Instalación y configuración de Samba.....	11
Instalación y configuración de Smbldaptools.....	15
Poblar el directorio LDAP.....	18
Configuración de NSS.....	19
Configuración de PAM.....	21
Gestión de usuarios y grupos.....	22
Creación de usuarios.....	22
Creación de grupos.....	22
Agregar usuario a un grupo.....	23
Migración de usuarios locales al LDAP.....	23
Preparar los clientes.....	24
Clientes Linux.....	24
Creación de un paquete Debian.....	24
Clientes Windows.....	28
Cambios para clientes Windows 7.....	30
Login de usuarios.....	31
Resultado.....	36
Administración mediante herramientas externas.....	37
Conclusión y posibles mejoras.....	42
Referencias.....	42

## Introducción

En el siguiente documento desarrollo como proyecto la implantación de un sistema de cuentas centralizadas heterogéneas.

### ¿Qué es un sistema de cuentas centralizadas?

Se trata de un servicio que pueden ofrecer muchos tipos de organizaciones para centralizar la información de sus usuarios, ofrecerles total accesibilidad a sus datos desde cualquier punto de conexión y en general facilitar la administración de los mismos.

### ¿Qué es un sistema heterogéneo?

En cuanto al sistema heterogéneo comenzaremos la explicación informando de lo fácil que es encontrar software para facilitar esta labor, como pueden ser Directorio Activo, OpenLDAP, Samba.

Sabemos que en referente al mundo empresarial, Directorio Activo lleva una gran ventaja gracias a su fácil instalación y administración. Nos podemos dar cuenta además de que está totalmente pensado para clientes con sistema operativo Windows, por lo que dificulta un poco la administración e integración de máquinas con otros sistemas operativos en el dominio. Además el método privativo de Microsoft con Directorio Activo lo convierte en una herramienta limitada ya que no te permite usarlo fácilmente para otros fines.

Es por ello por lo que surgieron los sistemas heterogéneos, son sistemas informáticos en los que no importa generalmente si se trata de un cliente Linux o Windows, ya que cualquiera de ellos pueden ser centralizados mediante un mismo servidor.

### Entonces ¿Cual nos conviene?

A la hora de plantear la implantación de un sistema de cuentas centralizadas hay que tener en cuenta muchas cosas. Principalmente el tipo de usuarios a los que daremos servicio y el presupuesto del que dispone la empresa para tal fin.

Si vemos el siguiente cuadrante:

Producto	Compatibilidad	Licencia de Servidor	Licencia de Cliente	Posibilidades que ofrece
Directorio Activo + Windows Server 2008 R2	Windows, Unix (instalando un complemento de compatibilidad)	600,00 € el más barato	100,00 € - licencia para 5 clientes	Limitadas por su sistema base
Samba + OpenLDAP + Debian Squeeze	Total	0,00 €	0,00 €	Casi ilimitadas y en continuo desarrollo

Aunque claramente Samba + OpenLDAP parece ser más conveniente, su dificultad de configuración le hace tener la cuota de mercado a un nivel más alto.

Es obvio no hablar de otros sistemas operativos en este proyecto puesto que no disponen de cuota de mercado suficiente en este sector.

Si analizamos más a fondo Directorio Activo, nos damos cuenta de que en realidad está compuesto básicamente por los siguientes elementos:

- Un servicio de resolución de nombres (DNS o WINS).
- Un servicio de directorio LDAP.
- Un servicio de autenticación NTLM o Kerberos (Según la versión).
- Un servicio de compartición de ficheros CIFS (Anteriormente denominado SMB).

Cualquier administrador de sistemas Linux, conociendo estos datos sería capaz de montar un servidor que ofreciera dichos servicios por separado. ¿Qué conseguiríamos con ello? Conseguríamos la posibilidad de sacarle el máximo rendimiento a cada uno de los servicios, además de la personalización y adaptación a nuestras necesidades.

## Descripción del proyecto

En este caso, durante el desarrollo del proyecto implantaré un sistema de cuentas centralizadas heterogéneas con las siguientes características:

- El servidor centralizará toda la información de los usuarios y máquinas. Dicha información será almacenada en el directorio LDAP. Incluyendo contraseñas y datos encriptados.
- Además el servidor contendrá los datos con los que los usuarios operan normalmente, ya sean datos, configuraciones, etc. De esto se encargará el servicio de compartición de ficheros utilizando el protocolo CIFS.
- También se centralizarán todos los controles de permisos de usuario y de fichero tanto para clientes Windows como para clientes Linux, haciendo uso de permisos NTFS y políticas de grupo.
- Los directorios home de cada usuario junto con su perfil serán auto-montados en el momento que el usuario inicie sesión en un equipo de la red.
- Se hará uso de autenticación NTLM para autenticar a los usuarios y máquinas.

## Herramientas

En otros tipos de proyecto, existen varias herramientas que pueden realizar las mismas funciones y proporcionar los mismos servicios, como por ejemplo un servicio SMTP. En este caso no podemos decir lo mismo ya que dentro del mundo Linux, suelen ser las siguientes las que se utilizan normalmente para este fin, ya que en algunos casos son las únicas y en otros son las más utilizadas por su eficiencia como es el caso de OpenLDAP.

- **Debian Squeeze 6.0** – Sistema Operativo Linux de servidor.
- **Samba 3.5** - Nos ofrecerá el servicio de resolución de nombres WINS y además realizará la función de controlador de dominio y la compartición de ficheros por CIFS/SMB.
- **OpenLDAP 2.4** - Nos proporcionará el servicio de directorio.
- **SmbLdapTools** – Nos facilitará los scripts para administrar el dominio.

- **PAM** - Controla los módulos de autenticación en Linux, y proporcionará la posibilidad de autenticarnos contra el servidor LDAP para los usuarios de las máquinas con dicho sistema.
- **NSS** - Resuelve los nombres de diferentes objetos del sistema en Linux, así como máquinas, metadatos de los ficheros, etc. Nos ayudará a resolver dichos datos proporcionados por el LDAP.

## **Debian Squeeze 6.0**

En general Debian ha sido siempre uno de los sistemas operativos más estables gracias al grupo de desarrolladores y usuarios que forman parte del proyecto y ayudan a corregir los errores de ejecución de todos sus paquetes. Quizás sea su filosofía a la hora de lanzar una distribución lo que le haya concedido a lo largo del tiempo la reputación de estabilidad, fiabilidad y seguridad que tiene hoy en día.

Otra de las ventajas vista desde el ámbito técnico es que no consume más recursos que los que necesita, quiero decir con esto que gracias a su origen libre Debian se amolda a cada circunstancia, existiendo versiones que se pueden lanzar en máquinas con limitaciones convirtiéndolo en un sistema operativo versátil y extremadamente utilizable.

## **Samba 3.5**

Es la implementación de CIFS en Linux más utilizada y más estable, además de ser la alternativa a Windows a la hora de compartir ficheros, también realiza funciones de controlador de dominio y gestión de usuarios.

## **OpenLDAP 2.4**

Es el sistema de directorio LDAP más utilizado en el mundo del software libre. Su configuración ha cambiado radicalmente en las últimas versiones, lo que nos da una ventaja a la hora de aprender directamente a partir de esta versión.

OpenLDAP está realmente extendido en todo el mundo y es una opción claramente fiable y recomendada para ser utilizada.

En este proyecto, lo utilizaremos para almacenar información sobre los usuarios, grupos y equipos.

## **SmbLdapTools**

Son scripts que nos facilitan la administración del servidor Samba. Nos ayuda a crear los usuarios, grupos, incluir máquinas, crear dominios nuevos, etc.

## **PAM**

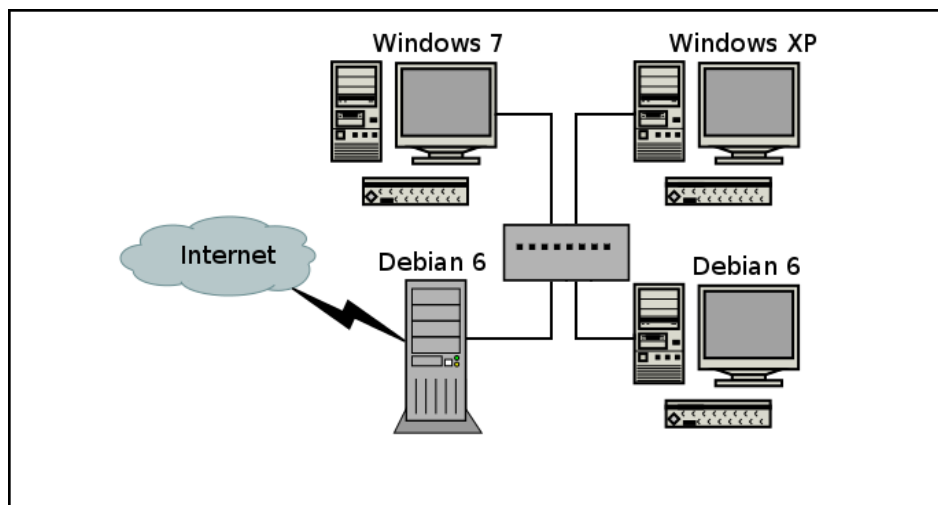
Herramienta que nos configura todo lo correspondiente al inicio de sesión en los equipos Linux.

La utilizaremos para que se utilice el directorio LDAP como fuente de usuarios.

## **NSS**

Herramienta que se utiliza en los equipos Linux para establecer la localización de los nombres de usuario, de grupos y de cualquier objeto. Lo configuraremos para que los extraiga del LDAP.

Para realizar las pruebas de funcionamiento, integraré tres máquinas como clientes, dos con Microsoft Windows (XP y 7), y una con Debian GNU/Linux. Este será el esquema de la red:



La red podría complicarse todo lo que se quisiera pero quizás se saldría un poco del fin que le quiero dar a este proyecto.

Posibles cuestiones que nos pueden surgir:

#### **¿Porqué utilizar Debian Squeeze?**

La verdadera elección cae entre utilizar Linux o Windows, y para este propósito se ha decidido utilizar Linux por su alto grado de compatibilidad de forma nativa con otras plataformas. Además de ofrecernos la posibilidad de desmenuzar la configuración de cada uno de los servicios.

#### **¿Porqué no utilizar un servidor DNS?**

Porque su instalación no sería necesaria en este caso, si que sería complementaria para el mejor funcionamiento de la red de la organización donde se implantase. En su lugar utilizaremos Samba como servidor WINS y controlador de dominio NetBIOS.

#### **¿Porqué no utilizar Kerberos?**

Sería útil utilizar Kerberos para este propósito ya que obtendríamos el mayor sistema de seguridad en la autenticación disponible por ahora, pero Samba en su versión 3 no tiene soporte para utilizar Kerberos como servidor de autenticación, sí para autenticarse contra él, pero no para utilizarlo como base de datos de contraseñas.

En su lugar utilizaremos autenticación NTLM, ofrecida por Microsoft Windows Server desde su versión 2000. Aunque no es tan seguro como Kerberos, es lo suficientemente seguro como para utilizarse aún para el fin que le vamos a dar.

## Objetivos

Comprender el funcionamiento de un sistema de cuentas centralizadas.

Lograr la implantación y testeo del sistema en un entorno en producción.

Aprender a utilizar las herramientas de administración.

Ser capaz de detectar posibles errores de seguridad en el sistema.

## Desarrollo

Para el desarrollo de este proyecto se utilizarán cuatro máquinas virtuales, dos con sistema operativo Debian Squeeze 6.0 (servidor y cliente), una con Windows XP y otra con Windows 7.

Lo primero que vamos a hacer será preparar nuestro servidor Debian, para ello contamos con una máquina con dicho sistema operativo recién instalado y sin interfaz gráfica, ya que tratándose de un servidor lo lógico es que lo administrásemos por terminal.

Estas son las características de nuestro servidor:

- Nombre de la máquina: kalimdor
- Nombre de dominio NetBIOS: AZEROTH
- Procesador Intel 1.6 GHz compartido con las demás máquinas virtuales.
- 512 MB de RAM.
- 10 GB de disco duro.
- 2 Tarjetas de red.
- Sistema operativo Debian Squeeze 6.0.

Como podéis ver, no son características de una máquina potente. No necesitaremos más.

Estas son las características de las máquinas que harán la función de clientes:

- Nombre de la máquina: rasganorte
  - Nombre de dominio NetBIOS: AZEROTH
  - Procesador Intel 1.6 GHz compartido con las demás máquinas virtuales.
  - 256 MB de RAM.
  - 10 GB de disco duro.
  - 1 Tarjeta de red.
  - Sistema operativo Debian Squeeze 6.0.
-

- Nombre de la máquina: terrallende
  - Nombre de dominio NetBIOS: AZEROTH
  - Procesador Intel 1.6 GHz compartido con las demás máquinas virtuales.
  - 256 MB de RAM.
  - 10 GB de disco duro.
  - 1 Tarjeta de red.
  - Sistema operativo Microsoft Windows XP.
- 

- Nombre de la máquina: infralar
- Nombre de dominio NetBIOS: AZEROTH
- Procesador Intel 1.6 GHz compartido con las demás máquinas virtuales.
- 512 MB de RAM.
- 10 GB de disco duro.
- 1 Tarjeta de red.
- Sistema operativo Microsoft Windows 7.

## **Configuración de la red**

Lo primero que debemos hacer es configurar la red en los equipos.

Comenzaremos configurando la red en el servidor, añadiendo lo siguiente en el fichero “/etc/network/interfaces”:

```
#nano /etc/network/interfaces

auto lo
iface lo inet loopback

#Interfaz wan
auto eth0
iface eth0 inet dhcp

#Interfaz lan
auto eth1
iface eth1 inet static
    address 10.0.0.1
    netmask 255.255.255.0

#Activamos el enrutamiento
up echo 1 >/proc/sys/net/ipv4/ip_forward
```

Reiniciamos la red.

```
#!/etc/init.d/networking restart
```



## Instalación y configuración de OpenLDAP.

Comenzaremos con la instalación de los servicios y su configuración. Para ello primariamente instalaremos la aplicación que nos hará realizar la función de servidor LDAP. Instalaremos OpenLDAP.

```
#dpkg-reconfigure debconf
```

Con la anterior instrucción establecemos el nivel de configuración de los diálogos a bajo nivel, lo que nos permitirá configurar desde su instalación el servidor LDAP. Posteriormente instalamos el paquete slapd.

```
#aptitude install slapd
```

Responderemos a las preguntas con estos datos:

- ¿Desea omitir la configuración de Slapd? **No**.
- Nombre de dominio DNS: **azeroth.com**.
- Nombre de la organización: **Azeroth**.
- Contraseña de administrador: **\*\*\*\*** (en mi caso y de prueba: “root”)
- Repetir contraseña: **\*\*\*\***.
- Motor de la base de datos: **HDB**.
- ¿Borrar la BD cuando se purgue Slapd? **Sí**.
- Permitir el protocolo LDAPv2: **No**.
- ¿Quiere que man y man-db se instalen 'setuid man'? **No**.

Volvemos a dejar el debconf como estaba (en alto).

```
#dpkg-reconfigure debconf
```

En este momento tendremos a nuestra disposición un acceso total al directorio. Podemos ver su contenido con el siguiente comando:

```
#slapcat
```

Que actualmente sólo contendrá dos objetos:

cn: dc=azeroth,dc=com (El dominio)

cn: cn=admin,dc=azeroth,dc=com. (El usuario de administración)

Para facilitar la administración del directorio LDAP instalaremos el siguiente paquete, el cual nos proporcionará herramientas para introducir y eliminar datos del directorio:

```
#aptitude install ldap-utils
```

El siguiente paso será preparar la estructura de nuestro directorio. Instalaremos el esquema de Samba que podemos obtener instalando el paquete correspondiente.

```
#aptitude install samba-doc
```

Procedemos a la instalación del esquema de Samba en el directorio LDAP:

```
#zcat /usr/share/doc/samba-doc/examples/LDAP/samba.schema.gz > \
/etc/ldap/schema/samba.schema
```

Creamos un fichero temporal con el siguiente contenido:

```
#nano /tmp/borraime.conf
include /etc/ldap/schema/core.schema
include /etc/ldap/schema/cosine.schema
include /etc/ldap/schema/nis.schema
include /etc/ldap/schema/inetorgperson.schema
include /etc/ldap/schema/samba.schema
```

Creamos un directorio temporal con el que trabajar mientras creamos el fichero ldif:

```
#mkdir /tmp/borraime.d
```

Creamos el fichero ldif:

```
#slapcat -f /tmp/borraime.conf -F /tmp/borraime.d/ -n0 -s \
"cn={4}samba,cn=schema,cn=config" > /tmp/cn=samba.ldif
```

Lo editamos para hacerle las siguientes modificaciones:

```
#nano /tmp/cn\samba.ldif
```

Borrarle el “{4}” del las 3 primeras líneas:

dn: cn={4}samba,cn=schema,cn=config	dn: cn=samba,cn=schema,cn=config
objectClass: olcSchemaConfig	=> objectClass: olcSchemaConfig
cn: {4}samba	cn: samba

Eliminar estas últimas:

```
structuralObjectClass: olcSchemaConfig
entryUUID: c2cc4d18-0a76-1030-829f-ef06145691e2
creatorsName: cn=config
createTimestamp: 20110504084634Z
entryCSN: 20110504084634.760215Z#000000#000#000000
modifiersName: cn=config
modifyTimestamp: 20110504084634Z
```

Finalmente lo añadimos con la herramineta ldapadd:

```
#ldapadd -Y EXTERNAL -H ldapi:/// -f /tmp/cn\samba.ldif
```

```
root@debian:~# ldapadd -Y EXTERNAL -H ldapi:/// -f /tmp/cn\samba.ldif
SASL/EXTERNAL authentication started
SASL username: gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth
SASL SSF: 0
adding new entry "cn=samba,cn=schema,cn=config"
root@debian:~# _
```

Así finalizamos con la configuración de OpenLDAP.

## Instalación y configuración de Samba

La instalación de samba es muy sencilla,

```
#aptitude install samba samba-client
```

Al igual que con OpenLDAP, nos hará una serie de preguntas, aunque en este caso podemos responder con los datos que sepamos, sin preocuparnos mucho con los resultados ya que posteriormente vamos a modificar el fichero de configuración y lo adaptaremos para que realice la función que necesitamos.

Una vez instalado, procederemos a configurar nuestro servicio Samba editando el fichero **/etc/samba/smb.conf** contendrá las siguientes secciones:

```
#nano /etc/samba/smb.confv

### Configuraciones globales. ###
[global]

# Establecemos la codificación de caracteres compatible tanto con Unix como con DOS.
dos charset = 850
Unix charset = ISO8859-1

# Nombre de dominio y nombre netBIOS del servidor.
workgroup = AZEROTH
realm = KALIMDOR

# Mensaje de bienvenida del servidor.
server string = %h server

# Política contra usuarios inexistentes.
map to guest = Bad User

# Fichero de alias de nombres de usuario. Se utilizará para mapear por ejemplo la
cuenta de root en Linux con la de Administrador en Windows.
username map = /etc/samba/smbusers

# Configuración de conexión con el servidor LDAP.

# Le indicamos que los usuarios se encuentran en el LDAP.
passdb backend = ldapsam:ldap://127.0.0.1/

# Le indicamos la cuenta de administrador del LDAP.
ldap admin dn = cn=admin,dc=azeroth,dc=com
ldap delete dn = Yes

# Le indicamos las unidades organizativas correspondientes a los componentes necesarios
en un dominio Windows.
ldap user suffix = ou=people
ldap group suffix = ou=group
ldap idmap suffix = ou=idmap
ldap machine suffix = ou=computer

# Le indicamos la base.
ldap suffix = dc=azeroth,dc=com

# Que no dispondremos de conexión SSL.
ldap ssl = no

# Le indicamos los diferentes scripts que usaremos para la administración de los
diferentes componentes del dominio.
```

```
add user script = /usr/sbin/smbldap-useradd -m %u
delete user script = /usr/sbin/smbldap-userdel %u
add group script = /usr/sbin/smbldap-groupadd -p %g
delete group script = /usr/sbin/smbldap-groupdel %g
add user to group script = /usr/sbin/smbldap-groupmod -m %u %g
delete user from group script = /usr/sbin/smbldap-groupmod -x %u %g
set primary group script = /usr/sbin/smbldap-usermod -g %g %u
add machine script = /usr/sbin/smbldap-useradd -w %u

# Le indicamos los usuarios administradores del dominio.
# Esto es muy importante ya que serán los usuarios que utilizaremos para agregar
componentes como las máquinas Windows.
admin users = adminnuevo

# Configuración de la red.
socket options = TCP_NODELAY SO_RCVBUF=8192 SO_SNDBUF=8192

# Interfaces por las que se ofrecerá el servicio de controlador de dominio.
interfaces = eth1 eth2 lo

# Redes u Hosts permitidos.
hosts allow = 127.0.0.1, 10.0.0.0/24

# Redes u Hosts denegados.
hosts deny = 0.0.0.0

# Puertos de escucha.
smb ports = 139 445
bind interfaces only = Yes
name resolve order = wins hosts lmhosts bcast

# Redes para las que se anuncia.
remote announce = 10.0.0.255

# Opciones de contraseñas.
# Permitir el cambio de contraseña.
pam password change = Yes

# Script para cambiar la contraseña.
passwd program = /usr/sbin/smbldap-passwd -u %u

# Formato de interacción con el usuario para cambiar la contraseña.
passwd chat = *Nueva*contraseña* %n\n *Repetir*nueva*contraseña* %n\n
*Contraseña*actualizada*correctamente.*

# Nombre del script para cargar las políticas de inicio de sesión en los clientes
Windows.
logon script = 'logon.bat %U'

# Configuración de los perfiles móviles en los clientes Windows.
# Ruta hacia el perfil.
logon path = \\%N\profiles\%U

# Unidad en la que se montará el /home del usuario.
logon drive = U:

# Permitir login de usuarios del dominio.
domain logons = Yes
```

```
os level = 65
# Configuración como PDC y servidor de WINS
preferred master = Yes
domain master = Yes
dns proxy = No
wins support = Yes
panic action = /usr/share/samba/panic-action %d
map acl inherit = Yes
case sensitive = No
hide unreadable = Yes

# Sincronización de contraseñas entre Unix y el dominio.
unix password sync = Yes

# Configuración del logging.
syslog = 0
log file = /var/log/samba/log.%m
max log size = 1000

# Sincronizar la hora con el servidor PDC
time server = Yes

# Mapear atributos de archivo de Unix a Windows
# Estas opciones requieren que los parametros create mask y directory mask
# tenga activo el bit de ejecución para "grupo" y "otros"
map hidden = Yes
map system = Yes

# Recursos compartidos
# Compartimos los /home de cada usuario.
[homes]
comment = Home Directories
valid users = %S
read only = No
create mask = 0611
directory mask = 0711
browseable = No

# Compartimos las impresoras.
[printers]
comment = All Printers
path = /var/spool/samba
create mask = 0611
directory mask = 0711
printable = Yes
browseable = No

[print$]
comment = Printer Drivers
path = /var/lib/samba/printers
create mask = 0611
directory mask = 0711

# Compartimos el directorio de inicio de sesión de los clientes Windows. En él se
# encuentran los scripts que configurarán nuestros clientes.
[netlogon]
path = /var/lib/samba/netlogon
browseable = No
create mask = 0611
```

```
directory mask = 0711
# Compartimos los perfiles
[profiles]
path = /var/lib/samba/profiles
force user = %U
read only = No
create mask = 0611
directory mask = 0711
guest ok = Yes
profile acls = Yes
browseable = No
csc policy = disable
# Compartimos una carpeta pública de uso común.
[public]
path = /tmp
read only = No
guest ok = Yes
create mask = 0611
directory mask = 0711
```

Ahora crearemos el fichero donde almacenaremos los alias.

```
#nano /etc/samba/smbusers
```

Con el siguiente contenido:

```
root = Administrador
root = Administrator
```

También crearemos el directorio y el script de inicio de sesión para los clientes Windows. Este script no es más que un fichero .bat que nos ayudará a establecer las políticas del dominio. Debe programarse en función de lo que queramos conseguir.

```
#mkdir /var/lib/samba/netlogon
#nano /var/lib/samba/netlogon/logon.bat
```

En el fichero smb.conf se definía con la siguiente línea: “logon script = 'logon.bat %U'”

En este caso tendrá un contenido similar a este:

```
@echo off
net time \\debian-pdc /set /yes
IF %1 == Administrador net use p: \\debian-pdc\root
IF %1 == guest net use p: \\debian-pdc\publico
```

Simplemente sincroniza la hora y comprueba si el usuario es “Administrador” o “invitado”.

Crearemos también el directorio donde tendremos los perfiles para los clientes Windows.

```
#mkdir /var/lib/samba/profiles
```

Y controlamos los permisos de ambos directorios:

```
#chown -Rf root:root /var/lib/samba/netlogon /var/lib/samba/profiles
#chmod 1777 /var/lib/samba/profiles
```

Ahora si comprobamos la configuración de Samba con testparm, deberá devolvernos un mensaje de

la función que desempeñaría nuestro servidor y de que no ha encontrado ningún error:

```
root@kalimdor:~# testparm
Load smb config files from /etc/samba/smb.conf
rlimit_max: rlimit_max (1024) below minimum Windows limit (16384)
Processing section "[homes]"
Processing section "[printers]"
Processing section "[print$]"
Processing section "[netlogon]"
Processing section "[profiles]"
Processing section "[public]"
Loaded services file OK.
Server role: ROLE_DOMAIN_PDC
Press enter to see a dump of your service definitions
_
```

Lo siguiente será reiniciar el servicio.

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

Ahora añadiremos la contraseña del usuario administrador del LDAP a samba para que en un futuro podamos administrar los objetos del directorio a través de samba, lo haremos con “smbpasswd -W”.

```
root@kalimdor:~# smbpasswd -W
Setting stored password for "cn=admin,dc=prueba,dc=com" in secrets.tdb
New SMB password:
Retype new SMB password:
root@kalimdor:~# _
```

La contraseña se almacena en el fichero “/var/lib/samba/passdb.tdb”, comprobaremos que lo hace de forma segura.

```
root@kalimdor:~# ls -l /var/lib/samba/passdb.tdb
-rw----- 1 root root 36864 may 20 10:07 /var/lib/samba/passdb.tdb
root@kalimdor:~# _
```

Sólo root puede leerla y reescribirla.

Así concluimos la instalación y configuración de samba.

## **Instalación y configuración de Smbldaptools**

Nos ofrece la posibilidad de crear el esquema de objetos en el LDAP necesarios para que nuestro servidor funcione como PDC de un dominio Windows.

Además nos facilita los scripts de administración del dominio.

Para instalarlo correremos el siguiente comando:

```
#aptitude install smbldap-tools
```

Al instalarlo no disponemos de ficheros de configuración, deberemos tirar de ficheros de ejemplo.

```
#zcat /usr/share/doc/smbldap-tools/examples/smbldap.conf.gz > /etc/smbldap-
tools/smbldap.conf
#cp /usr/share/doc/smbldap-tools/examples/smbldap_bind.conf /etc/smbldap-
```

```
tools/smbldap_bind.conf
```

Ahora controlaremos que los permisos sean los adecuados para ambos ficheros ya que contendrán críticos como las credenciales del usuario administrador del LDAP.

```
# chmod 640 /etc/smbldap-tools/smbldap.conf /etc/smbldap-tools/smbldap_bind.conf
# chown root:openldap /etc/smbldap-tools/smbldap.conf /etc/smbldap-\
tools/smbldap_bind.conf
```

Para configurar smbldap-tools editaremos inicialmente el fichero smbldap.conf. Deberemos añadirle el SID de nuestro controlador de dominio. Para obtener el SID introduciremos:

```
#net getlocalsid
```

NOTA: Puede que “net getlocalsid” nos informe de algún error, si lo hace comprobad bien el fichero /etc/samba/smb.conf, en especial el campo en que dice cual es el usuario administrador del LDAP. En cualquier caso **SIEMPRE** nos devolverá el SID.

```
root@kalimdor:~# net getlocalsid
SID for domain KALIMDOR is: S-1-5-21-2702463925-462337624-1789547561
root@kalimdor:~# _
```

Y se lo añadimos al fichero smbldap.conf en el que también tendremos que realizar las siguientes configuraciones:

```
#nano /etc/smbldap-tools/smbldap.conf

# El sid obtenido anteriormente.
SID="S-1-5-21-669132894-2586221759-3914214969"

# Nuestro dominio netBIOS
sambaDomain="AZEROTH"

# Informacion del servidor LDAP primario y esclavo, en este caso ambos son el mismo.
slaveLDAP="127.0.0.1"
slavePort="389"
masterLDAP="127.0.0.1"
masterPort="389"

# No utilizar conexión cifrada
ldapTLS="0"

# Sin importancia ya que no se utiliza TLS
verify="require"
cafile="/etc/smbldap-tools/ca.pem"
clientcert="/etc/smbldap-tools/smbldap-tools.pem"
clientkey="/etc/smbldap-tools/smbldap-tools.key"

# Sufijo LDAP
suffix="dc=azeroth,dc=com"

# Donde se almacenan los usuarios, grupos, computadoras y idmapdn
usersdn="ou=people,${suffix}"
computersdn="ou=computer,${suffix}"
groupsdn="ou=group,${suffix}"
idmapdn="ou=idmap,${suffix}"
sambaUnixIdPoolIdn="sambaDomainName=${sambaDomain},${suffix}"

# Default scope
scope="sub"
```



```
# Tipo de cifrado UNIX (CRYPT, MD5, SMD5, SSHA, SHA, CLEARTXT)
hash_encrypt="SSHA"
crypt_salt_format="%s"

# Especifico para cuentas UNIX, shell, ruta al home y demás
userLoginShell="/bin/bash"
userHome="/home/%U"
userHomeDirectoryMode="700"
userGecos="System User"
defaultUserGid="513"
defaultComputerGid="515"
skeletonDir="/etc/skel"
defaultMaxPasswordAge="365"

# Configuración específica para cuentas SAMBA
userSmbHome="//kalimdor/%U"
userProfile="//kalimdor/profiles/%U"
userHomeDrive="U:"
userScript="logon.bat %U"
mailDomain="azeroth.com"

# Especifico de smbldap-tools
with_smbpasswd="0"
smbpasswd="/usr/bin/smbpasswd"
with_slappasswd="0"
slappasswd="/usr/sbin/slappasswd"
```

También editaremos el fichero smbldap\_bind.conf:

```
#nano /etc/smbldap-tools/smbldap_bind.conf

slaveDN="cn=admin,dc=azeroth,dc=com"
slavePw="contrasena del cn=admin, ej: root"
masterDN="cn=admin,dc=azeroth,dc=com"
masterPw="contrasena del cn=admin, ej: root"
```

## **Poblar el directorio LDAP**

Para esto haremos uso de la herramienta `smbldap-populate` que convertirá la estructura de nuestro LDAP en la de un controlador de dominio NetBIOS. Añadirá las siguientes unidades organizativas:

`ou=people`: Definirá a los usuarios.

`ou=Groups`: Definirá a los grupos.

`ou=Computers`: Definirá a las máquinas.

`ou=Idmap`: Realizará el mapeo de cuentas Unix a cuentas Samba.

Por supuesto comprobará anteriormente si existe la base que nos identifica.

Si no existe la creará: `dc=azeroth,dc=com`.

En este caso nos avisa de que la entrada ya existe.

```
Populating LDAP directory for domain AZEROTH (S-1-5-21-2702463925-462337624-1789
547561)
(using builtin directory structure)

entry dc=azeroth,dc=com already exist.
adding new entry: ou=people,dc=azeroth,dc=com
adding new entry: ou=group,dc=azeroth,dc=com
adding new entry: ou=computer,dc=azeroth,dc=com
adding new entry: ou=idmap,dc=azeroth,dc=com
adding new entry: uid=root,ou=people,dc=azeroth,dc=com
adding new entry: uid=nobody,ou=people,dc=azeroth,dc=com
adding new entry: cn=Domain Admins,ou=group,dc=azeroth,dc=com
adding new entry: cn=Domain Users,ou=group,dc=azeroth,dc=com
adding new entry: cn=Domain Guests,ou=group,dc=azeroth,dc=com
adding new entry: cn=Domain Computers,ou=group,dc=azeroth,dc=com
adding new entry: cn=Administrators,ou=group,dc=azeroth,dc=com
adding new entry: cn=Account Operators,ou=group,dc=azeroth,dc=com
adding new entry: cn=Print Operators,ou=group,dc=azeroth,dc=com
adding new entry: cn=Backup Operators,ou=group,dc=azeroth,dc=com
adding new entry: cn=Replicators,ou=group,dc=azeroth,dc=com
entry sambaDomainName=AZEROTH,dc=azeroth,dc=com already exist. Updating it...

Please provide a password for the domain root:
Changing UNIX and samba passwords for root
New password: _
```

Nos pedirá la contraseña de administrador del dominio. En nuestro caso y recordando que esto no es más que una configuración de prueba, le ponemos “root”.

Ahora podemos comprobar los usuarios que existen en el LDAP con el siguiente comando:

```
root@kalimdor:~# pdbedit -L
root:0:root
nobody:65534:nobody
root@kalimdor:~# _
```

También podemos ver los detalles de un usuario en concreto:

```
root@kalimdor:~# pdbedit -Lv root
Unix username:      root
NT username:        root
Account Flags:       [U
User SID:            S-1-5-21-2702463925-462337624-1789547561-500
Primary Group SID:   S-1-5-21-2702463925-462337624-1789547561-513
Full Name:           root
Home Directory:      \\kalimdor\root
HomeDir Drive:       U:
Logon Script:         'logon.bat root'
Profile Path:         \\kalimdor\profiles\root
Domain:              AZEROTH
Account desc:
Workstations:
Munged dial:
Logon time:          0
Logoff time:         never
Kickoff time:        never
Password last set:   dom, 22 may 2011 19:32:40 CEST
Password can change: dom, 22 may 2011 19:32:40 CEST
Password must change: never
Last bad password    : 0
Bad password count   : 0
Logon hours          : FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
root@kalimdor:~# _
```

Y podremos conectarnos para hacer pruebas:

```
root@kalimdor:~# smbclient //localhost/netlogon -U root
Enter root's password:
Domain=[AZEROTH] OS=[Unix] Server=[Samba 3.5.6]
smb: \> exit
root@kalimdor:~# smbclient //localhost/netlogon -U administrator
Enter administrator's password:
Domain=[AZEROTH] OS=[Unix] Server=[Samba 3.5.6]
smb: \> exit
root@kalimdor:~# smbclient //localhost/netlogon -U administrador
Enter administrador's password:
Domain=[AZEROTH] OS=[Unix] Server=[Samba 3.5.6]
smb: \> exit
root@kalimdor:~# _
```

## Configuración de NSS

Para demostrar correctamente el funcionamiento de NSS, vamos a crear un nuevo usuario en el directorio usando smbldap-tools.

```
root@kalimdor:~# smbldap-useradd -a -m -P pepe
Cannot confirm uidNumber 1000 is free: checking for the next one
Changing UNIX and samba passwords for pepe
New password:
Retype new password:
root@kalimdor:~# _
```

Veamos el uidNumber y gidNumber de pepe, lo haremos realizando una búsqueda en el LDAP:

```
#ldapsearch -x -D "cn=admin,dc=azeroth,dc=com" -W -b \
"uid=pepe,ou=People,dc=azeroth,dc=com"
```

```
uid: pepe
uidNumber: 1001
gidNumber: 513
```

Este usuario es sólo para realizar pruebas, más adelante nos centraremos más en las operaciones con usuarios.

Si en este momento creamos un fichero y le asignamos como dueño al usuario “pepe”:

```
root@kalimdor:~# mkdir dirdeprueba
root@kalimdor:~# touch dirdeprueba/fichero
root@kalimdor:~# chown 1001:513 dirdeprueba/fichero
root@kalimdor:~# ls -l dirdeprueba/
total 0
-rw-r--r-- 1 1001 513 0 jun 11 19:46 fichero
root@kalimdor:~# _
```

Podemos comprobar que el sistema no reconoce el uid 1001 y el gid 513. Por lo que no reconoce que el fichero pertenezca al usuario “pepe”.

Para conseguir que lo haga instalaremos el paquete “libnss-ldap”, que permitirá que NSS se conecte al LDAP para extraer la información necesaria.

```
#aptitude install libnss-ldap
```

Nos pedirá el nombre y la contraseña del usuario root del LDAP.

Le ingresamos cualquiera ya que posteriormente vamos a deshabilitar la opción de autenticarse como root.

Una vez instalado nos vamos al fichero de configuración “/etc/libnss-ldap.conf” y comentamos la siguiente línea:

```
# The distinguished name to bind to the server with
# if the effective user ID is root. Password is
# stored in /etc/libnss-ldap.secret (mode 600)
# Use 'echo -n "mypassword" > /etc/libnss-ldap.secret' instead
# of an editor to create the file.
#rootbinddn cn=admin,dc=azeroth,dc=com
```

Y eliminamos el fichero “/etc/libnss-ldap.secret”.

```
#rm -r /etc/libnss-ldap.secret
```

Por último recargamos el servicio nscd que se encarga de cachear este tipo de nombres.

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

Si comprobamos ahora el propietario del fichero, podemos ver que aparece “pepe” como propietario y “Domain Users” como grupo principal.

```
root@kalimdor:~# ls -l dirdeprueba/
total 0
-rw-r--r-- 1 pepe Domain Users 0 jun 11 19:46 fichero
root@kalimdor:~# _
```

## Configuración de PAM

En este apartado utilizaremos la librería libpam-ldap para hacer que nuestros clientes linux se autenticuen usando el directorio LDAP.

Lo primero que se hará es salvar el estado de nuestro directorio /etc/pam.d/

```
#cp -r /etc/pam.d /etc/pam.d.old
```

Con esto nos aseguraremos de que si algo falla durante la instalación de libpam-ldap o la configuración de PAM, no nos deje el sistema de acceso irrecoverable.

Procedemos a instalar libpam-ldap

```
#aptitude install libpam-ldap
```

Nos hará las siguientes preguntas:

- URI del servidor LDAP: **ldapi:///**
- DN del servidor LDAP: **dc=azeroth,dc=com**
- Versión de LDAP a utilizar: **LDAPv3**
- ¿Permitir que la cuenta de root del LDAP se comporte como cuenta de root local?: **No**
- ¿Hace falta un usuario para acceder a las entradas del LDAP?: **No**

El siguiente paso es hacer que cuando un usuario se loguee, se cree su /home si este no existe.

Para ello vamos a añadir la siguiente línea al fichero “/etc/pam.d/common-session”:

```
# here are the per-package modules (the "Primary" block)
session [default=1] pam_permit.so
# here's the fallback if no module succeeds
session requisite pam_deny.so
# prime the stack with a positive return value if there isn't one already;
# this avoids us returning an error just because nothing sets a success code
# since the modules above will each just jump around
session required pam_permit.so
# and here are more per-package modules (the "Additional" block)
session required pam_unix.so
session optional pam_ldap.so
# end of pam-auth-update config
session required pam_mkhomedir.so
```

## Gestión de usuarios y grupos

### Creación de usuarios

Como vimos en el ejemplo anterior, a partir de ahora, cada vez que queramos crear un usuario del dominio habrá que hacerlo de la siguiente forma:

```
#smbldap-useradd -a -m -P <nombre_usuario>
```

- -a indica que el usuario podrá iniciar sesión en equipos Windows.
- -m indica que se creará su /home en el servidor.
- -P hará que nos pida una contraseña por teclado.

La primera vez que creamos un usuario, nos saldrá un mensaje que nos indica que no se puede confirmar que el uidNumber 1000 esté libre. Este mensaje no volverá a aparecer en futuras creaciones ya que tomará como referencia el uidNumber del último usuario que se añadió al directorio.

Ahora veremos cómo hacer Administradores del dominio a nuestros usuarios. Estos usuarios nos permitirán añadir máquinas Windows al dominio.

```
#smbldap-useradd -a -m -G 512 -P <nombre_usuario>
```

Básicamente es lo mismo que antes, pero indicándole el gidNumber 512, que hace referencia al grupo “Administradores del dominio”. También tenemos que editar el fichero “/etc/samba/smb.conf” y añadir al campo “admin users” el nombre del nuevo usuario administrador.

```
root@kalimdor:~# smbldap-useradd -a -m -G 512 -P admin dominio
Changing UNIX and samba passwords for admin dominio
New password:
Retype new password:
root@kalimdor:~# _
```

Para borrar un usuario lo haremos con smbldap-userdel.

```
#smbldap-userdel -r <nombre_usuario>
```

Con el parámetro -r borramos el /home del usuario.

### Creación de grupos

El proceso de creación de grupos es básicamente igual al de los usuarios. Para crearlo ejecutamos lo siguiente:

```
#smbldap-groupadd -a <nombre_grupo>
```

Puesto que no es un grupo local, no aparecerá en /etc/passwd, para verlo debemos hacerlo con el comando getent.

```
#getent group |grep <nombre_grupo>
```

Para borrar un grupo, utilizaremos smbldap-groupdel.

```
#smbldap-groupdel <nombre_grupo>
```

## Agregar usuario a un grupo

Para agregar un usuario a un grupo lo haremos con el siguiente comando:

```
#smbldap-usermod -group <nombre_grupo> <nombre_usuario>
```

Borrar un usuario de un grupo

Para eliminar un usuario de un grupo simplemente le pasamos el parámetro “-”.

```
#smbldap-usermod -group -<nombre_grupo> <nombre_usuario>
```

## Migración de usuarios locales al LDAP

Puede ocurrir que queramos conservar los usuarios y grupos de nuestro sistema operativo en el directorio para luegoarnos remotamente.

Para que esto ocurra debemos migrarlos al servidor. SmbLdapTools nos provee unos scripts que nos ayudarán también con esta función. Los scripts son los siguientes:

```
#zcat /usr/share/doc/smbldap-tools/examples/migration_scripts/smbldap-migrate-unix-\
accounts.gz > /usr/sbin/smbldap-migrate-unix-accounts
#zcat /usr/share/doc/smbldap-tools/examples/migration_scripts/smbldap-migrate-unix-\
groups.gz > /usr/sbin/smbldap-migrate-unix-groups
#chmod 755 /usr/sbin/smbldap-migrate-unix-accounts /usr/sbin/smbldap-migrate-unix-\
groups
```

Para ejecutarlos, lo haremos de la siguiente forma:

- Copiamos /etc/passwd y /etc/shadow a un directorio temporal.

```
#cp /etc/passwd /etc/shadow /tmp/
```

- Borramos de los ficheros temporales los usuarios que no queremos incluir en el directorio y ejecutamos el script correspondiente a los usuarios.

```
#smbldap-migrate-unix-accounts -a -P /tmp/passwd -S /tmp/shadow
```

Con esto migramos los usuarios, ahora migraremos los grupos:

- Volvemos a copiar el fichero local a un directorio temporal.

```
#cp /etc/group /tmp/
```

- Eliminamos los grupos que no nos hacen falta y ejecutamos el script.

```
# smbldap-migrate-unix-groups -a -G /tmp/group
```

Ahora sólo queda que los usuarios vuelvan a pertenecer a los grupos como anteriormente lo hacían en el sistema local. Como por ejemplo:

```
#smbldap-usermod --group audio,video,floppy,cdrom,plugdev,"Domain Users" pepe
```

## **Preparar los clientes**

Como era de esperar, son procedimientos distintos para máquinas clientes Linux que para Windows. Por lo que explicaremos los cambios que debe haber en cada una de ellas.

Antes de instalar ni configurar ningún componente, debemos estar seguros de que la configuración de red es la correcta.

## **Clientes Linux**

Los cambios en estas máquinas se parecen mucho a los realizados en el servidor durante la instalación del PDC. Tenemos que instalar y configurar los siguientes componentes:

- libnss-ldap
- libpam-ldap
- libpam-mkhomedir

La configuración es básicamente la misma excepto en la dirección del servidor LDAP en la que responderemos la IP del servidor en lugar de “localhost” o el socket Unix utilizado anteriormente.

## **Creación de un paquete Debian**

Es totalmente contraproducente repetir lo mismo en todos los clientes Linux teniendo en cuenta lo incómodo que es tocar los ficheros de configuración.

Para simplificar esto, vamos a crear un paquete .deb que nos configure y dependa de los paquetes necesarios.

Lo primero que vamos a hacer es crear un directorio de trabajo en /root por ejemplo.

```
#cd /root
#mkdir sys-ldap
#cd sys-ldap
```

Ahora, dentro de este, debemos crear principalmente dos directorios, “control” y “data”, este último en realidad no lo vamos a utilizar, pero forma parte de la estructura estándar de un paquete Debian.

```
#mkdir control data
```

También crearemos un fichero llamado “debian-binary”, con el contenido “2.0”.

Lo mejor es hacerlo de la siguiente forma:

```
#echo “2.0” >debian-binary
```

Una vez terminado con la estructura de directorios, comenzaremos a explicar el funcionamiento de cada uno de ellos.

Empezamos por “control”. Dentro de este, debemos crear otro fichero con el mismo nombre y la siguiente estructura:



```
root@rasganorte:~/sys-ldap# cat control/control
Package: sys-ldap
Architecture: all
Version: 0.1
Maintainer: José Manuel Ferrete Benítez <ferretel1989@gmail.com>
Section: net
Priority: extra
Homepage: http://virtuatopedia.blogspot.com
Depends: libnss-ldap, libpam-ldap, libpam-mkhomedir
Installed-Size: 4
Description: Paquete de autoconfiguración para la integración de una máquina en
un sistema de cuentas centralizadas con autenticación ldap.
root@rasganorte:~/sys-ldap# █
```

Aunque presenta una información bastante lógica, vamos a repasar cada uno de los campos.

Package – Representa el nombre del paquete.

Architecture – Hace referencia a las arquitecturas soportadas.

Version – La versión actual del paquete.

Maintainer – El mantenedor del paquete, al que se puede escribir o contactar si se detectan bugs.

Section – El tipo de función para la que está destinado este paquete.

Priority – Es la prioridad, en este caso, como no es un paquete necesario, se establece a extra.

Homepage – La web del paquete.

Depends – Información muy importante ya que hace referencia a los paquetes necesarios para que este funcione correctamente.

Installed-Size – El espacio en disco una vez instalado.

Description – La descripción de nuestro paquete.

En este directorio también se pueden crear los siguientes ficheros:

- preinst – Script que se ejecutará antes de la instalación.
- posinst – Script que se ejecutará después de la instalación.
- prerm – Script que se ejecutará antes de la desinstalación.
- postrm – Script que se ejecutará después de la desinstalación.

Estos scripts, son escritos en BASH y facilita la configuración del paquete. De hecho, necesitaremos utilizar dos de ellos, postinst y postrm.

Este será el contenido de nuestro fichero postinst:

```
#nano postinst
#!/bin/bash

#Pedimos los datos necesarios.
read -p "Dirección del servidor LDAP: " LDAP
read -p "Base de búsqueda: " BASE
read -p "Usuario administrador del ldap: " ADMIN
read -s -p "Contraseña: " PASSWD
```

```
#Creamos una copia de seguridad de los ficheros de configuración.
#Posteriormente, aplicamos los cambios pertinentes.

if [ -f /etc/ldap/ldap.conf ]
then
    mv /etc/ldap/ldap.conf /etc/ldap/ldap.conf.old
    echo "BASE $BASE" >/etc/ldap/ldap.conf
    echo "URI ldap://$LDAP" >>/etc/ldap/ldap.conf
fi

if [ -f /etc/nsswitch.conf ]
then
    cp /etc/nsswitch.conf /etc/nsswitch.conf.old
    sed -ie 's:compat:compat ldap:g' /etc/nsswitch.conf
    sed -ie 's:files dns:files wins dns:g' /etc/nsswitch.conf
fi

if [ -f /etc/libnss-ldap.conf ]
then
    mv /etc/libnss-ldap.conf /etc/libnss-ldap.conf.old
    echo "base $BASE" >/etc/libnss-ldap.conf
    echo "uri ldap://$LDAP" >>/etc/libnss-ldap.conf
    echo "ldap_version 3" >>/etc/libnss-ldap.conf
    echo "rootbinddn $ADMIN" >> /etc/libnss-ldap.conf
    echo "bind_policy soft" >>/etc/libnss-ldap.conf
    echo "pam_password crypt" >>/etc/libnss-ldap.conf
    echo "nss_base_passwd $BASE?sub" >>/etc/libnss-ldap.conf
    echo "nss_base_shadow $BASE?sub" >>/etc/libnss-ldap.conf
    echo "nss_base_group ou=group,$BASE?one" >>/etc/libnss-ldap.conf
    echo $PASSWD >/etc/libnss-ldap.secret
fi

if [ -f /etc/pam_ldap.conf ]
then
    mv /etc/pam_ldap.conf /etc/pam_ldap.conf.old
    echo "base $BASE" >/etc/pam_ldap.conf
    echo "uri ldap://$LDAP" >>/etc/pam_ldap.conf
    echo "ldap_version 3" >>/etc/pam_ldap.conf
    echo "rootbinddn $ADMIN" >> /etc/pam_ldap.conf
    echo "bind_policy soft" >>/etc/pam_ldap.conf
    echo "pam_password crypt" >>/etc/pam_ldap.conf
    echo "nss_base_passwd $BASE?sub" >>/etc/pam_ldap.conf
    echo "nss_base_shadow $BASE?sub" >>/etc/pam_ldap.conf
    echo "nss_base_group ou=group,$BASE?one" >>/etc/pam_ldap.conf
    echo $PASSWD >/etc/pam_ldap.secret
fi

cp -r /etc/pam.d /etc/pam.d.old
echo "session required pam_mkhomedir.so" >>/etc/pam.d/common-session
/etc/init.d/nscd restart
```

Y este será el del fichero postrm:

```
#nano postrm
#!/bin/bash

#Reemplazamos los ficheros de configuración antiguos por los actuales.
mv /etc/ldap/ldap.conf.old /etc/ldap/ldap.conf
mv /etc/nsswitch.conf.old /etc/nsswitch.conf
mv /etc/libnss-ldap.conf.old /etc/libnss-ldap.conf
mv /etc/pam-ldap.conf.old /etc/pam-ldap.conf
echo "" >/etc/libnss-ldap.secret
echo "" >/etc/pam_ldap.secret
mv /etc/pam.d.old /etc/pam.d
```

Ahora comenzaremos a empaquetar el contenido del .deb, para ello dentro de “control”, ejecutamos el siguiente comando:

```
#tar -czvf control.tar.gz ./
```

Puesto que no necesitamos pasar ningún fichero desde el paquete durante su instalación, en el directorio “data”, lo dejamos en blanco y ejecutamos el mismo comando.

```
#tar -czvf data.tar.gz ./
```

Ahora movemos los .tar al directorio padre y empaquetamos.

```
#cd /root/sys-ldap
#mv control/control.tar.gz ./
#mv data/data.tar.gz ./
#ar -rc sys-ldap_0.1_all.deb debian-binary control.tar.gz data.tar.gz
```

El nombre del .deb no puede ser cualquiera, debe seguir la siguiente estructura:

paquete\_version\_arquitectura.deb

Ahora, como todo paquete debian, para instalarlo:

```
#dpkg -i sys-ldap_0.1_all.deb
```

Si vemos el proceso de instalación del paquete:

```
root@rasganorte:~/sys-ldap# dpkg -i sys-ldap_0.1_all.deb
Seleccionando el paquete sys-ldap previamente no seleccionado.
(Leyendo la base de datos ... 68342 ficheros o directorios instalados actualment
e.)
Desempaquetando sys-ldap (de sys-ldap_0.1_all.deb) ...
Configurando sys-ldap (0.1) ...
Dirección del servidor LDAP: 10.0.0.1
Base de búsqueda: dc=azeroth,dc=com
Usuario administrador del ldap: cn=admin,dc=azeroth,dc=com
Contraseña: Restarting Name Service Cache Daemon: nscd.
root@rasganorte:~/sys-ldap# █
```

Podemos comprobar que se instala igual que cualquier otro.

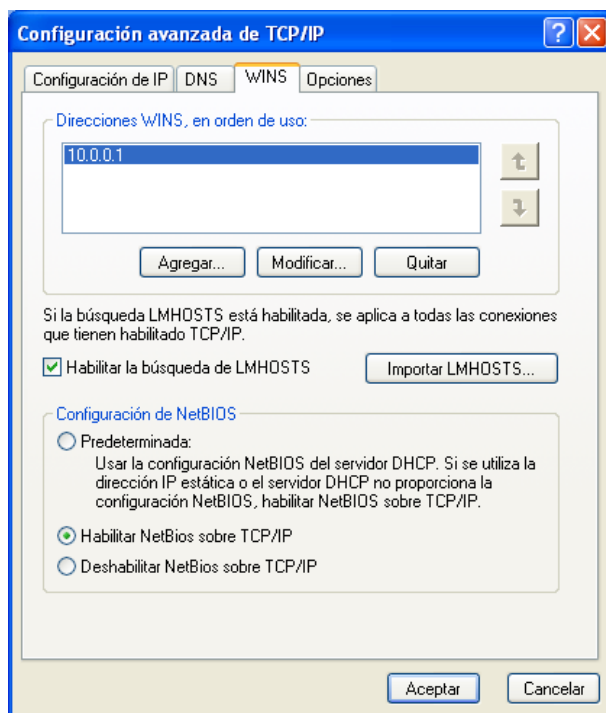
Para desinstalarlo:

```
#dpkg -r sys-ldap
```

## Clientes Windows

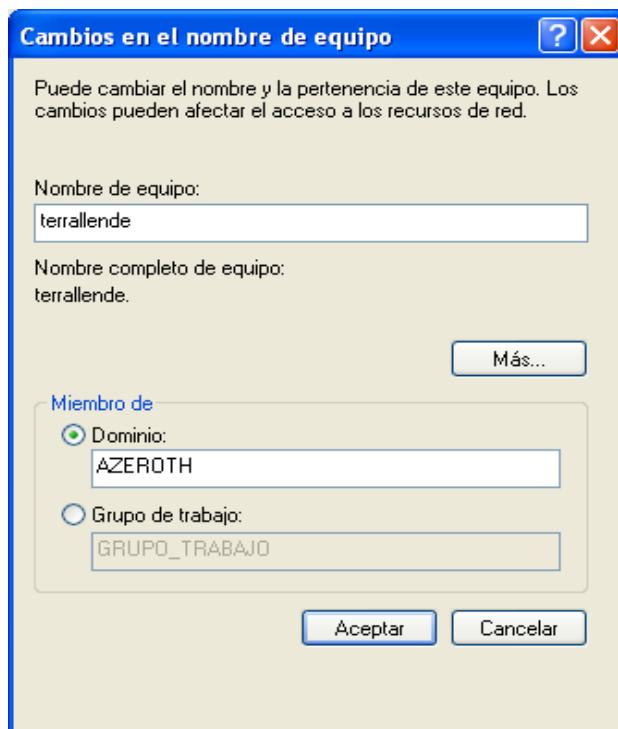
En este caso si cambia radicalmente la configuración, ya que se trata de un sistema totalmente distinto. El ejemplo lo veremos primero con la máquina Windows XP.

Una vez que tenemos configurada la IP y el nombre de host, nos vamos a “Opciones avanzadas” en las propiedades del dispositivo de red y añadimos nuestro servidor Samba como servidor WINS. También debemos habilitar NetBIOS sobre TCP/IP.

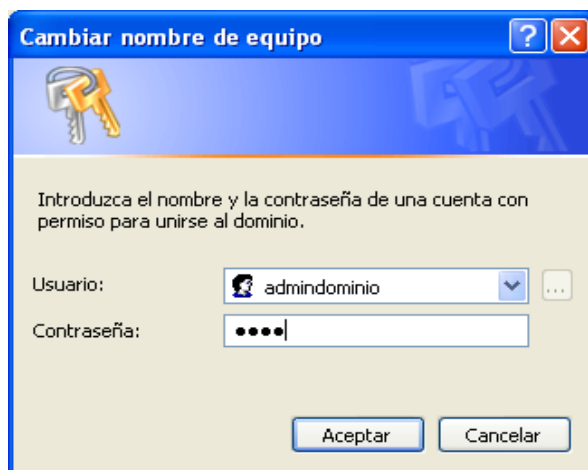


Una vez hecho esto, hacemos clic con el botón secundario del ratón sobre “Mi pc” y accedemos a las propiedades.

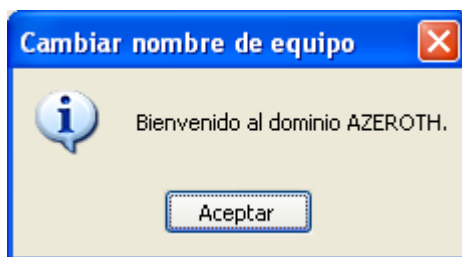
En la pestaña “Nombre del equipo”, hacemos clic en “Cambiar”, y establecemos “AZEROTH” como grupo de trabajo.



Ahora al hacer clic en “Aceptar” nos pedirá un nombre de usuario. Le indicaremos “admindominio” que es el que creamos anteriormente, pero podemos indicarle cualquiera que pertenezca al grupo “Administradores del dominio”.



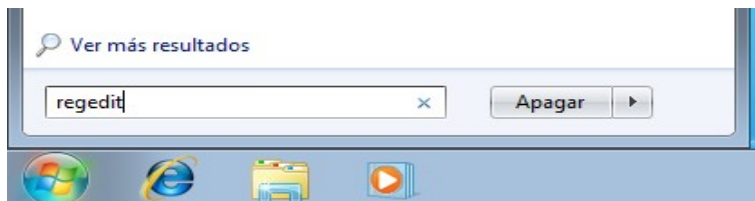
Y finalmente nos aparecerá la pantalla de bienvenida.



## Cambios para clientes Windows 7

De forma predeterminada, Windows 7 no acepta la autenticación NTLM, para habilitarla hay que realizar algunos cambios en el registro, el procedimiento sería el siguiente:

1. Vamos a Inicio => regedit.



2. En el “Editor del Registro”, nos vamos a :

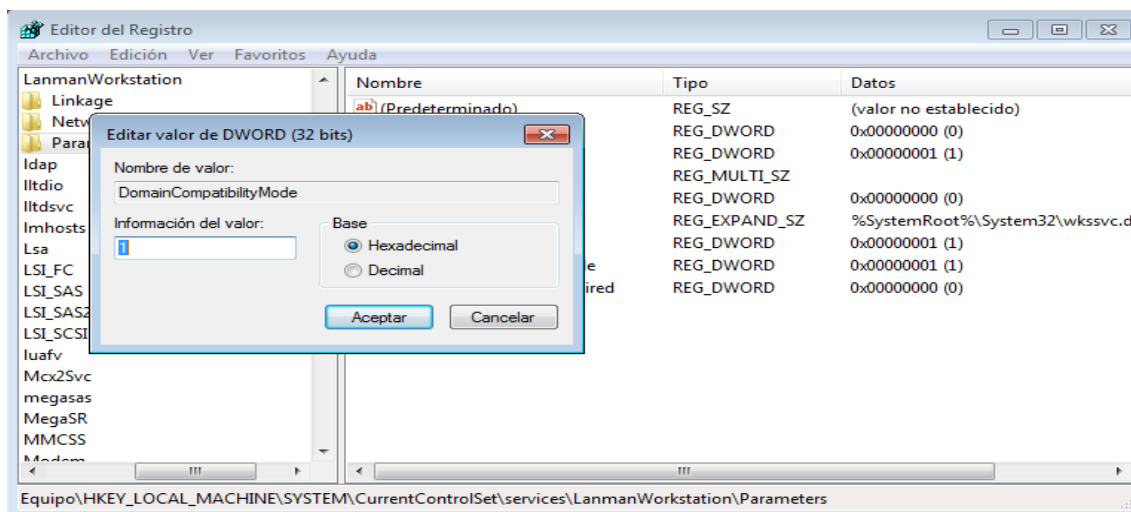
Equipo\HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\services\LanmanWorkstation

3. Agregar las siguientes claves:

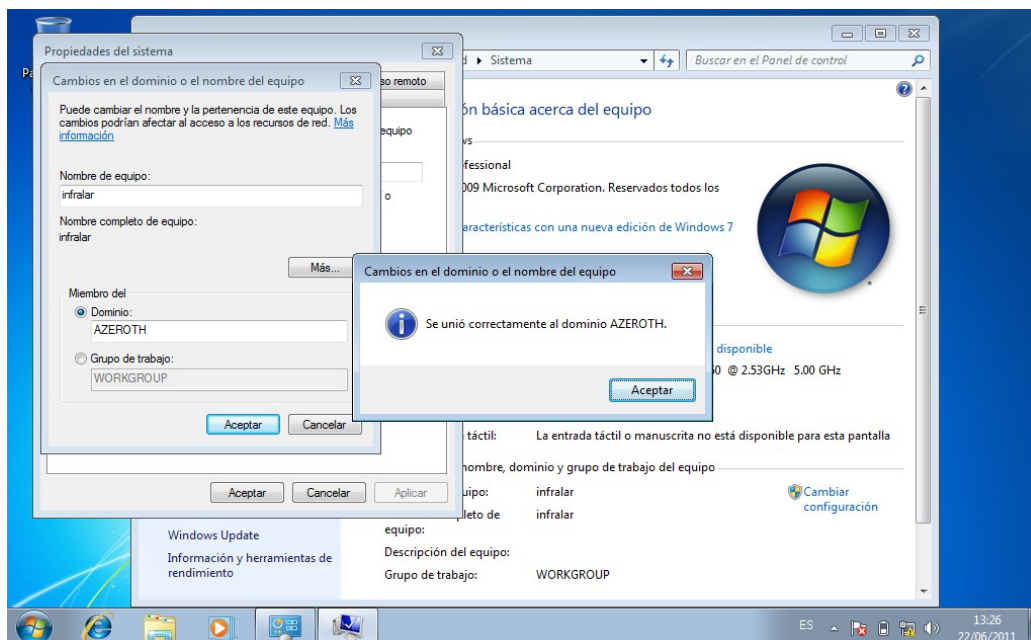
DWORD DomainCompatibilityMode = 1

DWORD DNSNameResolutionRequired = 0

4. Para ello, hacemos clic con el botón secundario en “Parameters” y seleccionamos Nuevo => Valor de DWORD (32 bits)



Para finalizar la configuración de Windows 7, seguimos los mismos pasos que en Windows XP y obtendremos el siguiente mensaje de bienvenida al dominio.



## Login de usuarios

El login de nuestros usuarios lo haremos de la misma forma que hasta ahora en los equipos linux sin interfaz.

Si nos fijamos, podemos ver como se crea el directorio /home/pepe, esto ha sido gracias a la línea que añadimos anteriormente al fichero "/etc/pam.d/common-session" correspondiente al módulo "pam\_mkhomedir.so".


```
root@kalimdor:~# login pepe
Contraseña:
Último inicio de sesión:sáb jun 11 19:36:00 CEST 2011en tty2
Linux kalimdor 2.6.32-5-amd64 #1 SMP Mon Mar 7 21:35:22 UTC 2011 x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Creando directorio '/home/pepe'.
pepe@kalimdor:~$ _
```

En los clientes Linux con interfaz, simplemente ingresamos el usuario y la contraseña correspondiente.

Veamos el inicio en “rasganorte”:



rasganorte

Otro...

Usuario: pepe

Cancelar Iniciar sesión

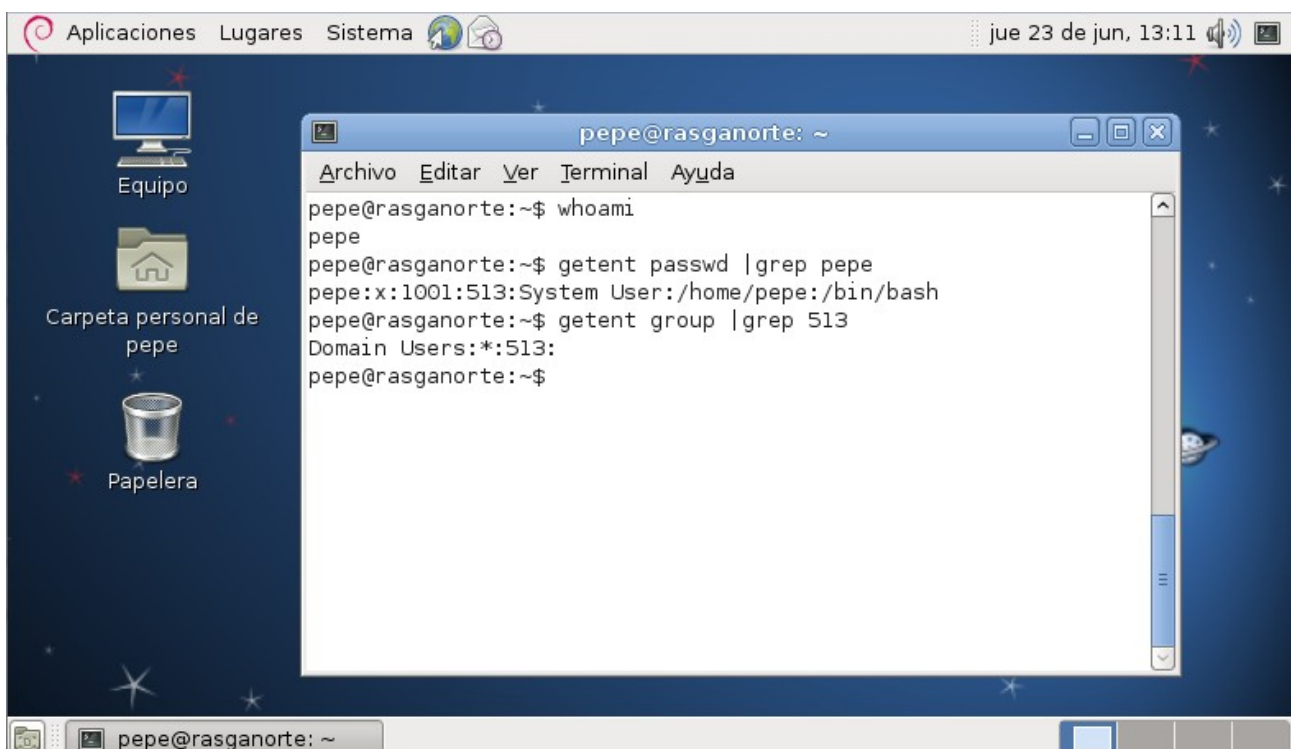


rasganorte

Otro...

Contraseña: ●●●●

Cancelar Iniciar sesión





En cuanto a los usuarios Windows, veremos como se inicia sesión en ambas máquinas (“terrallende” e “infralar”).

Primero veremos el inicio de sesión en “terrallende” (Windows XP):

Al iniciar el sistema, la primera pantalla que veremos será esta:



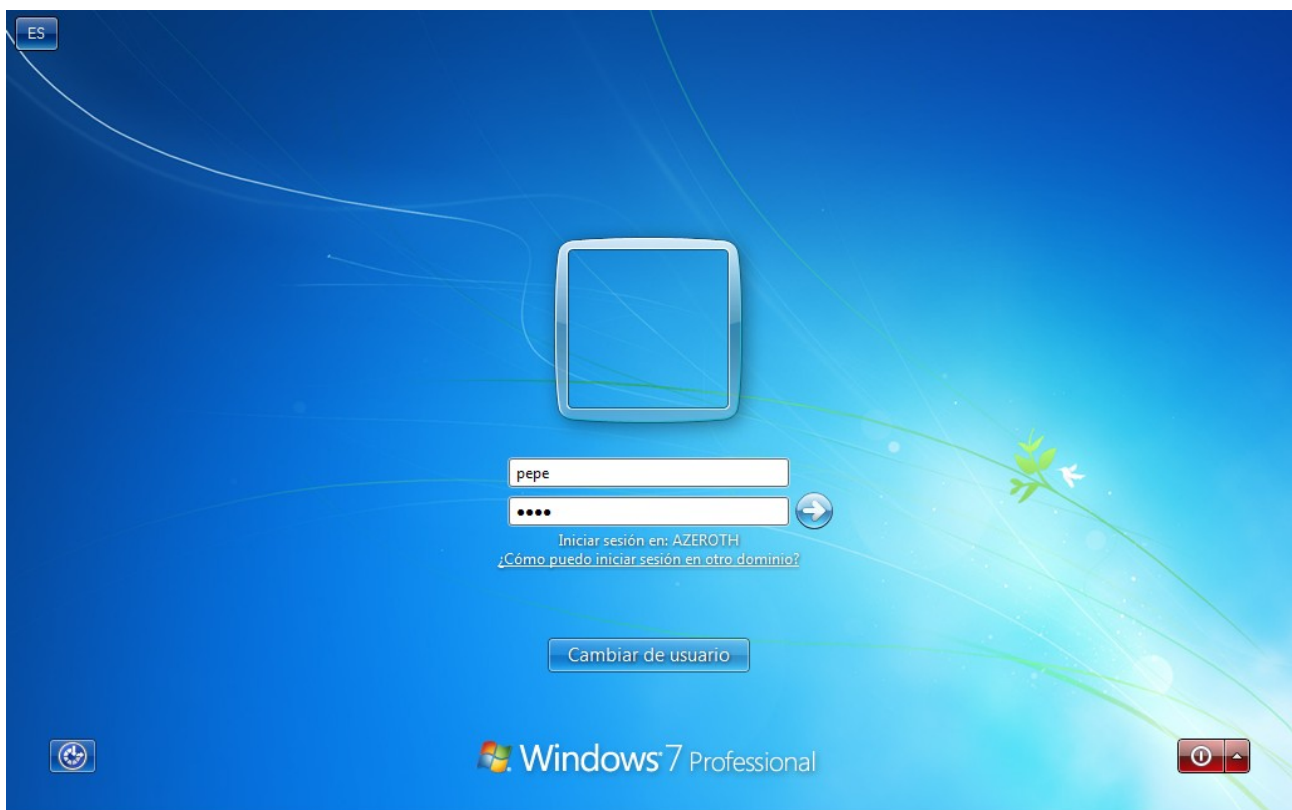
Hacemos clic en opciones y se nos desplegará un menú en el que podremos elegir a qué dominio conectarnos.



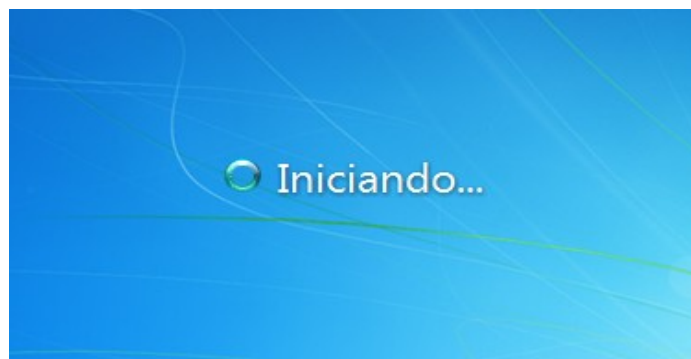
Dependiendo de los recursos del sistema, se nos cargará un escritorio gráficamente más elegante o no. En este caso, como se trata de una máquina virtual, se nos carga el estilo antiguo de Windows.



En cuanto a la máquina infralar, es incluso más fácil, simplemente ingresamos el usuario y la contraseña.



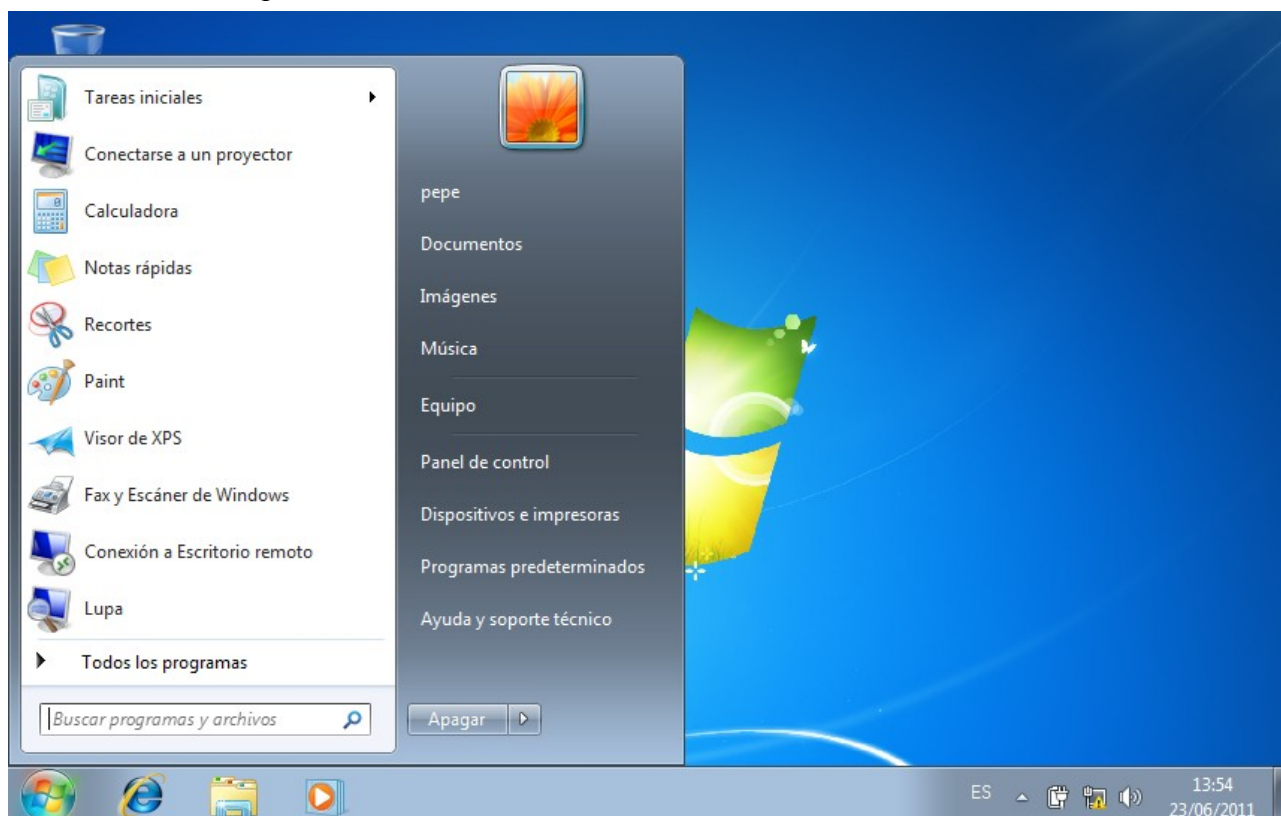
Nos aparecerá el mensaje “Iniciando...”.



Y el siguiente, que nos indica que el equipo está preparando el entorno.



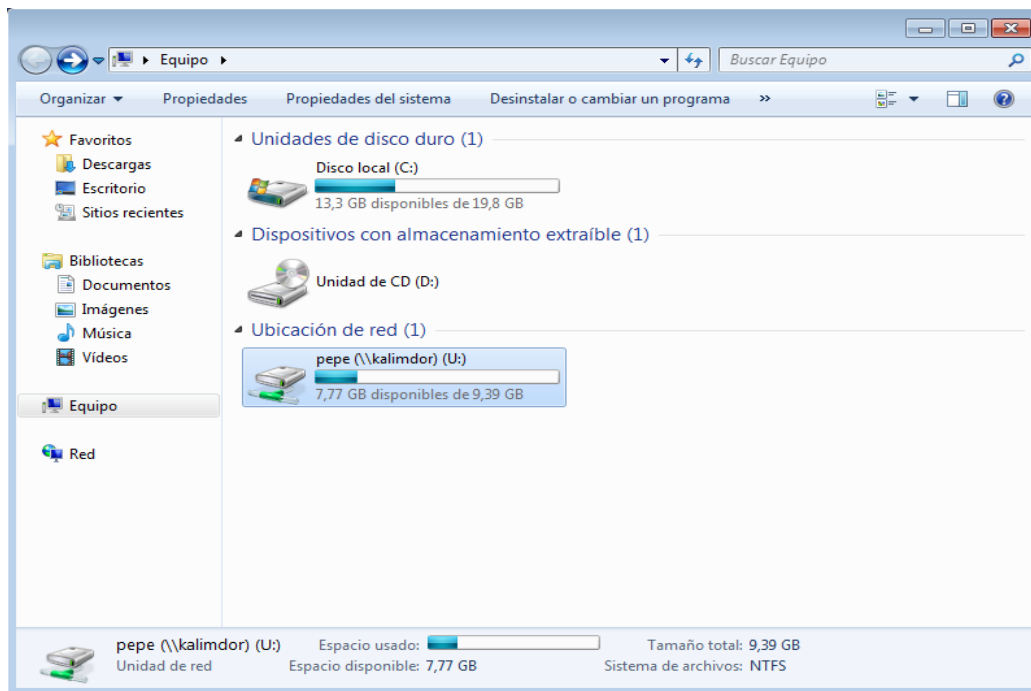
Este es el escritorio por defecto.



## Resultado

Como resultado, obtenemos la correcta integración de los equipos en el dominio y la posibilidad de compartir ficheros entre distintos sistemas.

Como ejemplo de esto pongo la siguiente captura, en la que se puede observar cómo el usuario “pepe” obtiene acceso a su /home de Linux, compartido por CIFS. Este home se monta automáticamente a la hora de iniciar sesión tanto en los equipos Linux, como Windows.



## Administración mediante herramientas externas

En esta sección veremos un ejemplo de administración utilizando una herramienta que opera directamente sobre el directorio LDAP sin importar el sistema en el que esté implantado.

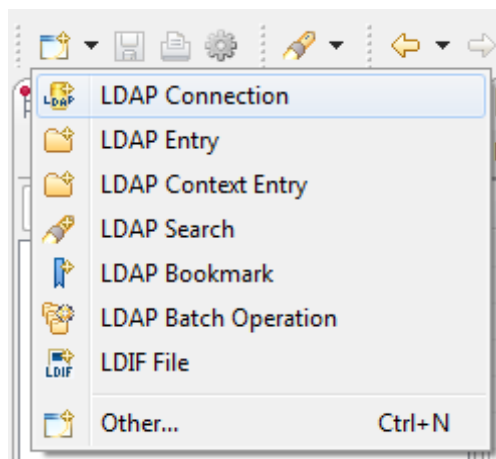
En este caso utilizaremos Apache Directory Studio.



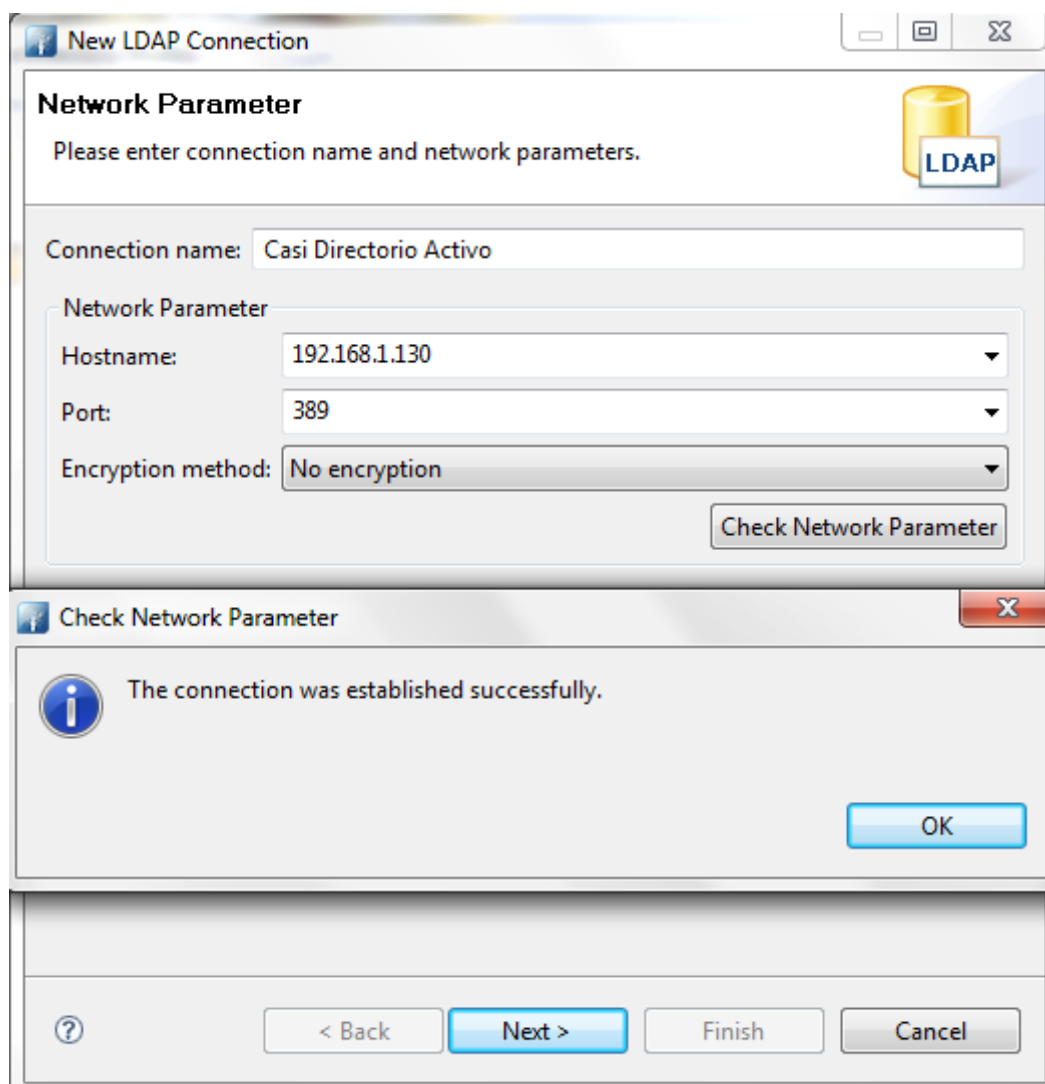
Plantearemos un posible problema a la hora de integrar una máquina en el dominio.

La solución podría ser eliminar esa máquina del directorio y volver a integrarla, para ello nos conectamos usando los siguientes datos.

En la pantalla de inicio, hacemos clic en el botón “Nuevo” y “LDAP Connection”.

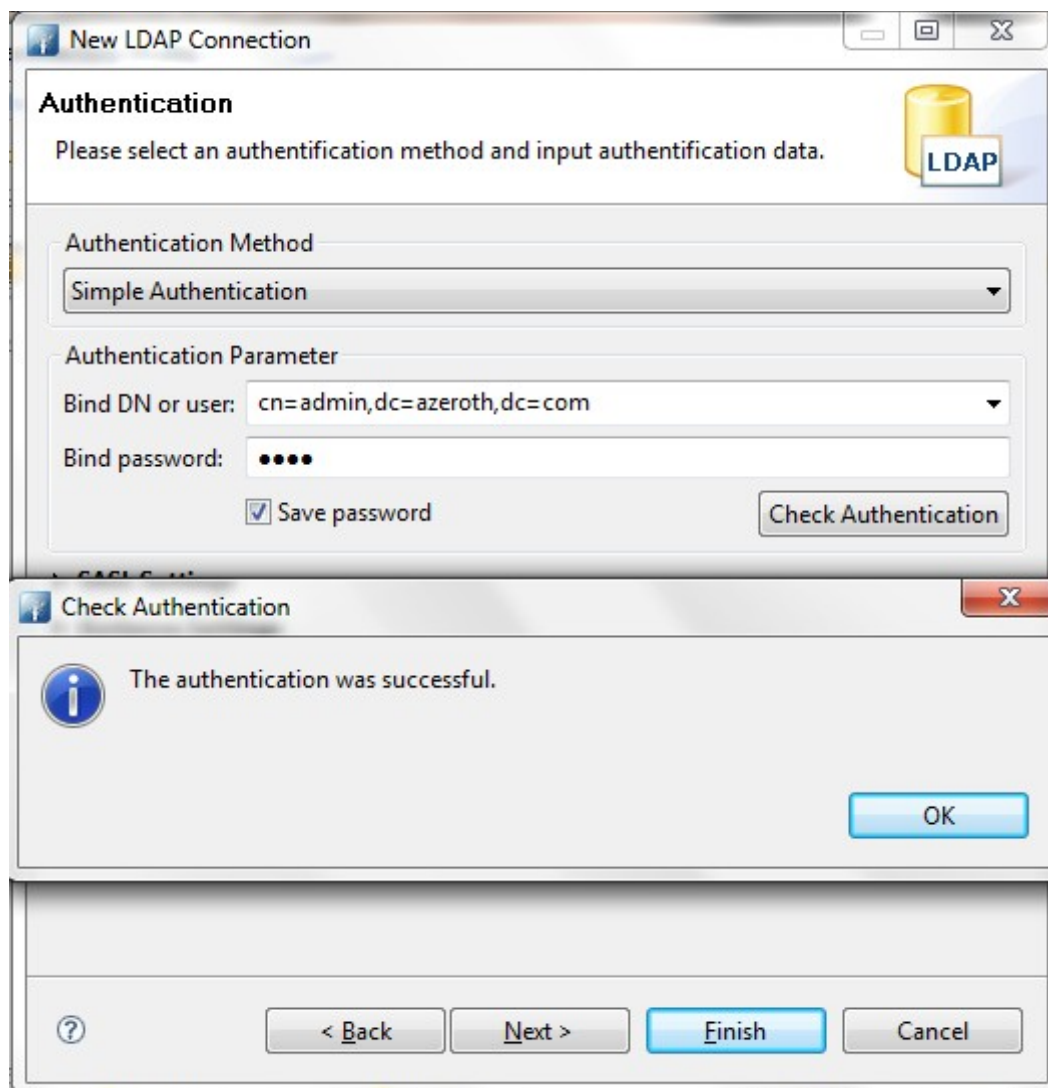


Nos pedirá los primeros datos de conexión, como la dirección del LDAP y el nombre de la conexión.



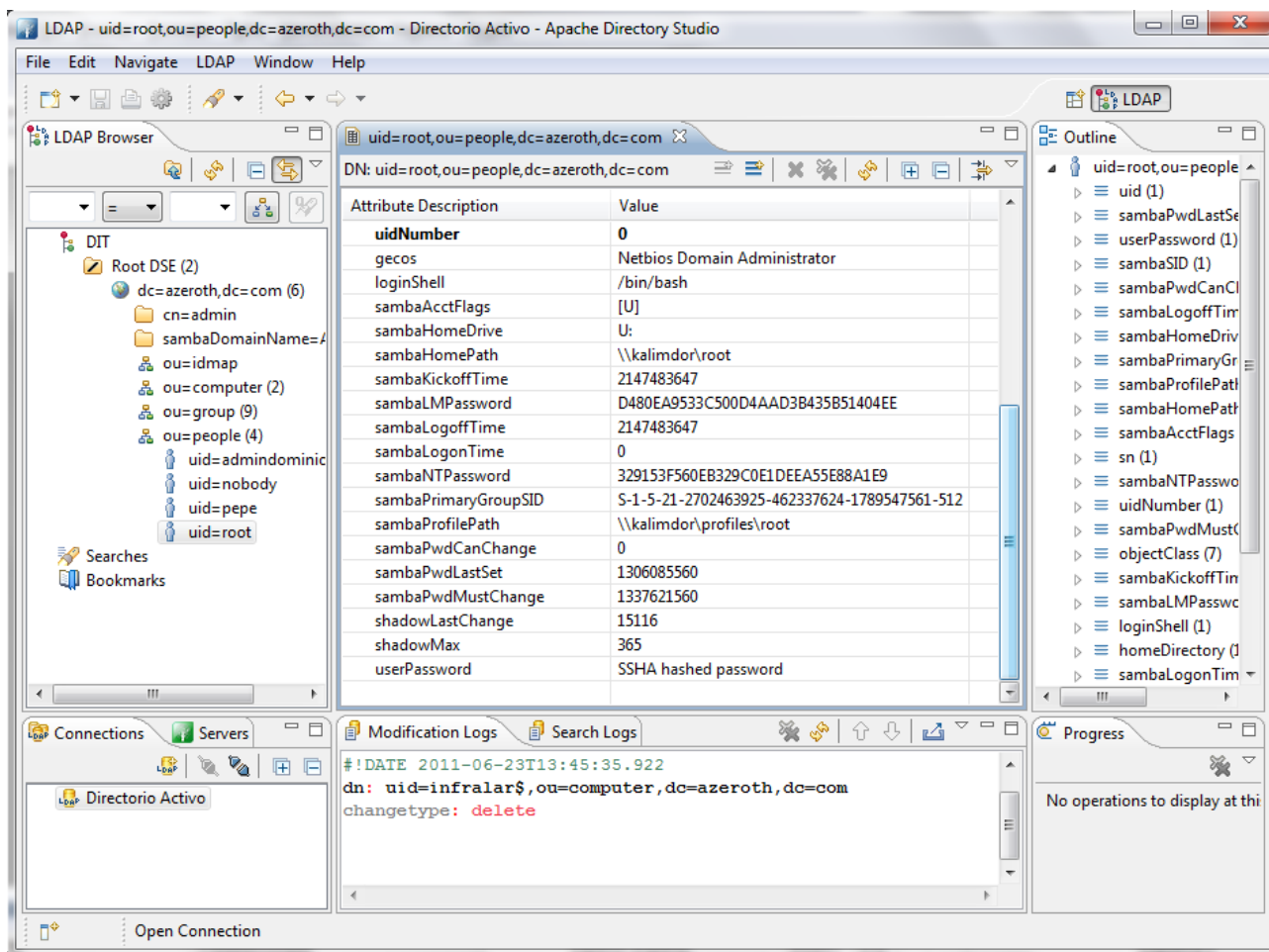
Hacemos clic en "Next >".

En la siguiente pantalla nos pedirá el usuario con el que conectarnos, y la contraseña.



Hacemos clic en “Finish”.

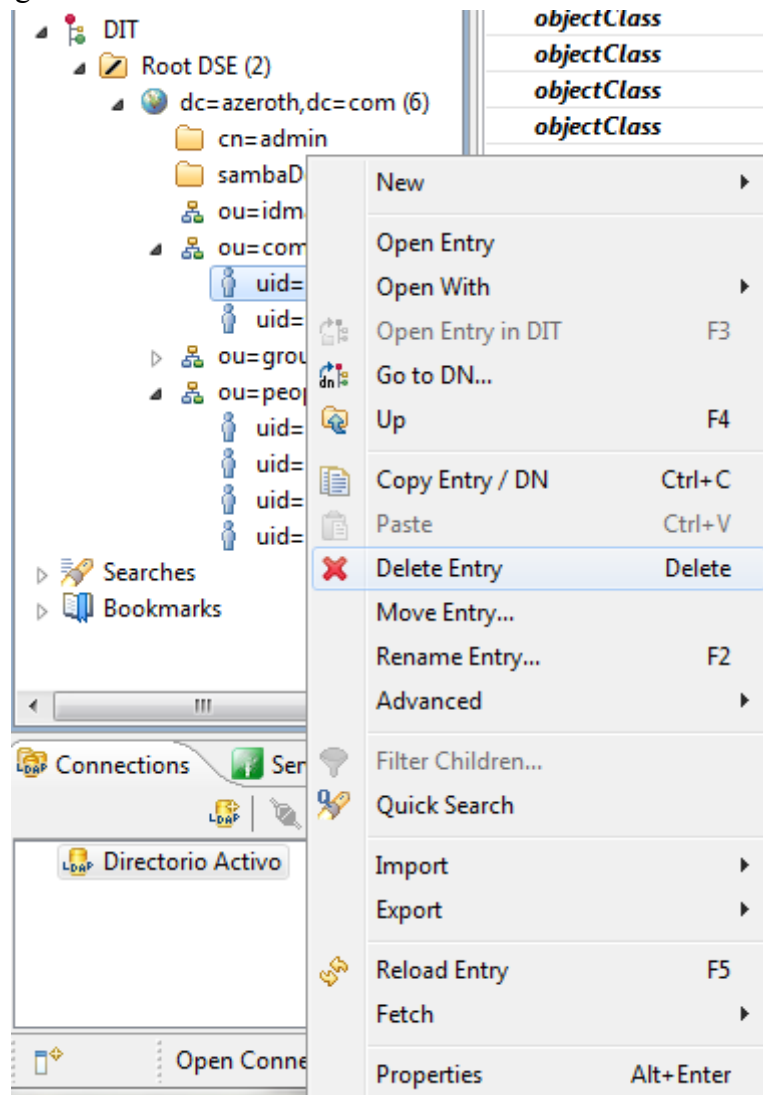
Este es el resultado.



Podemos ver la estructura del directorio, en la que aparecen los distintos tipos de objetos.



Para realizar cualquier acción simplemente hacemos clic con el botón secundario en cualquier elemento y se nos desplegará un menú con las opciones disponibles, en este caso podremos eliminar el equipo que se integró inadecuadamente.



Esto es todo en cuanto a la administración mediante herramientas externas.

## Conclusión y posibles mejoras

Como conclusión podemos sacar que es totalmente fiable y seguro utilizar estas herramientas para crear un controlador de dominio. No tiene nada que envidiar a los sistemas de servidores Windows y como ventaja mayoritaria obtenemos el ahorro de dinero que incluye el utilizar software libre.

Estoy seguro de que una vez expuesto este proyecto a casi cualquier empresa que necesite de un PDC, sería capaz de ver también las ventajas de montarlo con estas características.

Durante el desarrollo de este proyecto hemos aprendido a utilizar debidamente Samba, lo cual personalmente estoy muy orgulloso porque desde un principio pensé que era una herramienta muy interesante y útil.

También se ha visto cómo funciona el modo de inicio en los clientes Windows cuando estos están integrados en un dominio. Haciendo uso del recurso compartido netlogon, el cual desconocía.

Como posible mejora de seguridad, aunque con NTLM obtenemos un alto nivel, podemos mejorarlo incluyendo Kerberos a nuestra configuración.

Esto, actualmente no es posible puesto que Samba no admite autenticación de sus clientes mediante Kerberos, pero en su versión 4, la cual está en alfa aun, si lo permitirá.

Entonces se convertiría en una completa opción para reemplazar a Directorio Activo.

Para finalizar quiero decir que es bastante interesante conocer el funcionamiento de Samba ya que actualmente existen muchísimas empresas que quieren establecer este sistema de cuentas centralizadas y no conocen la existencia de esta alternativa.

## Referencias

Parte de la documentación utilizada para el desarrollo de este proyecto se ha obtenido de los siguientes enlaces:

<http://informatica.gonzalonazareno.org/plataforma/>

<http://albertomolina.wordpress.com/>

<http://www.josedomingo.org/web/>

<http://www.server-world.info/>

<http://www.esdebian.org/wiki/>

<http://linuca.org/>

<http://www.google.es/>