TEMA 3

Gestión de usuarios, NFS y LDAP

LDAP

P. Ruíz

http://somebooks.es/sistemas-operativos-red-2a-

    - Cuentas de usuario

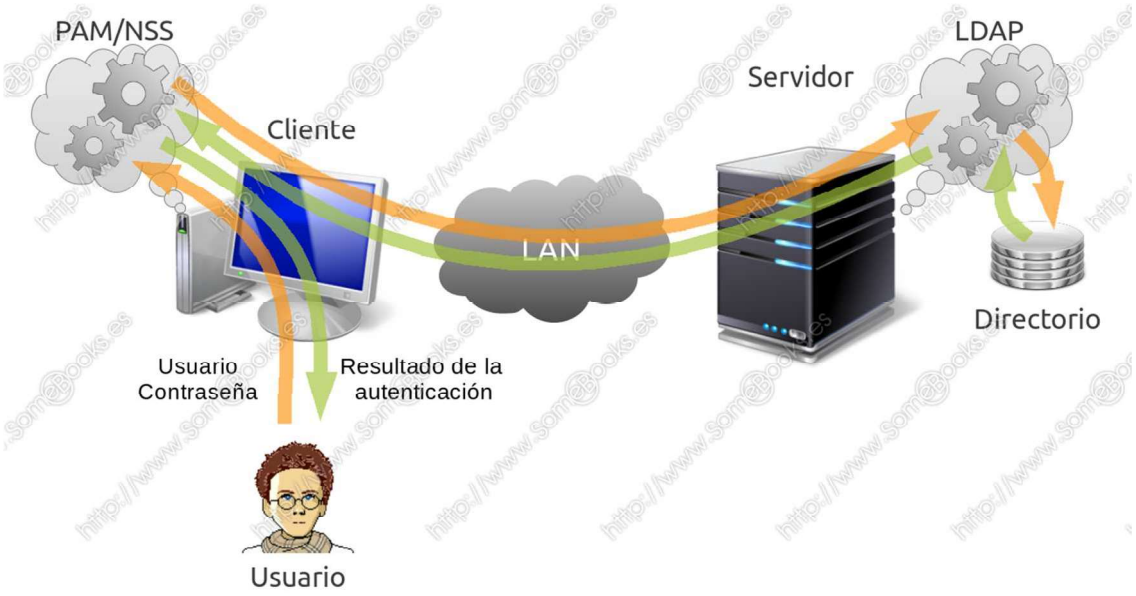
- Perfiles móviles de usuario usando NFS y LDAP

    - Crear una carpeta para guardar los perfiles móviles en el servidor
    - Exportar el contenido de la carpeta que tendrá los perfiles móviles
    - Crear una carpeta para guardar los perfiles móviles en cada cliente
    - Modificar el archivo /etc/fstab en cada cliente para montar la carpeta en el arranque
    - Indicar en el usuario LDAP la carpeta donde tendrá su perfil en el cliente
    - Comprobar que la configuración funciona correctamente

3.1. Introducción

Existen diferentes formas de autenticar clientes en una red *GNU/Linux*, pero una de las más usadas es la combinación de tres herramientas diferentes: *PAM*, *NSS* y *LDAP*.

La idea consiste en disponer de un servidor que facilite la autenticación de los clientes, de modo que éstos recurran al servidor cada vez que un usuario necesite identificarse. De esta forma, la cuenta de usuario no es específica de un equipo cliente, sino que será válida en cualquier equipo de la red que haya sido debidamente configurado.



De hecho, éste es el método que suele utilizarse en *GNU/Linux* para obtener una gestión de usuarios globales similar a la que ofrecen los *Servidores Windows* a través de una estructura de dominios.

En este capítulo aprenderemos antes de nada cuál es la función de cada uno de los componentes, tanto de forma individual como combinado con los demás. Después, aprenderemos a realizar la instalación y configuración de la estructura que hemos ilustrado en la figura anterior.

¿Qué es NSS?

NSS (***N**ame **S**ervice **S**witch*) es un servicio que permite la resolución de nombres de usuario y contraseñas (o grupos) mediante el acceso a diferentes orígenes de información. En condiciones normales, esta información se encuentra en los archivos locales del sistema operativo, en concreto en */etc/passwd*, */etc/shadow* y */etc/group*, pero puede proceder de otras fuentes, como *DNS* (*Domain Name System*), *NIS* (*Network Information Service*), *LDAP* (*Lightweight Directory Access Protocol*) o *WINS* (*Windows Internet Name Service*).

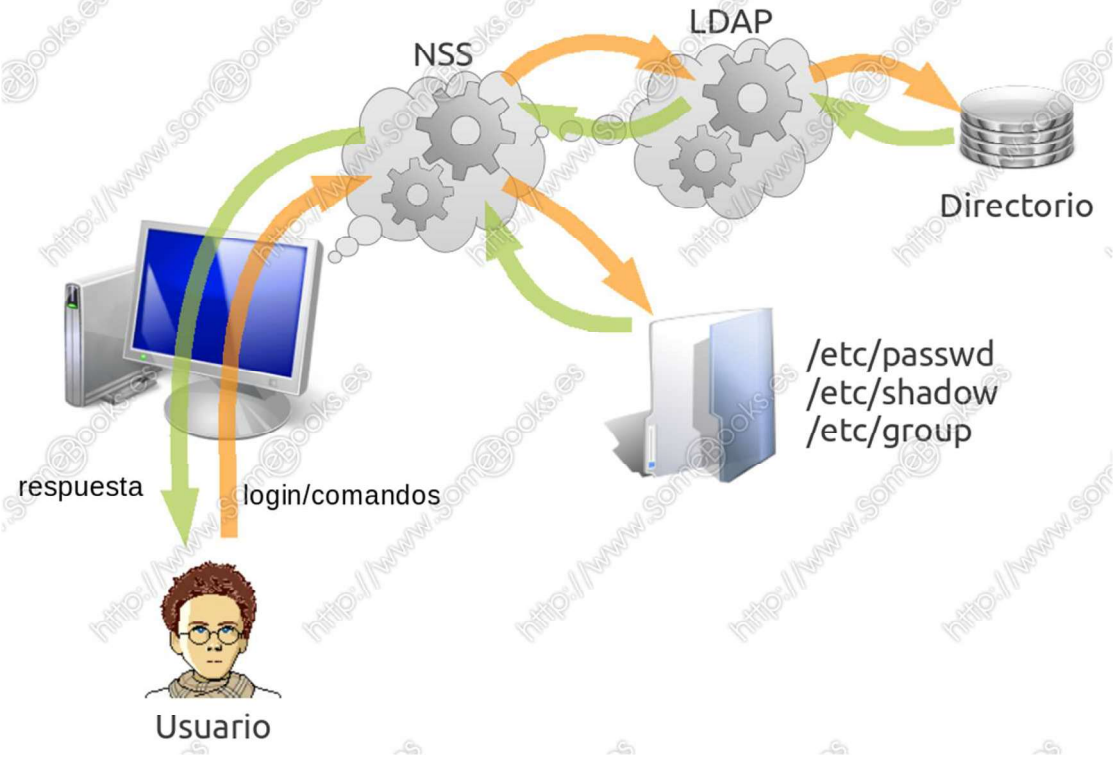
Como curiosidad, podemos decir que la versión de *NSS* para *GNU/Linux* está reescrita desde cero, sin tomar ni una línea de código de la versión de *Sun Microsystems*.

Los primeros sistemas operativos de tipo *Unix* accedían directamente a los archivos de configuración o código que dependía de la forma en que se producía la autenticación. Esto hacía que cualquier cambio en el modo de autenticación obligara a cambiar el sistema operativo.

Ultrix fue el primero en ofrecer una funcionalidad muy parecida a *NSS*, pero fue *Sun Microsystems* el primero en desarrollar *NSS* de una forma muy parecida a como lo conocemos en la actualidad. Por lo tanto, el primer sistema operativo que incorporó *NSS* fue *Solaris*.

Poco después, se portó a diferentes sistemas operativos, como *AIX*, *NetBSD*, *FreeBSD* o *GNU/Linux*.

El objetivo de *NSS* es que los programas o los comandos del sistema operativo puedan manejar información administrativa relacionada con los usuarios, las contraseñas y los grupos (incluidos aspectos como la caducidad de una contraseña o su nivel de complejidad) sin tener que conocer el lugar donde se encuentran almacenados.



¿Qué es PAM?

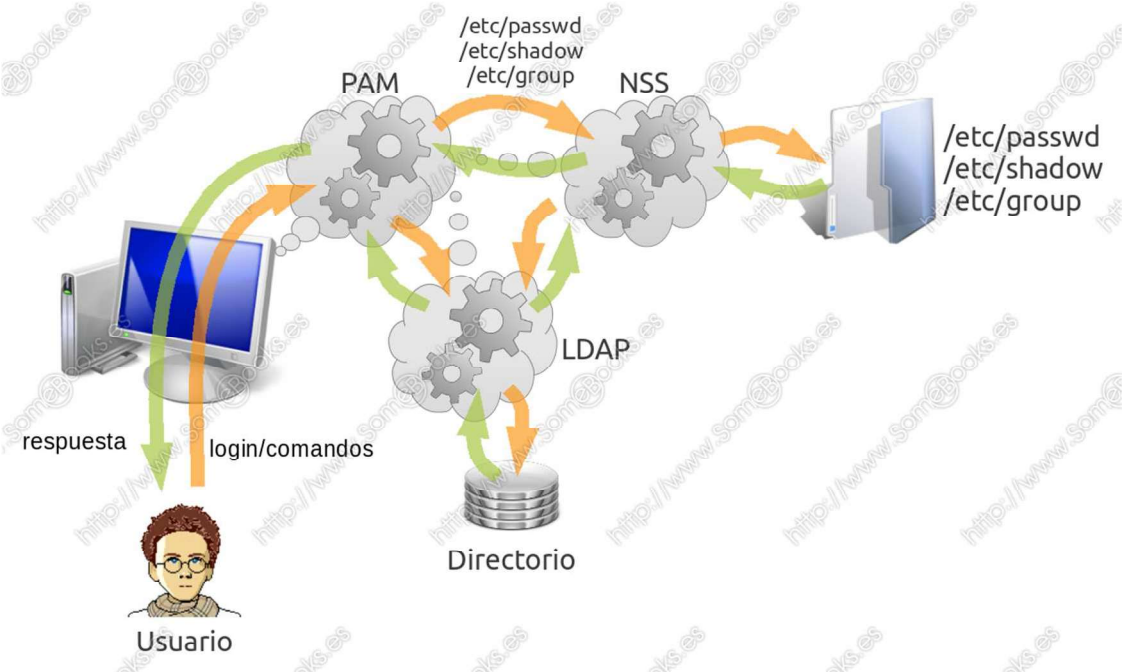
PAM (***P**luggable **A**uthentication **M**odules*) establece una interfaz entre los programas de usuario y distintos métodos de autenticación. De esta forma, el método de autenticación se hace transparente para los programas.

Como ocurrió con *NSS*, *PAM* surgió en *Sun Microsystems*, aunque, en este caso, como una propuesta a la *Open Software Foundation*. Fue *Red Hat* quien lo desarrolló como una herramienta de *Software libre* y lo incorporó por primera vez a la versión 3.0.4 de su sistema operativo en 1996.

La idea se basa en la creación de módulos de autenticación reemplazables, de forma que sea transparente para el sistema el uso de distintos métodos de autenticación. Esto hace que, sin realizar modificaciones en el sistema, podamos utilizar métodos que vayan desde el uso típico de un nombre de usuario y una contraseña, hasta dispositivos que faciliten la identificación biométrica de los usuarios (lectores de huellas, de voz, de imagen, etc.). Incluso incorpora opciones para aceptar contraseñas de un solo uso, restringir el acceso a determinados horarios o establecer políticas de autenticación específicas para cada usuario o grupos de usuarios.

Básicamente, *PAM* complementa en algunos aspectos el funcionamiento de *NSS* ya que mientras éste se centra en la búsqueda y mapeo de los usuarios, *PAM* controla la autenticación, el inicio de sesión y su configuración.

En la actualidad, *PAM* es el método que utilizan la mayoría de las aplicaciones y herramientas de *GNU/Linux* que necesitan relacionarse, de algún modo, con la autenticación de los usuarios.



¿Qué es LDAP?

LDAP es un protocolo que ofrece el acceso a un servicio de directorio implementado sobre un entorno de red, con el objeto de acceder a una determinada información. Puede ejecutarse sobre *TCP/IP* o sobre cualquier otro servicio de trasferencia orientado a la conexión.

LDAP son las siglas en inglés de **L**ightweight **D**irectory **A**ccess **P**rotocol (*Protocolo Ligero de Acceso a Directorios*) y podemos considerarlo como un sistema de almacenamiento de red (normalmente construido como una base de datos) al que se pueden realizar consultas.

El protocolo *LDAP* se creó originalmente en la *Universidad de Míchigan*, que publicó un primer desarrollo en 1993. Más tarde, *Tim Howes* y *Steve Killela*, dos de los diseñadores originales del proyecto comienzan a trabajar en una nueva versión bajo los auspicios de *IETF* (*Internet Engineering Task Force*) completando el desarrollo original.

La nueva versión (*LDAPv3*) se publicó en 1997 e integraba mecanismos de autenticación sencilla y una capa de seguridad. Después de esto, la *IETF* ha añadido numerosas extensiones y especificaciones propias que le han ido incorporando nuevas capacidades.

La respuesta es muy sencilla: *OpenLDAP* es un desarrollo del protocolo *LDAP*, implementado con la filosofía del software libre y código abierto.

El proyecto *OpenLDAP* se inició en agosto de 1998 y está sustentado por una entidad sin ánimo de lucro llamada *OpenLDAP Foundation*, creada por el desarrollador estadounidense *Kurt D. Zeilenga* para coordinar las actividades del proyecto.

OpenLDAP se publica bajo su propia licencia *OpenLDAP Public License* (http://www.openldap.org/software/release/license.html)

Como ocurría en el caso de *LDAP*, *OpenLDAP* está muy optimizado para ofrecer los mejores resultados en situaciones que requieran operaciones de lectura intensivas. De esta forma, un directorio *OpenLDAP* arrojará unos resultados muy superiores a los que ofrece una base de datos relacional optimizada, cuando realicemos operaciones de consulta intensivas sobre ambas.

Por el contrario, si utilizáramos un directorio *OpenLDAP* para guardar datos que sean actualizados de manera frecuente, los resultados obtenidos serían muy inferiores a los ofrecidos por una base de datos relacional.

No sólo podemos encontrar *OpenLDAP* en la mayoría de de las distribuciones *GNU/Linux*, sino que también lo encontramos para *Microsoft Windows*, *Apple OSX*, *Solaris*, *HP-UX*, *BSD*, etc.

3.2. ¿Cómo funcionan LDAP y OpenLDAP?

El modelo de información de *LDAP* se basa en entradas, entendiendo por entrada un conjunto de atributos identificados por un *nombre global único* (*Distinguished Name – DN*), que se utiliza para identificarla de forma específica. Las entradas se organizan de forma lógica y jerárquica mediante un *esquema de directorio*, que contiene la definición de los objetos que pueden formar parte del directorio.

Cada entrada del directorio representa un objeto, que puede ser abstracto o real: una persona, un mueble o una función en la estructura de una empresa, etc

Cada atributo de una entrada tendrá un tipo y un valor con el formato *atributo/valor* que permite caracterizar un aspecto del objeto que define la entrada. Estos atributos tienen nombres que hacen referencia a su contenido y pueden ser de dos tipos:

- *Atributos normales*: Son los atributos que identifican al objeto (nombre, apellidos, etc.).
- *Atributos operativos*: Son los atributos que utiliza el servidor para administrar el directorio (fecha de creación, tamaño, etc.).

Las entradas se indexan mediante el *nombre completo* (*dn*), que facilita la identificación singular a cada elemento del árbol. El nombre completo se formará con una serie de pares *atributo/valor*, separados por comas, que reflejan la ruta inversa desde la posición lógica del objeto hasta la raíz del árbol.

Para referirse al nombre completo suelen utilizarse las siglas *RDN*, que provienen del inglés *Relative Distinguished Name*.

Entre los atributos que suelen emplearse habitualmente, encontramos los siguientes, aunque puede haber muchos más:

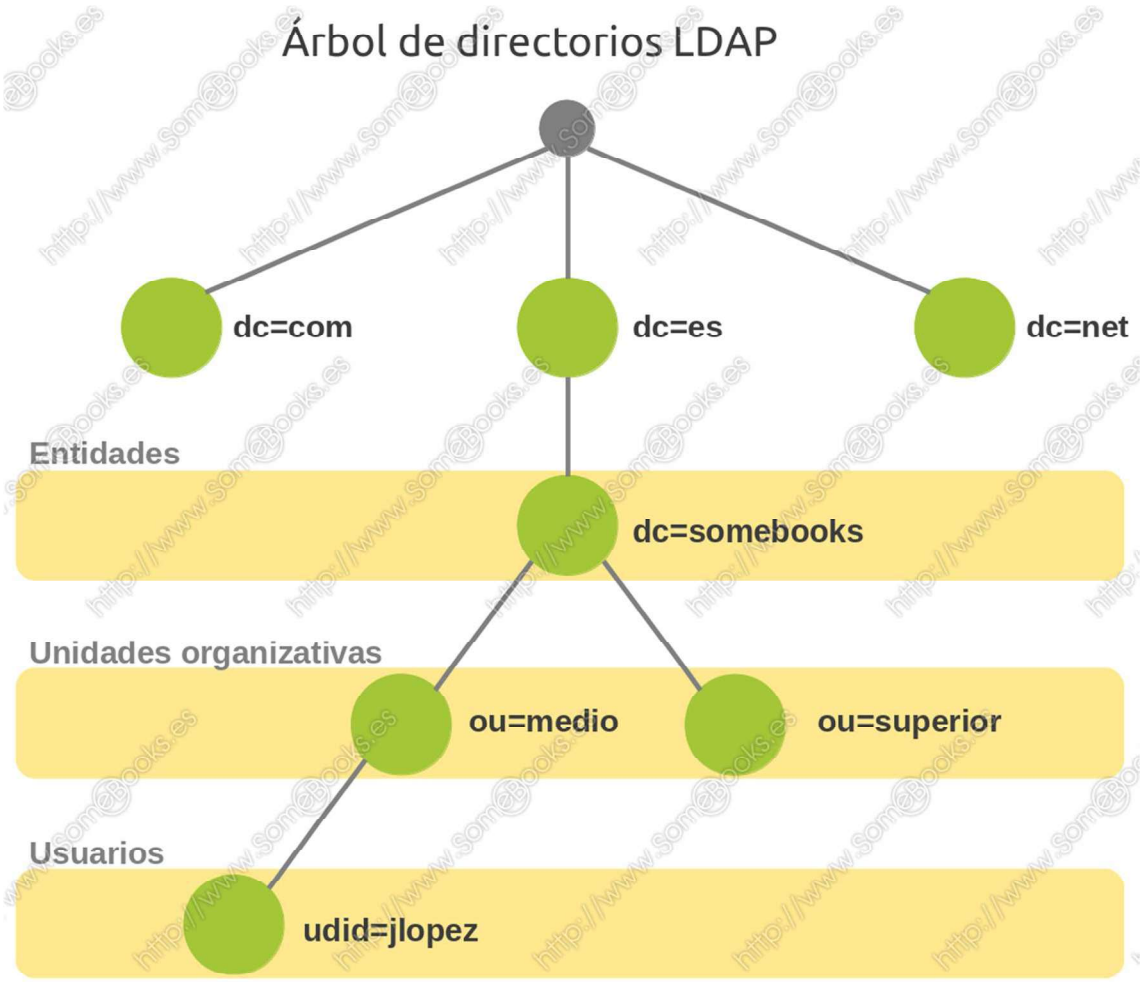
- *uid* (user id): Identificación única de la entrada en el árbol.
- *objectClass*: Indica el tipo de objeto al que pertenece la entrada.
- *cn* (common name): Nombre de la persona representada en el objeto.
- *givenname*: Nombre de pila.
- *sn* (surname): Apellido de la persona.
- *o* (organization): Entidad a la que pertenece la persona.
- *u* (organizational unit): El departamento en el que trabaja la persona.
- *mail*: dirección de correo electrónico de la persona.

Obviamente, los atributos anteriores hacen referencia a un tipo de objeto que representa a los miembros de una empresa. Para representar otros tipos de objetos, necesitaríamos atributos diferentes.

De esta forma, una entrada almacenada en el directorio *LDAP* podría tener el siguiente aspecto:

```
dn: uid=jlopez, ou=medio, dc=somebooks, dc=es
objectClass: person
cn: Juan Lopez
givenname: Juan
sn: Lopez
o: somebooks
u: medio
mail: juanlopez@somebooks.es
```

Como hemos dicho antes, las diferentes entradas se organizan a modo de árbol jerárquico que, normalmente, representa una estructura organizativa o geográfica en particular. Así, las entradas que representan comunidades autónomas aparecerán en la parte superior del árbol, debajo estarán las que representan provincias, después las ciudades, los departamentos, los usuarios, etc.



En la actualidad, las implementaciones de *LDAP* suelen utilizar *DNS* (*Domain Name Service*) para la estructura de los niveles superiores del árbol. En los niveles inferiores, sin embargo, las entradas representarán otro tipo de unidades organizativas, usuarios o recursos.

Por otra parte, gracias al uso de un atributo especial llamado *objectClass*, podemos controlar qué atributos son válidos y cuáles imprescindibles en una entrada. Los valores de *objectClass* establecen las reglas que debe seguir el valor de una entrada.

Lógicamente, *LDAP* establece operaciones para consultar o actualizar el directorio. Éstas nos permiten crear o eliminar entradas y modificar entradas existentes.

La mayor parte del tiempo, *LDAP* se utiliza para diversas consultas sobre la información que contiene, por lo que es común que la estructura de su base de datos se encuentre optimizada para la lectura en detrimento de la escritura.

Como vemos, *LDAP* puede utilizarse para organizar de forma unificada el acceso a la información representativa de una red. Sin embargo, es muy frecuente que también almacene la información de autenticación para los usuarios y/o recursos. De esta forma, se facilita el control de acceso sobre los datos contenidos en el servidor.

Aunque ya hemos visto al principio un esquema de funcionamiento mucho más detallado, podríamos representar el funcionamiento de LDAP de una forma más abstracta con el siguiente esquema:

Por último, *LDAP* incluye servicios de integridad y confidencialidad de los datos que contiene.

3.3. Instalar OpenLDAP en el servidor Ubuntu

En este apartado vamos a ver cómo se instala *OpenLDAP* en un equipo con el sistema operativo *Ubuntu 14.04 LTS*. Al final de este capítulo, también daremos por hecho que el sistema dispone del sistema de archivos *NFS* debidamente instalado y configurado para exportar la carpeta */home* (te recomiendo revisar el *capítulo 10: Instalar y configurar NFS en Ubuntu 14.04 LTS* para refrescar conocimientos). Sin embargo, esto sólo será necesario cuando necesitemos crear perfiles móviles de usuario en *Ubuntu* usando *NFS* y *LDAP*.

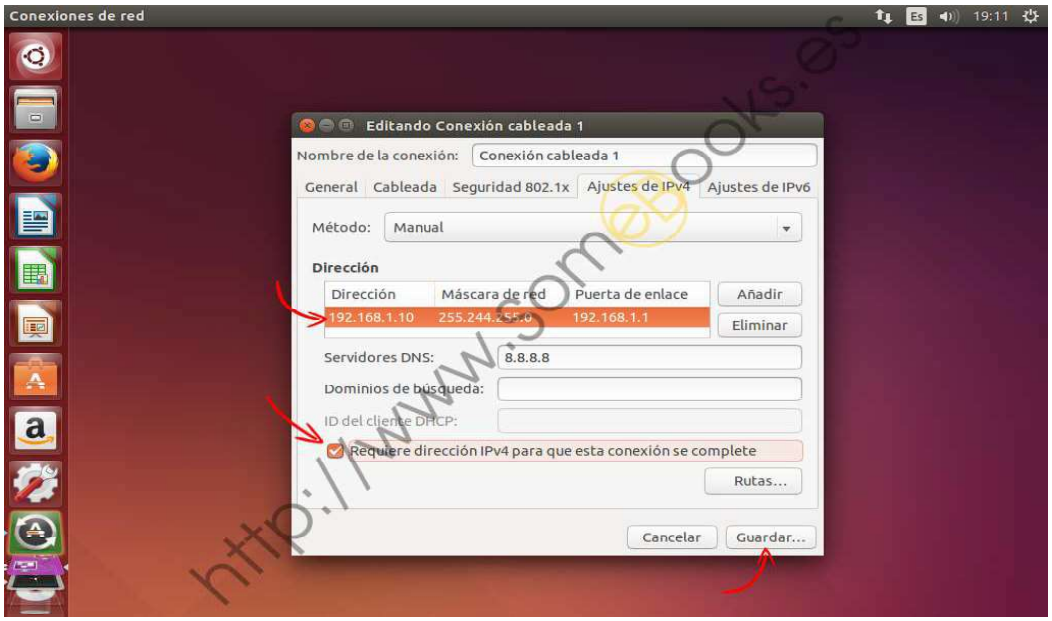
Configuraciones previas

Existen algunas cuestiones que deberemos tener en cuenta antes de instalar y configurar el sistema como servidor LDAP:

- Lo primero será asegurarnos de que el sistema tiene asignada una dirección IP estática. Si tienes dudas sobre cómo conseguirlo, puedes consultar el apartado 8.6. Ajustes tras la instalación de Ubuntu 14.04 LTS.

1

En este caso, la dirección asignada es **`192.168.1.10`**

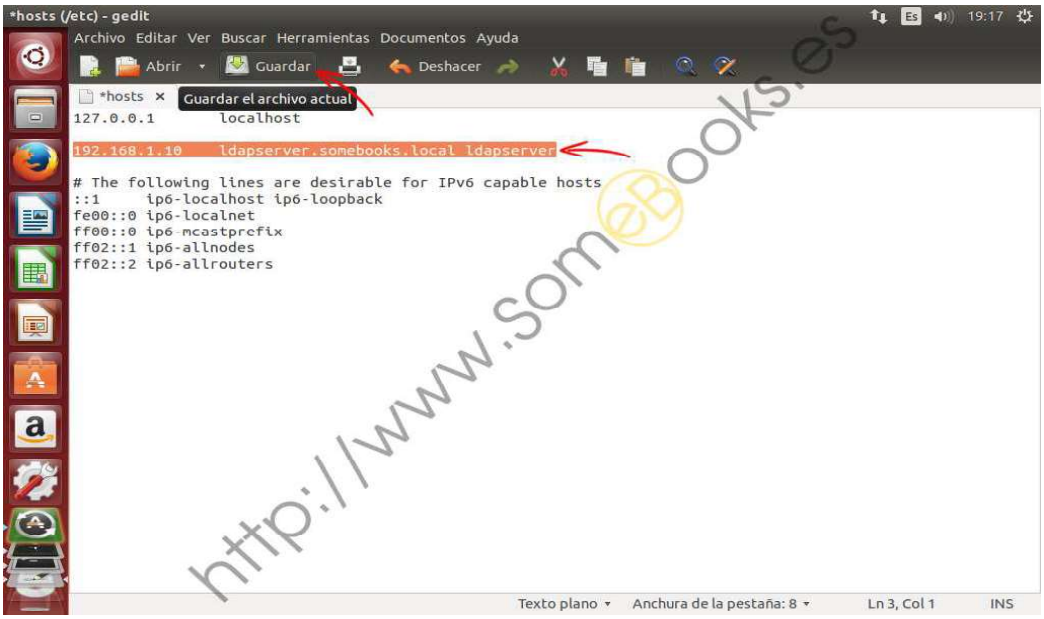


- También comprobaremos que los archivos **/etc/hostname** y **/etc/hosts** contienen los nombres adecuados para el servidor.En el caso de **/etc/hosts** deberá incluir una línea que relacione la dirección IP estática del servidor con los nombres lógicos que tenemos previsto utilizar. En definitiva, algo como esto (aunque, claro está, adaptada al nombre de tu servidor):

```
192.168.1.10     ldapserver.somebooks.local    ldapserver
```

2

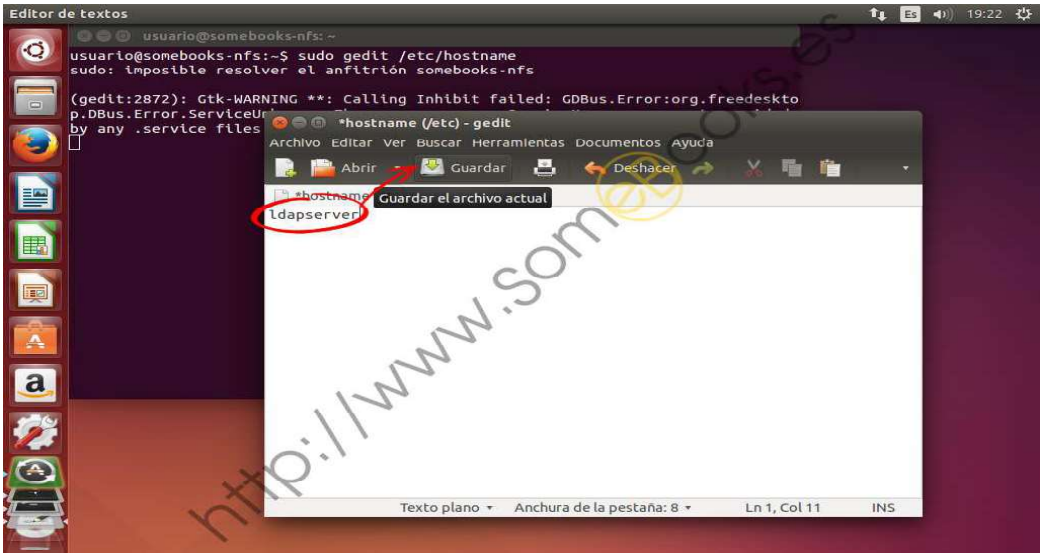
Editamos el archivo **/etc/hosts** y nos aseguramos de que contenga los valores adecuados



… Y en el caso de **/etc/hostname**, nos limitamos a incluir el nombre del equipo:

3

Editamos el archivo **/etc/hostname** y nos aseguramos de que contiene el valor adecuado.



Como antes, si tienes alguna duda sobre cómo hacerlo, puedes consultar el apartado 8.6. Ajustes tras la instalación de Ubuntu 14.04 LTS.

Instalar el software necesario

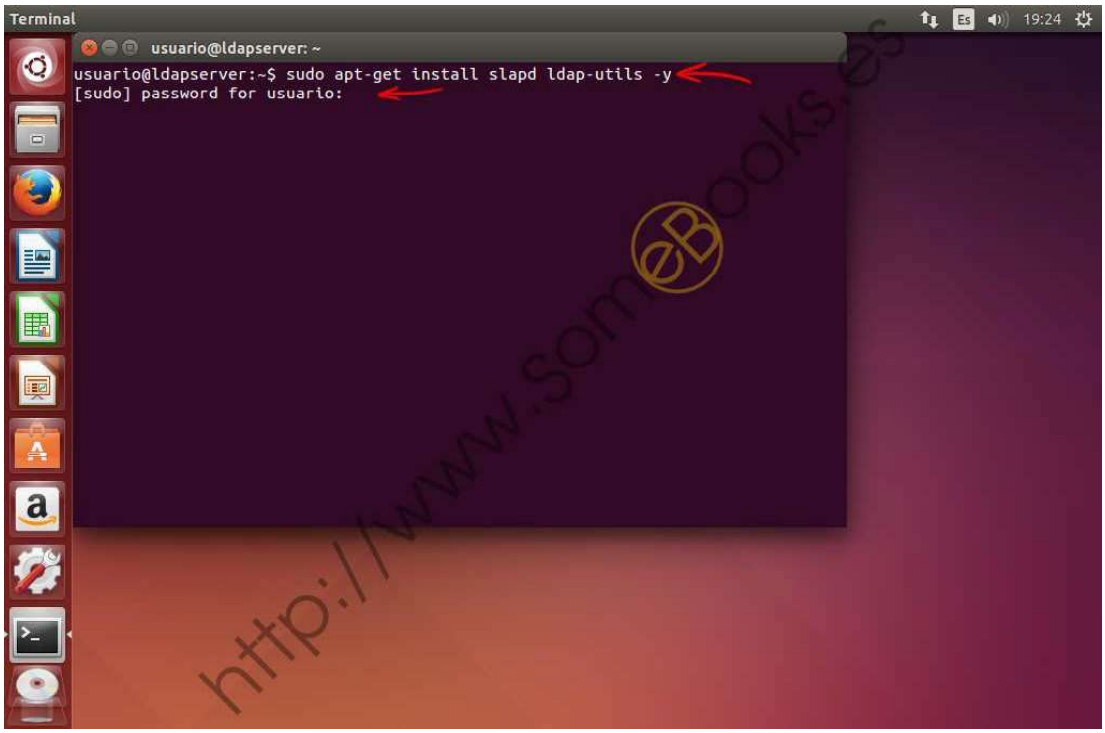
El proceso de instalación es realmente sencillo. Básicamente consiste en instalar el paquete *slapd*, aunque nosotros también instalaremos el paquete que contiene las utilidades de administración de *LDAP*: *ldap-utils*.

Como ambos paquetes se encuentran en los repositorios oficiales de *Ubuntu 14.04 LTS*, sólo tenemos que escribir en la terminal la siguiente orden:

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

1

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

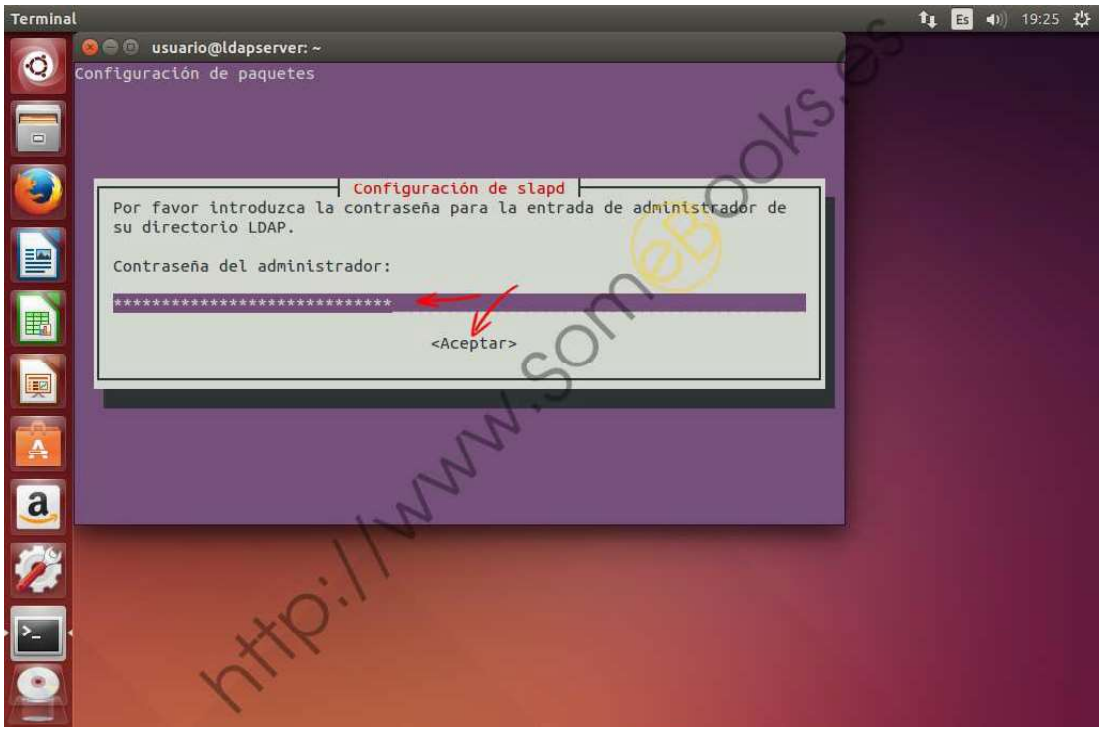


Durante la instalación, aparece en la consola un mensaje que nos solicita la contraseña de administración para *LDAP*. Como siempre, deberá ser una contraseña segura.

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

2

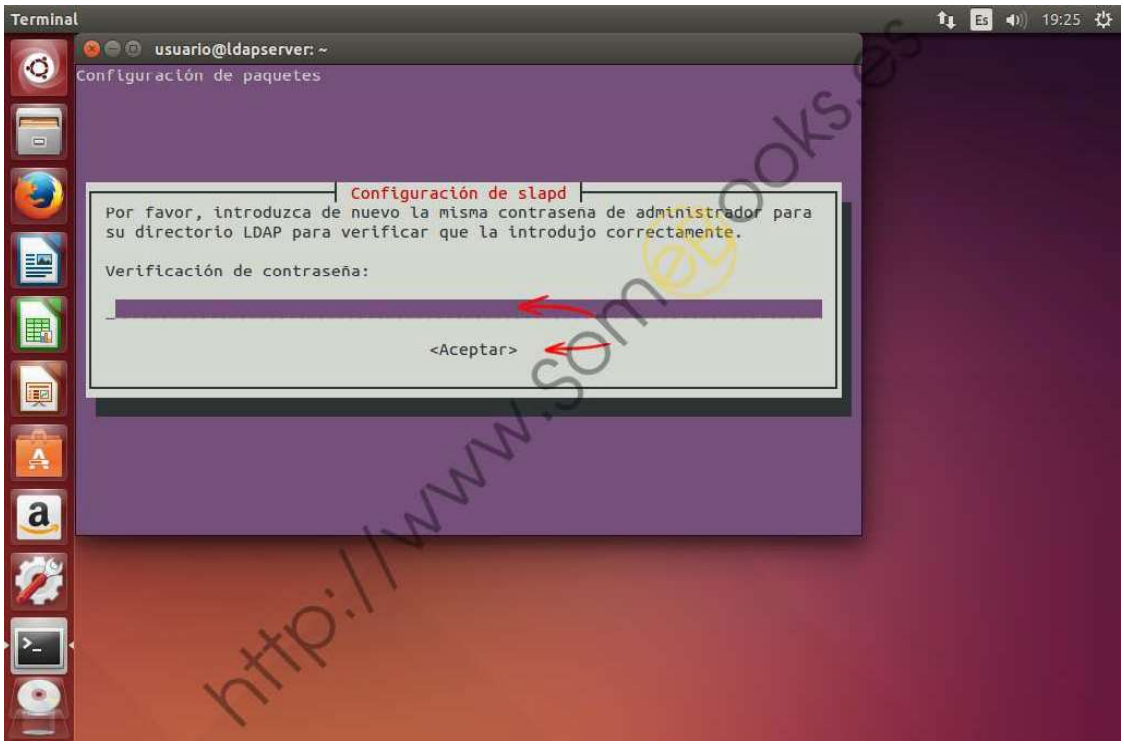
Cuando terminemos de escribir, pulsaremos la tecla *Intro*.



Como suele ocurrir cuando escribimos una contraseña, para evitar que hayamos cometido algún error tipográfico que después nos impida la entrada, el sistema nos pide que volvamos a escribirla.

3

Cuando acabemos, volvemos a pulsar la tecla *Intro*.



Al hacerlo, volveremos al aspecto normal de la terminal y la instalación seguirá su curso.

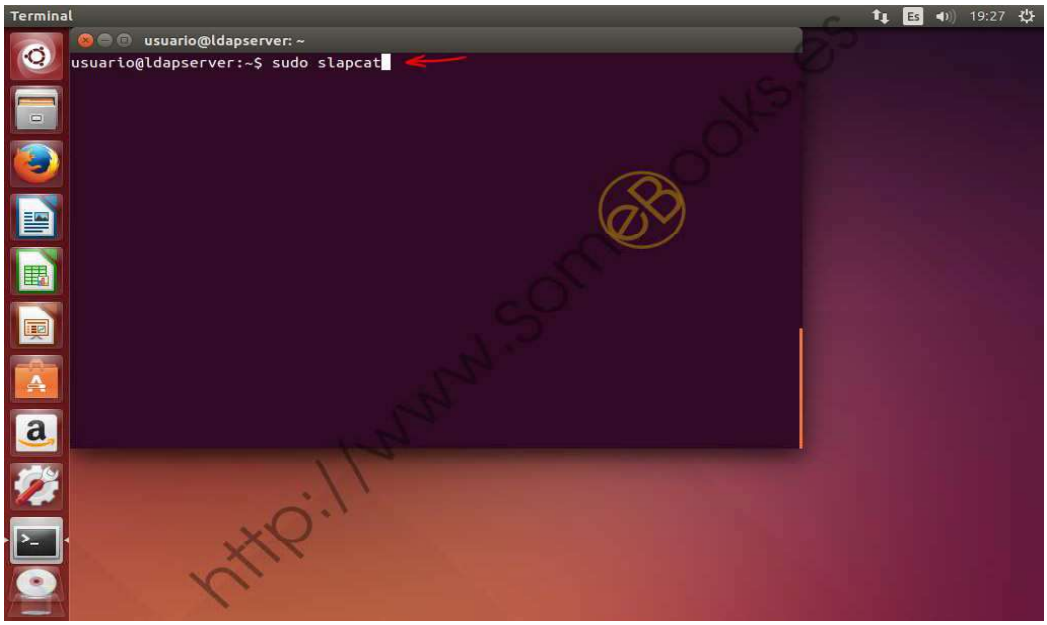
Una vez concluida la instalación, podemos comprobar que todo es correcto usando el comando **`slapcat`**.

El objetivo de este comando consiste en obtener la información de la base de datos *LDAP* y su salida se produce en formado *LDIF*, lo que nos facilitará exportar la estructura del directorio *LDAP* o, sencillamente, obtener una copia de respaldo de su contenido, sólo con redirigir su salida a un archivo.

Hablaremos de los archivos *LDIF* más adelante.

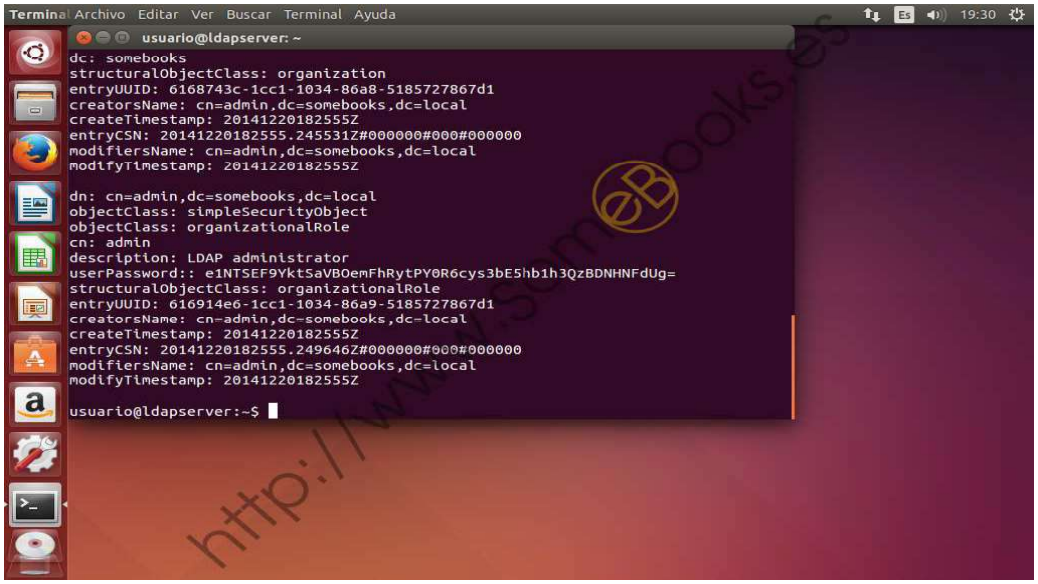
4

Ejecutamos el comando con privilegios de administrador.



5

Observando la salida, comprobamos que los datos asignados de forma predeterminada son correctos



3.4. Crear la estructura del directorio

Una vez instalado el servidor y comprobado su funcionamiento, el siguiente paso consistirá en comenzar a incluir contenido. Como cabe esperar, lo primero que debemos