MINISTERIO
DE EDUCACIÓN

INSTITUTO DE TECNOLOGÍAS
EDUCATIVAS

Redes de área local
Aplicaciones y Servicios Linux

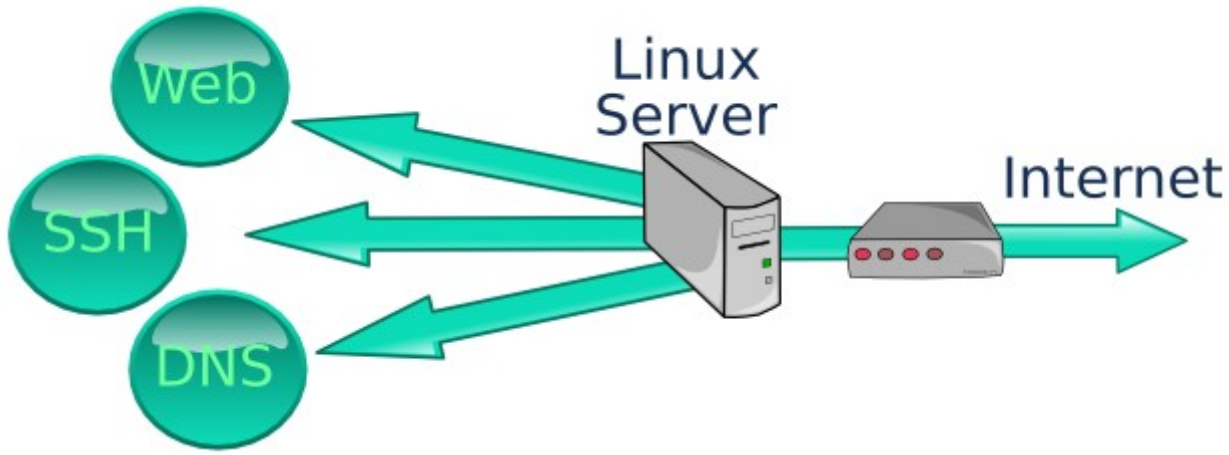
Introducción

C/ TORRELAGUNA, 58
28027 - MADRID

Índice de contenido

1.- Introducción

Con este curso lo que se pretende es ampliar todos los conocimientos adquiridos en el curso de Redes de área local en Centros Educativos, especialmente en lo referente a la explotación final de una red, mediante la instalación de un servidor de red con sistema operativo 'Linux', y equipos clientes también con 'Linux'. Nuestra intención es facilitar tareas habituales como la instalación del sistema operativo, aplicaciones en los equipos cliente, gestión de usuarios, personalización del entorno de trabajo de éstos, etc. Además, veremos la configuración y explotación del servidor web **Apache**, creando una pequeña **Intranet** en nuestro centro con la finalidad de liberar al responsable de la red del centro de muchas labores que hasta ahora realizaba de forma manual, así como poder ofrecer multitud de recursos para el profesorado y el alumnado.



Cuando hablamos del sistema operativo 'Linux' no nos ceñimos a una versión concreta, ya que cada pocos días aparecen versiones nuevas. Los contenidos que van a ser abordados en este curso podrían ser aplicables a cualquier versión de 'Linux', principalmente las basadas en Debian, y con alguna modificación a cualquier otra versión.

Se ha decidido elegir **Ubuntu** en lugar de otras distribuciones de Linux por estar basada en **Debian** y ser de instalación muy rápida y sencilla.

Planteamiento Inicial del curso

Enhorabuena, acabas de recibir el nombramiento de Responsable de Tecnologías de la Información y de la Comunicación (TIC) de tu centro y ahora debes poner en marcha un proyecto de actuación.

Tu Director, que confía plenamente en ti, sabe que no es una tarea sencilla y, por ello, se ha puesto en contacto con otro centro educativo y te ha conseguido un manual de la parte técnica que se implantó en él. Allí se recogen las actuaciones que se han llevado a cabo en la primera fase de su proyecto consistentes en la instalación, configuración y explotación de la red física de su centro.

Es cierto que cada centro educativo es diferente y, por lo tanto, este manual no resolverá exactamente tus necesidades, pues puede no ajustarse exactamente a tu contexto de trabajo. No obstante, seguro que tiene muchas semejanzas y, sin duda, podrá servirte.

Proyecto de implantación de las tecnologías de la información y de la comunicación (TIC) en el centro

Fase técnica: implantación, configuración y explotación de una red.

Nuestro centro educativo ha ido creciendo en los últimos años. Actualmente el número de profesores y

alumnos es elevado y un gran porcentaje de ellos están utilizando las Nuevas Tecnologías de la Información y de la Comunicación en su tarea diaria.

Fruto de la nueva complejidad del centro, la información entre el profesorado fluye de manera muy lenta, de forma que en ocasiones llega tarde y en otras se pierde. También ha crecido el número de equipos informáticos puestos a disposición de la Comunidad Educativa, tanto en número de equipos por aula, como en número de aulas y espacios departamentales. El número de alumnos que pasa por ellas es elevado y las diferentes configuraciones, fruto de la utilización inadecuada de los equipos, dificultan las tareas de mantenimiento y de aprendizaje. Se dispone, pues, de la instalación física de la red, pero no se obtiene el rendimiento adecuado de la misma, ni se explota al máximo su estructura física.

Por todo ello, y como responsable de las TIC en el centro, he decidido reestructurar el sistema de información interno, para crear una verdadera intranet y poder administrarla, dado que la situación actual resulta poco manejable. Esta primera actuación constituirá la fase técnica del Proyecto de Nuevas Tecnologías del Centro.

Del análisis pormenorizado del mismo se han detectado los problemas que, a mi juicio, son más graves y urgentes. También se desprende de este análisis la solución que se propone.

1. Fruto del crecimiento informático del centro y de la utilización cada vez más frecuente de dichos recursos informáticos resulta que la información se encuentra fragmentada en muchos lugares, el acceso a ella es complejo y lento y, en multitud de ocasiones, se accede a información desfasada, o ésta se ha perdido.

Se propone la instalación de un servidor central que proporcione acceso a la información, independientemente de donde se encuentre el usuario y que sea accesible para toda la comunidad educativa. Para ello, la información se centralizará en el servidor y se implantarán credenciales de autentificación, de tal forma que, a partir del perfil que se tenga, se pueda acceder a unos recursos o a otros, garantizando la integridad y seguridad en el acceso.

2. Es fundamental que el servidor sea un equipo al que no se pueda acceder para ejecutar programas o para realizar una tarea urgente; sin embargo, es un hecho cierto que se accede a los ordenadores críticos (secretaría, dirección, administración, etc) para urgencias de impresión, consultas u otras causas, con el consiguiente peligro de corrupción y borrado accidental de archivos.

Se propone que, una vez instalado y configurado el ordenador servidor, éste se aísle en un despacho próximo a secretaría o dirección, con la prohibición expresa de su utilización. Para las tareas de administración de la red, el coordinador de TIC utilizará una conexión remota.

3. Por otro lado, tanto el profesorado como el alumnado acostumbran a personalizar su escritorio, incluyendo accesos directos, carpetas y demás vínculos que considera importantes y cómodos en su tarea diaria. Pero todo ésto incomoda, en muchas ocasiones, a otros usuarios que también utilizan el mismo equipo. Esta situación es especialmente patente en las aulas de informática, donde parece que el alumnado compite en crear ámbitos de trabajo extravagantes e inútiles, con la consecuente pérdida de tiempo en la carga del sistema operativo.

Se propone la gestión centralizada de los usuarios del centro, tanto de profesorado como alumnado. La información se guardará en el servidor y será accesible independientemente del ordenador utilizado. De esta forma se dispondrá de un entorno de trabajo individualizado que se vinculará al usuario, siendo independiente del equipo que se quiera utilizar.

4. En algunos momentos el tráfico de la red es enorme, habiéndose detectado problemas de congestión sin saber el motivo y consecuentemente, sin poder tomar medidas para restablecer la normalidad. También se ha observado la tendencia a los cambios de configuración del alumnado de las direcciones IP, máscaras, puertas de enlace y direcciones DNS asignadas, lo que conlleva problemas de conectividad e inseguridad. Volver a realizar las configuraciones lleva demasiado tiempo. En ocasiones se produce el cambio de direcciones DNS por parte del proveedor, lo que implica un periodo de tiempo importante sin conexión, hasta conseguir volver a configurar totalmente los equipos.

Se propone la instalación de un servidor DHCP y DNS que además sea la puerta de salida a todas las conexiones externas, de tal forma que cualquier cambio en la configuración del proveedor se resuelva en el servidor y los cambios del alumnado se resuelvan simplemente reiniciando el equipo. También permitirá el

análisis del tráfico de la red al pasar todas las comunicaciones entrantes y salientes por el servidor.

5. El secretario del centro reiteradamente ha advertido que el gasto en fotocopias es elevado y, por lo tanto, es necesario tener un control de lo que se imprime y de su destino. Recientemente se ha suscrito un contrato con una empresa de reprografía que ha instalado en alquiler una impresora de red y se quiere que una gran parte de la impresión se realice a través de dicha impresora, dadas las ventajas económicas que supone frente al coste de los fungibles de las impresoras locales.

Se propone la instalación de la impresora de red para poder realizar el control que el secretario exige aunque sin rechazar la instalación de impresoras locales donde sea necesario.

6. La información existente debe ser salvaguardada de los desastres que habitualmente ocurren: virus, fallos de hardware, borrados accidentales y/o provocados, etc., máxime cuando se está llevando a cabo un sistema centralizado de información donde, además de los datos generales, se almacenan los perfiles y las carpetas personales de los usuarios.

Se propone la creación de un sistema de copias de seguridad automatizado que evite las pérdidas de información.

7. Muchos profesores se encuentran realizando cursos de aprendizaje y perfeccionamiento relacionados con las Nuevas Tecnologías (cursos de html, php, mysql, etc) y necesitan soporte para su desarrollo.

Se propone la instalación de software servidor web apache para facilitar y potenciar el aprendizaje del profesorado, permitiendo el uso de lenguajes de script (php) así como del acceso a bases de datos (MySQL). Se habilitará además un espacio web para el profesorado que lo solicite donde pueda hacer uso de todas estas herramientas y un servidor FTP.

8. Se ofrecerá a los profesores, como complemento a los Servicios Web prestados a los usuarios del centro, acceso a carpetas privadas y seguras (acceso https) disponibles vía web mediante autentificación.

Se propone la instalación de servidores de certificados (Entidad Certificadora OpenSSL,...) para el acceso a dichas páginas web seguras, de forma que se garantice la seguridad y privacidad de los datos allí alojados.

Para el desarrollo de esta fase técnica, es preciso partir de la realidad actual del centro que se concreta en:

Esquema Hardware

El centro dispone de ordenadores para el alumnado ubicados en varias aulas informáticas; así como también de ordenadores para el profesorado, repartidos en los despachos de Dirección y en los espacios departamentales.

El cableado de la red física del centro ya se encuentra instalado; su topología es en estrella, formando una única red física y lógica.

Salida a Internet con router ADSL.

Esquema Software

Los ordenadores de las aulas de informática son recientes, por lo que disponen de más memoria RAM. En ellos se instalará 'Linux'. Este sistema operativo puede comportarse también como un potente cliente de red que se comunicará perfectamente con el servidor.

Los ordenadores de los departamentos didácticos son más antiguos pero no queremos renunciar a las prestaciones de un potente y seguro cliente de red, por ello se instalará también 'Linux'. Si el hardware es demasiado obsoleto cabe la posibilidad de utilizarlo como terminal para conectarse por escritorio remoto directamente al servidor.

Para conseguir la integración de todos los equipos se utilizará un nuevo equipo que será el servidor. Sobre él se instalará un sistema operativo servidor, que también será 'Linux'. Éste será el centro de la Intranet y a través de él, se accederá a Internet.

Requisitos técnicos

Para poder realizar el curso es necesario disponer de ciertos elementos hardware y software que nos permitan desarrollarlo en su totalidad. En este apartado citaremos los requisitos necesarios para dicho fin.

Requisitos hardware

Para realizar muchas de las actividades que se plantean a lo largo del curso, es necesario disponer de dos PCs conectados en red. Si solo disponemos de un PC, existe la posibilidad de utilizar máquinas virtuales, pero lo recomendable es utilizar dos PCs.

Los requisitos hardware son distintos en función de la posible utilización o no de máquinas virtuales para la realización del curso. En el caso de que se disponga de dos PCs, los requisitos hardware son los siguientes:

Requisitos hardware utilizando dos PCs

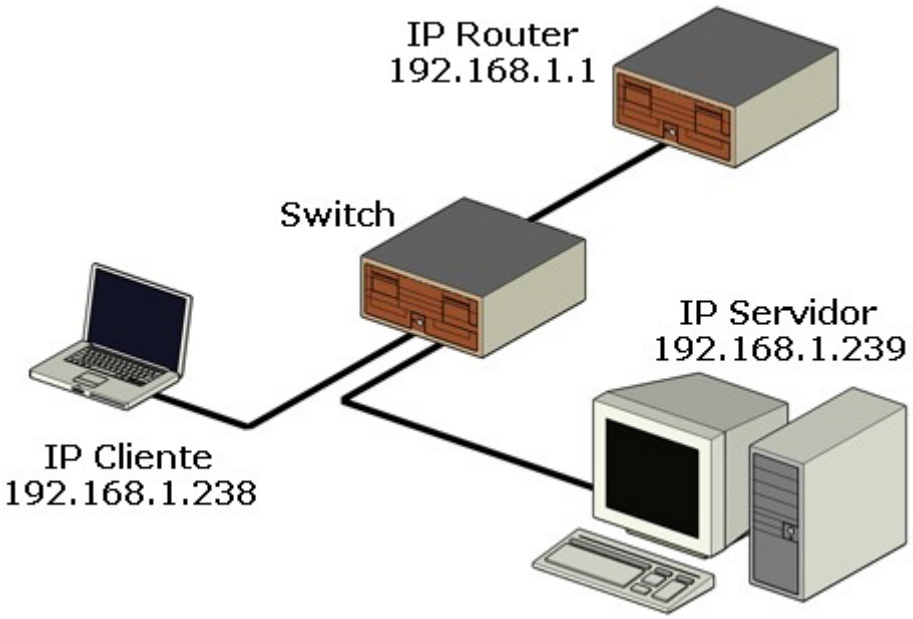
1.- Un ordenador Servidor con al menos los siguientes requisitos

- 384 Mb. de RAM
- 15 Gb. de Disco Duro
- Lector de DVD
- Dos tarjetas de red (una de ellas podría ser opcional ya que sólo es necesaria para realizar el apartado correspondiente al "Enrutamiento").

2.- Un ordenador Cliente con al menos los siguientes requisitos

- 128 Mb. de RAM
- 4 Gb. de Disco Duro
- Lector de CD
- Tarjeta de red

3.- Un dispositivo de conexión de red (hub, switch ó router ADSL) y 2 latiguillos UTP, o en su defecto, un cable UTP cruzado para unir los dos PCs.



Equipamiento recomendado para el curso (las IPs son de ejemplo)

En el PC servidor será necesario realizar la instalación de Ubuntu en el disco duro mediante el icono 'Instalar'. Será necesario disponer de 10 GB libres y el proceso tardará unos 15 minutos. Se recomienda disponer de 1 GB adicional en el disco duro formateado como swap para que Linux lo utilice como memoria virtual si fuera necesario.

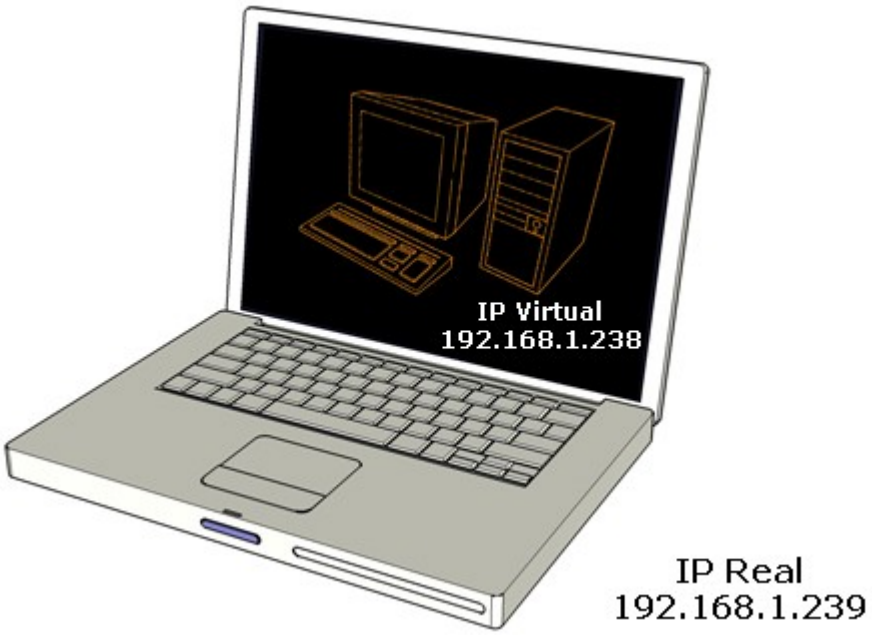
En el PC cliente se puede instalar cualquier versión de Linux basada en Debian. Una distribución que consume pocos recursos es DSL-Linux, que puede ser instalada en un PC con 128 MB de RAM. Si deseamos instalar ubuntu, el PC cliente deberá disponer de al menos 256 MB de RAM

No se debe realizar el curso usando el modo Live-CD ni en cliente ni en servidor ya que al apagar el PC se perderían nuestros cambios.

Requisitos hardware utilizando máquinas virtuales

Si no disponemos más que de un PC, para poder utilizar **máquinas virtuales**, nuestro PC debe satisfacer los siguientes requisitos:

- 512 Mb de RAM (recomendable 1 Gb)
- 15 Gb. de Disco Duro libres
- Lector de DVD
- Tarjeta de red



Equipamiento alternativo para el curso (las IPs son de ejemplo)

En el PC real será necesario realizar la instalación de Ubuntu. Posteriormente instalaremos el software de virtualización Vmware Player o Vmware Server (o VirtualBox o cualquier otro software de virtualización de PCs). El alumno deberá conocer el manejo del software de virtualización, pues no se da soporte técnico del mismo durante el curso. Una vez instalado el software de virtualización, podremos crear una máquina virtual e instalar Ubuntu en ella. Si utilizamos el software de Vmware, podremos utilizar la máquina virtual Ubuntu que viene con el DVD del curso. Tanto la máquina virtual como la real se pueden utilizar como servidor o como cliente indistintamente

Si nuestro PC trabaja en entorno Windows y de momento no queremos instalar Linux, existe la posibilidad de instalar Vmware Server o Vmware Player para Windows y utilizar dos máquinas virtuales, una que hará de servidor y otra que hará de cliente. En este caso, nuestro PC real deberá disponer de 1 GB de memoria RAM.

Se recomienda disponer de conexión a Internet de banda ancha. En caso de disponer de una conexión RTB, es necesario disponer de un modem externo ya que Linux tiene dificultades para reconocer los winmodems si no disponemos de drivers.

Requisitos software

El DVD del curso contiene, en la carpeta */REDES_LINUX/software*, una máquina virtual para Vmware con Ubuntu 9.04 preinstalado. El software de virtualización de Vmware (Vmware Player o Vmware Server) será necesario descargarlo desde su página web. También contiene la ISO de Ubuntu 9.04 por si queremos crear una máquina virtual nueva con VirtualBox o cualquier otro software de virtualización.

Versiones utilizadas durante el curso:

- **Para el PC servidor:** Ubuntu 9.04 Desktop Edition (Imagen CD-ISO incluída en la carpeta */REDES_LINUX/software* del DVD del curso).
- **Para el PC cliente:** Ubuntu 9.04 Desktop Edition (Imagen CD-ISO incluída en la carpeta */REDES_LINUX/software* del DVD del curso)) o cualquier otra basada en Debian.
- **Software de virtualización:** Vmware Player. Se puede descargar de http://www.vmware.com/es/. Versiones tanto para Linux como para Windows. Máquina virtual con Ubuntu 9.04 incluida en la carpeta */REDES_LINUX/software* del DVD del curso. Válida tanto para servidor como para cliente.

Máquinas Virtuales

Uno de los problemas más habituales con los que se encuentra un alumno que desea realizar un curso de Redes es que no suele disponer del material necesario para ello. La mayoría del alumnado no dispone en su domicilio de una red con al menos dos equipos disponibles, que son los requisitos hardware mínimos para poder llevar a cabo el curso. Parece, pues, complejo encontrar un entorno de **laboratorio** en el cual el alumno que realiza el curso pueda trabajar libremente y sin miedo ninguno a cometer de errores.

Esta falta de recursos puede desanimar a cualquier alumno que desee realizar el curso. Ahí es donde surgen las **máquinas virtuales** como auténtica revolución que permite, a cualquier usuario que posea un PC, disponer fácilmente de una red local **vitual** con la que poder trabajar en un entorno seguro.

Programas como **qemu**, **bochs** o **vmware** permiten definir en nuestro equipo máquinas virtuales, que son PCs independientes cuyo hardware es simulado mediante éstos programas. Dentro de una máquina virtual se puede instalar cualquier sistema operativo y será totalmente independiente del equipo real o **sistema anfitrión**.

No sólo eso, sino que además, si la configuración del direccionamiento IP asignado a cada equipo está en el mismo rango de direcciones, los dos equipos (el real y el virtual) estarán en una misma red, siendo accesibles, por tanto, los recursos compartidos que existieran en ambas máquinas y aunque físicamente sólo disponemos de un ordenador, lógicamente podemos pensar que disponemos de dos equipos conectados a un switch por tanto, a partir de este momento ya disponemos de una red con la que poder trabajar.

Un PC dentro de otro PC

Hemos de darnos cuenta que para llegar a construir esta red **virtual** no hemos necesitado ningún material **físico** de los indicados anteriormente y por tanto, para poder realizar el curso sólo necesitamos un ordenador con conexión a Internet (recurso este imprescindible cuando se desea realizar un curso de formación a distancia, independientemente de su temática) y un programa que permita correr máquinas

virtuales como vmware player.

Si el alumno desea utilizar máquinas virtuales para la realización del curso es imprescindible que tenga experiencia y conozca su manejo. Para más información sobre máquinas virtuales recomendamos leer el monográfico publicado en el observatorio del CNICE, haciendo clic aquí.

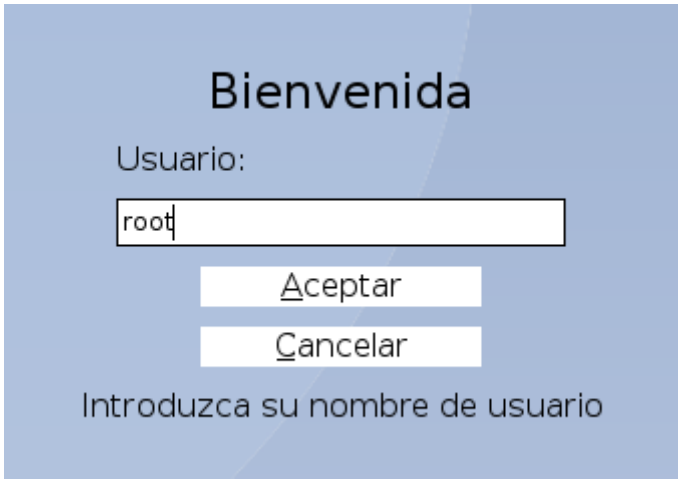
2.- Usuarios y grupos de usuarios en Unix

El sistema Unix es un sistema operativo multiusuario. Linux está basado en el sistema Unix. Para que múltiples usuarios puedan hacer uso del sistema de una forma segura y ordenada, es necesario que el sistema disponga de mecanismos de administración y seguridad para proteger los datos de cada usuario, así como para proteger y asegurar el correcto funcionamiento del sistema.

Cuentas de usuario

Para poder utilizar el sistema operativo Unix es necesario disponer de una cuenta de usuario que se compone de **nombre de usuario (login) y de contraseña (password)**. Las cuentas de usuario son creadas por el administrador que en Unix es un usuario especial llamado **root** (ver más abajo). Los usuarios deberán pertenecer al menos a un grupo de usuarios ya que obligatoriamente deben tener asignado un grupo principal o grupo primario.

Cuando un usuario entra en un sistema Unix, debe identificarse indicando su **nombre** de usuario (en inglés **'login'**) y su **contraseña** (en inglés **'password'**). Si se equivoca al introducir su nombre o su contraseña, el sistema le denegará el acceso y no podrá entrar.

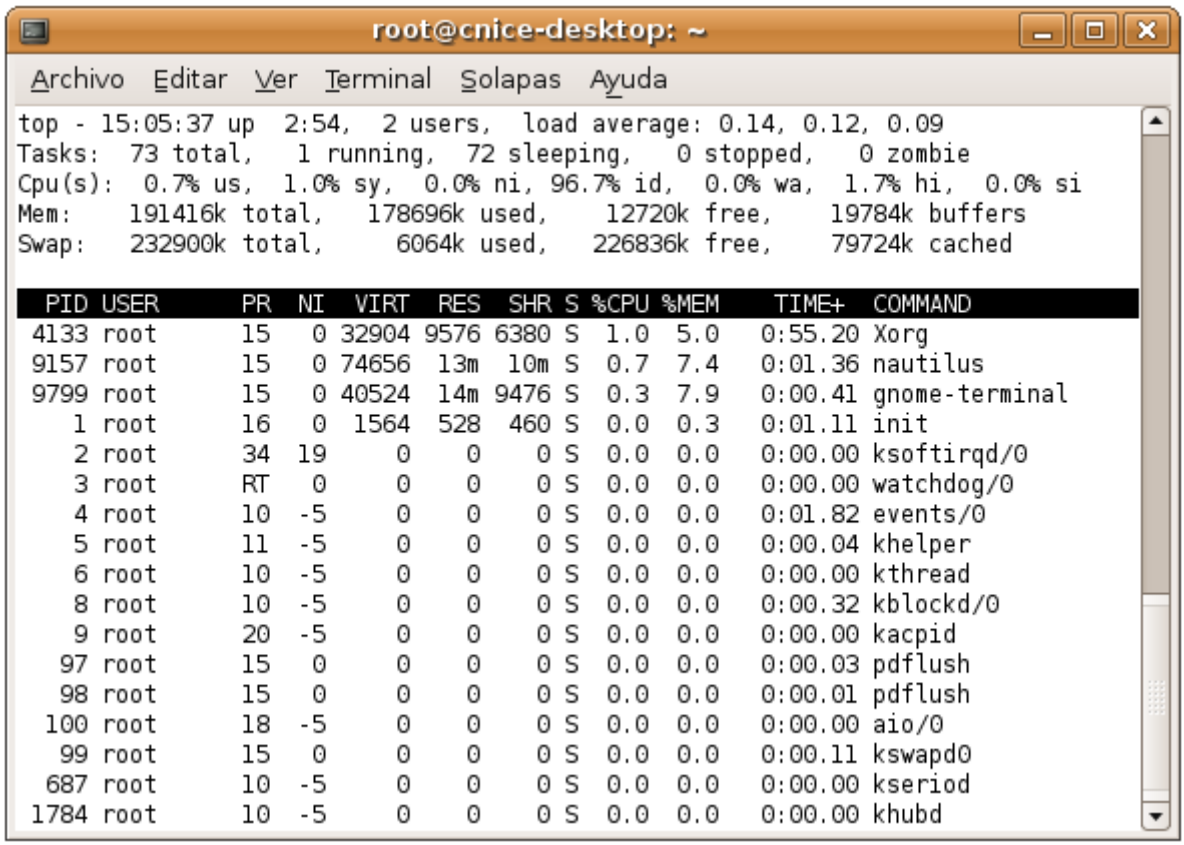


Inicio de sesión en Linux

Una vez se haya identificado de forma satisfactoria, el usuario podrá utilizar el sistema y ejecutar todas las aplicaciones que le sean permitidas, así como leer, modificar o borrar aquellos archivos sobre los cuales tenga permiso.

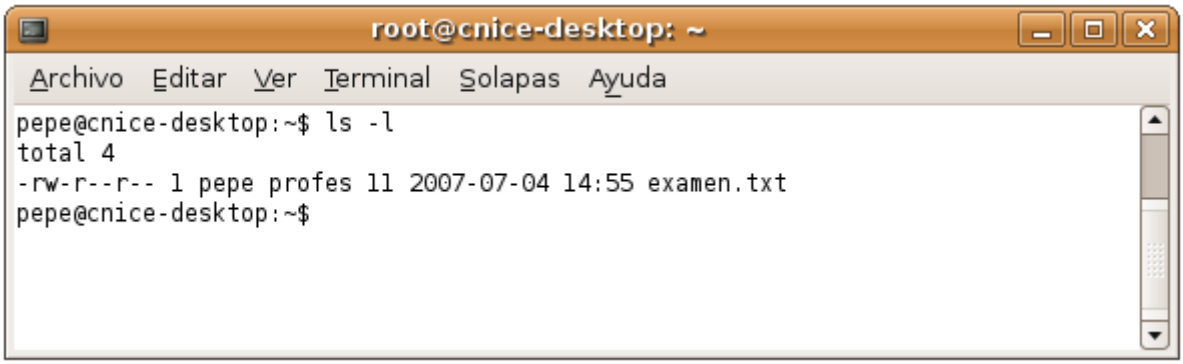
Las cuentas de usuario no solo ofrecen al usuario un nombre y una contraseña, también le proporciona una **ruta para almacenar sus documentos** y su perfil generalmente dentro de la carpeta /home/nombre-usuario y comúnmente denominada **carpeta home del usuario** y un **intérprete de comandos (shell)** que le permitirá ejecutar aplicaciones.

Cuando el usuario ejecuta una aplicación, el sistema carga la aplicación en memoria y la ejecuta. En el argot informático a las aplicaciones que se están ejecutando en un momento determinado se les denomina **procesos**. Los procesos en ejecución pertenecen a algún usuario. El sistema asigna a los procesos el usuario que los ejecuta. Ejemplo, si el usuario "pepe" ejecuta la aplicación "konqueror", en la lista de procesos del sistema aparecerá un nuevo proceso llamado "konqueror" cuyo propietario es "pepe". **Obligatoriamente, todos los procesos del sistema pertenecen a algún usuario**. Ejecutando el comando 'ps aux' podemos ver todos los procesos en ejecución. Si ejecutamos el comando 'top' lo veremos a tiempo real.



Mostrando procesos con top. La segunda columna indica el propietario

Cuando se crea un nuevo archivo, el propietario del archivo será el **usuario** que lo ha creado y el grupo del archivo será el **grupo principal** de dicho usuario. Ejemplo, si "pepe" cuyo grupo principal es "profes" crea un nuevo archivo llamado examen.txt, el propietario de examen.txt será "pepe" y el grupo propietario será "profes", o lo que es lo mismo, el archivo pertenecerá al usuario pepe y al grupo profes. **Obligatoriamente, todos los archivos del sistema pertenecen a algún usuario y a algún grupo**.



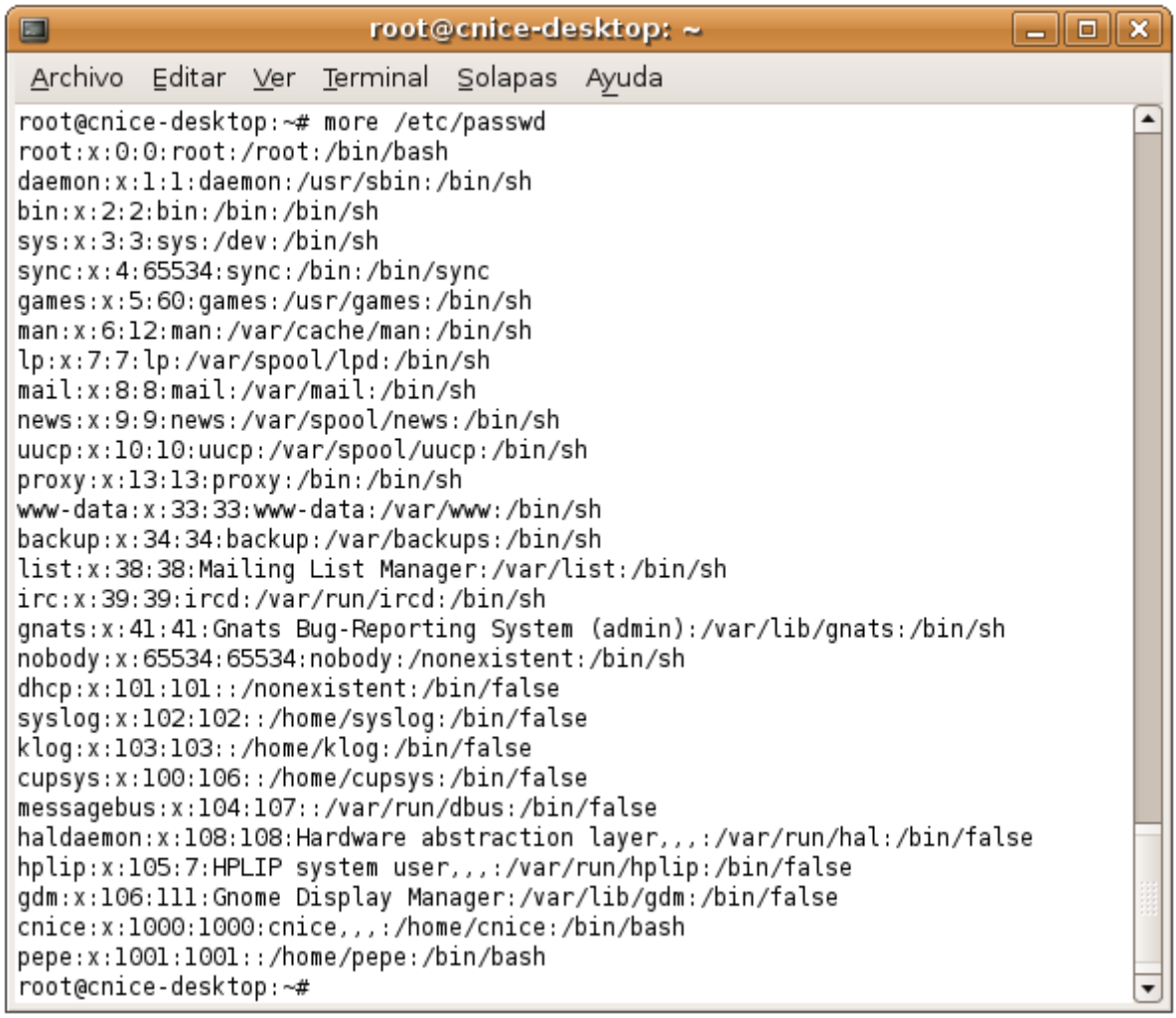
examen.txt pertenece al usuario pepe y al grupo profes

La cuenta de usuario le permite acceder al sistema tanto de forma presencial (sentado delante del ordenador) como de forma remota accediendo desde otro equipo por la red. Los permisos que tiene el usuario cuando utiliza el sistema presencialmente son los mismos que tiene cuando lo hace remotamente. Lo habitual es utilizar el sistema de forma remota ya que al ser Unix un sistema multiusuario, la única forma de que varios usuarios lo utilicen de forma simultánea es remotamente.

El sistema Unix codifica los usuarios con un número diferente a cada uno que es el **identificador de usuario (uid = User IDentifier)**. Internamente el sistema trabaja con el uid, no con el nombre del usuario. Normalmente a los usuarios que creemos se les asignan uids desde 1000 en adelante. Los números uid menores que 100 se reservan para usuarios especiales del sistema.

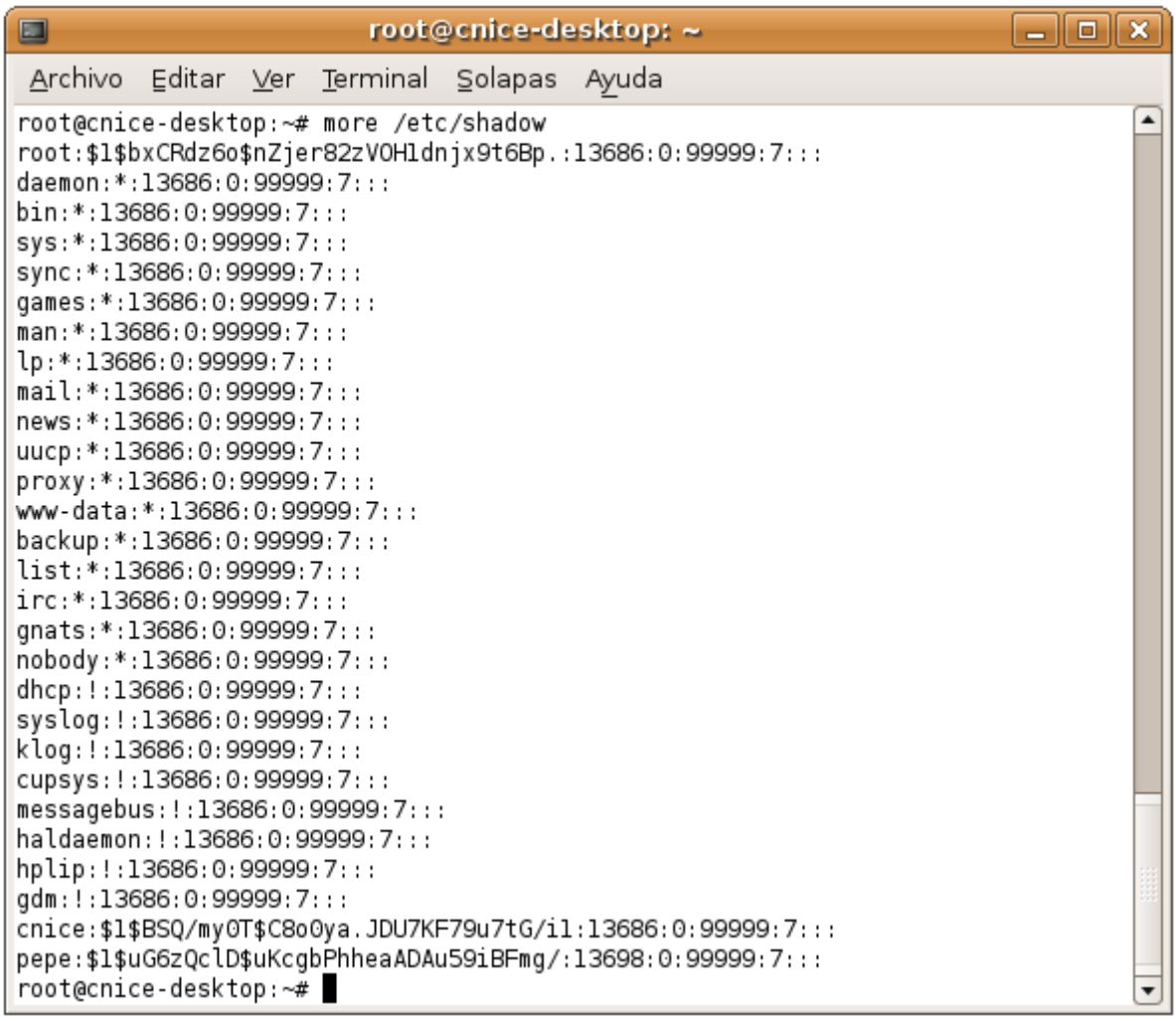
En Unix por defecto, la información de los usuarios de un sistema se guarda en el archivo **/etc/passwd**. Es un archivo de texto que puede visualizarse con cualquier editor. Cada linea del archivo /etc/passwd

almacena los parámetros de un usuario. Solo puede modificarlo el administrador (root). A continuación mostramos el archivo passwd:



Volcado del archivo /etc/passwd

Las contraseñas de cada usuario se guardan encriptadas con un sistema de codificación irreversible, en el archivo **/etc/shadow** que también es un archivo de texto.



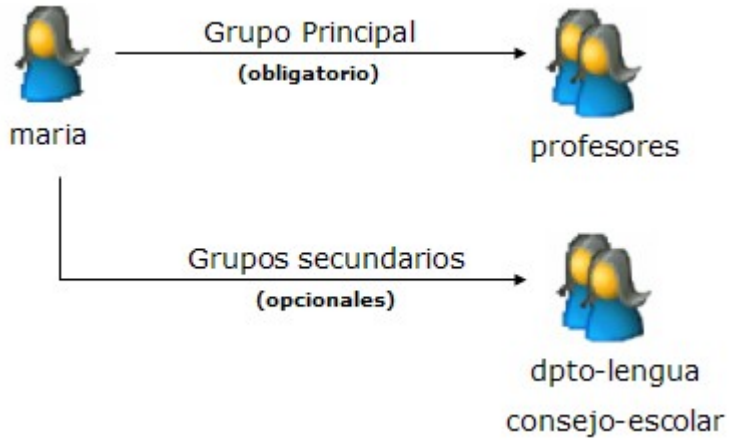
Volcado del archivo /etc/shadow

Grupos de usuarios

Para poder administrar los permisos de los usuarios de una forma más flexible, el sistema Unix permite la organización de usuarios en grupos y establecer permisos a los grupos.

Ejemplo, si en un centro educativo el grupo "profesores" tiene acceso a ciertas carpetas, cuando demos de alta un profesor nuevo, tan solo tendremos que añadirle al grupo "profesores" para que pueda acceder a todas esas carpetas. Es lo que se denomina administración de permisos por grupos.

Todos los usuarios pertenecen al menos a un grupo que es el **grupo principal del usuario**, también llamado grupo primario del usuario, pero pueden pertenecer a más grupos. En caso de que pertenezcan a más grupos, éstos serán **grupos secundarios**.



Todo usuario debe pertenecer a un grupo principal obligatoriamente

Los grupos pueden contener varios usuarios. Los grupos de usuarios solo pueden contener usuarios, nunca podrán contener a otros grupos.

El sistema Unix codifica los grupos de usuarios con un número diferente a cada uno que es el **identificador de grupo (gid = Group IDentifier)**. Internamente el sistema trabaja con el gid, no con el nombre del grupo. Normalmente a los grupos que creemos se les asignan gids desde 1000 en adelante. Los números gid menores que 100 se reservan para grupos especiales del sistema.

En Unix por defecto, la información de los grupos de un sistema se guarda en el archivo **/etc/group**. Es un archivo de texto que puede visualizarse con cualquier editor. Cada linea del archivo /etc/group almacena los parámetros del grupo y los usuarios que contiene. Solo puede modificarlo el administrador (root). Las contraseñas de los grupos se guardan encriptadas con un sistema de codificación irreversible, en el archivo **/etc/gshadow** que también es un archivo de texto.

Usuario root

El usuario root, a veces llamado **superusuario**, es el usuario administrador del sistema. Está identificado con el **número de usuario cero (uid=0)** y tiene permisos sobre todo el sistema sin ningún tipo de restricción. El usuario root puede acceder a cualquier archivo, ejecutar, instalar y desinstalar cualquier aplicación, modificar los archivos de configuración del sistema y administrar usuarios. Si dispones de la contraseña de root tendrás control total sobre todo el sistema.

Administración de usuarios y grupos

En nuestro centro tanto el profesorado como el alumnado acostumbran a personalizar su escritorio, incluyendo accesos directos, carpetas y demás vínculos que considera importantes y cómodos en su tarea diaria. Pero todo esto incomoda, en muchas ocasiones, a otros usuarios que también utilizan el mismo equipo. Esta situación es especialmente patente en las aulas de informática, donde parece que el alumnado compite en crear ámbitos de trabajo extravagantes e inútiles, con la pérdida de tiempo en la carga del sistema operativo y que posteriormente repercute en nuestro trabajo al tener que estar reinstalando equipos, eliminando programas, etc.

Para resolver el problema anterior es necesario disponer de una base de datos de usuarios y grupos donde poder asignar o denegar permisos de acceso a los recursos autenticados de nuestro centro. Ello permitirá personalizar los entornos de trabajo de cada usuario, dando solución al problema planteado tanto por el profesorado como por el alumnado.

La administración de usuarios y grupos **solamente puede realizarlas el usuario root** utilizando los comandos de gestión de usuarios. Las tareas y los comandos para realizarlas son:

- Creación de usuarios / useradd
- Modificación de usuarios / usermod

- Eliminación de usuarios / userdel
- Creación de grupos / groupadd
- Modificación de grupos / groupmod
- Eliminación de grupos / groupdel
- Añadir usuarios a un grupo / adduser
- Quitar usuarios de un grupo / deluser

Creación de usuarios

El comando useradd permite añadir un usuario indicando como parámetros la información particular para crear el usuario en la misma líne de comandos. La sintaxis es:

```
#useradd [opciones] nombre-usuario
```

Entre las opciones más destacables tenemos:

- -g: Grupo principal que queremos tenga el usuario (debe existir)
- -d: Carpeta home del usuario. Suele ser /home/nombre-usuario
- -m: Crear carpeta home si es que no existe.
- -s: Intérprete de comandos (shell) del usuario. Suele ser /bin/bash

Ejemplo, si deseamos crear un usuario llamado 'pedro' cuyo grupo principal sea 'profesores', cuya carpeta home sea /home/pedro y su intérprete de comandos sea /bin/bash, ejecutaremos el siguiente comando:

```
// Crear un usuario
     # useradd -g profesores -d /home/pedro -m -s /bin/bash pedro
```

De ésta manera habremos creado al usuario pedro y su carpeta home. Si no utilizamos la opción -m, no se creará la carpeta home del usuario; en tal caso tendríamos que crearla manualmente. Tan solo nos quedará establecer su contraseña con el comando passwd:

```
// Establecer la contraseña del usuario
     # passwd pedro
```

Entonces el sistema nos preguntará dos veces la contraseña que queremos asignar a pedro.

El comando useradd permite crear muchos usuarios automáticamente mediante **archivos de comandos (scripts).**

Se recomienda que el nombre de usuario sea en minúsculas y además de letras también puede contener números y algún signo como guiones normales y guiones bajos. Debemos recordar que unix distingue entre mayúsculas y minúsculas, es decir, Pepe es distinto de pepe.

Modificación de usuarios

Se utiliza el comando **usermod** y permite cambiar el nombre del usuario, su carpeta home, su intérprete de comandos, los grupos a los que pertenece y algunos otros parámetros.

```
// Cambiar el home de un usuario
     # usermod -d /home/carpeta_pedro pedro
```

Eliminación de usuarios

Se realiza con el comando userdel seguido del nombre del usuario. Con la opción -r eliminará también su carpeta home, ejemplo:

```
// Eliminación de un usuario
     # userdel -r pedro
```

Eliminaría el usuario pedro y su carpeta home.

Creación de grupos

El comando groupadd permite añadir un grupo indicando como parámetro el nombre del grupo. Ejemplo, si deseamos crear un grupo llamado 'alumnos' ejecutaremos:

```
// Añadir un grupo
     # groupadd alumnos
```

Modificación de grupos

El comando **groupmod** permite modificar el nombre de un grupo o el gid del mismo. La sintaxis es:

```
# groupmod [-g nuevo-gid] [-n nuevo-nombre] nombre-grupo

// Cambiar el gid del grupo profesores
     # groupmod -g 2000 profesores
```

Eliminación de grupos

Se realiza con el comando groupdel seguido del nombre del grupo, ejemplo:

```
// Eliminación de un grupo
     # groupdel profesores
```

Eliminaría el grupo profesores. Si algún usuario tuviera dicho grupo como grupo primario, el comando groupdel **no** eliminará el grupo.

Añadir usuarios a un grupo

Se utiliza el comando adduser seguido del nombre del usuario y del nombre del grupo al que queremos añadirle, ejemplo:

```
// Añadir un usuario a un grupo
     # adduser juan profesores // añade juan al grupo profesores
```

Quitar usuarios de un grupo

Se utiliza el comando **deluser** seguido del nombre del usuario y del nombre del grupo del que queremos quitarle, ejemplo:

```
// Quitar a un usuario de un grupo
     # deluser juan profesores // quita a juan del grupo profesores
```

Para más información de todos estos comandos se puede consultar la ayuda del manual ejecutando man seguido del nombre del comando, ejemplo man adduser.

Herramienta gráfica de administración de usuarios

Ubuntu dispone de una herramienta gráfica de administración de usuarios que es 'users-admin'. Para ejecutarla podemos abrir una consola de root y ejecutar users-admin o si hemos iniciado sesión como root, podemos pulsar Alt+F2 y ejecutar users-admin.

Se trata de una herramienta bastante intuitiva que dispone de una pestaña para administrar usuarios y otra para administrar grupos.

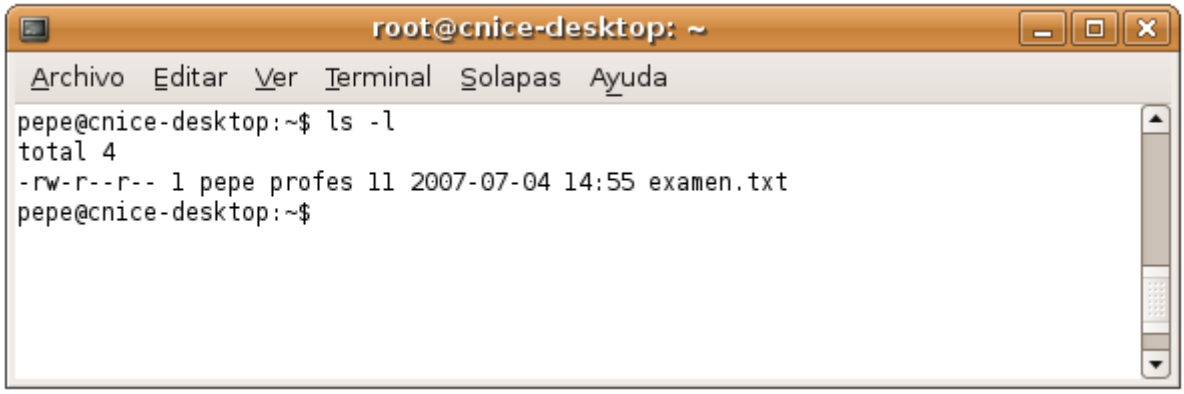
Permisos de archivos y carpetas

Usuario propietario y grupo propietario de un archivo

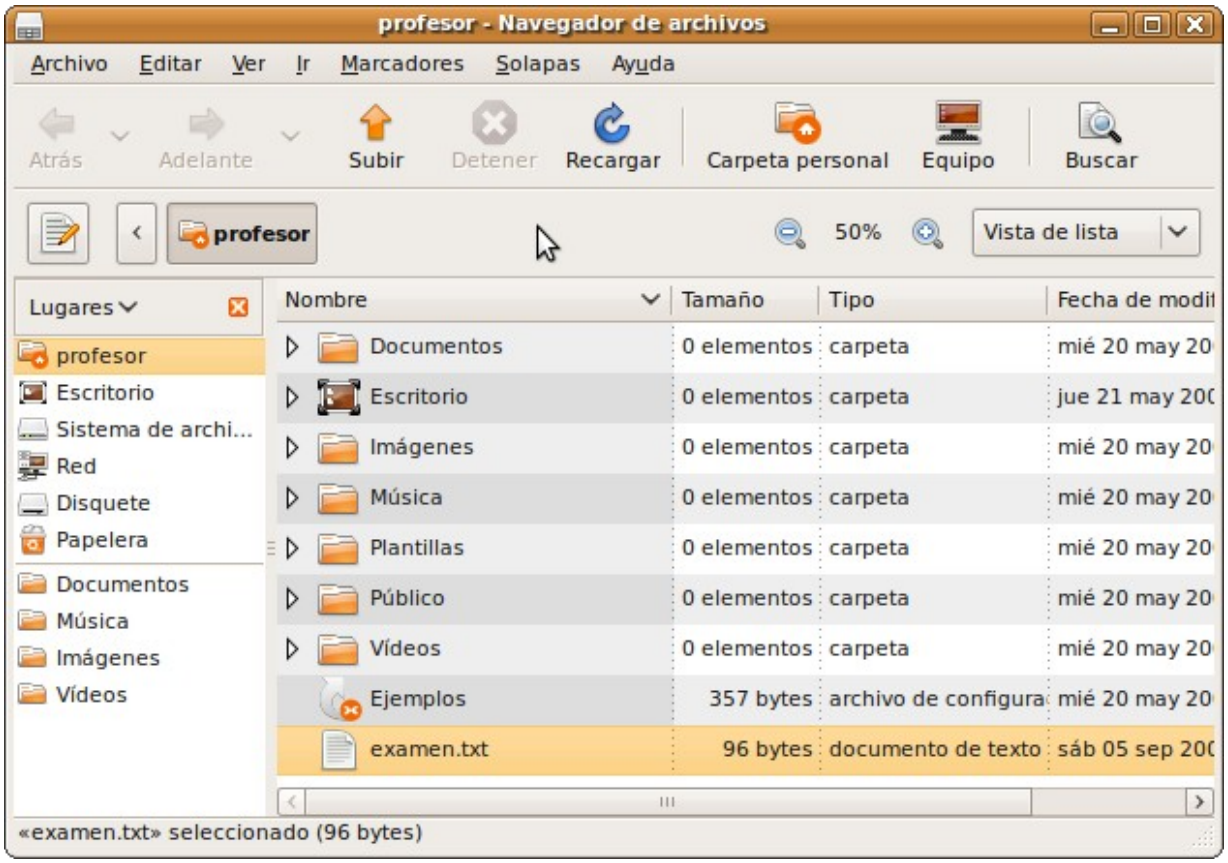
Anteriormente se ha comentado que en Unix todos los archivos pertenecen obligatoriamente a un usuario y a un grupo. Cuando un usuario crea un nuevo archivo, el propietario del archivo será el usuario que lo ha creado y el grupo del archivo será el grupo principal de dicho usuario.

Ejemplo, si un usuario llamado 'pepe' cuyo grupo principal es el grupo 'profesores' crea un nuevo archivo, el propietario del archivo será 'pepe' y el grupo propietario del archivo será 'profesores', o lo que es lo mismo, el archivo pertenecerá al usuario pepe y al grupo profesores. Obligatoriamente, todos los archivos del sistema pertenecen a algún usuario y a algún grupo.

Con el comando **ls** añadiendo la opción **-l** (formato largo) podemos visualizar el usuario propietario y el grupo propietario del archivo, ejemplo:



Comprobamos que el usuario propietario es pepe y el grupo propietario es profesores. La misma información podemos verla desde el administrador de archivos si vamos a la carpeta /home/pepe y mostramos las columnas correspondientes:



Tipos de permisos

En los Sistemas Unix, la gestión de los permisos que los usuarios y los grupos de usuarios tienen sobre los archivos y las carpetas, se realiza mediante un sencillo esquema de tres tipos de permisos que son:

- Permiso de lectura
- Permiso de escritura
- Permiso de ejecución

El significado de éstos permisos difiere si se tienen sobre archivos o sobre carpetas. A continuación veremos el significado para cada uno de los casos:

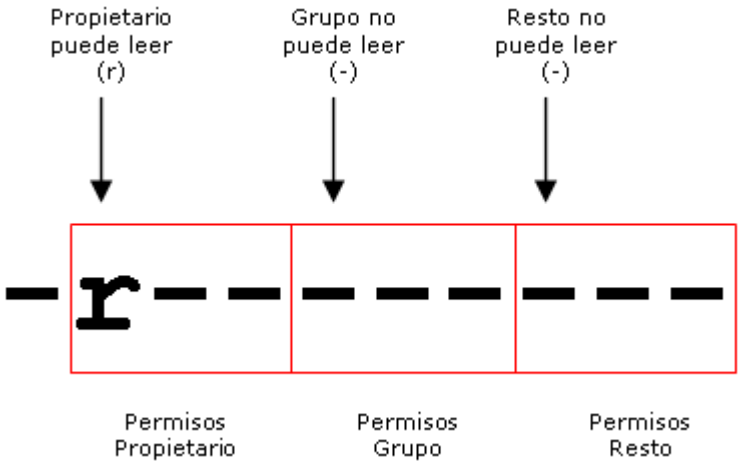
Permiso de lectura

Cuando un usuario tiene permiso de lectura de un archivo significa que puede leerlo o visualizarlo, bien sea

con una aplicación o mediante comandos. Ejemplo, si tenemos permiso de lectura sobre el archivo examen.txt, significa que podemos ver el contenido del archivo. Si el usuario no tiene permiso de lectura, no podrá ver el contenido del archivo.

Cuando un usuario tiene permiso de lectura de una carpeta, significa que puede visualizar el contenido de la carpeta, es decir, puede ver los archivos y carpetas que contiene, bien sea con el comando 'ls' o con un explorador de archivos como Konqueror. Si el usuario no tiene permiso de lectura sobre la carpeta, no podrá ver lo que contiene.

El permiso de lectura se simboliza con la letra 'r' del inglés 'read'.

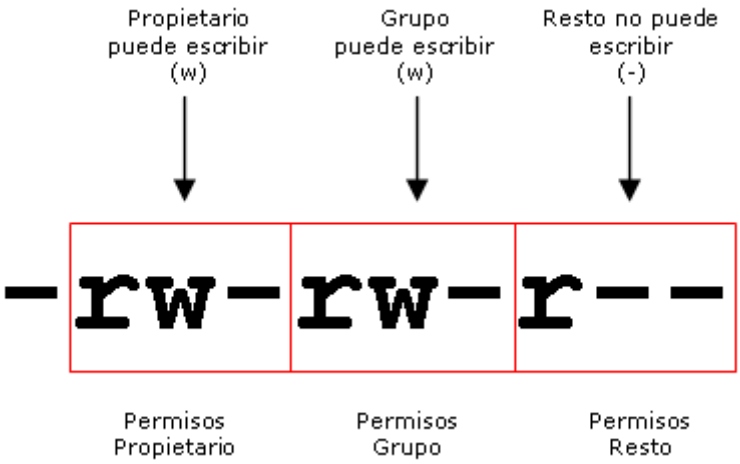


Permiso de escritura

Cuando un usuario tiene permiso de escritura sobre un archivo significa que puede modificar su contenido, e incluso borrarlo. También le da derecho a cambiar los permisos del archivo mediante el comando chmod así como cambiar su propietario y el grupo propietario mediante el comando chown. Si el usuario no tiene permiso de escritura, no podrá modificar el contenido del archivo.

Cuando un usuario tiene permiso de escritura sobre una carpeta, significa que puede modificar el contenido de la carpeta, es decir, puede crear y eliminar archivos y otras carpetas dentro de ella. Si el usuario no tiene permiso de escritura sobre la carpeta, no podrá crear ni eliminar archivos ni carpetas dentro de ella.

El permiso de escritura se simboliza con la letra 'w' del inglés 'write'.



Permiso de ejecución

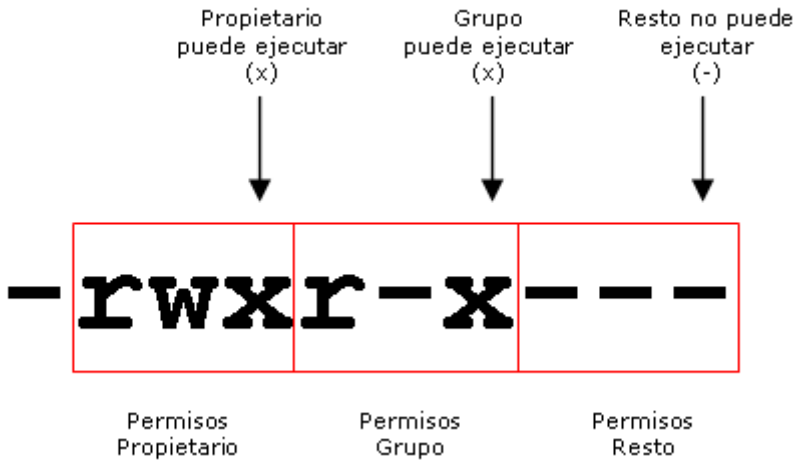
Cuando un usuario tiene permiso de ejecución de un archivo significa que puede ejecutarlo. Si el usuario no dispone de permiso de ejecución, no podrá ejecutarlo aunque sea una aplicación.

Los únicos archivos ejecutables son las aplicaciones y los archivos de comandos (scripts). Si tratamos de

ejecutar un archivo no ejecutable, dará errores.

Cuando un usuario tiene permiso de ejecución sobre una carpeta, significa que puede entrar en ella, bien sea con el comando 'cd' o con un explorador de archivos como Konqueror. Si no dispone del permiso de ejecución significa que no puede ir a dicha carpeta.

El permiso de ejecución se simboliza con la letra 'x' del inglés 'eXecute'.



¿A quién se puede otorgar permisos?

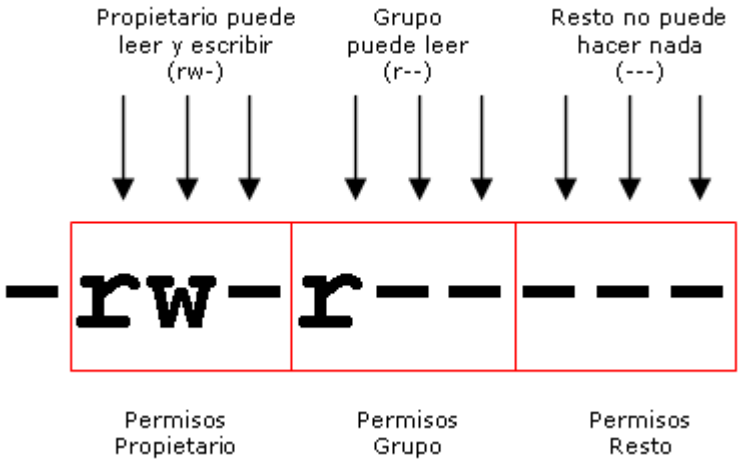
Los permisos solamente pueden ser otorgados a tres tipos o grupos de usuarios:

- Al usuario propietario del archivo
- Al grupo propietario del archivo
- Al resto de usuarios del sistema (todos menos el propietario)

Se pueden dar permisos de lectura, escritura, ejecución ó combinación de ambos al usuario propietario del archivo, al grupo propietario del archivo o al resto de usuarios del sistema. En Unix no existe la posibilidad de asignar permisos a usuarios concretos ni a grupos concretos, tan solo se puede asignar permisos al usuario propietario, al grupo propietario o al resto de usuarios.

Ejemplo, si disponemos de un archivo llamado 'examen.txt' cuyo propietario es 'pepe' y cuyo grupo propietario es 'profesores', se pueden dar permisos de lectura, escritura, ejecución ó combinación de ambos al usuario 'pepe', al grupo 'profesores' y al resto de usuarios, pero no podrémos dar permisos a otros usuarios distintos de pepe (juan, luis, pedro,...) ni a otros grupos (alumnos, directivos, personal,...) ya que el esquema Unix no lo permite.

Supongamos que la siguiente figura representa los permisos de examen.txt:



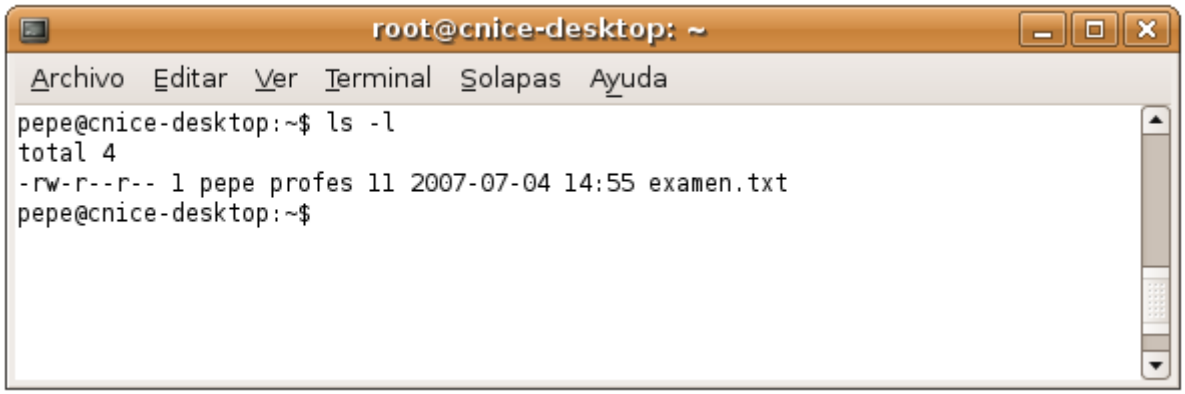
El usuario propietario (pepe) podrá leer y escribir en el documento. Los pertenecientes al grupo profesores podrán leerlo y el resto no podrá hacer nada.

Si deseo que otros usuarios tengan algún permiso sobre el archivo 'examen.txt', no me quedará más remedio que incluirlos en el grupo profesores u otorgar el permiso al resto de usuarios pero si hago esto último, absolutamente todos los usuarios del sistema gozarán del permiso, por eso no se recomienda salvo que eso sea nuestra intención.

Para poder cambiar permisos sobre un archivo, es necesario poseer el permiso de escritura sobre el mismo. El usuario root puede modificar los permisos de cuaquier archivo ya que tiene acceso total sin restricciones a la administración del sistema.

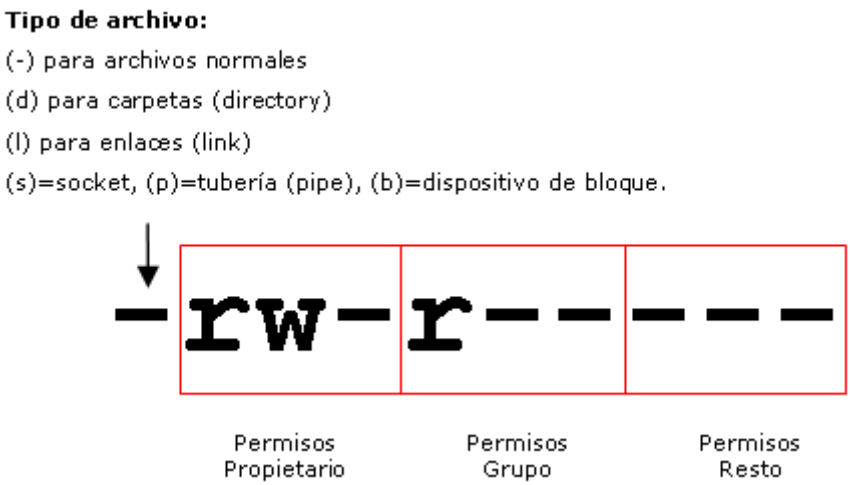
Visualizar los permisos de un archivo o carpeta

Con el comando **ls -l** podemos visualizar los permisos de los archivos o carpetas. Al ejecutar el comando aparecen todos los archivos, uno por línea. El bloque de 10 caracteres del principio simboliza el tipo de archivo y los permisos.



El primer caracter indica de qué **tipo de archivo** se trata. Si es un guión '-' significa que se trata de un **archivo normal**, la letra 'd' significa que se trata de una **carpeta** (directory), la letra 'l' significa que se trata de un **enlace** (link). Otros valores son s, p, b que se refieren a sockets, tuberías (pipe) y dispositivos de bloque respectivamente.

Los 9 caracteres siguientes simbolizan los **permisos del usuario propietario** (3 caracteres), los **permisos del grupo propietario** (3 caracteres) y los **permisos del resto de usuarios** (3 caracteres). Vienen codificados con las letras r, w y x que se refieren a los permisos de lectura, escritura y ejecución. Si en lugar de aparecer dichas letras aparecen guiones significa que se carece de dicho permiso. Ejemplo, si los diez primeros caracteres son -rw-r----- significa que es un archivo normal, que el usuario propietario dispone de permisos de lectura y escritura pero no de ejecución, que el grupo propietario dispone tan solo de permiso de lectura y el resto de usuarios no dispone de ningún permiso. Veámoslo en la siguiente imagen:



Permisos de lectura y escritura para el propietario y lectura para el grupo

En el siguiente ejemplo vemos que pepe tiene permiso de lectura y escritura y que el resto solo tiene permiso de lectura tanto sobre el archivo 'apuntes.doc' como sobre el archivo 'examen.txt'.

```
// Visualización de permisos
     root@knoppix37:/home/pepe# ls -l

     total 8

     -rw-r--r--  1 pepe profesores 359 2005-09-28 18:02 apuntes.doc

     -rw-r--r--  1 pepe profesores  11 2005-09-27 19:26 examen.txt
```

Cambio de permisos

Para cambiar los permisos de un archivo o una carpeta es necesario disponer del permiso de escritura (w) sobre dicho archivo o carpeta. Para hacerlo, se utiliza el comando **chmod**. La sintaxis del comando chmod es la siguiente:

```
 #chmod [opciones] permiso nombre_archivo_o_carpeta
```

Los permisos se pueden representar de dos formas. La primera es mediante las iniciales de a quién va dirigido el permiso (usuario=u, grupo=g, resto=o (other)), seguido de un signo + si se quiere añadir permiso o un signo - si se quiere quitar y seguido del tipo de permiso (lectura=r, escritura=w y ejecución=x). Ejemplos:

```
// Dar permiso de escritura al usuario propietario sobre el archivo 'examen.txt'
     # chmod u+w examen.txt



// Quitar permiso de escritura al resto de usuarios sobre el archivo
'examen.txt'
     # chmod o-w examen.txt



// Dar permiso de ejecución al grupo propietario sobre el archivo
'/usr/bin/games/tetris'
     # chmod g+x /usr/bin/games/tetris



// Dar permiso de lectura al grupo propietario sobre el archivo 'examen.txt'
     # chmod g+r examen.txt



// Se pueden poner varios permisos juntos separados por comas
     # chmod u+w,g-r,o-r examen.txt



// Se pueden poner varios usuarios juntos
     # chmod ug+w examen.txt
```

La segunda forma de representar los permisos es mediante un código numérico cuya transformación al

binario representaría la activación o desactivación de los permisos. El código numérico está compuesto por tres cifras entre 0 y 7. La primera de ellas representaría los permisos del usuario propietario, la segunda los del grupo propietario y la tercera los del resto de usuarios.

En binario, las combinaciones representan el tipo de permisos. El bit más a la derecha (menos significativo) se refiere al permiso de ejecución (1=activar y 0=desactivar). El bit central se refiere al permiso de escritura y el bit más a la izquierda se refiere al permiso de lectura. La siguiente tabla muestra las 8 combinaciones posibles:

Código	Binario	Permisos efectivos
0	0 0 0	- - -
1	0 0 1	- - x
2	0 1 0	- w -
3	0 1 1	- w x
4	1 0 0	r - -
5	1 0 1	r - x
6	1 1 0	r w -
7	1 1 1	r w x

Si deseamos otorgar sólo permiso de lectura, el código a utilizar es el 4. Si deseamos otorgar sólo permiso de lectura y ejecución, el código es el 5. Si deseamos otorgar sólo permiso de lectura y escritura, el código es el 6. Si deseamos otorgar todos los permisos, el código es el 7. Si deseamos quitar todos los permisos, el código es el 0. Ejemplos:

```
// Dar todos los permisos al usuario y ninguno ni al grupo ni al resto
     # chmod 700 examen.txt


// Dar al usuario y al grupo permisos de lectura y ejecución y ninguno al resto
     # chmod 550 examen.txt


// Dar todos los permisos al usuario y lectura y ejecución al grupo y al resto
     # chmod 755 /usr/bin/games/tetris


// Dar todos los permisos al usuario y de lectura al resto, sobre todos los
archivos
     # chmod 744 *


// Cambiar permisos a todos los archivos incluyendo subcarpetas
     # chmod -R 744 *
```

Existe la posibilidad de cambiar los permisos utilizando el explorador de archivos. Para ello tan solo hay que

seleccionar los archivos o carpetas y haciendo clic sobre la selección con el botón derecho del ratón > **Propiedades**, nos aparecerá la ventana de propiedades. Haciendo clic en la pestaña **Permisos** podremos establecer los permisos de una forma sencilla y haciendo clic en 'Permisos avanzados' de una forma avanzada.

Estableciendo permisos desde el administrador de archivos

Bits SUID y SGID

El **bit SUID** es una extensión del permiso de ejecución. Se utiliza en escasas ocasiones y sirve para que cuando un usuario ejecute una aplicación, ésta se ejecute con permisos del usuario propietario en lugar de hacerlo con los del usuario que ejecuta la aplicación, es decir, es equivalente a que sea ejecutada por el propietario.

Para activar el **bit SUID**, se puede ejecutar el comando **chmod u+s nombre_archivo** o sumar 4000 al número en octal si utilizamos dicho sistema. También se puede hacer lo mismo para el grupo, es el denominado **bit SGID** sumando 2000 al número en octal. Activar los bits SUID ó SGID puede ocasionar problemas de seguridad sobre todo si el propietario es root.

Si aplicamos el **bit SGID a una carpeta**, todas las subcarpetas y archivos creados dentro de dicha carpeta tendrán como grupo propietario el **grupo propietario de la carpeta** en lugar del grupo primario del usuario que ha creado el archivo. Es una ventaja cuando varias personas pertenecientes a un mismo grupo, trabajan juntas con archivos almacenados en una misma carpeta. Si otorgamos permisos de lectura y escritura al grupo, los archivos podrán ser modificados por todos los miembros del grupo y cuando cualquiera de ellos cree un archivo, éste pertenecerá al grupo.

Máscaras

Cuando se crea un archivo, los permisos originales por defecto son 666 y cuando se crea una carpeta, los permisos por defecto son 777. Dichos permisos por defecto pueden modificarse con el comando **umask**.

Con **umask** podemos definir la máscara de permisos, cuyo valor original es 000. El permiso por defecto será el resultado de restar del permiso original, el valor de la máscara. Si deseamos que los archivos se

creen con permisos 644 (lo más habitual), pondremos máscara 022 ya que 666-022=644. En el caso de las carpetas, el permiso efectivo será 755 ya que 777-022=755. Si analizamos el valor de la máscara en binario, cada bit a '1' desactiva un permiso y cada bit a '0' lo activa, es decir, si tiene un valor 022 (000 010 010) cuando creemos una carpeta, tendrá permisos rwxr-xr-x y cuando creemos un archivo tendrá permisos rw-r--r-- ya que el permiso de ejecución para archivos hay que fijarle con chmod al tener los archivos el permiso original 666.

Cada usuario tiene su máscara. Se puede fijar la máscara por defecto para todos los usuarios en el archivo /etc/profile o para cada usuario en el archivo /home/usuario/.bashrc

```
// Ejemplo de uso de umask
     pepe@3[pruebas]$ umask

     0002

     pepe@3[pruebas]$ mkdir nueva-carpeta

     pepe@3[pruebas]$ ls -l

     drwxrwxr-x    2 pepe     profes       1024 Feb 12 19:46 nueva-carpeta

     pepe@3[pruebas]$ umask 022

     pepe@3[pruebas]$ mkdir otra-carpeta

     pepe@3[pruebas]$ ls -l

     drwxrwxr-x    2 pepe     profes       1024 Feb 12 19:46 nueva-carpeta

     drwxr-xr-x    2 pepe     profes       1024 Feb 12 19:46 otra-carpeta

     pepe@3[pruebas]$
```

La modificacion con umask de la mascara por defecto no afecta a los archivos y carpetas existentes sino **solo a los nuevos que cree ese usuario a partir de ese momento.**.

Grupos privados de usuario

Para hacer más flexible el esquema de permisos Unix, se recomienda utilizar grupos privados de usuario. Consiste en crear un **nuevo grupo** con el mismo nombre del usuario, cada vez que se crea un **nuevo usuario** y hacer que el grupo principal del nuevo usuario sea el nuevo grupo.

Ejemplo, si creamos un **usuario pepe**, crearemos también un grupo llamado pepe y haremos que el grupo primario del usuario pepe sea el **grupo pepe**.

En el siguiente ejemplo observamos que el UID del usuario pepe es 1002 y que su grupo principal es el 1003 que corresponde al GID del grupo pepe. También vemos que si creamos un nuevo archivo, pertenecerá al **usuario pepe** y al **grupo pepe**.

```
// Ejemplo: Usuario pepe y grupo pepe
     pepe@3[pruebas]$ more /etc/passwd |grep pepe

     pepe:x:1002:1003::/home/pepe:

     pepe@3[pruebas]$ more /etc/group |grep pepe

     pepe:x:1003:
```

```
pepe@3[pruebas]$ ls > archivo.txt

pepe@3[pruebas]$ ls -l

-rw-rw-r--    1 pepe     pepe           12 Feb 12 20:17 archivo.txt

pepe@3[pruebas]$
```

Aunque parezca inservible, la creación de un grupo personal para cada usuario, permitirá crear otros grupos mediante los cuales, diferentes personas puedan trabajar de forma colaborativa sobre los archivos dentro de una carpeta concreta. Veámoslo mejor con un ejemplo:

Supongamos que creamos una carpeta llamada 'examenes' que pertenezca al grupo profesores. Si establecemos el bit SGID en dicha carpeta con el comando 'chmod g+s examenes', todos los archivos que se creen dentro de dicha carpeta tendrán como grupo propietario el grupo profesores. Si todos los usuarios utilizan máscara 002, los permisos de los archivos serán 664 con lo cual, cualquier integrante del grupo profesores podra visualizar y modificar los archivos.

El problema de usar la máscara 002 es que cualquiera que pertenezca al grupo principal de un usuario, tendría acceso de escritura sobre sus archivos, pero esto no sucederá nunca ya que cada usuario tiene su propio grupo principal y nadie más pertenece a él.

Cambiar usuario propietario y grupo propietario

Para poder cambiar el usuario propietario y el grupo propietario de un archivo o carpeta se utiliza el comando chown (change owner). Para ello hay que **disponer de permisos de escritura** sobre el archivo o carpeta. La sintaxis del comando es:

```
# chown nuevo_usuario[.nuevo_grupo] nombre_archivo
```

En el siguiente ejemplo vemos una secuencia de comandos en la que inicialmente comprobamos que el archivo 'examen.txt' pertenece al usuario pepe y al grupo profesores. Posteriormente hacemos que pertenezca al usuario luis y luego hacemos que pertenezca al usuario pedro y al grupo alumnos:

```
// Cambiar propietario y grupo propietario
     root@knoppix37:/home/pepe# ls -l

     total 4

     -rw-rw-r--  1 pepe profesores 11 2005-09-28 20:15 examen.txt

     root@knoppix37:/home/pepe# chown luis examen.txt

     root@knoppix37:/home/pepe# ls -l

     total 4

     -rw-rw-r--  1 luis profesores 11 2005-09-28 20:15 examen.txt

     root@knoppix37:/home/pepe# chown pedro.alumnos examen.txt

     root@knoppix37:/home/pepe# ls -l

     total 4

     -rw-rw-r--  1 pedro alumnos 11 2005-09-28 20:15 examen.txt
```

```
root@knoppix37:/home/pepe#
```

3.- Servidor DHCP

En nuestro centro educativo la configuración y modificación de las direcciones IP de los equipos de las distintas dependencias es un verdadero quebradero de cabeza, pues obliga al administrador de la red a desplazarse hasta el lugar donde se encuentra ubicado el equipo en cuestión para proceder a su configuración IP, sin la garantía de que no se pueda cometer un error al especificar dicha configuración. En muchos casos es el alumnado quién realiza cambios con afán investigador en dicho direccionamiento, en otros casos son las circunstancias o los movimientos de las ubicaciones físicas de los equipos los que obligan a realizar modificaciones en la dirección IP o puerta de enlace, por ejemplo. Estos cambios crean conflictos a medida que la red crece, de modo que parece lógico instalar un sistema más cómodo de direccionamiento, según el cual cada máquina que inicie sesión en nuestro centro reciba dinámicamente del servidor una dirección IP, una máscara, una puerta de enlace y un servidor DNS que le permitan la salida a Internet así como el acceso a todos los servicios de nuestra Intranet, de forma que cuando sea preciso realizar cualquier cambio en la configuración IP de dichos equipos, sea realizado desde el servidor sin necesidad de desplazarse físicamente hasta la dependencia correspondiente.

Definición de Servidor DHCP

El protocolo de configuración dinámica de host (DHCP, Dynamic Host Configuration Protocol) es un estándar TCP/IP diseñado para simplificar la administración de la configuración IP de los equipos de nuestra red.

Si disponemos de un servidor DHCP, la configuración IP de los PCs puede hacerse de forma automática sin necesidad de hacerlo manualmente.

Un servidor DHCP es un servidor que recibe peticiones de clientes solicitando una configuración de red IP. El servidor responderá a dichas peticiones proporcionando los parámetros que permitan a los clientes autoconfigurarse. Para que un PC solicite la configuración a un servidor, en la configuración de red de los PCs hay que seleccionar la opción 'Obtener dirección IP automáticamente'.

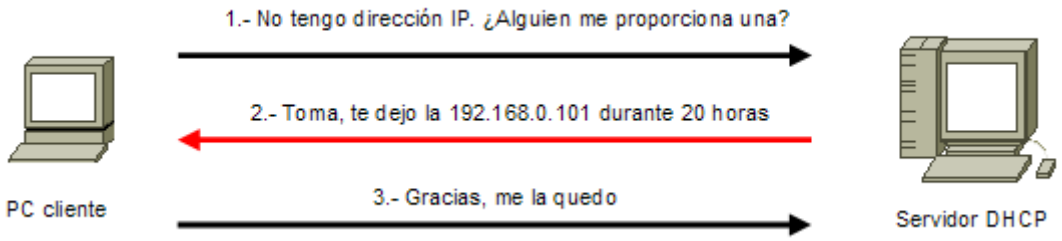
El servidor proporcionará al cliente al menos los siguientes parámetros:

- Dirección IP
- Máscara de subred

Opcionalmente, el servidor DHCP podrá proporcionar otros parámetros de configuración tales como:

- Puerta de enlace
- Servidores DNS
- Muchos otros parámetros más

El servidor DHCP proporciona una configuración de red TCP/IP segura y evita conflictos de direcciones repetidas. Utiliza un modelo cliente-servidor en el que el servidor DHCP mantiene una administración centralizada de las direcciones IP utilizadas en la red. Los clientes podrán solicitar al servidor una dirección IP y así poder integrarse en la red.



El servidor solo asigna direcciones dentro de un rango prefijado. Si por error hemos configurado manualmente una IP estática perteneciente al rango gestionado por nuestro servidor DHCP, podría ocurrir que dicha dirección sea asignada dinámicamente a otro PC, provocándose un **conflicto de IP**. En ese caso el cliente solicitará y comprobará, otra dirección IP, hasta que obtenga una dirección IP que no esté asignada actualmente a ningún otro equipo de nuestra red.

La primera vez que seleccionamos en un PC que su configuración IP se determine por DHCP, éste pasará a

convertirse en un **cliente DHCP** e intentará localizar un servidor DHCP para obtener una configuración desde el mismo. Si no encuentra ningún servidor DHCP, el cliente no podrá disponer de dirección IP y por lo tanto no podrá comunicarse con la red. Si el cliente encuentra un servidor DHCP, éste le proporcionará, para un periodo predeterminado, una configuración IP que le permitirá comunicarse con la red. Cuando haya transcurrido el 50% del periodo, el cliente solicitará una renovación del mismo.

Cuando arrancamos de nuevo un PC cuya configuración IP se determina por DHCP, pueden darse dos situaciones:

- Si la concesión de alquiler de licencia ha caducado, el cliente solicitará una nueva licencia al servidor DHCP (la asignación del servidor podría o no, coincidir con la anterior).
- Si la concesión de alquiler no ha caducado en el momento del inicio, el cliente intentará renovar su concesión en el servidor DHCP, es decir, que le sea asignada la misma dirección IP.

Antes de comenzar con los procesos de instalación y configuración de nuestro servidor DHCP, vamos a definir algunos términos que utilizaremos a lo largo de dicho proceso.

Ámbito servidor DHCP: Un ámbito es un agrupamiento administrativo de equipos o clientes de una subred que utilizan el servicio DHCP.

Rango servidor DHCP: Un rango de DHCP está definido por un grupo de direcciones IP en una subred determinada, como por ejemplo de 192.168.0.1 a 192.168.0.254, que el servidor DHCP puede conceder a los clientes.

Concesión o alquiler de direcciones: es un período de tiempo que los servidores DHCP especifican, durante el cual un equipo cliente puede utilizar una dirección IP asignada.

Reserva de direcciones IP: Consiste en reservar algunas direcciones IP para asignárselas siempre a los mismos PCs clientes de forma que cada uno siempre reciba la misma dirección IP. Se suele utilizar para asignar a servidores o PCs concretos la misma dirección siempre. Es similar a configurar una dirección IP estática pero de forma automática desde el servidor DHCP. En el servidor se asocian direcciones MAC a direcciones IP. Es una opción muy interesante para asignar a ciertos PCs (servidores, impresoras de red, PCs especiales...) siempre la misma IP.

Instalación del servidor DHCP

Para instalar los archivos necesarios de nuestro servidor DHCP podemos utilizar apt-get desde una consola de root:

```
// Instalación del servidor DHCP
     # apt-get install dhcp3-server
```

De esta forma instalaríamos los programas necesarios para disponer de nuestro propio servidor DHCP.

Configuración del servidor DHCP

Tal y como se ha comentado anteriormente, un servidor DHCP proporciona direcciones IP y otros parámetros a los clientes DHCP de forma que su configuración se puede determinar de manera automática sin tener que hacerlo manualmente lo cual es especialmente útil cuando el número de PCs de nuestra red local es grande.

El servidor DHCP deberá saber qué rangos de direcciones IP puede 'alquilar' y qué parámetros adicionales (puerta de enlace, servidores DNS, etc...) debe proporcionar a los clientes para que la configuración de los

mismos sea completa y sea la deseada.

Una configuración TCP/IP mínima debe contener al menos la dirección IP y la máscara de subred, por lo tanto, esos son los dos mínimos datos que un servidor DHCP puede proporcionar a un cliente, no obstante, un servidor DHCP suele proporcionar muchos más parámetros:

- Dirección IP
- Máscara de subred
- Dirección de difusión o broadcast
- Puerta de enlace
- Servidores DNS, etc...

Además, existen una serie de parámetros que definen las condiciones del 'alquiler' o cesión de la configuración IP hacia un cliente como son:

- Tiempo de cesión por defecto
- Tiempo de cesión máximo, y algunos parametros más.

Esta información compone la configuración del servidor DHCP.

Archivo de configuración del servidor DHCP

Al igual que todas las aplicaciones en Linux, el servidor DHCP dispone de su propio archivo de configuración. Se trata del archivo:

```
// Archivo de configuración del servidor DHCP
     /etc/dhcp3/dhcpd.conf
```

Este archivo de configuración consta de una primera parte principal donde se especifican los parámetros generales que definen el 'alquiler' y los parámetros adicionales que se proporcionarán al cliente.

El resto del archivo de configuración consta de una serie de secciones que especifican principalmente rangos de direcciones IPs que serán cedidas a los clientes que lo soliciten (sección subnet) y especificaciones concretas de equipos (sección host). Los parámetros de las secciónes deberán ir entre llaves '{' y '}'.

Los valores de los parámetros especificados al principio del archivo se aplican como valores por defecto al resto de secciones aunque si dentro de una sección se redefine alguno de los parámetros, se aplicará éste ignorándose el valor por defecto.

Los rangos de direcciones IP se especifican en secciones que empiezan con la palabra clave 'subnet' seguido de la dirección de red de la subred, continua con la palabra 'netmask' seguido de la máscara de red. A continuación estará la lista de parámetros para dicha sección encerrados entre llaves.

Ejemplo, supongamos que en nuestra red local disponemos de direciones pertenecientes a la subred 192.168.1.0/24 (/24 significa máscara de subred 255.255.255.0 ó lo que serían 24 'unos' en binario) y deseamos que nuestro servidor DHCP alquile direcciones del rango comprendido entre la dirección 192.168.1.60 y 192.168.1.90. La sección subnet que debemos crear será:

```
// Rango de cesión
      subnet 192.168.1.0 netmask 255.255.255.0 {

        range 192.168.1.60  192.168.1.90;

     }
```

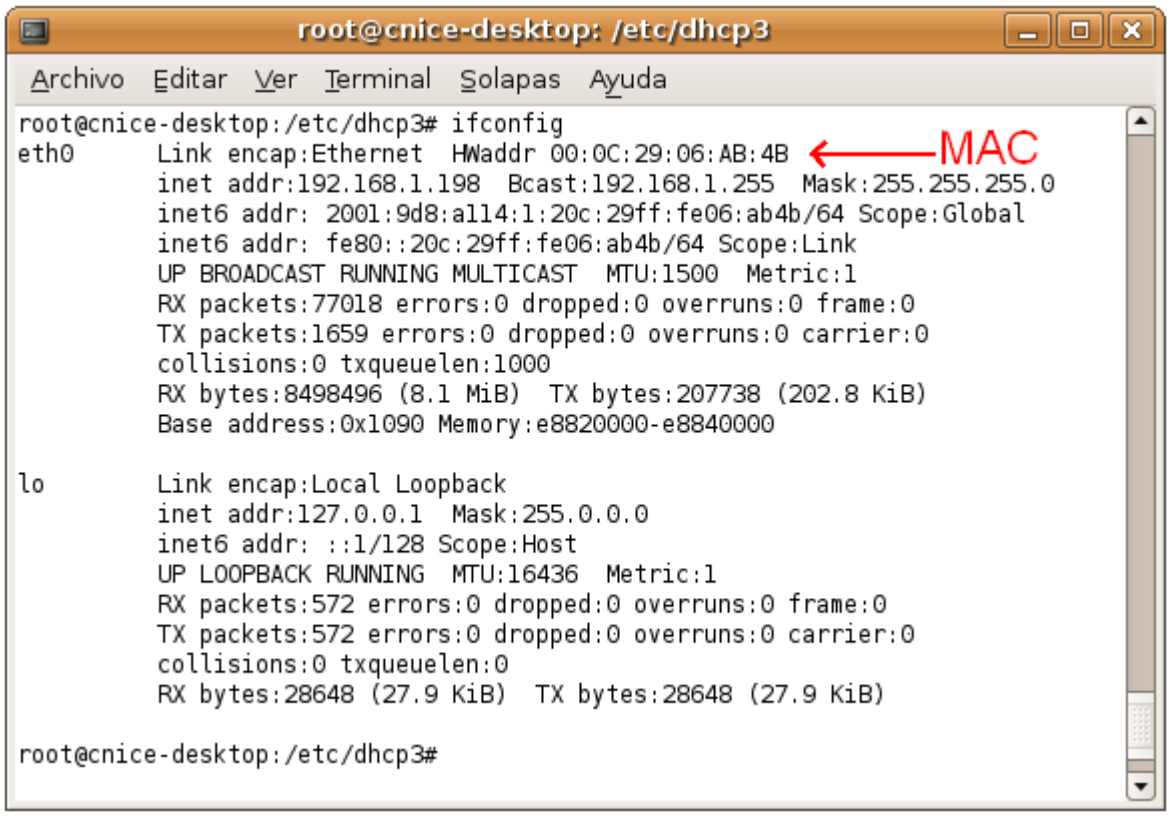
Atencion: El rango de cesión debe pertenecer a la misma subred a la que pertenece la IP del servidor, es necesario para que los clientes puedan comunicarse con el servidor DHCP para procesar las renovaciones.

Ejemplo, si un servidor tiene la IP 192.168.1.1/24, no puede ceder direcciones del rango 10.0.0.0/8 porque dicho rango está fuera del alcance de la subred del servidor.

Si además de proporcionar al cliente la dirección IP y la máscara deseamos que le proporcione también la dirección de la puerta de enlace y las direcciones de dos servidores DNS para que pueda navegar por Internet, la sección subnet que debemos crear será:

```
// Rango de cesión y parámetros adicionales
      subnet 192.168.1.0 netmask 255.255.255.0 {

       option routers 192.168.1.254;

       option domain-name-servers 80.58.0.33, 80.58.32.97;

       range 192.168.1.60  192.168.1.90;

     }
```

Existe la posibilidad de establecer una configuración concreta a un cliente concreto identificándolo por la dirección MAC de su tarjeta de red. Recordemos que la dirección MAC (MAC address) es un número único, formado por 6 octetos, grabado en la memoria ROM de las tarjetas de red ethernet y viene fijado de fábrica. Se suelen escribir los 6 octetos en hexadecimal separados por dos puntos ':'. Todas las tarjetas de red tienen una dirección MAC única en el mundo. Es como un número de serie. Los tres primeros octetos indican el fabricante y los tres siguientes el número de serie en fabricación. En Linux se puede averiguar la dirección MAC mediante el comando ifconfig. En Windows 2000 y XP se puede utilizar el comando ipconfig y en Windows 95 y 98 el comando winipcfg.



Ejecución de ifconfig en Linux. La MAC es la HWaddr

Para establecer una configuración de equipo es necesario crear una sección host. Ejemplo, si deseamos que el cliente cuya dirección MAC sea 00:0c:29:c9:46:80 se configure siempre (reserva de dirección IP) con

la dirección IP 192.168.1.50 y puerta de enlace 192.168.1.254, que su nombre de dominio sea "ieslapaloma.com" y el servidor de nombres netbios sea "192.168.1.250" la sección host que debemos crear será:

```
// Crear una reserva de dirección IP
     host Profesor5 {

             hardware ethernet 00:0c:29:c9:46:80;

             fixed-address 192.168.1.50;

             option routers 192.168.1.254;

             option domain.name "ieslapaloma.com";

             option netbios-name-servers 192.168.1.250;

     }
```

Cuando el PC cuya dirección MAC sea '00:0c:29:c9:46:80' solicite una dirección IP al servidor DHCP, recibirá la 192.168.1.50.

Archivo dhcpd.conf comentado

A continuación mostramos un sencillo archivo dhcpd.conf comentado línea por línea: (Todas las líneas que comienzan por almoadilla (#) son líneas de comentarios y son ignoradas por el servidor dhcp. Todas las líneas que especifican parámetros deben terminar en punto y coma ';')

```
// Ejemplo de archivo dhcp.conf
     # Sample configuration file for ISC dhcpd for Debian

     # $Id: dhcpd.conf,v 1.4.2.2 2002/07/10 03:50:33 peloy Exp $

     # Opciones de cliente y de dhcp aplicables por defecto a todas las
     secciones

     # Estas opciones pueden ser sobreescritas por otras en cada sección
```

```
     option domain-name-servers 195.53.123.57; # DNS para los clientes
     (atenea)

     option domain-name "ieslapaloma.com"; # Nombre de dominio para los
     clientes

     option subnet-mask 255.255.255.0; # Máscara por defecto para los
     clientes

     default-lease-time 600; # Tiempo en segundos del 'alquiler'

     max-lease-time 7200; # Máximo tiempo en segundos que durará el
     'alquiler'
```

```
# Especificación de un rango

subnet 192.168.1.0 netmask 255.255.255.0 {

  range 192.168.1.60  192.168.1.80; # Rango de la 60 a la 80 inclusive

  option broadcast-address 192.168.1.255; # Dirección de difusión

  option routers 192.168.1.254; # Puerta de enlace

  option domain-name-servers 80.58.0.33; # DNS (ej: el de telefónica)

  default-lease-time 6000; # Tiempo

}
```

```
# Configuración particular para un equipo

host aula5pc6 {

        hardware ethernet 00:0c:29:1e:88:1d; # Dirección MAC en
cuestión

        fixed-address 192.168.1.59; # IP a asignar (siempre la misma)

}
```

Nota: Si nuestro servidor tiene varias interfaces de red, será necesario indicar la interfaz o interfaces por las cuales se va a ofrecer el servicio DHCP. Para ello, tendremos que editar el archivo /etc/default/dhcp3-server. Ejemplo, si nuestro servidor disponde de la interfaz eth0 y la interfaz eth1, y queremos ofrecer el servicio por ambas interfaces, tendremos que editar el archivo /etc/default/dhcp3-server:

```
//Ofrecer DHCP por eth0 y eth1
//Editar /etc/default/dhcp3-server y añadir parámetro INTERFACES:
     INTERFACES="eth0 eth1"
```

Para otras opciones de configuración del servidor DHCP, se puede consultar la página del manual de dhcpd.conf:

```
// Página del manual de la configuración del servidor DHCP
     $ man dhcpd.conf
```

Si el servidor DHCP da un error al intentar arrancarlo, casi siempre es porque el rango de cesión está en un rango diferente de la dirección IP del servidor. No obstante, examinando las últimas líneas del archivo log del sistema quizás te dé alguna pista de lo que puede ocurrir. Para ello ejecuta el comando

```
//Ver las últimas 20 líneas del archivo log del sistema
     tail -n 20 /var/log/syslog
```

Arranque y parada manual del servidor DHCP

El servidor DHCP, al igual que todos los servicios en Debian, dispone de un script de arranque y parada en la carpeta /etc/init.d.

```
// Arrancar el servidor DHCP
     root@cnice-desktop:/# /etc/init.d/dhcp3-server start
```

```
// Parar el servidor DHCP
     root@cnice-desktop:/# /etc/init.d/dhcp3-server stop
```

```
// Reiniciar el servidor DHCP
     root@cnice-desktop:/# /etc/init.d/dhcp3-server restart
```

Arranque automático del servidor DHCP al iniciar el sistema.

Para un arranque automático del servicio al iniciar el servidor, debemos crear los enlaces simbólicos correspondientes tal y como se indica en el apartado Arranque automático de servicios al iniciar el sistema.

4.- Servidor DNS

El número de ordenadores en el centro educativo, cada vez es mayor. Aunque hayamos elegido un direccionamiento IP que relacione la asignación de direcciones con la ubicación física de los PCs, sería mucho más cómodo poder referirse a todos los PCs del centro utilizando nombres en lugar de direcciones IPs. Un servidor DNS en la red local, nos permitirá crear una asociación directa Nombre de PC <-> Dirección IP en nuestra red, que nos facilitará la identificación de nuestros equipos.

En las redes TCP/IP, cada PC dispone de una dirección IP para poder comunicarse con el resto de PCs. Es equivalente a las redes de telefonía en las que cada teléfono dispone de un número de teléfono que le identifica y le permite comunicarse con el resto de teléfonos.

Trabajar con direcciones IP es incómodo para las personas, ya que requeriría conocer en todo momento las direcciones IP de los equipos a los que queremos conectarnos. En su lugar utilizamos **nombres** que son más fáciles de recordar y utilizar como por ejemplo **enebro.pntic.mec.es, www.google.es, www.mec.es,** etc...

Cada equipo y cada servidor conectado a Internet, dispone de una dirección IP y de un nombre perteneciente a un dominio. Internamente, la comunicación entre los PCs se realiza utilizando direcciones IP por eso es necesario algún sistema que permita, a partir de los nombres de los PCs, averiguar las direcciones IPs de los mismos. Ejemplo, cuando queremos acceder a la página web del MEC (Ministerio de Educación), en la barra de direcciones del navegador escribimos:

http://www.mec.es

nuestro PC tendrá que averiguar cual es la IP correspondiente a **www.mec.es** y una vez que ha averiguado que su IP es **195.53.123.85**, se conecta con el servidor para adquirir la página web principal y mostrarla al usuario. Si en el navegador escribimos:

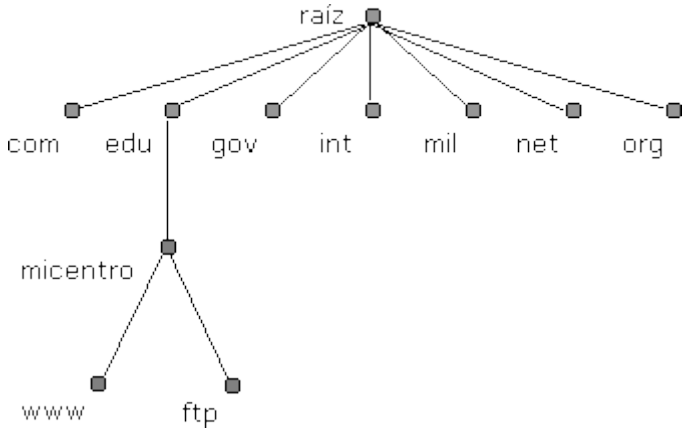
http://193.147.0.29

ahorraremos el paso de averiguar la IP y directamente nos mostrará la página web del MEC.

Un **servidor DNS** es un servidor que permite averiguar la IP de un PC a partir de su nombre. Para ello, el servidor DNS dispone de una base de datos en la cual se almacenan todas las direcciones IP y todos los nombres de los PCs pertenecientes a su dominio.

No existe una base de datos única donde se almacenan todas las IPs existentes en el mundo, sino que cada servidor almacena las IPs correspondientes a su dominio. Los servidores DNS están dispuestos jerárquicamente de forma que cuando nuestro servidor más inmediato no puede atender nuestra petición, éste la traslada al DNS superior.

En el proceso de resolución de un nombre, hay que tener en cuenta que los servidores DNS funcionan frecuentemente como clientes DNS, consultando a otros servidores para resolver completamente un nombre consultado.



En este curso configuraremos un servidor DNS local. Las entradas existentes en nuestro DNS no serán visibles en Internet solamente servirán a los equipos de nuestra red local. De esta forma, cuando un usuario de nuestra red intente acceder a un recurso local, podrá utilizar **nombres** en lugar de direcciones IP. Si el

usuario desea acceder fuera de nuestra red local a algún recurso en Internet, el DNS local nunca podrá llevar a cabo dicha resolución y se la traslada al siguiente servidor DNS (que sí estará en Internet) en su jerarquía de servidores DNS, hasta que la petición sea satisfecha.

Con servidor DNS en nuestra red local, si hacemos un ping a un PC cuyo nombre es "equipo10" y cuya IP es 192.168.0.40; podemos lanzar el comando "ping" indistintamente contra dicha IP o contra el nombre del equipo en el dominio:

- ping 192.168.0.40
- ping equipo10.micentro.edu

en ambos casos obtendremos respuesta. Esto es muy útil cuando las estaciones de trabajo reciben su IP por DHCP ya que puede ocurrir que desconozcamos la IP que tiene cierto equipo pero sí conocer su nombre en el dominio, que será invariable.

Otro ejemplo donde el servidor DNS tomará protagonismo será cuando deseemos acceder a un servidor web instalado en nuestro servidor; si hemos denominado al sitio web como "www", podremos introducir en el DNS una entrada que identifique "www" como 192.168.0.220 (dirección IP de nuestro servidor web), de modo que cuando introduzcamos la URL "www.micentro.edu" accederemos a nuestro servidor web. Lo mismo sería aplicable al servidor ftp o cualquier otro servicio.

Antes de comenzar con los procesos de instalación y configuración de nuestro DNS, vamos a definir algunos términos que utilizaremos a lo largo de dicho proceso.

Zona de Búsqueda Directa: Las resoluciones de esta zona devuelven la dirección IP correspondiente al recurso solicitado. Realiza las resoluciones que esperan como respuesta la dirección IP de un determinado recurso.

Zona de Búsqueda Inversa: Las resoluciones de esta zona buscan un nombre de equipo en función de su dirección IP; una búsqueda inversa tiene forma de pregunta, del estilo "¿Cuál es el nombre DNS del equipo que utiliza la dirección IP 192.168.0.20?".

Reenviador DNS: Servidor DNS designado por otros servidores DNS internos para su uso en consultas para resolver nombres de dominio DNS externos o fuera del dominio local.

Linux dispone de varios paquetes que software que permiten poner en marcha un servidor DNS. En este capítulo hablaremos de dos de ellos: el paquete **dnsmasq** que es un sencillo servidor DNS ideal para redes pequeñas como las que podemos encontrar en los centros educativos y el paquete **bind** que es un completo servidor DNS utilizado por muchos servidores DNS en Internet.

Servidor DNS sencillo con dnsmasq

El paquete dnsmasq permite poner en marcha un servidor DNS de una forma muy sencilla. Simplemente instalando y arrancando el servicio dnsmasq, sin realizar ningún tipo de configuración adicional, nuestro PC se convertirá en un servidor caché DNS y además, resolverá los nombres que tengamos configurados en el archivo /etc/hosts de nuestro servidor. La resolución funcionará tanto en sentido directo como en sentido inverso, es decir, resolverá la IP dado un nombre de PC y el nombre del PC dada la IP. Adicionalmente, dnsmasq dispone de servidor DHCP y permite resolver los nombres de los PCs a los que les ha asignado dirección IP dinámica. A lo largo de esta sección veremos todas estas posibilidades que nos ofrece dnsmasq.

Instalación del servidor dnsmasq

Para instalar la última versión de dnsmasq, podemos hacerlo con apt-get desde una consola de root:

```
// Instalación del servidor dnsmasq
     # apt-get install dnsmasq
```

De esta forma instalaríamos los programas necesarios para disponer de un sencillo servidor DNS. Tan solo será necesario configurarlo y ponerlo en marcha.

Arranque y parada del servidor dnsmasq

El servicio dnsmasq, al igual que todos los servicios, dispone de scripts de arranque y parada en la carpeta /etc/init.d. Debemos ejecutarlos desde una consola de root.

```
// Arrancar o reiniciar el servidor dnsmasq
     # /etc/init.d/dnsmasq restart



// Parar el servidor dnsmasq
     # /etc/init.d/dnsmasq stop
```

Para un arranque automático del servicio al iniciar el servidor, debemos crear los enlaces simbólicos correspondientes tal y como se indica en el apartado Arranque automático de servicios al iniciar el sistema.

Configuración básica de dnsmasq

Para que dnsmasq pueda ser un servidor caché DNS, es necesario que nuestro servidor tenga en el archivo de /etc/resolv.conf configurado al menos un servidor DNS externo. Normalmente los servidores DNS externos nos los proporciona el operador de telecomunicaciones que nos da servicio de Internet. Por ejemplo, Telefónica tiene unos DNSs, Orange tiene otros, ONO tiene otros, Tele2 otros, etc... Aunque podemos utilizar los de cualquier operador, lo mejor es configurar los del nuestro, porque responderá más rápido.

Servidores DNS de Telefónica:
- DNS primario 80.58.0.33
- DNS alternativo (por si falla el primario) 80.58.32.97

Servidores DNS de Orange:
- DNS primario 62.36.225.150
- DNS alternativo 62.37.228.20

Para que nuestro servidor utilice los DNS externos, debemos añadirlos en /etc/resolv.conf. En el caso de Telefónica, deberemos añadir en /etc/resolv.conf las siguientes líneas:

```
// Ejemplo: Utilización de los DNS externos de Telefónica
// Añadir en /etc/resolv.conf del servidor
     nameserver 80.58.0.33

     nameserver 80.58.32.97
```

Una vez introducidos los DNS externos en /etc/resolv.conf, debemos comprobar si dichos DNS externos funcionan correctamente y responden a las peticiones. Para ello haremos una consulta al DNS mediante el comando nslookup. También podríamos utilizar el comando host o el comando dig:

```
// Probar DNS externo
// Ejecutar en una consola del servidor
     $ nslookup www.unican.es
```

Si el DNS funciona, nos dirá cual es la IP del servidor de la Universidad de Cantabria, www.unican.es.

En este punto, ya tendremos en nuestro servidor un **servidor DNS caché** funcionando. Para probar su funcionamiento, configuraremos el archivo /etc/resolv.conf del resto de los PCs de nuestra red pero en lugar de indicar los DNS de Telefónica, indicaremos el nuestro. Si nuestro servidor tiene la IP 192.168.1.239, lo añadiremos en el archivo /etc/resolv.conf de cada PC

```
// Añadir en /etc/resolv.conf del PC cliente
     nameserver 192.168.1.239
```

Al igual que hemos hecho anteriormente, podemos comprobar si nuestro servidor DNS funciona correctamente, haciendo una consulta mediante el comando nslookup:

```
// Probar nuestro servidor DNS
//Ir al PC cliente, abrir una consola de comandos y ejecutar:
     $ nslookup www.unican.es
```

Si nuestro servidor DNS funciona, nos responderá con la IP del servidor de la Universidad de Cantabria.

Ahora que ya tenemos el servidor DNS caché funcionando, iremos más allá. El siguiente paso será editar el archivo /etc/hosts de nuestro servidor, para que nuestro DNS resuelva también los nombres y las IPs de nuestra red. Si los PCs de nuestra red disponen de IP fija y queremos que dnsmasq resuelva sus nombres e IPs, tan solo tenemos que añadir los nombres y las IPs en el archivo hosts del servidor y sería como disponer de un **DNS maestro** para nuestra red:

```
//Añadir en /etc/hosts del servidor las IPs y los nombres de nuestros PCs
//Se pueden añadir varios nombres en la misma línea. Separar con un tabulador
     192.168.1.239   www.ieslapaloma.com     proxy   www

     192.168.1.238   impresora

     192.168.1.1             router

     192.168.1.101   a1pc1   aula1pc1

     192.168.1.102   a1pc2   aula1pc2

     192.168.1.103   a1pc3   aula1pc3

     192.168.1.104   a1pc4   aula1pc4

     192.168.1.105   a1pc5   aula1pc5

     192.168.1.106   a1pc6   aula1pc6

     192.168.1.107   a1pc7   aula1pc7

     192.168.1.108   a1pc8   aula1pc8

     192.168.1.109   a1pc9   aula1pc9

     192.168.1.110   a1pc10  aula1pc10
```

Si desde un PC de nuestra red hacemos una consulta al DNS preguntando por otro PC de nuestra red, dnsmasq resolverá en el servidor y devolverá la IP configurada en el archivo hosts del servidor:

```
// Probar nuestro servidor DNS con nombres de nuestra red
// Ejecutar en una consola del PC cliente
     $ nslookup aula1pc1
```

Cada vez que modifiquemos el archivo /etc/hosts del servidor, deberemos ejecutar **"/etc/init.d/dnsmasq restart"** para reiniciar el servicio dnsmasq y recargue la información contenida en dicho archivo.

De esta manera, tan solo editando el archivo /etc/hosts del servidor, dispondremos de un sencillo servidor DNS para nuestra red lo que nos permitirá referirnos a nuestros PCs utilizando sus nombres que son mucho más fáciles de recordar que las direcciones IP.

Servidor DNS y servidor DHCP

Cuando las IPs de los PCs de nuestra red son dinámicas, se nos presenta un problema para utilizar un servidor DNS ya que el mismo PC, hoy puede tener una IP y mañana puede tener otra IP diferente. Dicho problema se puede resolver de tres formas:

Utilizando un servidor DNS dinámico: Los PCs, al recibir la IP del servidor DHCP, informarán al servidor DNS dinámico de la IP que les ha sido asignada de forma dinámica y así poder asociar de forma correcta el nombre con la IP que tiene en un momento dado. El inconveniente de este método es que nos obliga a instalar en los PCs un servicio que informe al servidor DNS dinámico de los cambios de IP de cada PC. Es similar al sistema utilizado por los servidores DNS dinámicos de Internet como www.no-ip.org o www.dyndns.com. Aquí no hablaremos de servidores DNS dinámicos porque las dos soluciones siguientes son más sencillas.

Utilizando reservas de DHCP: En el servidor DHCP existe la posibilidad de establecer una configuración concreta a un cliente concreto identificándolo por la dirección MAC de su tarjeta de red. Si configuramos tantas reservas de IPs como PCs hay en nuestra red, podríamos configurar a cada PC la IP que deseemos. Esto sería como tener IPs fijas en nuestra red, pero asignadas por DHCP. Esta idea no es para nada descabellada y nos permitiría controlar en todo momento la IP de cada PC.

Utilizando el servidor DHCP de dnsmasq: Dnsmasq, además de ofrecernos un servidor DNS, nos ofrece también un servidor DHCP fácilmente configurable que además resolverá los nombres de los PCs de nuestra red aún cuando sus IPs hayan sido configuradas por DHCP. Para configurar el servidor DHCP de dnsmasq debemos editar el archivo de configuración **/etc/dnsmasq.conf** y añadir una línea como esta: **dhcp-range=ip-inicial,ip-final, tiempo de cesión**. Ejemplo, si queremos que el DHCP utilice el rango desde 192.168.1.201 hasta 192.168.1.230 y que la cesión dure 24 horas, editaremos /etc/dnsmasq.conf y añadiremos la siguiente línea:

```
//Editar /etc/dnsmasq.conf para establecer el rango DHCP
//Añadir la siguiente línea:
     dhcp-range=192.168.1.201,192.168.1.230,24h
```

Cuando los PCs clientes pidan una IP al servidor DHCP, normalmente el cliente suministrará su nombre de PC. Dicho nombre será utilizado por dnsmasq para asociarlo a la IP que le ha sido asignada al PC y así resolver correctamente cualquier consulta DNS.

A medida que el servidor DHCP va concediendo IPs a todos los PCs que se la solicitan, éste va almacenándolas en el archivo de concesiones **/var/lib/misc/dnsmasq.leases** donde guarda la fecha y la hora de la cesión en formato %s (ver man date para información sobre dicho formato) la MAC del cliente, la IP concedida al cliente y el nombre del PC cliente siempre y cuando el cliente haya enviado su nombre de PC.

```
//Archivo donde aparecen las IPs asignadas a cada PC
     /var/lib/misc/dnsmasq.leases
```

Para que dnsmasq pueda conocer el nombre del cliente, éste deberá enviar su nombre cuando realiza la

petición DHCP. En los clientes Linux, el nombre que envía el PC cliente, suele almacenarse en el parámetro send host-name del archivo de configuración del cliente dhcp: /etc/dhcp3/dhclient.conf. Ejemplo, si nuestro PC se llama aula1pc1, deberemos configurarlo en el cliente dhcp:

```
//Configurar en /etc/dhcp3/dhclient.conf el nombre que envía el cliente al
servidor DHCP:
     send host-name aula1pc1
```

Lo normal es que dicho nombre coincida con el nombre del PC almacenado en el archivo /etc/hostname.

Como dnsmasq dispone de servidor DNS y servidor DHCP, no es necesario instalar otro servidor DCHP ni otro servidor DNS, por tanto, podríamos desinstalar el paquete dhcp3-server y el paquete bind.

Instalación del servidor DNS bind

Si con las posibilidades que nos ofrece dnsmasq no son suficientes para nuestra red y necesitamos un servidor DNS más completo, podemos utilizar el paquete **bind9**. Para instalarle, podemos hacerlo con apt-get desde una consola de root:

```
// Instalación del servidor DNS bind
     # apt-get install bind9
```

De esta forma instalaríamos los programas necesarios para disponer de un completo servidor DNS con bind. Tan solo será necesario configurarlo y ponerlo en marcha.

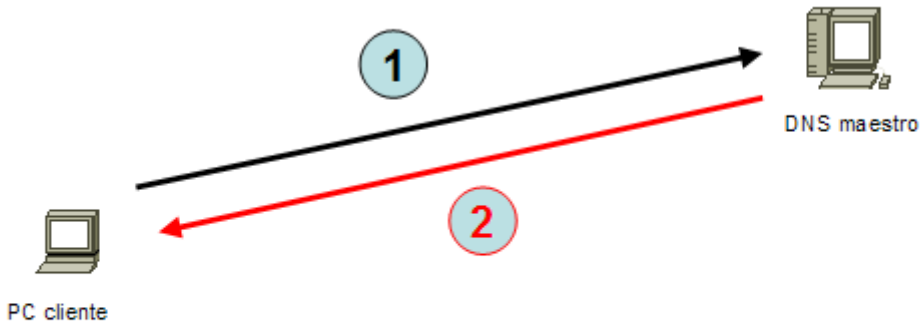
Configuración del servidor DNS

El servidor DNS bind admite tres modos de funcionamiento

- Servidor DNS maestro
- Servidor DNS esclavo
- Servidor caché DNS

Servidor DNS maestro

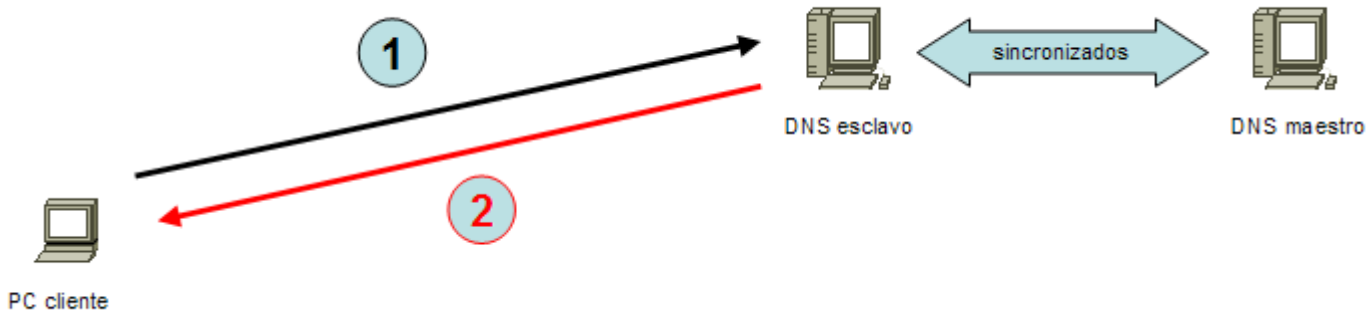
En este modo de funcionamiento, nuestro servidor se comporta como un auténtico servidor DNS para nuestra red local. Atenderá directamente a las peticiones de resolución de direcciones pertenecientes a la red local y reenviará a servidores DNS externos las peticiones del resto de direcciones de Internet.



Consulta a un DNS maestro

Servidor DNS esclavo

Un servidor esclavo actuará como un servidor espejo de un servidor DNS maestro. Permanecerá sincronizado con el maestro. Se utilizan para repartir las peticiones entre varios servidores aunque las modificaciones solo se realicen en el maestro. En redes locales salvo por razones de disponibilidad, es raro que exista la necesidad de tener dos servidores DNS ya que con uno será suficiente.



Consulta a un DNS esclavo

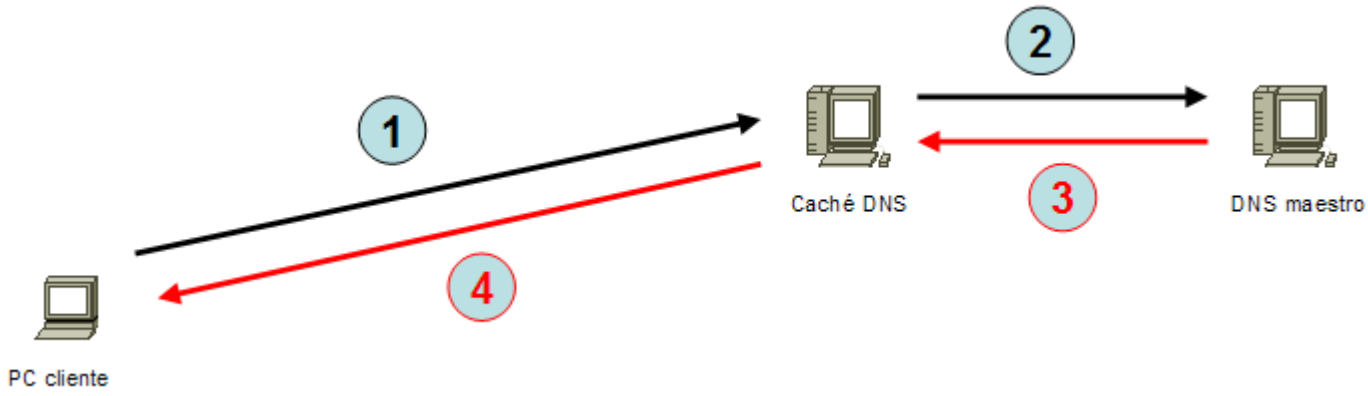
Servidor caché DNS

En este modo de funcionamiento, nuestro servidor se comporta como si fuera un auténtico servidor DNS para nuestra red local aunque realmente no sea un servidor DNS propiamente dicho. Cuando recibe una petición de DNS por parte de un cliente de nuestra red, la trasladará a un DNS maestro que puede estar en nuestra red o fuera, almacenará en una memoria caché la respuesta y a la vez la comunicará a quien hizo la petición. Si un segundo cliente vuelve a realizar la misma petición, como nuestro servidor tiene la respuesta almacenada en su memoria caché, responderá inmediatamente sin tener que cursar la petición a ningún servidor DNS de Internet.

Disponer de un servidor caché DNS en nuestra red local aumenta la velocidad de la conexión a Internet pues cuando navegamos por diferentes lugares, continuamente se están realizando peticiones DNS. Si nuestro caché DNS almacena la gran mayoría de peticiones que se realizan desde la red local, las respuestas de los clientes se satisfarán prácticamente de forma instantánea proporcionando al usuario una sensación de velocidad en la conexión.

Es un modo de funcionamiento de sencilla configuración ya que prácticamente lo único que hay que configurar son las direcciones IP de un DNS primario y de un DNS secundario. Muchos routers ADSL ofrecen ya este servicio de caché, tan solo hay que activarlo y configurar una o dos IPs de servidores DNS

en Internet. En los PCs de nuestra red local podríamos poner como DNS primario la IP de nuestro router y como DNS secundario una IP de un DNS de Internet.



Consulta a un cache DNS. En caso de fallo, se redirecciona hacia un DNS maestro

Archivos de configuracion del DNS

El archivo de configuración del DNS es el archivo /etc/bind/named.conf, pero este hace referencia a otros cuantos archivos como por ejemplo:

Archivo	Descripción
named.conf	*Archivo principal de configuración*
named.conf.options	*Opciones genéricas*
named.conf.local	*Especificación particular de este servidor DNS*
db.127	*Especificación dirección de retorno*
db.root	*DNSs de nivel superior*
otros	*db.0, db.255, db.empty, db.local, rndc.conf, rndc.key, zones.rfc1918*

Configuración como caché DNS

Por defecto, al instalar el paquete bind está preconfigurado como servidor caché DNS. Tan solo será necesario editar el archivo /etc/bind/named.conf.options y en la sección forwarders añadir las IPs de dos servidores DNS donde redirigir las peticiones DNS:

```
// Configuración como caché DNS
// Añadir IPs de los DNS de nuestro proveedor en /etc/bind/named.conf.options
     options {

               forwarders {

               80.58.0.33; 80.58.32.97;

               };

     };
```

Configuración DNS maestro

Por razones de accesibilidad y organizativas, deseamos asignar un nombre a todos los equipos de nuestra

red, para lo que instalaremos un servidor DNS privado con un **dominio ficticio**, por ejemplo 'ieslapaloma.com'. Todos los PCs de nuestra red pertenecerán a dicho dominio ficticio que funcionará solo en nuestra red interna, no en Internet. En tal caso el nombre completo de los PCs terminará con 'ieslapaloma.com', por ejemplo: aula5pc2.ieslapaloma.com. Lo ideal en una situación así es disponer de un servidor DNS que sea maestro de nuestro dominio, es decir, maestro del dominio interno 'ieslapaloma.com'.

Nuestro servidor DNS maestro para nuestro dominio ficticio interno 'ieslapaloma.com' será capaz de resolver peticiones internas de nombres de este dominio, tanto de forma directa como de forma inversa, es decir, si recibe una consulta acerca de quién es aula5pc7.ieslapaloma.com deberá devolver su IP, pongamos por ejemplo 192.168.0.107. Si la consulta es una consulta DNS inversa acerca de quién es 192.168.0.107, deberá responder aula5pc7.ieslapaloma.com. Por ello deberemos añadir en el archivo /etc/bind/named.conf.local la especificación de maestro para el dominio y para la resolución inversa, por ejemplo:

```
// Añadir en /etc/bind/named.conf.local
     // Archivo para búsquedas directas

     zone "ieslapaloma.com" {

             type master;

             file "/etc/bind/ieslapaloma.db";

     };



     // Archivo para búsquedas inversas

     zone "0.168.192.in-addr.arpa" {

             type master;

             file "/etc/bind/192.rev";

     };
```

Evidentemente será necesario crear los archivos ieslapaloma.db y 192.rev que especificarán la asociación entre nombres y direcciones IP de nuestra red en un sentido y en otro respectivamente.

Archivo de zona de búsqueda directa

Supongamos que en nuestra red local tenemos un aula llamada aula5 con 12 PCs con IPs que van desde la 192.168.0.101 hasta 112 y cuyos nombres van desde aula5pc1 hasta aula5pc10, luego un servidor web (pc11) y un servidor de correo electrónico que además es servidor DNS (pc12). El archivo de configuración DNS de nuestro dominio podría ser así:

```
// Archivo /etc/bind/ieslapaloma.db
     ;

     ; BIND data file for ieslapaloma.com

     ;

     @       IN      SOA     ieslapaloma.com. root.ieslapaloma.com. (
```

```
                              1         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Default TTL
                IN      NS              dns.ieslapaloma.com.
                IN      MX      10      mail.ieslapaloma.com.
aula5pc1        IN      A       192.168.0.101
aula5pc2        IN      A       192.168.0.102
aula5pc3        IN      A       192.168.0.103
aula5pc4        IN      A       192.168.0.104
aula5pc5        IN      A       192.168.0.105
aula5pc6        IN      A       192.168.0.106
aula5pc7        IN      A       192.168.0.107
aula5pc8        IN      A       192.168.0.108
aula5pc9        IN      A       192.168.0.109
aula5pc10       IN      A       192.168.0.110
www             IN      A       192.168.0.111
dns             IN      A       192.168.0.112
mail            IN      A       192.168.0.112
```

Las primeras líneas son unos parámetros relacionados con la actualización del DNS (número de serie y periodos de actuación). Las dos siguientes líneas indican quién es el servidor primario (NS = Name Server) y quien procesa el correo electrónico del dominio (MX = Mail eXchange). Las siguentes líneas especifican las IPs de los distintos PCs componentes del dominio (A = Address).

Si olvidamos algún punto y coma, dará errores y no funcionará correctamente. Para revisar los archivos disponemos de los comandos named-checkconf y named-checkzone que analizan que esté correcta la sintaxis de los mismos.

Archivo de zona de búsqueda inversa

Para poder realizar consultas inversas (de IP a nombre) será necesario crear el siguiente archivo:

```
// Archivo /etc/bind/192.rev
```

```
;
; BIND reverse data file for 192.168.0.0
;
@       IN      SOA     ieslapaloma.com. root.ieslapaloma.com. (
                              1         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Default TTL
        IN      NS      dns.ieslapaloma.com.
101             IN              PTR          aula5pc1.ieslapaloma.com.
102             IN              PTR          aula5pc2.ieslapaloma.com.
103             IN              PTR          aula5pc3.ieslapaloma.com.
104             IN              PTR          aula5pc4.ieslapaloma.com.
105             IN              PTR          aula5pc5.ieslapaloma.com.
106             IN              PTR          aula5pc6.ieslapaloma.com.
107             IN              PTR          aula5pc7.ieslapaloma.com.
108             IN              PTR          aula5pc8.ieslapaloma.com.
109             IN              PTR          aula5pc9.ieslapaloma.com.
110             IN              PTR         aula5pc10.ieslapaloma.com.
111             IN              PTR             www.ieslapaloma.com.
112             IN              PTR             dns.ieslapaloma.com.
112             IN              PTR             mail.ieslapaloma.com.
```

Una vez configurado nuestro servidor DNS, debemos indicar a nuestro servidor Linux que el servidor DNS es él mismo, lo cual se especifica en el archivo /etc/resolv.conf.

```
// Indicamos que nosotros mismos somos servidores DNS
//  y por defecto buscamos en nuestro dominio
// Editar /etc/resolv.conf del servidor DNS
      nameserver 127.0.0.1
```

```
search ieslapaloma.com
```

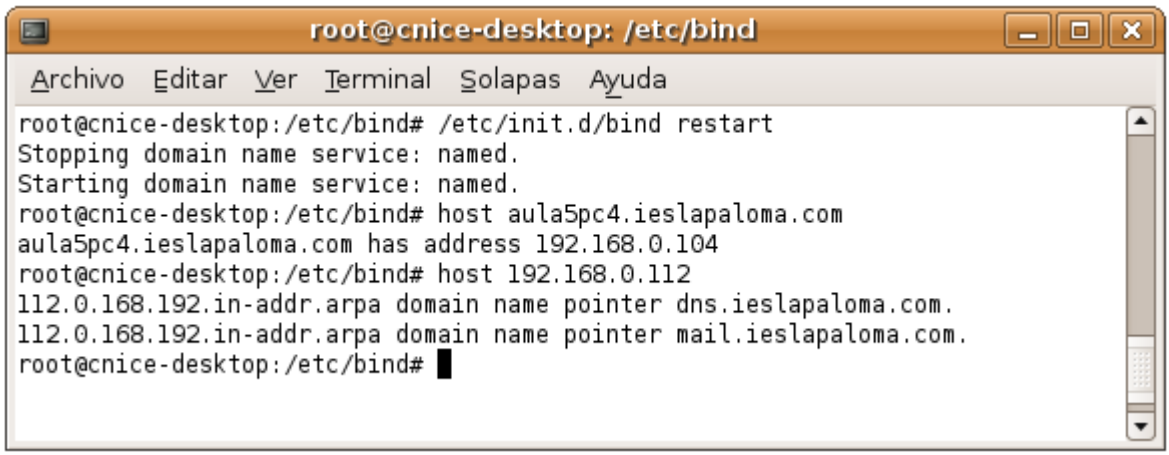
En el resto de PCs de la red, indicaremos que el servidor DNS es 192.168.0.112

```
// En el resto de PCs de la red indicamos quién es el DNS
// Editar /etc/resolv.conf del resto de PCs de la red
      nameserver 192.168.0.112
```

Tan solo nos faltará poner en marcha nuestro servidor de nombres ejecutando en el servidor el script de inicio correspondiente:

```
// Arranque del servidor DNS
     # /etc/init.d/bind9 restart
```

y, mediante el comando **host**, el comando **dig** o el comando **nslookup** hacer alguna consulta de prueba:



DNS funcionando correctamente

Configuración DNS esclavo

Si deseamos configurar nuestro servidor DNS para que actue como esclavo de un servidor DNS maestro, la configuración es mucho más sencilla que en el caso anterior ya que únicamente será necesario indicar en el DNS esclavo quién es el servidor DNS maestro, y en el DNS maestro, la IP del DNS esclavo.

Ejemplo, supongamos que el nombre del DNS maestro es dns.ieslapaloma.com (IP 192.168.0.112) y que el nombre del DNS esclavo es dns2.ieslapaloma.com. En el archivo 'ieslapaloma.db' de zona de búsqueda directa añadiremos la línea del segundo dns justo debajo de donde está la del primero:

```
// Añadir línea en /etc/bind/ieslapaloma.db del maestro

     ....

             IN      NS              dns.ieslapaloma.com.

             IN      NS              dns2.ieslapaloma.com. // Nueva línea

     ....
```

de esta forma indicaremos que existen más servidores DNS para dicha zona. Lo mismo haremos en el archivo '192.rev' de la zona inversa:

```
// Añadir línea en /etc/bind/192.rev del maestro

     ....

             IN      NS              dns.ieslapaloma.com.

             IN      NS              dns2.ieslapaloma.com. // Nueva línea

     ....
```

En el archivo /etc/bind/named.conf.local del servidor DNS esclavo debemos indicar que se trata de un servidor esclavo y también debemos indicar quién es el maestro:

```
// Añadir en /etc/bind/named.conf.local del esclavo
     zone "ieslapaloma.com" {

             type slave;

             file "/etc/bind/ieslapaloma.db";

             masters { 192.168.0.112; };

     };

     zone "0.168.192.in-addr.arpa" {

             type slave;

             file "/etc/bind/192.rev";

             masters { 192.168.0.112; };

     };
```

En el archivo /etc/bind/named.conf.local del servidor DNS maestro podemos utilizar also-notify para mantener los DNS sincronizados. Con also-notify pasamos los cambios de zonas en el maestro al esclavo:

```
// Archivo /etc/bind/named.conf.local del maestro
     zone "ieslapaloma.com" {

             type master;

             file "/etc/bind/ieslapaloma.db";

             also-notify {ip_del_esclavo;}

     };

     zone "0.168.192.in-addr.arpa" {

             type master;
```

```
            file "/etc/bind/192.rev";

            also-notify {ip_del_esclavo;}

    };
```

De ésta forma dispondremos en la red de un servidor DNS esclavo que podrá satisfacer las peticiones DNS al igual que lo haría el maestro. Es interesante si el número de peticiones es muy elevado y se requiere distribuir la carga entre los dos servidores, o si deseamos disponer de servicio DNS de alta disponibilidad de forma que aunque el servidor maestro deje de funcionar, el servidor esclavo podrá seguir ofreciendo el servicio.

Cada vez que hagamos un cambio en los archivos /etc/bind/ieslapaloma.db y /etc/bind/192.rev del maestro, debemos acordarnos de actualizar el parámetro serial (incrementar en una unidad) para que los dns dependientes del maestro sepan que ha cambiado y actualicen su información para mantenerse perfectamente sincronizados.

Arranque y parada manual del servidor DNS

El servidor DNS, al igual que todos los servicios en Debian, dispone de un script de arranque y parada en la carpeta /etc/init.d.

```
// Arranque del servidor DNS
     # /etc/init.d/bind9 start
```

```
// Parada del servidor DNS
     # /etc/init.d/bind9 stop
```

```
// Reinicio del servidor DNS
     # /etc/init.d/bind9 restart
```

Arranque automático del servidor DNS al iniciar el sistema.

Para un arranque automático del servicio al iniciar el servidor, debemos crear los enlaces simbólicos correspondientes tal y como se indica en el apartado Arranque automático de servicios al iniciar el sistema.

5.- Entidad Certificadora

Una entidad certificadora (en inglés CA Certification Authority) es alguien que puede firmar certificados de usuarios y garantizar su autenticidad. Por ejemplo en España, una entidad certificadora es la FNMT - Fábrica Nacional de Moneda y timbre http://www.cert.fnmt.es

Los certificados permiten identificar y autenticar a sus titulares (usuarios, equipos, servidores,...), siempre y cuando estén firmados por una CA de confianza. Ejemplo, el usuario Pepe puede tener un certificado firmado por la FNMT que le sirve para autentificarse en la Agencia Tributaria. La Agencia Tributaria le permitirá el acceso ya que confía en los certificados firmados por la FNMT.

Si confiamos en una CA, debemos aceptar (instalar) su certificado raíz y de ésta forma confiaremos en todos los certificados firmados por dicha CA. Un certificado raíz es un certificado autofirmado por una CA.

Cuando accedemos a una página web segura mediante el protocolo https, el servidor deberá demostrar su autenticidad mediante un certificado firmado por una CA de nuestra confianza. Si la CA no es de nuestra confianza, el navegador preguntará al usuario si desea continuar o por el contrario, cancela la comunicación.

La comunicación se realiza de forma segura ya que se utilizan algoritmos de cifrado asimétrico. Para saber más del cifrado asimétrico, consultar el apartado Autentificacion segura con OpenLDAP

Nuestro servidor Linux puede comportarse como una CA y ofrecer certificados a un solicitante. Crearemos nuestra propia CA para poder utilizar páginas web seguras en nuestro servidor web Apache y para otros

servicios como LDAP, mediante el protocolo SSL. Nuestra CA no será válida en Internet y sólo tendrá vigencia en el ámbito de nuestro dominio (ejemplo: 'ieslapaloma.com') pero obviamente es suficiente para el fin que pretendemos.

Instalación y configuración de OpenSSL

A nuestro servidor no acceden solamente los alumnos sino que también lo hacen los profesores y dentro de ellos los miembros del equipo directivo, accediendo a documentos privados y confidenciales. Nosotros como administradores debemos garantizar que esa información siga siendo privada, para lo cual vamos a definir en el servidor carpetas seguras que mediante el protocolo SSL proporcionen el cifrado de los datos que se intercambian entre el ordenador servidor y el cliente, como hacen en los bancos y cajas de ahorro para garantizar el acceso a nuestras cuentas.

Para ello debemos disponer de un certificado de seguridad que puede ser expedido por una entidad certificadora con el consiguiente coste económico o bien crear y utilizar nuestra propia entidad certificadora, que expedirá certificados válidos en su ámbito de actuación; el dominio de nuestro centro, ámbito suficiente para lograr la seguridad en nuestra Intranet.

Instalación de OpenSSL

Utilizaremos apt-get para instalar el software que necesitamos para crear una entidad certificadora. Deberemos instalar el paquete openssl:

```
// Instalación de OpenSSL
     # apt-get install openssl
```

Configuración de OpenSSL

El archivo de configuración de openssl es **/etc/ssl/openssl.cnf**. En dicho archivo únicamente vamos a configurar los valores por defecto de nuestra organización para que el resto de aplicaciones y programas que usen openssl tomen dichos valores por defecto de forma automática. Dichos valores debemos configurarlos en la sección [req_distinguished_name]. En el resto de secciones no es necesario que modifiquemos nada ya que nos sirve con las opciones configuradas por defecto.

```
// Configuración particular de nuestra CA. Archivo /etc/ssl/openssl.cnf


     [ req_distinguished_name ]

     countryName = Country Name (2 letter code)

     countryName_default = ES

     countryName_min = 2

     countryName_max = 2

     stateOrProvinceName = State or Province Name (full name)

     stateOrProvinceName_default = Soria

     localityName = Soria

     0.organizationName = Organization Name (eg, company)
```

```
0.organizationName_default = I.E.S. La Paloma

# we can do this but it is not needed normally #1.organizationName =
Second Organization Name (eg, company)

#1.organizationName_default = World Wide Web Pty Ltd

organizationalUnitName = I.E.S. La Paloma

#organizationalUnitName_default =

commonName = www.ieslapaloma.com

commonName_max = 64

emailAddress = root@ieslapaloma.com

emailAddress_max = 64
```

En las siguientes secciones se utiliza openssl para permitir páginas web seguras y autentificación segura:

- Acceso_a_carpetas_seguras
- Autentificacion_segura_OpenSSL_y_OpenLDAP

6.- Servidor Web Apache

El servidor web apache es una de las aplicaciones estrella del mundo Linux. Es el servidor web más implantado entre los distintos servidores que ofertan servicios web en Internet.

Entre las características más significativas destacamos:

- Es modular
- Permite crear servidores virtuales
- Permite crear servidores seguros https
- Permite crear sitios privados
- Permite crear sitios de usuario

En este curso haremos uso de éstas y otras características de apache.

Organización del sitio web

La organización que realizaremos de nuestro servidor Apache, será la clásica en los sistemas Unix: la **página web de la intranet** se almacenará en la carpeta raíz del servidor web, las **páginas de los usuarios** se almacenarán en la carpeta home de cada usuario y para albergar las **páginas web de los distintos departamentos** didácticos del centro, lo más práctico es crear nuevos usuarios con el nombre del departamento.

Espacio web para la Intranet

Por defecto, la carpeta raiz del servidor web es la carpeta /var/www. Todos los documentos que se encuentren dentro de la carpeta raíz del servidor web, serán accesibles vía web. Dentro del raiz de documentos crearemos la **página web de nuestra intranet.**

Carpeta principal del servidor web (DocumentRoot)

- **Carpeta raíz del servidor web:** /var/www
- **Acceso a la web principal:** http://ip-del-servidor ó http://nombre-del-servidor

Para acceder vía web a la página almacenada en la carpeta raíz del servidor, desde un navegador debemos acceder directamente con la dirección IP a: http://ip-del-servidor o bien utilizando el nombre del mismo si tenemos el DNS funcionando: http://nombre-del-servidor. Si no tenemos el DNS funcionando, podemos añadir el nombre y la IP en /etc/hosts para resolver localmente.

Espacio web para cada usuario

Cada usuario del sistema dispondrá de un espacio web que se almacena dentro de su carpeta home en una carpeta llamada 'public_html'. Si dicha carpeta no existe, el propio usuario puede crearla y copiar dentro de ella su página web. Los permisos recomendados son 644 para que el 'grupo' y el 'resto' de usuarios tengan acceso de lectura y así se puedan visualizar las páginas.

Para acceder vía web a la página de un usuario, desde un navegador debemos acceder directamente con la dirección IP a: http://ip-del-servidor/~login-usuario/

El caracter '~'comúnmente conocido como gusanillo y que se obtiene con Alt Gr + 4 sirve para indicar a apache que debe servir la página desde el home del usuario (en Linux el 'gusanillo' equivale a la carpeta home). Ejemplo, si hemos creado un usuario javier y éste ha creado la carpeta /home/javier/public_html y ha copiado en ella su página web, desde cualquier PC de la red podremos acceder a dicha carpeta yendo a la dirección http://ip-del-servidor/~javier/. Para que la página aparezca automáticamente, es necesario crear un archivo llamado index.html.

Carpetas web de los usuarios

- **Carpeta web de javier:** /home/javier/public_html
- **Acceso a la web de javier:** http://ip-del-servidor/~javier/

Espacio web para los departamentos

Para proporcionar espacio web a los departamentos, lo más sencillo es crear un usuario para cada departamento. Podemos crear los usuarios: matematicas, lengua, ingles, plastica (sin acentos), etc... Al igual que cada usuario del sistema, dispondrán de un espacio web dentro de su carpeta home en una carpeta llamada 'public_html'. Si dicha carpeta no existe, habrá que crearla y copiar dentro de ella la página web del departamento.

Para acceder vía web a la página del departamento, desde un navegador debemos acceder directamente con la dirección IP a: http://ip-del-servidor/~departamento. Ejemplo, si hemos creado un usuario matematicas y hemos creado la carpeta /home/matematicas/public_html y copiado en ella la web del departamento de matemáticas, desde cualquier PC de la red podremos acceder a dicha web yendo a la dirección http://ip-del-servidor/~matematicas. Para que la página aparezca automáticamente, es necesario crear un archivo llamado index.html.

Carpetas web de los departamentos

- **Carpeta web del dpto. de matemáticas:** /home/matematicas/public_html
- **Acceso a la web de dpto. de matemáticas:** http://ip-del-servidor/~matematicas/

De la misma manera, se pueden crear usuarios para proporcionar espacio web a otros órganos del centro, p.ej: ccp, orientacion, equipodirectivo, conserjeria, etc... para que dispongan de su propio espacio web.

Espacio web seguro

Además crearemos un sitio web virtual seguro en el servidor web Apache para poder tener acceso vía SSL a contenidos que deseamos que sean seguros, es decir, accesibles en el navegador mediante el protocolo "https", será la carpeta /var/www/websegura

Carpeta web segura

- **Carpeta web segura:** /var/www/websegura
- **Acceso a la web segura:** https://ip-del-servidor/websegura/

Dentro de esta estructura la mayoría de los contenidos serán públicos y cualquier usuario podrá acceder a ellos. Sin embargo, algunas de las carpetas serán privadas y solo se tendrá acceso a ellas identificándose con nombre de usuario y contraseña.

Instalación de Apache2

Disponer de un servidor web en el centro nos permitirá alojar nuestras propias páginas y aplicaciones web de forma que den servicio tanto desde dentro de la intranet como desde Internet. Serán la base que facilitará el acceso a la información por parte de la comunidad educativa.

```
// Instalación de apache2
     # apt-get install apache2
```

Con lo cual se instalarán los archivos necesarios para que funcione nuestro servidor web. Se instalará apache v2.

Configuración de Apache

Los archivos de configuración de apache2 se encuentran en la carpeta **/etc/apache2**. El archivo principal de configuración es **/etc/apache2/apache2.conf**. Antes de realizar cualquier cambio en este archivo, es

conveniente realizar una copia de seguridad del mismo ya que si apache encuentra algún error en el archivo de configuración, no arrancará.

Se pueden configurar infinidad de parámetros. Aquí, para poner en marcha el servidor, editaremos el archivo apache2.conf y añadiremos únicamente el siguiente parámetro:

```
// Añadir en apache2.conf
     ServerName www.ieslapaloma.com
```

Para que los PCs de la red local sepan que www.ieslapaloma.com es nuestro servidor web, debemos crear una entrada 'www' hacia su dirección IP en el servidor DNS, o bien editar el archivo /etc/hosts agregando la línea: '192.168.1.239 www.ieslapaloma.com' (si la IP del servidor fuera 192.168.1.239). Si no, no quedará más remedio que acceder utilizando la dirección IP del servidor.

Arranque y parada del servidor web apache

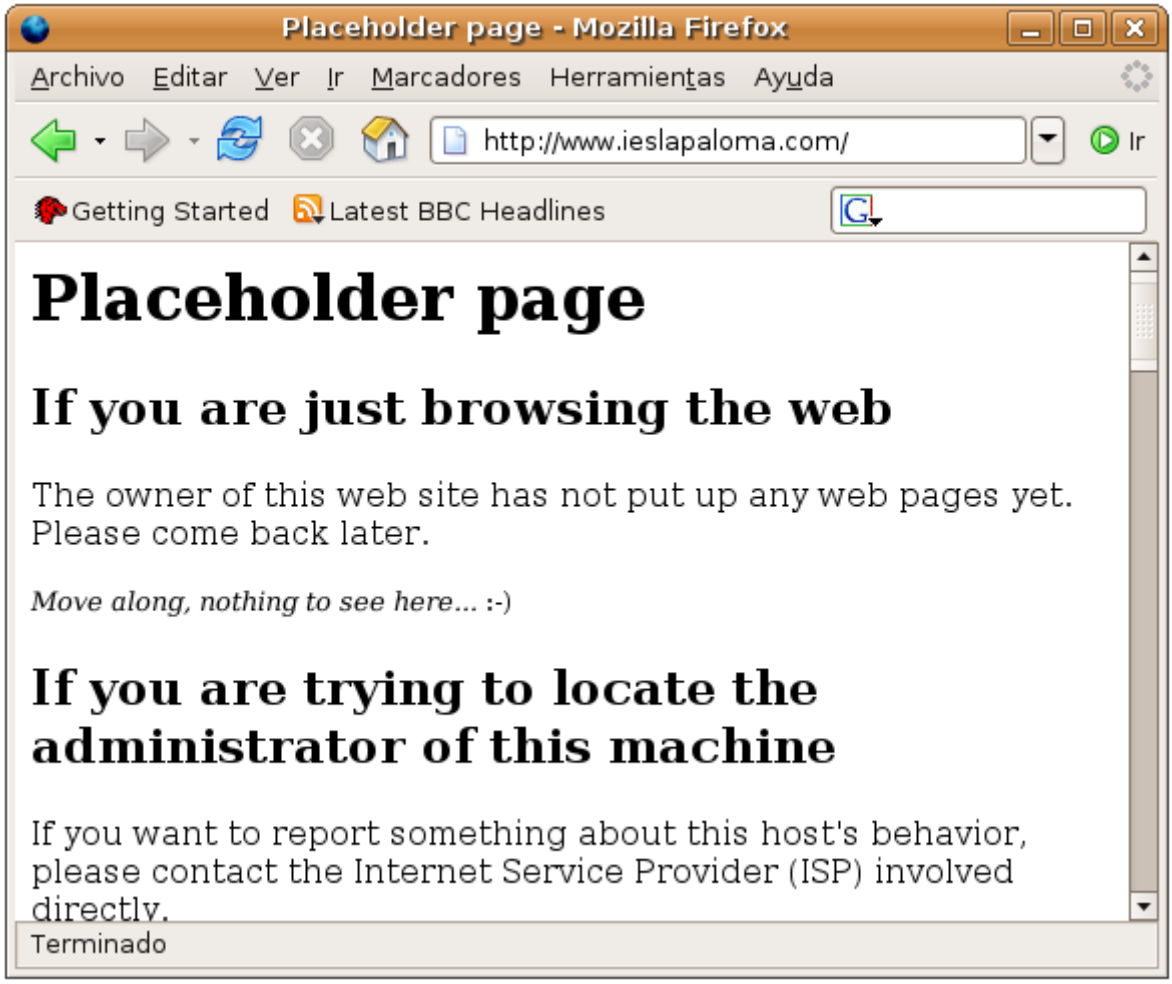
El servidor web apache2, al igual que todos los servicios en Debian, dispone de un script de arranque y parada en la carpeta /etc/init.d.

```
// Arrancar o reiniciar el servidor apache2
     # /etc/init.d/apache2 restart

// Parar el servidor apache
     root@cnice-desktop:/# /etc/init.d/apache stop
```

Arranque automático del servidor Web Apache al iniciar el sistema.

Para un arranque automático del servicio al iniciar el servidor, debemos crear los enlaces simbólicos correspondientes tal y como se indica en el apartado Arranque automático de servicios al iniciar el sistema.

Para comprobar que apache funciona perfectamente, desde el navegador de cualquier estación de trabajo de nuestro centro, debemos dirigirnos a 'http://ip-del-servidor'. Si tenemos el DNS funcionando, podemos acceder a 'http://www.ieslapaloma.com', visualizando la siguiente pantalla:



Si no disponemos de servidor DNS, podemos editar el archivo /etc/hosts y añadir la dirección IP del servidor e indicar el nombre, tal que así:

```
//Resolver nombres de dominio de forma local
//Añadir en /etc/hosts una línea similar a esta:
     192.168.1.239   www.ieslapaloma.com
```

Lo que siempre funcionará es ir con la dirección IP. Ejemplo, si la dirección IP de nuestro servidor fuera 192.168.1.239, podemos ir con el navegador a la dirección http://192.168.1.239 y obtendremos el mismo resultado. Podemos personalizar nuestra página modificando el archivo index.html que hay dentro de la carpeta /var/www.

Como vemos en la pantalla anterior, la instalación de Apache se produjo de forma adecuada, así pues hemos completado este apartado satisfactoriamente.

Activar el espacio web de los usuarios

Como hemos comentado anteriormente, cada usuario dispone de un espacio web que se almacena en la carpeta public_html dentro de su carpeta home. Si la carpeta public_html no existe, el propio usuario la puede crear y almacenar en ella su sitio web. La carpeta public_html deberá tener permisos 755. Si queremos que la carpeta public_html se genere de forma automática al dar de alta al usuario, se puede crear en /etc/skel. Para que apache procese los espacios web de los usuarios, es necesario activar el módulo **userdir** mediante el siguiente comando:

```
// Activar el espacio web de los usuarios
```

```
# a2enmod userdir
```

El acceso por web será mediante la URL: http://ip-del-servidor/~login-usuario/

Acceso a carpetas seguras

Introducción

Una página web segura o un sitio web seguro es un sitio web que utiliza el protocolo https en lugar de utilizar el protocolo http.

El protocolo https es idéntico al protocolo http con la excepción de que la transferencia de información entre el cliente (navegador web) y el servidor (servidor web) viaja a través de Internet cifrada utilizando robustos algoritmos de cifrado de datos proporcionados por el paquete OpenSSL.

Los algoritmos de cifrado utilizados reunen las características necesarias para garantizar que la información que sale desde el servidor hacia el cliente, esté cifrada y solamente pueda ser descifrada por el cliente y que la información que sale desde el cliente hacia el servidor, esté cifrada y solamente pueda ser descifrada por el servidor. Si durante la transferencia de la información un 'hacker' hiciera copia de los paquetes de datos e intentara descifrarlos, los algoritmos garantizarían que no podría hacerlo por fuerza bruta (probando todas las claves posibles) en un plazo mínimo de varios años.

Durante la transmisión, se utilizan algotirmos de cifrado simétricos, pero para intercambiar las claves de cifrado, hay una sesión inicial de cifrado asimétrico.

Módulo ssl para apache2

Al instalar apache2 se instala también el módulo ssl para apache2, por lo que no es necesario instalar ningún paquete adicional. Tan solo debemos generar un certificado para el servidor y activar el módulo ssl.

Generar el certificado

Para que nuestro servidor pueda servir páginas seguras con el protocolo https, necesita un certificado. Dicho certificado permitirá que nuestro servidor utilizar cifrado asimétrico para intercambiar las claves de cifrado con los clientes, antes de iniciar una trasmisión segura de información. Inicialmente, el cliente deberá aceptar el certificado del servidor, ya que generaremos un certificado autofirmado. Si queremos evitarlo, deberíamos contratar un certificado a una entidad certificadora confiable, pero tiene un coste que no merece la pena soportar en un entorno educativo. Para generar nuestro certificado autofirmado, ejecutaremos el comando:

```
// Generar certificado autofirmado
     # make-ssl-cert /usr/share/ssl-cert/ssleay.cnf
     /etc/ssl/certs/apache2.pem
```

Durante la ejecución de comando make-ssl-cert, quizás nos pregunte algunas sencillas preguntas como el nombre del servidor, el país, etc... y después se creará el archivo /etc/ssl/certs/apache.pem que contiene la claves que permitirán al servidor utilizar cifrado asimétrico. El siguiente paso será configurar un servidor virtual para que utilice dicho certificado.

Crear servidor virtual seguro en apache2

Primero crearemos una carpeta de nombre 'websegura' dentro de '/var/www'. Dicha carpeta será el raíz de

documentos (DocumentRoot) de nuestro servidor virtual seguro, de modo que todo lo que coloquemos en dicha carpeta deba ser accedido vía 'https'. Eso lo indicaremos más adelante mediante el parámetro SSLRequireSSL. El protocolo https utiliza el puerto 443, por lo tanto, tendremos habilitar dicho puerto para que apache lo utilice. Si ya está habilitado el puerto 443, no hacer nada.

```
// Habilitar puerto 443. Añadir en /etc/apache2/ports.conf
    Listen 443
```

Después debemos crear el servidor virtual en apache. Dicho servidor virtual dispondrá de una url de acceso diferente a la de nuestra web principal (websegura.ieslapaloma.com en nuestro ejemplo) y será accesible mediante https, por tanto tendremos que habilitar SSL e indicar la ruta del archivo que contiene el certificado. Todo ello lo haremos editando el archivo /etc/apache2/sites-available/default:

```
// Añadir al final en /etc/apache2/sites-available/default


    NameVirtualHost websegura.ieslapaloma.com:443

    <VirtualHost websegura.ieslapaloma.com:443>

            ServerName websegura.ieslapaloma.com

            DocumentRoot /var/www/websegura

            SSLEngine On

            SSLCertificateFile /etc/ssl/certs/apache2.pem

            ErrorLog /var/log/apache2/error.log

            CustomLog /var/log/apache2/access.log combined

    </VirtualHost>



    <Directory "/var/www/websegura">

        Options Indexes FollowSymlinks MultiViews

        AllowOverride None

        Order allow,deny

        Allow from all

        SSLRequireSSL

    </Directory>
```

Posteriormente debemos habilitar el módulo ssl del servidor apache:

```
// Habilitar el módulo ssl
     # a2enmod ssl
```

Finalmente reiniciamos el servidor apache:

```
// Reinicio de apache
      # /etc/init.d/apache2 restart
```

Probando el acceso a la página web segura

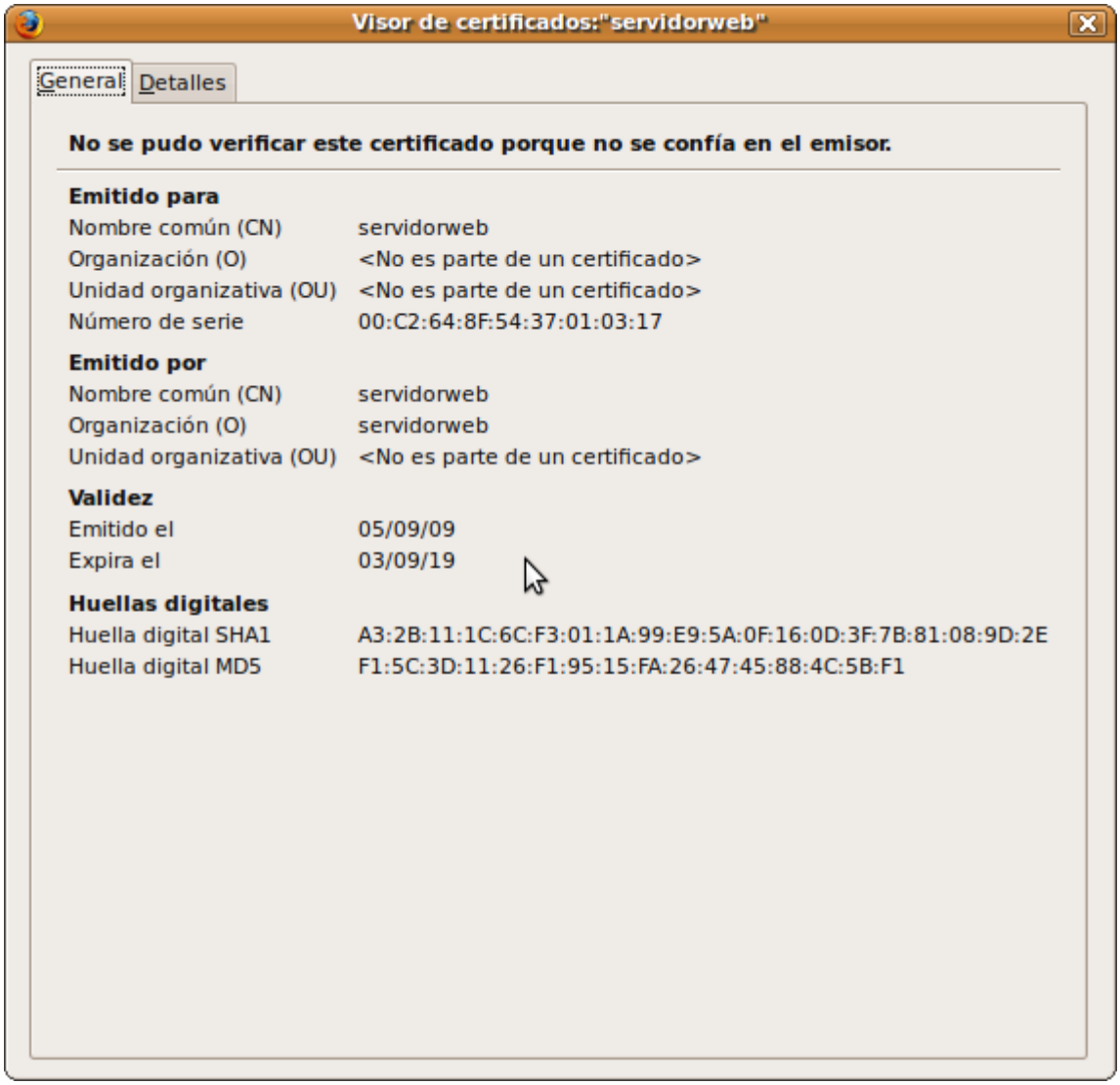
Nota: Si no tenemos un DNS funcionando, debemos incluir en /etc/hosts una línea para resolver localmente el nombre de nuestro servidor por su IP, porque en este caso, navegar con la dirección IP no funcionará. Ejemplo:

```
//Resolver el nombre localmente. Añadir en /etc/hosts
     192.168.1.239  websegura.ieslapaloma.com
```

Para acceder a las páginas seguras de nuestro servidor web, tecleamos desde el navegador 'https://websegura.ieslapaloma.com'. Lo primero que se muestra es la alerta de seguridad que nos indica que el certificado no está emitido por una CA en la que confiamos:

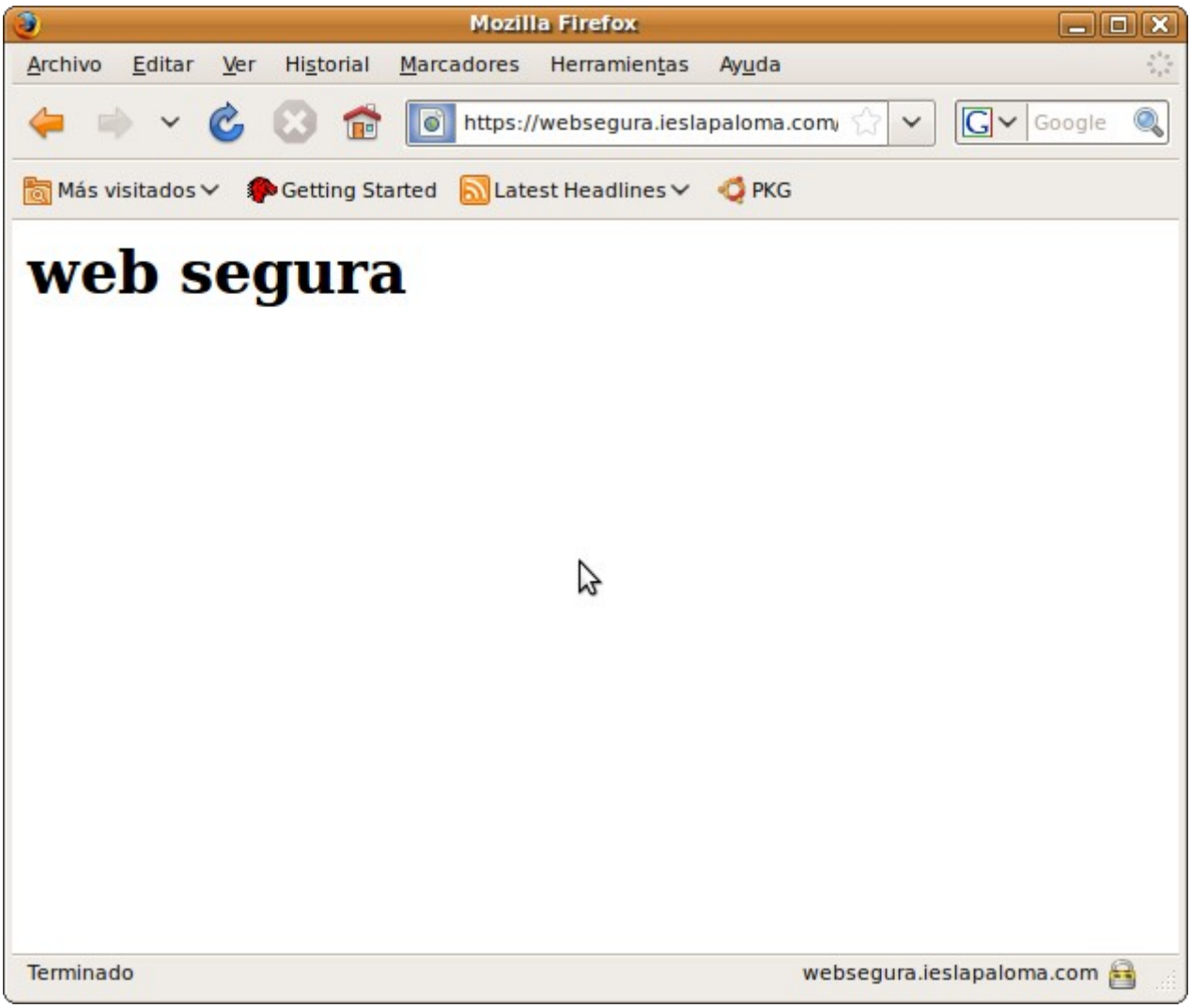
Para continuar debemos ir a añadir una excepción > obtener certificado. Si pulsamos sobre el botón 'Ver'

veremos la información tanto del certificado como de la entidad certificadora que lo firma:

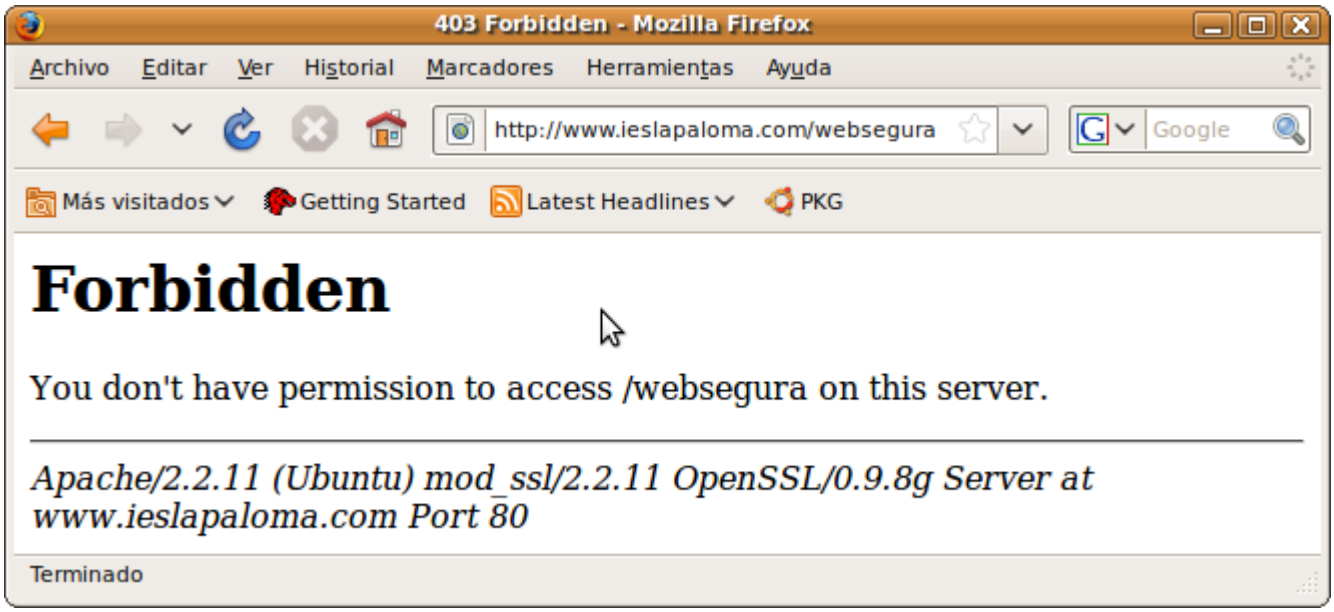


Si aceptamos el certificado significa que, a pesar de estar firmado por una entidad certificadora que no es de confianza para el navegador (lo hemos firmado nosotros mismos), lo aceptamos. Tendremos que indicar al navegador si aceptamos el certificado para siempre o solo para ahora. Como tenemos la seguridad de que el certificado es bueno porque acabamos de crearle nosotros mismos, podemos aceptarlo para siempre y así el navegador no volverá a preguntarnos más sobre él ya que hemos indicado manualmente que confiamos en este certificado:

Ahora ya tenemos acceso a la web segura mediante el protocolo https lo que nos garantiza que la información de la página segura, antes de salir del servidor, ha sido cifrada y por tanto la trasferencia de datos desde el servidor a nuestro navegador se ha producido de forma segura. Al llegar a nuestro navegador, se han descifrado los datos. El candado cerrado que aparece abajo a la derecha en el navegador, indica que la trasferencia de datos se ha realizado de forma segura.



Como sabemos la ruta de la carpeta segura, si intentamos acceder a la carpeta segura utilizando el protocolo http yendo con el navegador a 'http://www.ieslapaloma.com/websegura', apache denegará el acceso ya que en '/etc/apache2/sites-available/default' se ha especificado que la carpeta debe ser accedida mediante https:



Carpetas seguras de usuario

Si en el centro existiera la necesidad de que los profesores dispongan de una carpeta web segura donde poder colocar contenidos accesibles vía SSL, como serán casos excepcionales, una solución sencilla es crear una carpeta dentro de la carpeta '/var/www/websegura' para dicho profesor y para que éste tenga acceso de forma autónoma a subir contenidos a dicha carpeta, se le puede crear un usuario adicional cuyo home sea la carpeta correspondiente, ejemplo, para el profesor Javier podemos crear otro usuario llamado javier-s (javier-seguro) cuyo home sea /var/www/websegura/javier. Podría subir contenidos por ftp utilizando el usuario javier-s. El acceso a los contenidos desde un navegador sería yendo a la dirección https://websegura.ieslapaloma.com/javier

Este proceso habría que hacerlo para todos los profesores o departamentos de nuestro centro que requieran de carpeta segura.

Archivos log de apache

Por defecto, apache utiliza dos archivos de registro: access.log y error.log que están almacenados en la carpeta /var/log/apache2.

En el archivo **/var/log/apache2/access.log**, apache va registrando todos los accesos que los PCs hacen al servidor web y en cada línea de dicho archivo va almacenando la IP, la fecha y la hora, el comando HTTP enviado por el cliente, la url solicitada y la versión del navegador y el sistema operativo. Analizando este archivo podemos ver las veces que se ha descargado una página o un archivo, o las IPs más activas. Este archivo de registro es utilizado por los programas que presentan estadísticas de acceso al servidor web como awstats.

En el archivo **/var/log/apache2/error.log**, apache registra todas las incidencias o errores que se van produciendo. Ejemplo, cuando un cliente solicita una página inexistente o cuando un cliente intenta entrar en una carpeta prohibida o protegida. Si estamos configurando algo en apache (carpetas privadas, carpetas seguras, servidores web virtuales, alias, etc...) y no funciona, una buena idea es hacer pruebas y examinar el archivo error.log ya que nos puede dar pistas para encontrar la solución a nuestro problema.

```
//Ver últimas 20 líneas del access.log para ver quien está accediendo
      # tail -n 20 /var/log/apache2/access.log
```

Acceso a carpetas privadas con autentificación por LDAP

Otra posibilidad muy interesante es que los profesores e incluso el sitio web de la Intranet de nuestro centro, puedan disponer de carpetas privadas accesibles mediante el navegador pero no por cualquier usuario. Por ejemplo, los profesores podrían disponer de una carpeta donde almacenar información confidencial accesible desde la web -notas, por ejemplo-. Así mismo puede ocurrir que queremos tener en el servidor web de nuestra intranet, páginas a las que sólo puedan tener acceso de lectura los profesores del centro. En el capítulo LDAP > Carpetas privadas en Apache, explicamos cómo hacerlo.

Apache+PHP+MySQL+PHPMyAdmin

Para poder aprovechar al máximo las características del servidor web apache, es muy conveniente que pueda ejecutar scripts en servidor y pueda acceder a bases de datos.

Las aplicaciones web más interesantes como los gestores de contenidos para crear y mantener sitios web

dinámicos, wikis, blogs, foros-web, repositorios de arhivos, etc... requieren de lenguaje script en servidor y sistema gestor de bases de datos.

En el desarrollo web del mundo Linux el lenguaje script en servidor más utilizado es el lenguaje php y el sistema gestor de bases de datos más utilizado es mysql. Phpmyadmin es una excelente herramienta para administrar bases de datos mysql.

Más información sobre cómo instalar y configurar php, mysql y phpmyadmin en:

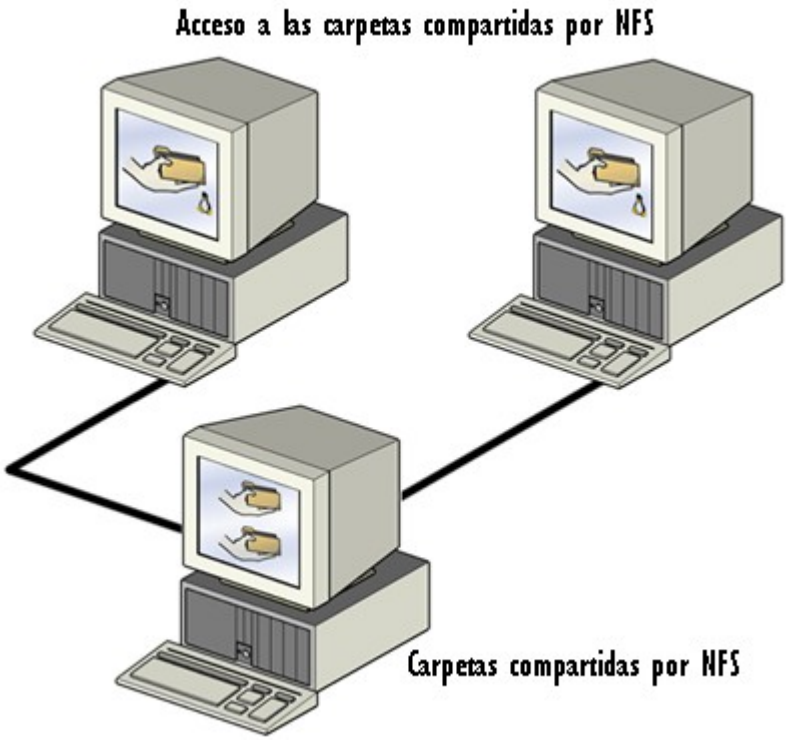
- Instalacion_y_configuracion_de_PHP
- Instalacion_y_configuracion_de_MySQL
- Instalacion_y_configuracion_de_PHPMyAdmin
-

7.- NFS

NFS es el sistema que utiliza Linux para compartir carpetas en una red. Mediante NFS, un servidor puede compartir sus carpetas en la red. Desde los PCs de los usuarios se puede acceder a dichas carpetas compartidas y el resultado es el mismo que si estuvieran en su propio disco duro. NFS son las siglas en inglés de Network File System que podríamos traducir como Sistema de Archivos en Red.

Básicamente **NFS** permite, a PCs que utilizan Linux, compartir y conectarse a carpetas compartidas entre sí. Es el sistema nativo que utiliza Linux para compartir y acceder a carpetas compartidas en la red.

Existen otras alternativas para compartir carpetas en una red como **samba, ssh o ftp,** pero el sistema recomendado para compartir carpetas entre sistemas Linux es NFS.



Compartir carpetas por NFS

Instalación de NFS

Para poder disfrutar del servicio de compartir carpetas en la red mediante NFS, en el PC servidor es necesario instalar el paquete del **servidor NFS**. Lo normal es que todos los PCs dispongan del paquetes servidor de NFS ya que en cualquier momento puede existir la necesidad de tener que compartir una carpeta desde cualquier PC, aunque lo habitual es que el único que comparta sea el servidor. Que un PC de un usuario tenga instalado el paquete del servidor NFS, no significa que automáticamente esté compartiendo su sistema de archivos en la red. Para ello es necesario configurar y arrancar el servicio.

Si deseamos instalar la última versión disponible, podemos hacerlo con apt-get desde una consola de root:

```
// Instalación de NFS
     # apt-get install nfs-common nfs-kernel-server
```

Configuración del servidor NFS

Antes de arrancar el servicio NFS, es necesario indicar qué carpetas deseamos compartir y si queremos que los usuarios accedan con **permisos de solo lectura o de lectura y escritura**. También existe la posibilidad de establecer desde qué PCs es posible conectarse. Estas opciones se configuran en el archivo /etc/exports

```
// Archivo de configuración del servidor NFS
     /etc/exports
```

En cada línea del archivo de configuración del servidor NFS /etc/exports, se puede especificar:

- La carpeta que se quiere compartir
- El modo en que se comparte (solo lectura 'ro' o lectura y escritura 'rw')
- Desde qué PC o PCs se permite el acceso (nombre o IP del PC o rango de IPs)

A continuación mostramos un sencillo archivo /etc/exports para configurar algunas carpetas compartidas

```
// Ejemplo de archivo /etc/exports de configuración del servidor NFS:

     # Compartir la carpeta home del servidor

     #  en modo lectura y escritura y accesible desde la red 192.168.0.0/24

     /home           192.168.0.0/255.255.255.0(rw)



     # Compartir carpeta tmp a todos como 'solo-lectura'

     /tmp            *(ro)



     # Compartir carpeta /var/log a un PC como 'solo-lectura'

     /var/log        192.168.0.211(ro)
```

Nota: Los permisos de compartición por NFS no excluyen a los permisos del sistema unix sino que **prevalecen los más restrictivos**. Si una carpeta está compartida con permiso NFS de lectura y escritura pero en los permisos del sistema solo disponemos de permiso de lectura, no podremos escribir. Si una carpeta está compartida con permisos NFS de lectura y disponemos de permisos de lectura y escritura en el sistema, tampoco podremos escribir. Para poder escribir necesitaremos disponer permiso de lectura y escritura tanto en los permisos del sistema como en los permisos de compartición NFS. De igual forma, si compartimos la carpeta /home con permisos de lectura y escritura pero el usuario pepe solo tiene acceso a la carpeta /home/pepe, no podrá acceder a ninguna otra carpeta dentro de /home ya que los permisos del sistema se lo impedirán.

Cuando se comparte por NFS, se recomienda restringir al máximo los permisos. Si los usuarios no tienen la

necesidad de escribir, debemos compartir con permiso de 'solo lectura'. Si los usuarios solo se conectan desde nuestra red 192.168.0.0/24, debemos permitir el acceso solo desde dicha red.

Arranque y parada de NFS

Arranque y parada manual

Para que el servidor NFS funcione, es necesario que esté arrancado el servicio **portmap**, por lo tanto, la primera acción será iniciar portmap por si no estuviera arrancado:

```
// Iniciar portmap
     # /etc/init.d/portmap start
```

Para poner en marcha el servicio NFS, o cada vez que modifiquemos el archivo /etc/exports, debemos reiniciar el servidor **NFS**, mediante el comando:

```
// Reinicio del servidor NFS
     # /etc/init.d/nfs-kernel-server restart
```

Si deseamos detener el servidor **NFS**, debemos ejecutar:

```
// Parada del servidor NFS
     # /etc/init.d/nfs-kernel-server stop
```

Arranque automático de NFS al iniciar el sistema.

Para un arranque automático del servicio al iniciar el servidor, debemos crear los enlaces simbólicos correspondientes tal y como se indica en el apartado Arranque automático de servicios al iniciar el sistema.

Acceso a carpetas compartidas por NFS

Para poder acceder desde un PC a una carpeta compartida por NFS en un servidor, lo primero que tenemos que hacer es instalar los paquetes portmap y nfs-common que nos permitirán acceder como clientes:

```
// Instalar portmap y nfs-common y reiniciar portmap
     # apt-get install portmap nfs-common

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

Ahora ya estaremos en condiciones de **montar** la carpeta compartida en nuestro sistema de archivos. De ésta manera, el acceso a la carpeta compartida es exactamente igual que el acceso a cualquier otra carpeta de nuestro disco duro.

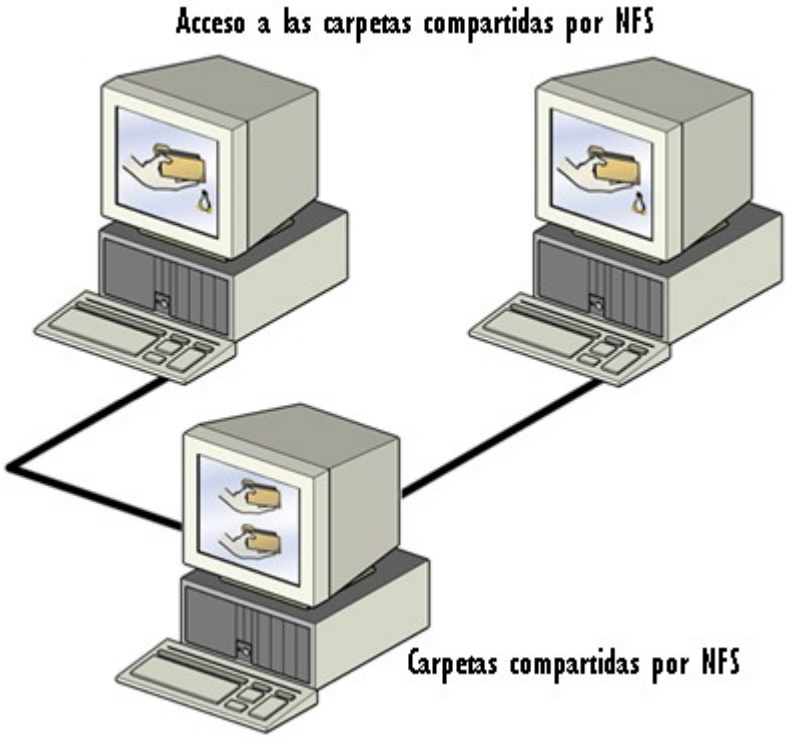
Ejemplo, supongamos que un servidor comparte por NFS una carpeta llamada /fotos. En el PC cliente podemos crear una carpeta llamada /fotos-servidor y montar sobre ella la carpeta compartida en el servidor. Para ello, en el cliente y como root ejecutaríamos el siguiente comando:

```
// Montar carpeta compartida por NFS
     # mount -t nfs ip-del-servidor:/fotos /fotos-servidor
```

A partir de éste momento, podemos comprobar que nuestra carpeta /fotos-servidor contiene la información de la carpeta /fotos del servidor. Si disponemos de permisos de lectura y escritura, podemos incluso crear o modificar los archivos dentro de nuestra carpeta /fotos-servidor y los cambios se estarán guardando realmente en la carpeta /fotos del servidor.

Para realizar el montaje, debemos hacerlo sobre una carpeta existente en nuestro sistema. Si dicha carpeta de nuestro sistema contiene archivos, estos no estarán accesibles ya que la carpeta nos mostrará los archivos remotos.

Si al intentar montar la carpeta NFS no funciona suele ser por una de estas tres razones: por un **problema en la red**, un **problema en el servidor** o un **problema en el cliente**. Para averiguar si el problema es del servidor o no, podemos intentar montar por NFS la carpeta en el propio servidor, usando la IP 127.0.0.1. Si funciona entonces el problema estará en la red o en el cliente. Si hacemos ping del servidor al cliente y no hay cortafuegos, el problema será en el cliente. Podemos intentar hacer una reinstalación del cliente igual que la instalación en el servidor. Ejecuta en el cliente los siguientes comandos: apt-get install nfs-common nfs-kernel-server, luego /etc/init.d/portmap restart, después /etc/init.d/nfs-kernel-server restart y finalmente intentar montar la carpeta.



Si deseamos que nuestro PC monte siempre de forma automática una carpeta compartida por NFS cuando iniciemos nuestro Linux, existe la posibilidad de añadir en el archivo /etc/fstab una línea como por ejemplo:

```
# Montaje automático al iniciar el PC
```

```
#Añadir en /etc/fstab

ip-del-servidor:/fotos /fotos-servidor nfs
```

De ésta manera, cuando arranquemos nuestro PC, la carpeta /fotos del servidor quedará automáticamente montada sobre nuestra carpeta /fotos-servidor y no tendremos que ejecutar el comando mount para nada.

Consejos

Es conveniente que los datos de los usuarios se almacenen de forma centralizada en el servidor en lugar de hacerlo en los PCs de los usuarios. Ésto permite al usuario acceder a sus archivos aunque utilice un PC diferente al habitual, además, será más sencillo realizar copias de seguridad y si el PC del usuario se estropea, no perderá información. Lo ideal es que los PCs de usuario no almacenen la carpeta home de cada usuario sino que dicha carpeta esté compartida en el servidor. El servidor así mismo deberá centralizar las cuentas de usuario mediante un servidor LDAP y los PCs clientes deberán estar configurados para montar el home de forma remota y autentificar a los usuarios a través del servidor LDAP.

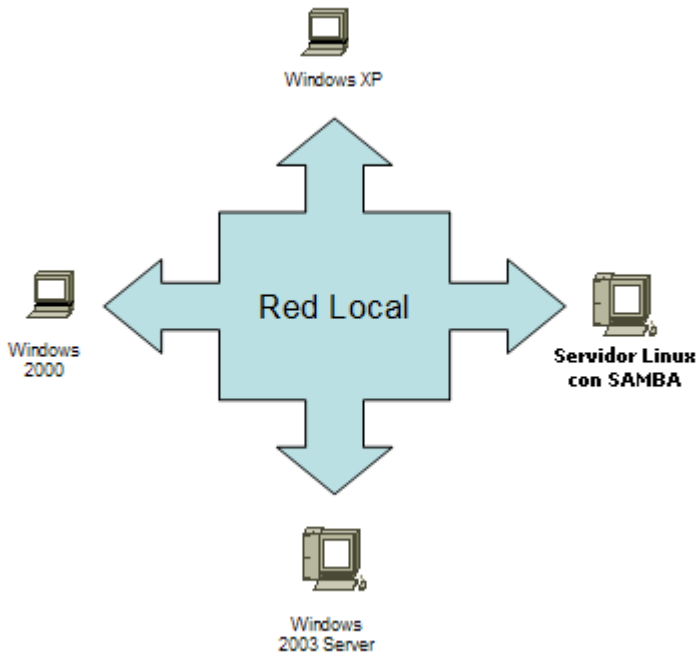
Para su uso práctico en el centro educativo, además de exportar la carpeta /home para que cada usuario tenga acceso a su espacio de trabajo, se pueden crear en el servidor tres carpetas de uso común cuyos permisos dependan del tipo de usuario. En una de ellas deberán tener permisos de lectura y escritura todos los usuarios: alumnos y profesores. A dicha carpeta se la puede llamar **comun-alumnos** y serviría para que los alumnos intercambien archivos entre ellos y con los profesores. En otra carpeta, deberían tener permisos de lectura y escritura solamente los profesores. Se podría llamar **comun-profesores** y serviría para que los profesores se intercambien archivos confidenciales entre sí. Finalmente, en otra carpeta deberían tener solo permisos de lectura los profesores y los alumnos. Se podría llamar **documentos-centro** y serviría para que el administrador mantenga un repositorio de documentos o aplicaciones de propósito general para el centro, aunque ésto último quizás sea más adecuado hacerlo mediante un servidor web.

8.- Samba

Samba son un conjunto de aplicaciones libres para Linux que implementan el protocolo de comunicación SMB utilizado por los sistemas operativos Microsoft Windows para compartir carpetas e impresoras.

Básicamente **samba** permite a PCs que utilizan Linux conectarse a carpetas compartidas en PCs con Windows y compartir carpetas como si de un Windows se tratara. Gracias a **samba**, en una red podemos tener PCs con Windows y PCs con Linux de forma que puedan intercambiar información en carpetas compartidas de la misma forma que se haría si todos los PCs fueran Windows.

Cuando en una misma red conviven sistemas Unix con equipos Windows, se utiliza **samba** para integrarlos y poder intercambiar información. Como alternativa, sería posible utilizar protocolos estándar como el ftp que es utilizado tanto equipos Windows como equipos Linux.



Red con sistemas windows y linux

Las funcionalidades de **samba** no se quedan solo en una simple compartición de archivos e impresoras sino que permite a un PC con Linux comportarse como un controlador de dominio de Windows para redes Microsoft con prestaciones superiores a las que nos ofrecería un servidor con Windows NT Server 4.0. En las páginas siguientes veremos como hacer que un PC con Linux haga las veces de controlador de dominio de nuestra red Windows.

Instalación de samba

La 'suite' completa de samba se compone de varios paquetes. Se pueden localizar en http://packages.debian.org buscando 'samba' en la descripción de los paquetes. Destacamos los más importantes:

- samba - Servidor de archivos e impresoras tipo LanManager para Unix.
- samba-common - Archivos comunes de samba utilizados para clientes y servidores.
- smbclient - Cliente simple tipo LanManager para Unix.
- swat - Herramienta de administración de Samba via web

- samba-doc - Documentación de Samba.
- smbfs - Comandos para montar y desmontar unidades de red samba
- winbind: Servicio para resolver información de usuarios y grupos de servidores Windows NT

Instalaremos los paquetes necesarios para disfrutar del servicio. Para ello ejecutaremos:

```
// Instalar samba
     # apt-get install samba samba-common smbclient samba-doc smbfs
```

así tendremos instalados y actualizados a la última versión, los paquetes básicos para disfrutar del servicio SAMBA.

Configuración de samba

Introducción

Samba, al igual que todas las aplicaciones para Linux, dispone de un archivo de texto para su configuración. Se trata del archivo:

```
// Archivo de configuración de samba
     /etc/samba/smb.conf
```

Aunque el archivo de configuración de **samba** es bastante extenso, para empezar a disfrutar de samba, tenemos que hacer muy pocos cambios. El archivo de configuración se divide en secciones identificadas por un nombre entre corchetes. Hay tres secciones especiales que son **[global], [homes] y [printers]**. La sección principal es la sección **[global]** que nos permite configurar los parámentros generales del servicio. La sección **[homes]** nos permitirá compartir las carpetas home de cada usuario, para que cada usuario pueda acceder a su carpeta home por la red. La sección **[printers]** nos permitirá compartir impresoras. Para compartir una carpeta, debemos crear una sección nueva. El nombre de la sección, será el nombre del recurso compartido. Ejemplo, si queremos compartir la carpeta /home/comun-profes y llamar al recurso compartido **profes**, debemos crear una sección llamada **[profes]**. Para facilitar la configuración de **samba** existe una herramienta llamada swat que permite, vía web, configurar la aplicación.

Puesto que editando el archivo **smb.conf** se pueden configurar más de 300 parámetros, dando lugar a miles de configuraciónes, nos limitaremos a analizar los parámetros más relevantes y a la compartición de archivos e impresoras directamente.

Archivo smb.conf

Podemos ver un ejemplo del archivo de configuración de **samba** haciendo clic »aqui.

A continuación analizaremos un sencillo archivo smb.conf:

```
# Ejemplo de archivo de configuración de samba smb.conf



       #Sección global, parámetros generales

       [global]
```

```
    # Seguridad por usuarios
        security = user
    # Grupo de trabajo 'Aula5'
        workgroup = Aula5
    # Las contraseñas se deberán enviar encriptadas
        encrypt passwords = yes
    # Samba será servidor wins
        wins support = yes
    # Nivel y longitud máxima del archivo de registro
        log level = 1
        max log size = 1000
    # Por defecto, lectura y escritura
        read only = no
    # Se comparten también las impresoras
        load printers = yes

    # Sección homes, carpetas home de usuarios
    [homes]
    # Comentario
        comment = Carpetas home
    # No explorables
        browsable = no
    # Máscara de creación de archivos (rxw------)
        create mask = 0700
    # Máscara de creación de carpetas
        directory mask = 0700

# Sección printers, impresoras
```

```
    [printers]
        path = /var/tmp
        printable = yes
        min print space = 2000

# Carpeta común profesores
    [profesores]
    # Ruta de la carpeta compartida
        path = /home/samba/profesores
    # Explorable
        browsable = yes
    # Lectura y escritura
        read only = no
    # Máscara de creación de archivos (rxwrxw---)
        create mask = 0770
    # Máscara de creación de carpetas
        directory mask = 0770

    # Carpeta común alumnos
    [alumnos]
        browsable = yes
        read only = no
        path = /home/samba/profesores/alumnos

    # Carpeta común del centro (solo lectura)
    [programas]
        browsable = yes
        read only = yes
```

```
# Se admiten invitados

    guest ok = yes

    path = /home/samba/programas



# Parámetros impresora

[laserjet5]

    path = /tmp

# Se permite imprimir

    printable = yes
```

Todas las líneas que comienzan por almoadilla (#) o punto y coma (;) son líneas de comentarios y son ignoradas por **samba**.

Como hemos comentado anteriormente, el archivo smb.conf está dividido en secciones identificadas con corchetes [ ] . Ninguna de las secciones son obligatorias aunque normalmente suelen tener las siguientes secciones:

Sección [global]

En la sección [global] se configuran los parámetros generales (globales) que determinarán el modo de comportamiento general del servidor **samba**. Todos los parámetros que se omitan tomarán el valor predefinido por defecto. Existen unos 300 parámetros que se pueden configurar en ésta sección. A continuación exponemos los parámetros más significativos y ejemplo de valor:

- hosts allow = 192. 127.
    - Permite especificar desde qué direcciones IPs se podrá acceder al servicio. Ej.: Si ponemos 192.168. significa todas las que empiecen por 192.168.
    - Se pueden poner IPs concretas
- hosts deny = 10.
    - Igual que hosts allow pero para especificar los rangos no permitidos
- security = share
    - Permite determinar el modo de compartición de recursos de **samba**. Hay cinco opciones posibles: share, user, domain, server y ads.
    - 'Share' significa compartir los recursos con contraseña (como W95, 98,...).
    - 'User' gestiona los permisos por usuario (como W2000 y WXP).
    - 'Domain' gestiona los permisos por dominio.
    - 'Server' indica que los permisos son gestionados por otro servidor.
    - 'Ads' hace que samba se comporte como un miembro de un dominio Active Directory y por lo tanto requiere un servidor W2000 Server o W2003 Server.
    - **Samba** no puede actuar como controlador de dominio de Active Directory, es decir, no puede sustituir a Windows 2000 Server, pero sí puede actuar como controlador de dominio de Windows NT.
- domain logons = yes
    - Para que **samba** sea autentificador del dominio. En este caso, habrá que poner 'security = user' porque no tiene sentido que el **samba** sea servidor de dominio y que comparta los recursos con contraseña.
- domain master = yes
    - Para que **samba** sea controlador de dominio. Lo lógico es que domain logons esté a 'yes'
- encrypt passwords = yes

        - Hace que **samba** solo reconozca passwords encriptados. Las primeras versiones de W95 enviaban las contraseñas en texto plano pero tanto las últimas versiones de Windows 95 como W98, 2000 y XP las encriptan. Se puede impedir que W98 las encripte cambiando un valor del registro (ver encription.txt en **samba**) pero lo recomendable es que se envíen encriptadas para impedir que otros usuarios puedan descubrirlas capturando paquetes de datos (sniffing). Los password encriptados de **samba** se guardan en otro archivo:
- smb passwd file = /etc/smbpasswd
    - Archivo que guarda las contraseñas encriptadas de acceso a **samba**. Para que un usuario pueda acceder a **samba** debe existir en el sistema pero no tiene por qué coincidir la contraseña de un usuario en el sistema linux con la de **samba** aunque es aconsejable.
- logon script = INICIO.BAT
    - Indica el script que ejecutarán los clientes windows al loguearse
- password server = 192.168.0.10
    - Indica qué servidor autentificará a los usuarios
- wins server = 192.168.0.10
    - Indica quién es el servidor de nombres wins
- wins support = yes
    - Hace que nuestro samba sea servidor wins
- load printers = yes
    - Para que automáticamente comparta todas las impresoras del sistema

Sección [homes]

En ésta sección se configuran los parámetros para compartir la carpeta home (carpeta donde se almacena el perfil y todos los documentos) de cada usuario. Esta sección es opcional. Si no existe, no se compartirán las carpetas home de cada usuario. Se utiliza cuando se desean crear perfiles móviles de forma que cuando se identifique el usuario en cualquiera de los PCs de la red, se mapee de forma automática su perfil.

Sección [printers]

En ésta sección se configuran los parámetros para compartir las impresoras o colas de impresión disponibles en el servidor.

Una sección por cada carpeta compartida

Cada vez que se comparte una carpeta, hay que crear una sección denominada como se desee ya que dicho nombre será el nombre del recurso compartido. Ejemplo, si deseamos compartir la carpeta /home/samba/alumnos crearemos una sección [alumnos] donde se configurará dicho recurso compartido con los parámetros específicos para dicho recurso. Parámetros destacables:

- browseable = yes
    - Indica si el recurso compartido será visible cuando se escanea la red, por ejemplo haciendo clic en 'Mis sitios de red' en Windows
- create mask = 0770
    - Establece la máscara de creación de archivos, igual con directory mask para la creación de carpetas
- guest ok = yes
    - Indica que cualquier usuario sin contraseña tiene permiso de acceso
- valid users = pepe, juan
    - Indica qué usuarios pueden acceder al recurso

Consejos

Es conveniente crear en /home una carpeta llamada **samba** y que cuelguen de ella todas las carpetas compartidas, para tener todos los datos de usuario dentro de /home y sea sencillo hacer las copias de seguridad.

Si somos servidores de dominio y vamos a tener en nuestra red clientes Windows, es conveniente crear un

recurso compartido llamado **netlogon** para poder almacenar scripts de inicio y archivos de políticas ya que los clientes Windows están preconfigurados para acceder a dicho recurso compartido:

```
// Si samba es controladores de dominio se recomienda crear recurso 'netlogon'
      [netlogon]

       path = /home/samba/netlogon

       public = no

       writeable = no

       browsable = no
```

Si deseamos almacenar los drivers de impresora para los clientes Windows crearemos una sección [print$]

Samba analiza cada 60 segundos el archivo smb.conf y si ha habido cambios, estos tomarán efecto. Es conveniente crear una copia de seguridad del archivo smb.conf antes de hacer ningún cambio para poder retornar al estado anterior en caso de que hagamos una modificación incorrecta del archivo que impida que arranque el servicio.

Para comprobar que nuestro archivo smb.conf está correcto, podemos utilizar el comando testparm que analiza cada línea en busca de errores.

Para tener una descripción detallada de todos los parámetros se puede consultar la página del manual de smb.conf:

```
// Página del manual de smb.conf
     $ man smb.conf
```

Arranque y parada de samba

Arranque y parada manual

Samba, al igual que todos los servicios en Debian, dispone de un script de arranque y parada en la carpeta /etc/init.d.

Si deseamos iniciar o reiniciar el servidor **samba**, debemos ejecutar:

```
// Iniciar o reiniciar el servidor samba
     # /etc/init.d/samba restart
```

Este comando reiniciará los dos demonios (procesos residentes) necesarios que necesita **samba** para su funcionamiento: nmbd y smbd.

Si deseamos detener el servidor **samba**, debemos ejecutar:

```
// Parada del servidor samba
     # /etc/init.d/samba stop
```

Si deseamos reiniciar el servidor **samba**, debemos ejecutar:

Arranque automático de samba al iniciar el sistema.

Para un arranque automático del servicio al iniciar el servidor, debemos crear los enlaces simbólicos correspondientes tal y como se indica en el apartado Arranque automático de servicios al iniciar el sistema.

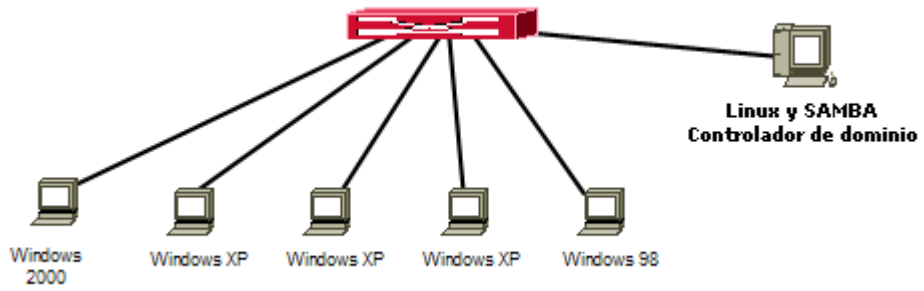
Unión de equipos al dominio

Introducción

Un **Dominio de Windows** es una agrupación lógica de PCs en los cuales existe al menos un servidor especial llamado **Controlador del Dominio** en el que se centralizan las tareas de administración de todos los PCs de la red.

Ejemplo, si en un aula disponemos de 10 PCs con Windows XP formando un grupo de trabajo (WORKGROUP), en cada uno de ellos deberemos crear las cuentas de usuario que necesitemos. A menudo se utilizan cuentas genéricas (alumno, profesor, ...). Si más adelante deseamos crear una nueva cuenta, por ejemplo, 'responsable' para el responsable del aula, no nos quedará más remedio que ir PC por PC creando dicha cuenta. Si vamos más alla y deseamos que cada profesor y cada alumno disponga de su cuenta de usuario, deberíamos crearlas en cada uno de los PCs lo que se convierte en una tarea disparatada con resultados poco prácticos.

Para centralizar las cuentas de usuarios, lo más razonable es crear un dominio de Windows. Para ello necesitamos de un PC que sea Controlador del Dominio. En dicho PC es necesario instalar un sistema operativo capaz de actuar como Controlador de Dominio. Podemos elegir entre: Windows NT Server, Windows 2000 Server, Windows 2003 Server o Linux con samba. El resto de PCs deberán **unirse al dominio** de forma que cuando un usuario se loguée (identifique con su nombre y contraseña) en cualquiera de los PCs, éstos envían de forma encriptada la información al controlador del dominio y será éste quien valide o invalide el acceso. Las cuentas de usuario se crean en el controlador pero sirven para autentificarse en cualquiera de los PCs pertenecientes al dominio.



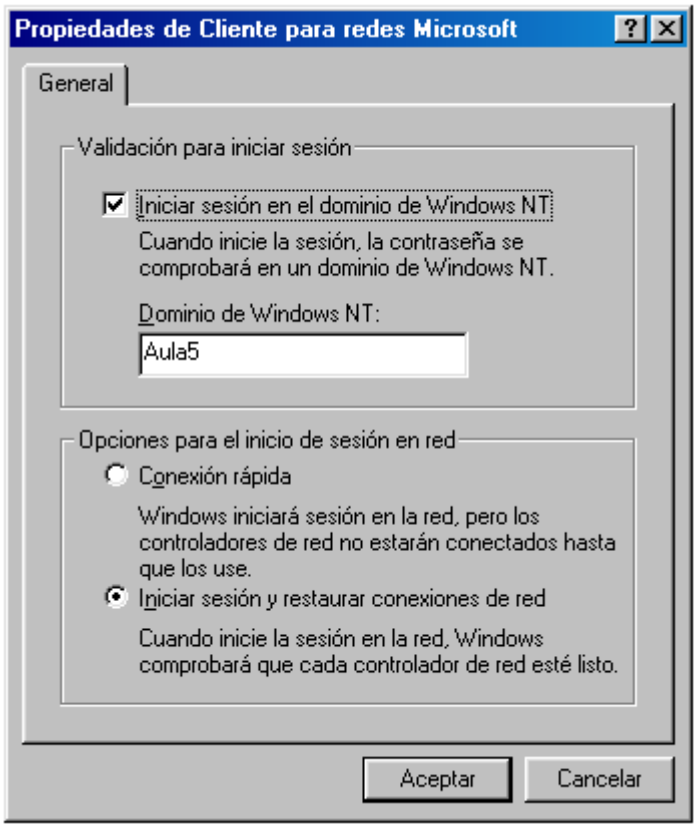
Red con un dominio Windows controlado mediante samba

Lo normal es que el Controlador de Dominio almacene también los documentos de los usuarios, por ello será interesante compartir la carpeta 'home' de cada usuario creando la sección 'homes' en el archivo de configuración de samba. Así no solo centralizaríamos las cuentas de usuario sino también sus documentos, lo que facilitaría la realización de las copias de seguridad, pues en los PCs de los usuarios no sería necesario salvaguardar nada al estar todo en el servidor.

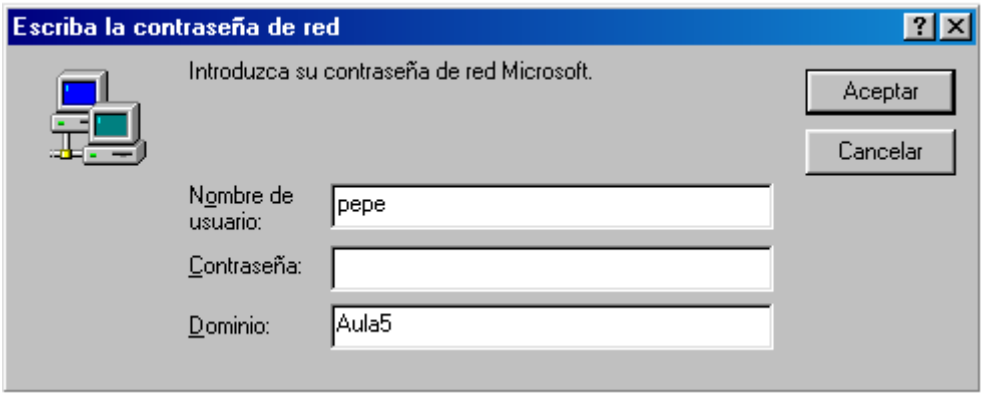
Unión de equipos con Windows 95 o Windows 98 al dominio

Para hacer que un PC con Windows 95 o Windows 98 pertenezca a un dominio, es necesario realizar los siguientes pasos:

- Clic con el derecho en entorno de red > Propiedades > Seleccionar 'Cliente para redes Microsoft' > Propiedades. Aparecerá la siguiente ventana:



Debemos seleccionar la casilla 'Iniciar sesión en el dominio de Windows NT' y debemos escribir el nombre del dominio que deberá coincidir con el parámetro 'workgroup = nombre' en el archivo de configuración de samba. De ésta forma, la próxima vez que reinicie el PC con Windows 95 o Windows 98, la ventana de identificación del usuario será así:



El usuario deberá introducir un nombre y una contraseña de una cuenta existente en el Controlador de Dominio ya que de lo contrario no podrá utilizar el PC.

Unión de equipos con Windows 2000 o Windows XP al dominio

Para unir al dominio PCs con Windows 2000 ó Windows XP, es necesario previamente crear en el servidor samba una cuenta de usuario para el equipo a unir. Supongamos que el PC que vamos a unir al dominio, se llama 'aula5pc7', deberémos crear en el servidor un usuario llamado 'aula5pc7$' (terminado en dolar)

ejecutando como root el siguiente comando:

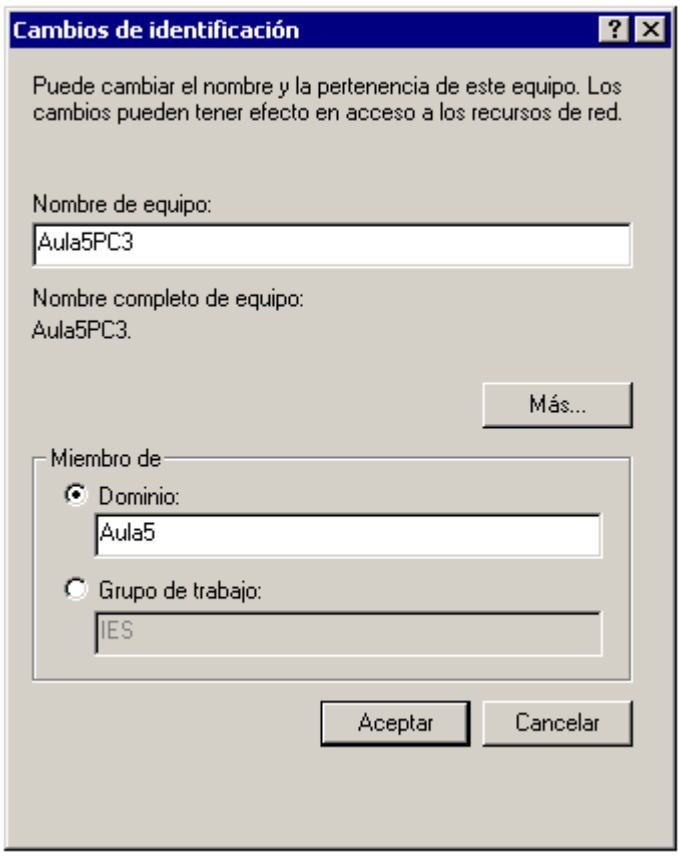
```
#useradd -g 100 -d /dev/null aula5pc7$ && passwd -l aula5pc7$ && smbpasswd -a -m aula5pc7
```

- Opciones useradd:
  - -g: indica el grupo inicial de dicho usuario. 100 corresponde al grupo 'users'
  - -d: indica la carpeta home del usuario (no necesitamos)
- Opciones passwd:
  - -l: indica que el password del usuario quede bloqueado para que nadie pueda hacer login con ese nombre de usuario.
- Opciones smbpasswd:
  - -a: indica que añada al usuario
  - -m: indica que es una cuenta de máquina (equipo)

De ésta forma habremos creado en Linux y en samba un usuario para el equipo que se va a unir al dominio.

Después deberemos ir al PC y, en el caso de Windows 2000 haremos:

- Clic con el derecho en Mi PC > Propiedades > identificación de red > Propiedades. Nos aparecerá la siguiente ventana:



Debemos seleccionar la opción 'Dominio' y debemos escribir el nombre del dominio.

En el caso de PCs con Windows XP haremos:

- Clic con el derecho en Mi PC > Propiedades > Nombre de equipo > Cambiar, y el procedimiento es similar al anterior.

Gestión de usuarios, grupos y permisos de samba

Samba es un servicio que requiere de administración de usuarios para poder gestionar los permisos de éstos. En función del usuario que acceda, samba se comportará de una forma u otra ya que cuando accede un usuario normal, generalmente tiene unos permisos limitados y cuando accede un usuario administrador, deberá disponer de todos los permisos.

Para que esa administración sea posible, samba dispone de su propia base de datos de 'usuarios samba' pero como los usuarios utilizan otros recursos del servidor como carpetas e impresoras, es necesario que estén creados en el sistema Unix. Resumiendo, podemos decir que **para poder ser usuario de samba, es necesario disponer de una cuenta de usuario en Unix y de una cuenta de usuario en samba.**

Todo lo relativo a la creación y administración de usuarios y grupos en Unix se puede consultar en el apartado 'Usuarios del sistema Unix'. Clic aquí para acceder al índice.

Gestión de usuarios de samba

La gestión de usuarios de samba se realiza con el comando smbpasswd. Con él podremos crear y eliminar usuarios, cambiar su contraseña y algunas cosas más.

Creación de un usuario de samba

Para crear un usuario de samba debemos utilizar el comando smbpasswd, pero antes debemos haber creado el usuario en Unix. Ejemplo, supongamos que queremos crear en Unix al usuario pepe:

```
// Creación de un usuario en unix
     # useradd pepe
```

Si deseamos que pepe pueda disfrutar de los servicios samba, debemos crear a pepe como usuario de samba ejecutando el siguiente comando:

```
// Creación de un usuario de samba
     # smbpasswd -a pepe
```

Con la opción -a indicamos que añada al usuario. Acto seguido nos preguntará dos veces la contraseña que deseamos poner al usuario. Lo razonable es que sea la misma contraseña que tiene el usuario en Unix. A continuacion veremos un ejemplo de utilización:

```
root@knoppix36:~# smbpasswd -a pepe

New SMB password: // Establecemos contraseña

Retype new SMB password: // Repetimos la contraseña'''

Added user pepe.
```

Eliminar un usuario de samba

Para eliminar un usuario de samba debemos ejecutar smbpasswd con la opción -x, ejemplo:

```
// Eliminar un usuario de samba
```

```
# smbpasswd -x pepe
```

Inmediatamente el usuario habrá desaparecido de la base de datos de 'usuarios samba' aunque seguirá siendo un usuario de Unix.

Otras opciónes de smbpasswd

- -d: Deshabilitar un usuario
- -e: Habilitar un usuario
- -n: Usuario sin password. Necesita parámetro **null passwords = yes** en sección 'global' del archivo de configuración de samba.
- -m: Indica que es una cuenta de máquina (equipo)

Para más información se puede consultar la página del manual de smbpasswd:

```
// Manual de smbpasswd
     $ man smbpasswd
```

Gestión de grupos y permisos con samba

La gestión de grupos y permisos de usuarios y grupos es sustancialmente diferente en Sistemas Unix y en Sistemas Microsoft Windows.

En los Sistemas Unix, la gestión de los permisos que los usuarios y los grupos de usuarios tienen sobre los archivos se realiza mediante un sencillo esquema de tres tipos de permisos (lectura, escritura y ejecución) aplicables a tres tipos de usuarios (propietario, grupo propietario y resto). Este sencillo esquema se desarrolló en los años 70 y aún hoy resulta adecuado para la gran mayoría de los sistemas en red que podamos encontrar en cualquier tipo de organización, desde pequeñas redes a las más grandes. Es cierto que tiene algunas limitaciones pero la ventaja de ser sencillo hace que su administración sea fácil y su rendimiento muy elevado.

En los Sistemas Microsoft Windows, la gestión de los permisos que los usuarios y los grupos de usuarios tienen sobre los archivos, se realiza mediante un complejo esquema de listas de control de acceso (ACLs = Access Control Lists) para cada carpeta y cada archivo. El sistema de ACLs tiene la ventaja de ser mucho más flexible que el sistema Unix ya que se pueden establecer más tipos de permisos, establecer permisos solo a algunos usuarios y algunos grupos, denegar permisos, etc..., pero como hemos comentado anteriormente, en la mayoría de los casos, con las prestaciones del Sistema Unix es suficiente. En el lado contrario, el sistema de ACLs es más complejo de administrar y más lento ya que antes de acceder a las carpetas o archivos, el sistema debe comprobar listas mientras que en Unix hace una operación lógica de los bits que especifican los permisos lo cual es muchísimo más rápido.

Samba tiene también implementado el sistema de ACLs y se gestiona utilizando el comando smbcacls, no obstante, la recomendación es utilizar el sistema de gestión de permisos de Unix. Aunque existan carpetas compartidas con samba, en última estancia imperan los permisos de Unix. Por ejemplo, si tenemos compartida una carpeta llamada 'profesores' con permisos de escritura para el grupo profesores, todos los usuarios que pertenezcan al grupo profesores podrán realizar cambios en la carpeta, pero si dentro de dicha carpeta existe otra llamada 'confidencial' sobre la cual no tiene permiso para entrar el grupo profesores, ningún profesor podrá ver su contenido aunque esté dentro de una carpeta compartida.

Para realizar una gestión eficaz de usuarios, grupos y permisos, se recomienda utilizar los permisos de Unix que permiten asignar permisos de lectura, escritura y ejecución al usuario propietario del archivo, al grupo propietario del archivo y al resto de usuarios del sistema.

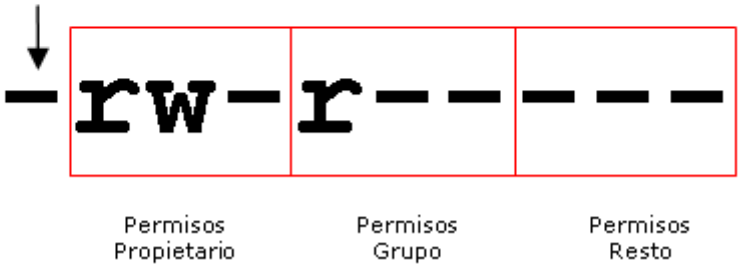
Tipo de archivo:

(-) para archivos normales

(d) para carpetas (directory)

(l) para enlaces (link)

(s)=socket, (p)=tubería (pipe), (b)=dispositivo de bloque.



Todo lo relativo a la administración de usuarios, grupos y permisos en Unix se puede consultar en el apartado 'Usuarios del sistema Unix'. Clic aquí para acceder al índice.

Sobrecarga de permisos

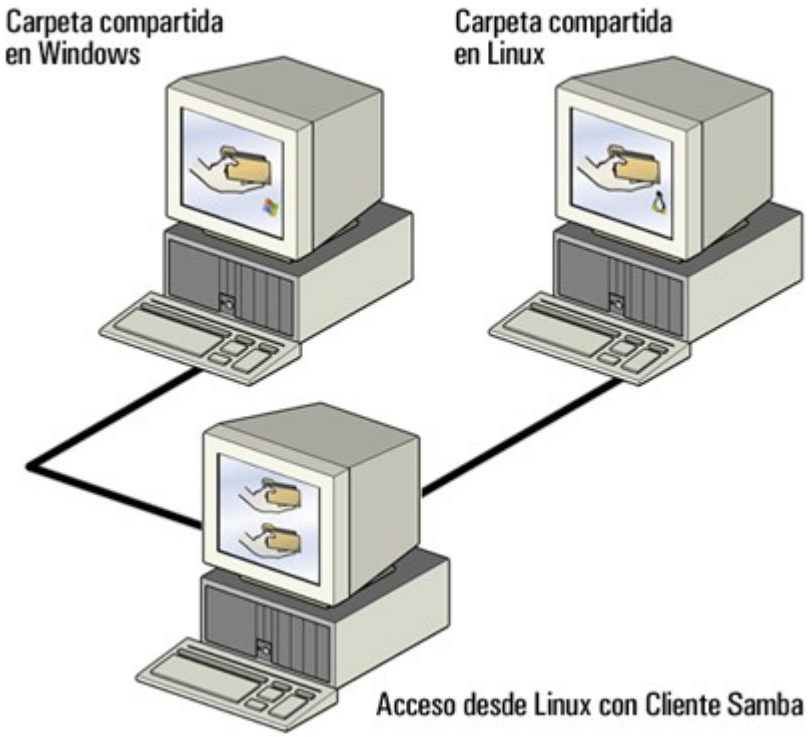
Puede ocurrir que exista contradicción entre los permisos del sistema Unix y los permisos del recurso compartido en **samba**, por ejemplo, podemos tener una carpeta compartida llamada **almacen-software** con permisos Unix de lectura, escritura y ejecución para todos, en cambio si en el archivo de configuración de **samba**, dicho recurso tiene el parámetro **read only = yes**, no será posible realizar cambios ya que está compartido con permiso de 'solo lectura'. Cuando los permisos Unix se contradicen con los permisos **samba**, el permiso efectivo es el más restrictivo de los dos.

Para simplificar la administración de los permisos, se recomienda no ser restrictivos en los permisos de recurso compartido con samba y aplicar los permisos en el Sistema Unix, de ésta forma, además de ser efectivos cuando accedemos a través de **samba**, también lo seguirán siendo si accedemos de otra forma como por **ssh, ftp**, o nos sentamos en la consola del servidor.

Cliente samba

Descripción

Samba dispone de un cliente que permite a PCs con Linux acceder a carpetas compartidas en PCs con windows y PCs Linux con servidor samba.



Instalación

El cliente se encuentra en el paquete smbclient instalable con apt-get:

```
// Instalación del cliente samba
     # apt-get install smbclient
```

Utilización

El cliente se utiliza ejecutando el comando smbclient seguido del nombre del recurso compartido, ejemplo, si deseamos acceder a la carpeta compartida 'alumnos' en 'servidor5', ejecutaremos:

```
// Conectando a un recurso compartido
     $ smbclient //servidor5/alumnos
```

Una vez que accede a la carpeta compartida, es como un cliente de ftp. Podemos ejecutar los comandos típicos del ftp como put, get, ls, cd, etc...

Para averiguar lo que comparte un PC:

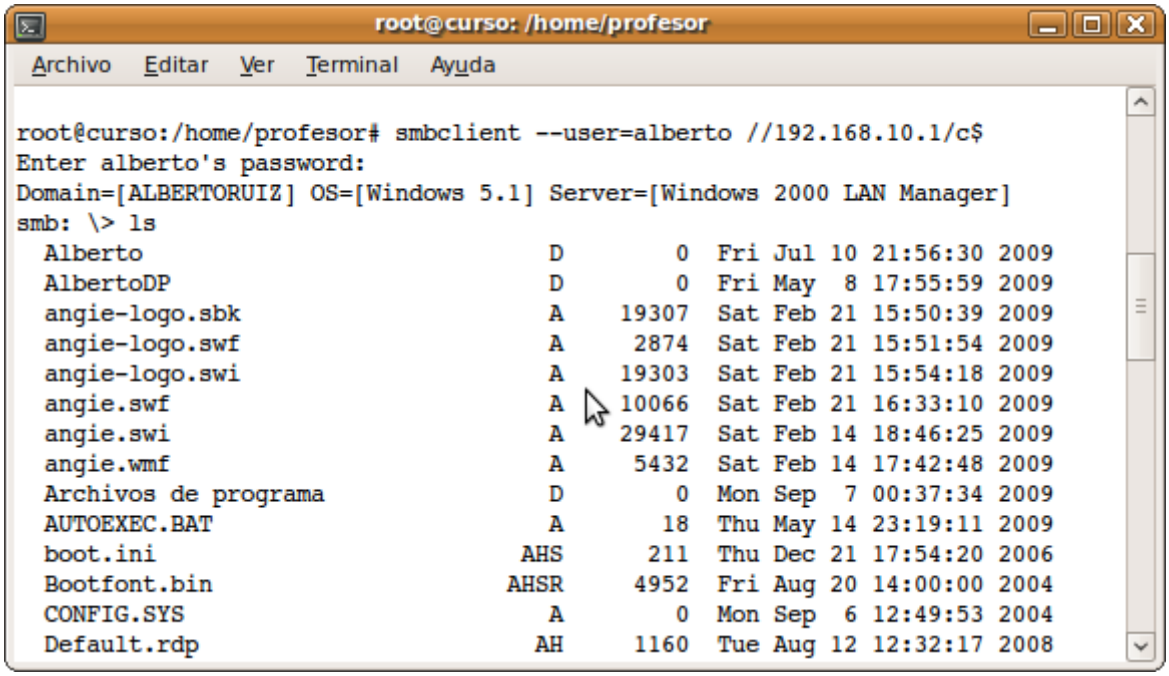
```
// Ver recursos compartidos
     $ smbclient --list servidor5
```

Nos mostraría una lista con todo lo que comparte 'servidor5'. Se puede poner directamente la dirección IP en lugar del nombre del PC. Si está protegido con contraseña, es necesario añadir la opción --user seguida

del nombre de usuario y al ejecutar el comando pedirá la contraseña.

```
// Conectando a un recurso compartido que requiere autentificación
     $ smbclient --user=profesor //servidor5/examenes
```

Ejemplo: supongamos que deseamos acceder a la carpeta compartida c$ en un servidor cuya IP es 192.168.10.1, como usuario alberto. Deberemos ejecutar el comando mostrado en la siguente figura:



Como es un poco engorroso trabajar de esa forma, existe la posibilidad de montar las unidades de red en carpetas de nuestro sistema como si se tratara de una carpeta local. Ejemplo, si queremos acceder desde el **pcprofesor** a una carpeta compartida con el nombre de **profesores** en el servidor, ejecutaremos:

```
// Montar una carpeta compartida sobre nuestro sistema de archivos
     $ smbmount //servidor/profesores /mnt/profesores -o username=juan
     %manzana
```

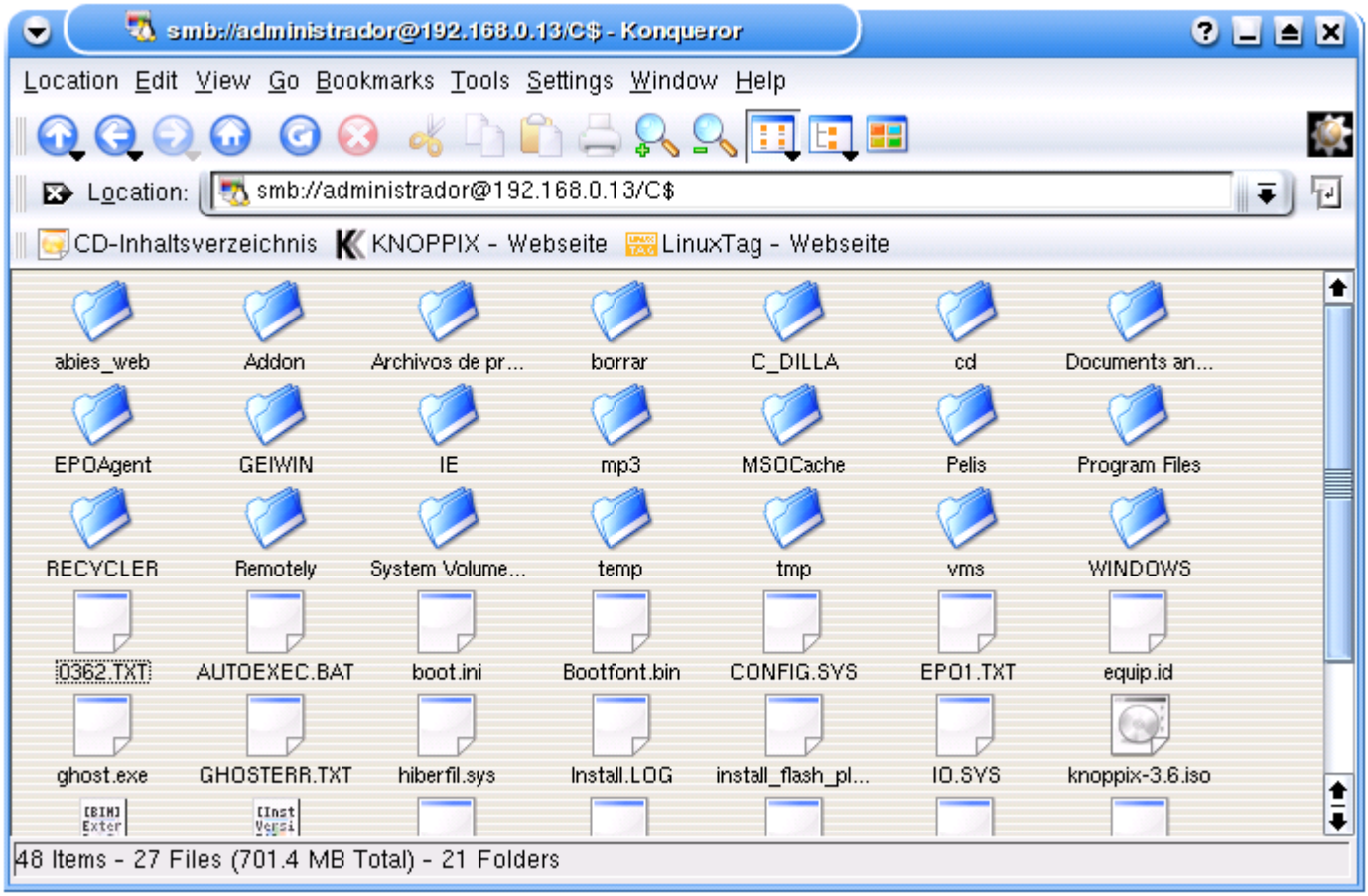
El parámetro '-o' nos permite añadir opciones como en este caso que hemos proporcionado directamente en el comando el nombre de usuario y la contraseña. De no haberlo hecho, hubiera utilizado el nombre del usuario que lanza el comando y nos habría pedido la contraseña. Si deseamos que una carpeta compartida se conecte siempre de forma automática cuando iniciemos nuestro Linux, existe la posibilidad de añadir en el archivo /etc/fstab una línea como por ejemplo:

```
# Montaje automático al iniciar el servidor
     #Añadir en /etc/fstab

     //servidor/profesores /mnt/profesores smbfs
     username=juan,password=manzana
```

Acceso desde Nautilus o desde Konqueror

Konqueror es un navegador con funciones de explorador de archivos para entorno gráfico KDE. Desde Konqueror se puede acceder a carpetas compartidas con samba y a carpetas compartidas en PCs windows, para ello hay que escribir en la barra de direcciones el recurso al que se desea acceder y el usuario con el que se accede siguiendo una sintaxis concreta.

Ejemplo, si deseamos acceder a un recurso de nombre c$ compartido en el servidor cuya IP es 192.168.0.13, con el usuario administrador y contraseña manzana10, escribiremos lo siguiente:
smb://administrador:manzana10@192.168.0.13/c$

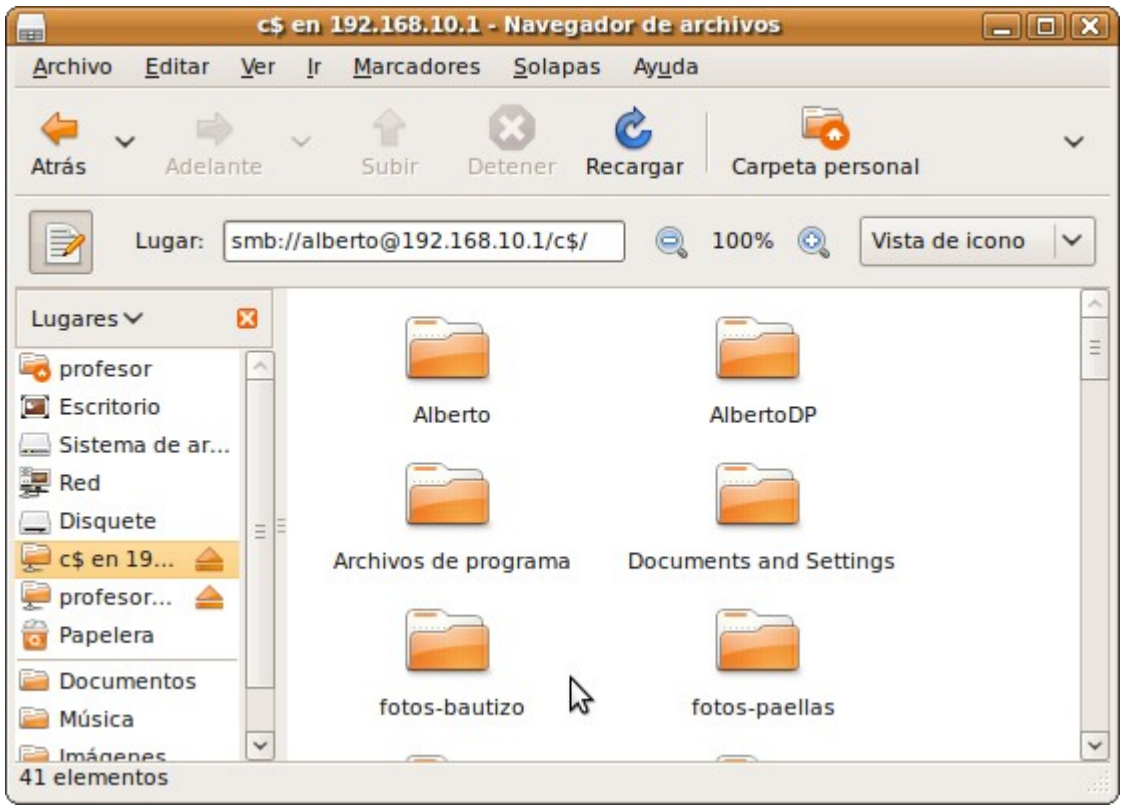


Si no queremos que otras personas la vean en la barra de direcciones, podemos omitir la contraseña. En tal caso nos la pedirá al conectar. También podemos omitir el nombre de usuario. En lugar de escribir la dirección IP, podemos utilizar el nombre del PC.

Truco: Konqueror también se puede utilizar como cliente de ftp, la sintaxis sería la misma:
ftp://juan:manzana@enebro.pntic.mec.es/public_html

Nautilus es un navegador similar a Konqueror, pero para entorno gráfico GNOME. Desde Nautilus también se puede acceder a carpetas compartidas con samba y a carpetas compartidas en PCs windows, para ello hay que pulsar Ctrl+L para que aparezca la barra de direcciones y escrir la ruta del recurso como en el caso de Konqueror. No debemos ejecutar Nautilus como usuario root ya que, por seguridad, quedarán deshabilitadas las funciones de acceso a carpetas en red.

Ejemplo, si deseamos acceder a un recurso de nombre 'datos' compartido en el servidor cuya IP es 192.168.1.244, con el usuario 'administrador', escribiremos lo siguiente:
smb://administrador@192.168.1.244/datos


c$ en 192.168.10.1 - Navegador de archivos
Archivo Editar Ver Ir Marcadores Solapas Ayuda
Atrás
Adelante
Subir
Detener
Recargar
Carpeta personal
Lugar:
smb://alberto@192.168.10.1/c$/
100%
Vista de icono
Lugares
profesor
Escritorio
Sistema de ar...
Red
Disquete
c$ en 19...
profesor...
Papelera
Documentos
Música
Imágenes
Alberto
AlbertoDP
Archivos de programa
Documents and Settings
fotos-bautizo
fotos-paellas
41 elementos

9.- Otros servicios

Instalación del servidor de shell seguro - SSH

El servidor de shell seguro o SSH (Secure SHell) es un servicio muy similar al servicio telnet ya que permite que un usuario acceda de forma remota a un sistema Linux pero con la particularidad de que, al contrario que telnet, las comunicaciones entre el cliente y servidor viajan encriptadas desde el primer momento de forma que si un usuario malintencionado intercepta los paquetes de datos entre el cliente y el servidor, será muy dificil que pueda extraer la información ya que se utilizan sofisticados algoritmos de encriptación.

La popularidad de ssh ha llegado a tal punto que el servicio telnet prácticamente no se utiliza. Se recomienda no utilizar nunca telnet y utilizar ssh en su lugar.

Para que un usuario se conecte a un sistema mediante ssh, deberá disponer de un cliente ssh. Desde la primera conexión, y mediante encriptación asimétrica, las comunicaciones se encriptan incluido el proceso de autentificación del usuario cuando proporciona su nombre y su contraseña. También se proporciona una clave de encriptación simétrica para encriptar las comunicaciones del resto de la sesión mediante encriptación simétrica por su menor necesidad de procesamiento.

Para instalar el servidor y el cliente ssh debemos instalar mediante apt-get el paquete ssh que contiene tanto la aplicación servidora como la aplicación cliente:

```
// Instalación de servidor ssh y cliente ssh
     root@cnice-desktop:# apt-get install ssh
```

Los archivos de configuración son:

- /etc/ssh/ssh_config: Archivo de configuración del cliente ssh
- /etc/ssh/sshd_config: Archivo de configuración del servidor ssh

Arranque y parada manual del servidor ssh

El servidor ssh, al igual que todos los servicios en Debian, dispone de un script de arranque y parada en la carpeta /etc/init.d.

```
// Iniciar o Reiniciar el servidor ssh
     root@cnice-desktop:# /etc/init.d/ssh restart
```

```
// Parar el servidor ssh
     root@cnice-desktop:# /etc/init.d/ssh stop
```

Arranque automático del servidor ssh al iniciar el sistema.

Para un arranque automático del servicio al iniciar el servidor, debemos crear los enlaces simbólicos correspondientes tal y como se indica en el apartado Arranque automático de servicios al iniciar el sistema.

Conexión al servidor mediante ssh

Para conectar desde un PC cliente al servidor mediante ssh, debemos ejecutar el comando ssh seguido del nombre ó dirección IP del servidor. La conexión se realizará con el mismo nombre de usuario que estemos utilizando en el PC cliente. Ejemplo, supongamos que jessica, desde el PC llamado aula5pc3, quiere conectarse al servidor cuya IP es 192.168.1.239:

```
// Conexión por ssh
     jessica@aula5pc3:~$ ssh 192.168.1.239

     The authenticity of host '192.168.1.239 (192.168.1.239)' can't be
     established.

     RSA key fingerprint is
     51:70:3f:9c:ac:49:52:74:88:f5:45:a6:ae:f0:9c:8a.

     Are you sure you want to continue connecting (yes/no)? yes

     Warning: Permanently added '192.168.1.239' (RSA) to the list of known
     hosts.

     Password: // Introducir contraseña de jessica

     jessica@cnice-desktop:~$ // Ya estámos en el servidor
```

La primera vez que se conecte alguien desde dicho PC cliente, se instalará el certificado de autentificación del servidor, lo cual es normal si se trata de la primera vez. A la pregunta 'Are you sure you want to continue connecting (yes/no)?' debemos responder 'yes' ya que de lo contrario la comunicación se cortará. Si ya nos hemos conectado anteriormente otras veces y vuelve a realizar ésta pregunta, significa que alguien se está haciendo pasar por el servidor (nuestro servidor ha sido hackeado) o que se ha reconfigurado el servidor (cambio de nombre, IP, etc...)

Si deseamos conectarnos al servidor utilizando un nombre de usuario diferente, debemos incluir el nombre de usuario antes del nombre o IP del servidor y separado por una arroba '@'. Ejemplo, supongamos que jessica, desde el PC llamado aula5pc3, quiere conectarse como miguel al servidor cuya IP es 192.168.1.239:

```
// Conexión por ssh como otro usuario
     jessica@cliente:~$ ssh miguel@192.168.1.239

     Password: // Introducir contraseña de miguel en el servidor

     miguel@servidor:~$ // Ya estámos en el servidor como miguel
```

Desde PCs con Windows es posible conectarse por ssh a servidores Linux mediante el programa **Putty**. Se trata de un cliente ssh para Windows que permite acceder en modo texto al sistema Linux desde sistemas Windows.

Servicios adicionales

El paquete ssh no solamente nos proporciona conexión remota sino que proporciona otros servicios como:

Ejecución remota de aplicaciones gráficas

Mediante ssh existe la posibilidad de ejecutar aplicaciones gráficas en el servidor y manejarlas y visualizarlas en el cliente. El servidor ssh deberá tener activada la redirección del protocolo X, es decir, deberá tener el siguiente parámetro en el archivo de configuración /etc/ssh/ssh_config:

```
// Habilitar la redirección X en /etc/ssh/sshd_config
     X11Forwarding yes
```

Ejemplo: supongamos que en nuestro terminal tenemos Damn Small Linux (que no dispone del gimp) y deseamos conectarnos a otro PC que sí que tiene instalado el editor gráfico gimp, los pasos que haremos serán:

```
// Ejecutar aplicaciones gráficas
     jessica@cliente:~$ ssh -X cnice@192.168.1.239 // -X para redirigir
     Xwindows.

     cnice@cnice-desktop:~$ gimp // Ejecutamos el gimp
```

El resultado será que desde nuestro Linux sin gimp estamos manejando el gimp que se está ejecutando en el PC remoto:

Desde PCs con Windows es posible conectarse por ssh a servidores Linux de forma gráfica mediante **Cygwin**. Se trata de un conjunto de programas libres que simulan un 'Unix para Windows' con servidor gráfico X y cliente ssh para Windows entre otras cosas, que permite acceder en modo gráfico al sistema Linux desde sistemas Windows. Otros servidores X gratuitos para Windows son Xming y Mocha.

Servidor de ftp seguro

El paquete ssh también incorpora un servidor ftp seguro y un cliente ftp seguro. Para activar el servidor ftp seguro tan solo hay que tener arrancado el servidor ssh.

El cliente ftp seguro es el comando sftp que funciona igual que el comando ftp. También podemos utilizarlo desde el navegador Nautilus escribiendo sftp://nombre-del-usuario@nombre-del-servidor por ejemplo en la url: sftp://profesor@miservidor

Copia remota de archivos

También se dispone de el comando scp que permite copiar archivos desde y hacia el servidor remoto desde el cliente. Ejemplo, si deseamos copiar el archivo /etc/hosts del servidor cuya IP es 192.168.1.239 e identificándonos como jessica en la carpeta actual de nuestro PC, ejecutaremos el siguiente comando:

```
// Copiar un archivo del servidor a nuestro PC
     root@cliente:~# scp jessica@192.168.1.239:/etc/hosts .

     Password: // Introducimos la contraseña de jessica en el servidor

     hosts                    100%  443     0.4KB/s   00:00 // Archivo
     copiado

     root@cliente:~#



// Copiar un archivo de nuestro PC al servidor

// La carpeta de destino debe existir en el servidor
     root@cliente:~# scp miarchivo.txt jessica@192.168.1.239:/home/jessica/
     pruebas/

     Password: // Introducimos la contraseña de jessica en el servidor

     miarchivo.txt                  100%  443     1.6KB/s   00:00 //
     Archivo copiado

     root@cliente:~#



// Copiar una carpeta y subcarpetas de nuestro PC al servidor
```

```
root@cliente:~# scp -r /datos/*.*
jessica@192.168.1.239:/pruebas/datos/

Password: // Introducimos la contraseña de jessica en el servidor

datos/*.*                    100%  443     50.6KB/s   00:03 // Archivos
copiados

root@cliente:~#
```

Desde PCs con Windows es posible utilizar el programa **WinSCP** que permite copiar archivos desde y hacia el servidor. Se trata de un cliente que utiliza el protocolo ssh para acceder al sistema de archivos del servidor Linux desde sistemas Windows.

Identificación por certificado

Para evitar tener que introducir continuamente la contraseña cuando deseamos conectar con un servidor remoto por ssh, existe la posibilidad de autentificarse por certificado, para ello debemos:

1. Crear un certificado de usuario en el PC cliente
2. Copiar el certificado en el PC servidor

Para que el servidor ssh acepte la autentificación por medio de certificado, deberá tener activada la opción PubkeyAuthentication yes, es decir, deberá tener el siguiente parámetro en el archivo de configuración /etc/ssh/sshd_config:

```
// Permitir autentificación por certificado
     PubkeyAuthentication yes
```

Crear un certificado en el PC cliente

Para crear un certificado que permita autentificar al usuario, debemos ejecutar el comando ssh-keygen. Dicho comando creará dentro de nuestra carpeta home, en una carpeta llamada '.ssh', dos archivos: uno llamado id_rsa que será la clave privada de nuestro certificado y otro llamado id_rsa.pub que será la clave pública de nuestro certificado. Éste último archivo será el que hay que copiar en el servidor remoto.

```
// Creación de un certificado
     miguel@cliente:~$ ssh-keygen -t rsa

     Generating public/private rsa key pair.

     Enter file in which to save the key (/home/miguel/.ssh/id_rsa):

     // Archivo del certificado. Podemos dejar el que viene por defecto

     Created directory '/home/miguel/.ssh'.

     Enter passphrase (empty for no passphrase): // Opcional

     Enter same passphrase again:

     Your identification has been saved in /home/miguel/.ssh/id_rsa.

     Your public key has been saved in /home/miguel/.ssh/id_rsa.pub.
```

```
The key fingerprint is:

c8:a4:fe:0c:19:78:8e:7d:05:5b:13:df:37:17:e8:ea
miguel@dsl.ieslapaloma.com

miguel@dsl:~$
```

Copiar el certificado en el PC servidor

Para poder identificarse en el servidor como miguel desde el cliente, debemos copiar el archivo id_rsa.pub que hemos creado en el cliente, en la carpeta home de miguel en el servidor dentro de una carperta llamada '.ssh' en un archivo llamado authorized_keys. Para copiar dicho archivo del cliente al servidor, podemos hacerlo con scp. Supongamos que el cliente se llama 'cliente' y el servidor se llama 'servidor':

```
// Copia del certificado y prueba de la conexión
// Nota: el símbolo ~ en Linux es la carpeta home del usuario
     miguel@cliente:~$ scp ~/.ssh/id_rsa.pub
     miguel@servidor:~/.ssh/authorized_keys

     Password:  // Va a ser la última vez que introduzcamos la contraseña

     id_rsa.pub                 100%  242     0.2KB/s   00:00 // Copiado

     miguel@cliente:~$ ssh miguel@servidor // Probamos la conexión

     miguel@servidor:~$ // Ya estamos en el servidor sin necesidad de
     contraseña
```

Instalación y configuración de PHP

Introducción

PHP es, junto con mysql, el complemento ideal del servidor web apache ya que dota al servidor de un lenguaje script de ejecución en el servidor lo que facilita la creación de aplicaciones web y sitios web dinámicos.

Instalación de PHP

Para instalar PHP en nuestro servidor podemos utilizar apt-get. El paquete a instalar depende de la versión que deseemos instalar y la versión de apache. Lo normal es que utilicemos la versión 2 de apache y que instalemos la versión 5 de php. En tal caso deberíamos instalar libapache2-mod-php5:

```
// Instalación de php5 para apache 2
     # apt-get install libapache2-mod-php5
```

Al instalar libapache2-mod-php5 mediante apt-get, automáticamente se configura para integrarse perfectamente en apache, creando los archivos necesarios en la carpeta de módulos disponibles de apache

(/etc/apache2/mods-available) y creando los enlaces necesarios para habilitarlos en la carpeta de módulos habilitados de apache (/etc/apache2/mods-enabled).

Si vamos a conectar a bases de datos mysql desde php, necesitamos instalar el módulo php5-mysql:

```
// Instalación del módulo php5-mysql
     # apt-get install php5-mysql
```

Además, tendremos que editar el archivo /etc/php5/apache2/php.ini y añadir la línea **extension=mysql.so** como veremos en el siguiente apartado.

Configuración de PHP

El archivo de configuración de php5 es el archivo:

```
// Archivo de configuración de php5
     /etc/php5/apache2/php.ini
```

Los parámetros más destacables a configurar son:

- **Safe Mode = Off** (Modo Seguro. Si el Modo seguro está desactivado, se habilitan todas las funciones del php. Para un uso educativo es mejor ser funcional y no activar el modo seguro. Si el Modo seguro está activado, se deshabilitan todas las funciones del php consideradas peligrosas. Para servicios de hosting se recomienda activar el modo seguro.)
- **Display errors = On** (Mostrar Errores. Muestra los errores en las mismas páginas, cuando les haya. Cuando hay errores en los scritps, es más fácil encontrarlos si se muestran en las páginas)
- **max_execution_time=30** (Tiempo máximo en segundos, de ejecución de un script)
- **post_max_size=8M** (Tamaño máximo de datos que se pueden enviar al servidor mediante POST)
- **upload_max_filesize = 8M** (Tamaño máximo de archivo que se puede subir al servidor)
- **extension=mysql.so** (Activa el acceso a bases de datos mysql desde php)

Probando PHP

Una vez instalado y configurado, antes de probar debemos reiniciar el servidor web apache:

```
// Reiniciando apache
     /etc/init.d/apache restart
```

Ahora crearemos una página php que utilice la función phpinfo que además de comprobar que apache y php están funcionando, nos mostrará una información de la versión. Crearemos el siguiente archivo:

```
// Probando PHP. Crear archivo /var/www/phpinfo.php - permisos 644
     <HTML>

          <H1>Probando PHP</H1>

          Salida del comando phpinfo:



          <?

               phpinfo();
```

```
?>


</HTML>
```

Ahora tan solo necesitamos arrancar el navegador e ir a la URL: http://ip-del-servidor/phpinfo.php. Si nos aparece la información de la versión de php significa que está correctamente instalado.

Instalación y configuración de MySQL

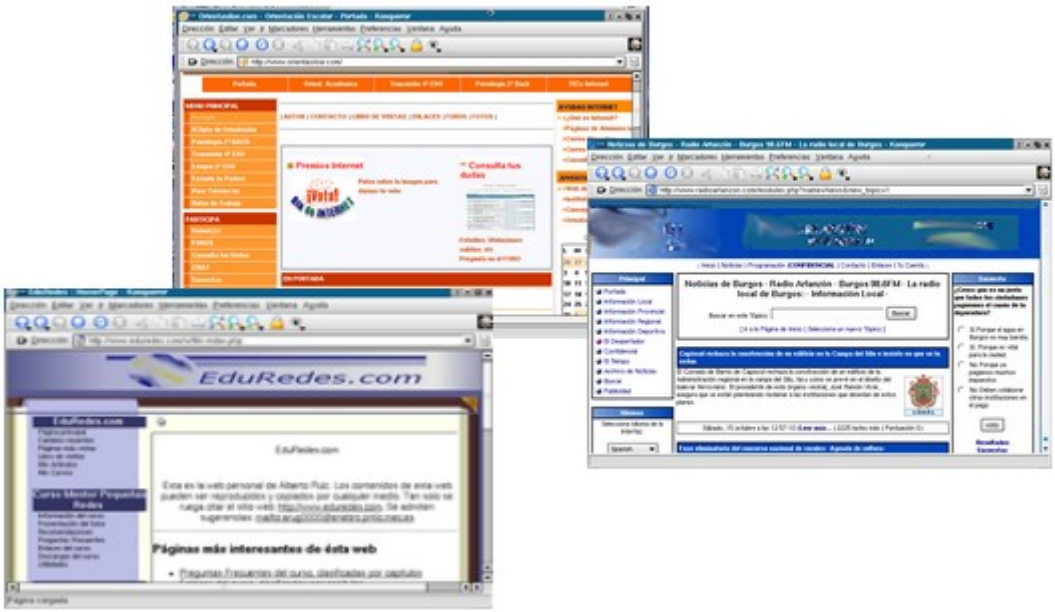
Introducción

MySQL es un SGBD (Sistema Gestor de Bases de Datos) relacionales muy completo y muy utilizado tanto en entornos Linux como en entornos Windows, principalmente para el desarrollo de aplicaciones web. Entre sus principales prestaciones destacamos:

- Fácil instalación
- Fácil administración
- Rápido
- Completo
- Multiplataforma

Por éstas razones, casi todas las aplicaciones web desarrolladas en lenguaje php que requieran de base de datos, utilizan mysql.

Si disponemos de un servidor web con soporte php y base de datos mysql, tendremos la arquitectura ideal para crear un portal dinámico utilizando gestores de contenidos como PHPNuke, drupal o Tikiwiki y herramientas orientadas a crear sitio web para entornos educativos como Mambo o Claroline, así como aplicaciones web orientadas al trabajo colaborativo y al desarrollo rápido de contenidos como Wikis y Blogs.



Instalación de mysql

Para la instalación del servidor y el cliente de mysql, debemos instalar los paquetes mysql-server, mysql-common y mysql-client mediante apt-get. Se instalará la versión 5 de mysql:

```
// Instalación de mysql
     # apt-get install mysql-server mysql-common mysql-client
```

Arranque y parada del SGBD mysql

El servidor de datos mysql, al igual que todos los servicios en Debian, dispone de un script de arranque y parada en la carpeta /etc/init.d.

```
// Iniciar o reiniciar el servidor mysql
     # /etc/init.d/mysql restart



// Parar el servidor mysql
     # /etc/init.d/mysql stop
```

Arranque automático del servidor MySQL al iniciar el sistema.

Para un arranque automático del servicio al iniciar el servidor, debemos crear los enlaces simbólicos correspondientes tal y como se indica en el apartado Arranque automático de servicios al iniciar el sistema.

Configuración del SGBD mysql

El archivo de configuración de mysql es el archivo:

```
// Archivo de configuración de mysql
     /etc/mysql/my.cnf
```

En dicho archivo se configuran aspectos generales como la ruta donde se almacenarán los archivos de la base de datos, el puerto a utilizar y algún otro aspecto pero para hacer un uso normal de mysql, no es necesario realizar ninguna modificación del archivo original.

Administración del SGBD mysql

Mysql es un SGBD completo que permite crear usuarios y establecer permisos sobre bases de datos, tablas y campos deseados a dichos usuarios. Los permisos pueden ser de consulta, inserción, modificación y borrado de datos, creación, modificación y eliminación de tablas y bases de datos y de administración de usuarios y permisos, lo que hace a mysql ser un SGBD muy flexible y muy completo.

Quizás la primera acción que se debería hacer nada más arrancar el SGBD mysql sería poner una contraseña al usuario root ya que inicialmente no tiene contraseña. Para ello debemos iniciar mysql con el comando:

```
// Iniciar el servidor de bases de datos mysql
     # /etc/init.d/mysql start
```

Posteriormente iniciamos el cliente de mysql como root y cuando aparezca el prompt de mysql (mysql>) ejecutamos una orden grant para establecer la contraseña de root:

```
// Ejecutar cliente de mysql y cambiar contraseña de root
     # mysql -u root    // Accedemos sin contraseña

     Welcome to the MySQL monitor.  Commands end with ; or \g.

     Your MySQL connection id is 5 to server version: 4.0.20-log



     Type 'help;' or '\h' for help. Type '\c' to clear the buffer.



     mysql> grant all privileges on *.* to root@localhost identified by
     'secreta' with grant option;

     Query OK, 0 rows affected (0.00 sec)



     mysql> quit

     Bye
```

De ésta manera habremos puesto como contraseña de root la palabra 'secreta'. La próxima vez que entremos, debemos añadir la opción -p para que nos pida la contraseña ya que de lo contrario no nos dejará entrar:

```
// Entrando como root con contraseña
     # mysql -u root -p
```

Ahora debemos introducir la contraseña para acceder y tener acceso global al SGBD mysql.

Administración del SGBD mysql vía web

La herramienta de administración de mysql vía web es phpmyadmin. Para más información consultar el apartado Instalacion_y_configuracion_de_PHPMyAdmin

Instalación y configuracion de phpmyadmin

Introducción.

Phpmyadmin es una excelente herramienta de administración de mysql vía web. Para poder utilizar phpmyadmin se requiere disponer de un servidor web con soporte php.

La herramienta permite que cualquier usuario de la base de datos que disponga de algún permiso, acceda y haga uso de dichos permisos. Identificándose con el usuario administrador de la base de datos (root) o con cualquier otro usuario que disponga de todos los privilegios, estarán habilitadas todas las características de

la herramienta.

Instalación de phpmyadmin

La instalación de phpmyadmin se puede realizar de forma automática con apt-get, pero se trata de un paquete que no se encuentra en el 'repositorio principal' (main) de ubuntu, sino que se encuentra en el 'repositorio universo' (universe). Para que apt-get pueda instalar paquetes del repositorio universo, es necesario editar el archivo /etc/apt/sources.list y quitar la almohadilla de las líneas:

```
// Quitar almohadilla delante de las líneas, para descomentarlas
deb http://es.archive.ubuntu.com/ubuntu/ dapper universe
deb-src http://es.archive.ubuntu.com/ubuntu/ dapper universe
```

Posteriormente, tendremos que actualizar en nuestro PC, el contenido de los repositorios:

```
// Actualizar contenido de los repositorios
     # apt-get update
```

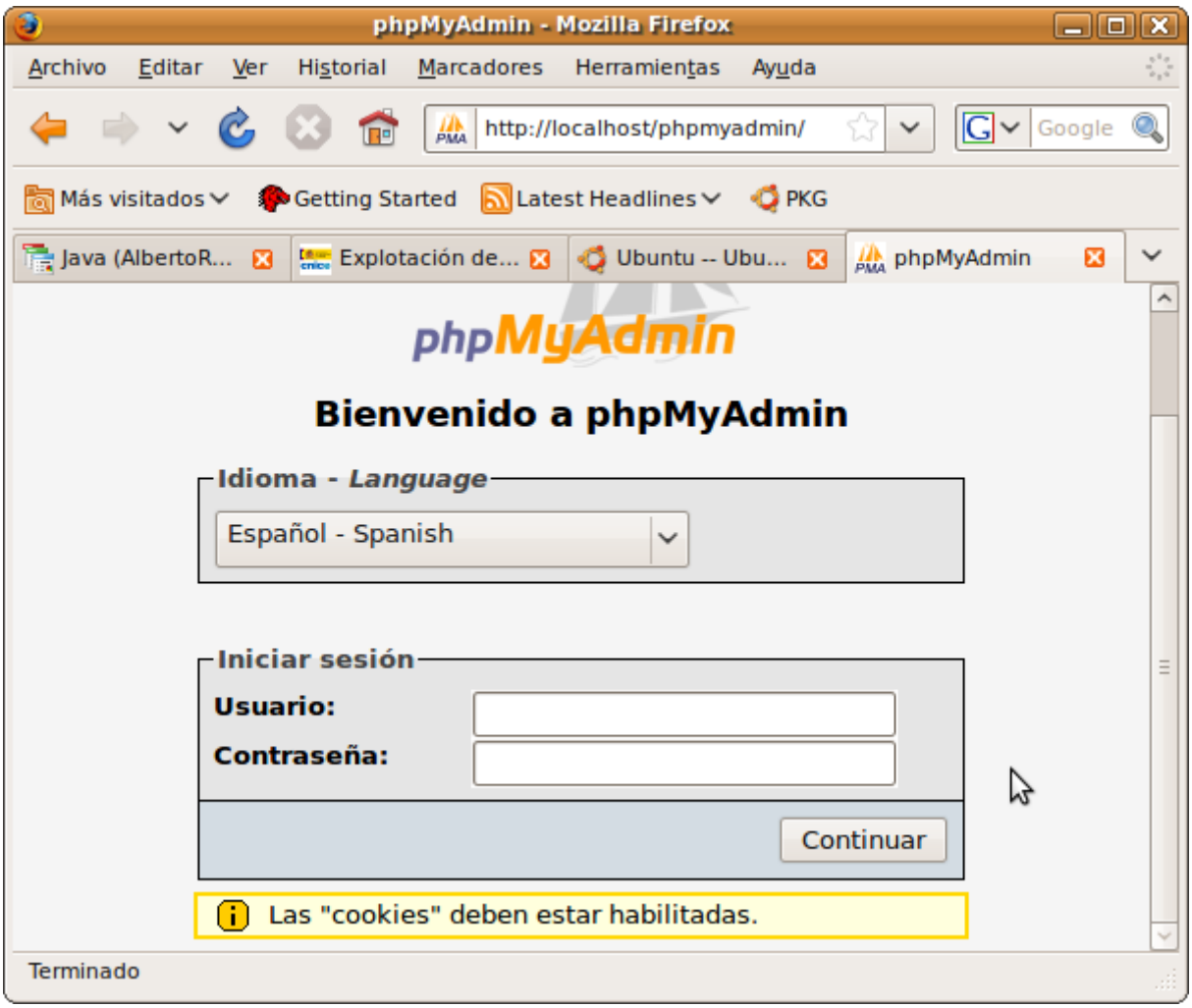
Ahora sí, podremos instalar el paquete 'phpmyadmin' mediante el comando:

```
// Instalación de phpmyadmin
     # apt-get install phpmyadmin
```

El programa de instalación crea un enlace simbólico en el DocumentRoot del servidor web para que la aplicación pueda ser accesible desde la url: http://ip-del-servidor-web/phpmyadmin/index.php. Si no se viera la aplicación en dicha url, quizás sea por algún aspecto de la configuración de apache. En tal caso, lo más sencillo sería mover la carpeta de phpmyadmin directamente dentro del DocumentRoot del servidor y asignar al usuario www-data que es el usuario con el que se ejecuta el apache, para que apache pueda acceder a dicha carpeta:

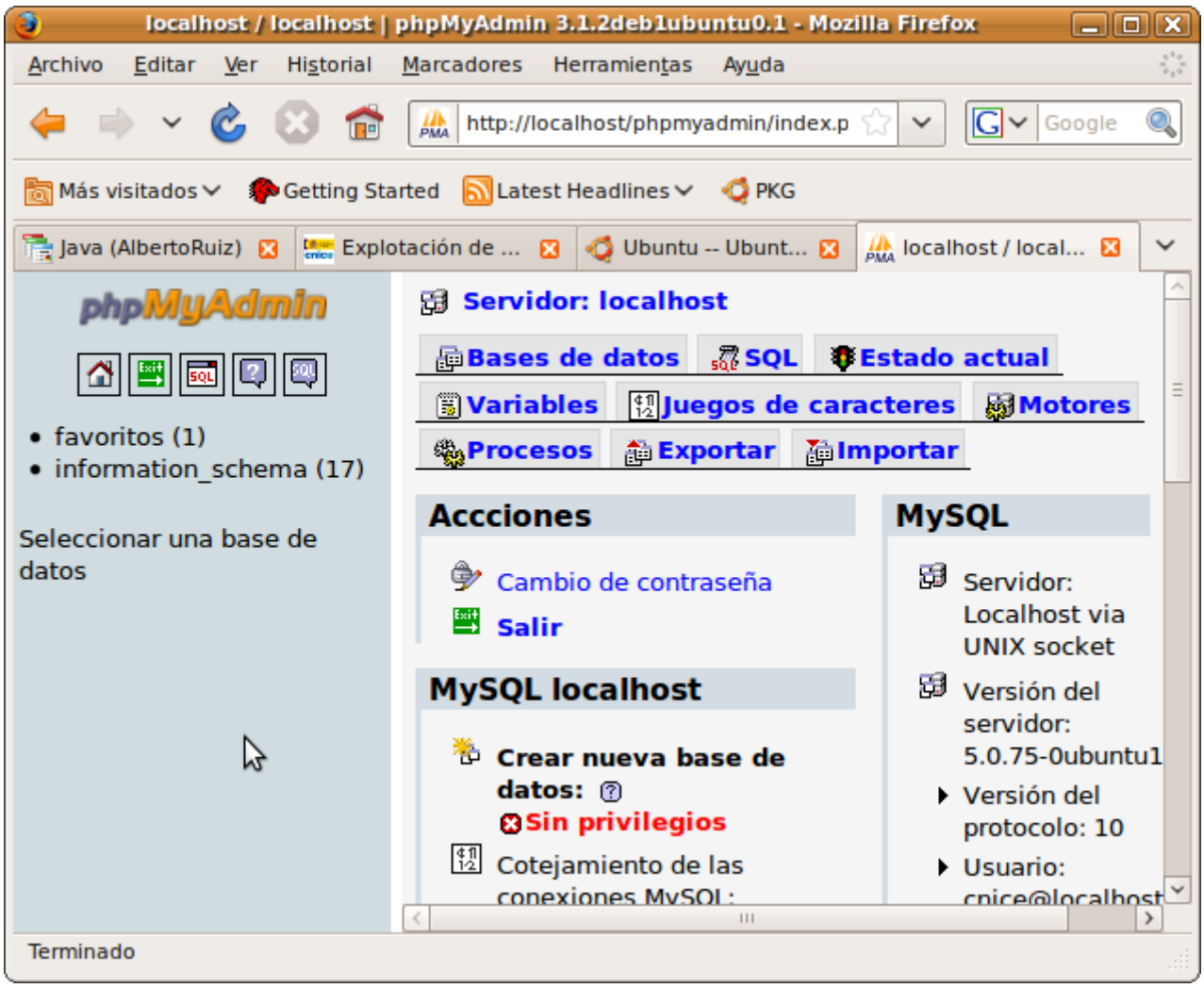
```
// Colocar phpmyadmin en el servidor web y asignar propietario a www-data
     # mv /usr/share/phpmyadmin /var/www/ (en nuestro caso)

     # chown -R www-data /var/www/phpmyadmin
```

De ésta forma, es seguro que accediendo a http://ip-del-servidor-web/phpmyadmin/index.php nos aparecerá la página de autentificación de phpmyadmin:



Página de autentificación de phpmyadmin

Una vez que nos identifiquemos con un usuario y contraseña válidos, accederemos a la página principal de phpmyadmin desde la que podremos crear una nueva base de datos o realizar consultas y modificaciones sobre bases de datos ya existentes:



Página principal de phpmyadmin

Configuración de phpmyadmin

El archivo de configuración de phpmyadmin es el archivo:

```
// Archivo de configuración de phpmyadmin
     /etc/phpmyadmin/config.inc.php
```

En dicho archivo de configuración hay que establecer los parámetros que permitirán a phpmyadmin conectar con mysql, que son:

host

En este parámetro habrá que indicar la IP del servidor mysql. Si el servidor web y el servidor mysql son la misma máquina, se deberá poner 'localhost' ó 127.0.0.1. En el caso de que sean máquinas diferentes, se deberá poner la IP del servidor mysql.

port

Aquí se especifica el puerto de conexión al servidor mysql. El puerto por defecto por el que sirve datos el servidor mysql es el 3306. Si en la configuración de mysql (archivo /etc/mysql/my.cnf) no se ha cambiado, no es necesario especificarlo ya que se usará el puerto 3306 por defecto.

auth_type

Para que phpmyadmin pueda acceder a mysql, es necesario autentificarse. Se admiten tres formas de autentificación:

- **config:** permite que el nombre de usuario y la contraseña se especifiquen en el archivo config.inc.php
- **http:** el usuario deberá introducir nombre y contraseña para acceder a la ruta web
- **cookie:** el usuario deberá introducir nombre y contraseña para acceder a la aplicación

user y password

En el caso de que hayamos elegido tipo de autentificación 'config', será necesario proporcionar el nombre de usuario y la contraseña con el que phpmyadmin accederá a mysql. En tal caso, la línea del password puede quedar comentada.

```
// Configuración por defecto en config.inc.php
     Corresponden a las siguentes líneas en el archivo config.inc.php:

     $cfg['Servers'][$i]['host']          = 'localhost'; // MySQL hostname
     or IP

     $cfg['Servers'][$i]['port']          = '';          // MySQL port-
     blank default port

     $cfg['Servers'][$i]['auth_type']     = 'cookie';    // Authentication
     method

     $cfg['Servers'][$i]['user']          = 'root';      // MySQL user

     $cfg['Servers'][$i]['password']      = '';   // MySQL pass (only
     'config')
```

Práctica: Mis Favoritos on line

En la siguiente práctica pondremos en marcha una aplicación web basada en php y mysql. Dicha aplicación la hemos bautizado como **Mis Favoritos on-line** y permitirá añadir, eliminar y visualizar mis direcciones de Internet favoritas. Los datos se almacenarán en una base de datos mysql.

Paso 1: Creación de la base de datos

Crearemos en mysql una base de datos llamada 'favoritos' y nos situaremos en ella.

```
// Crear base de datos
     mysql> CREATE DATABASE favoritos;

     Query OK, 1 row affected (0.00 sec)



     mysql> USE favoritos;

     Database changed
```

Paso 2: Creación de la tabla

Crearemos una tabla llamada 'favoritos' en la base de datos 'favoritos'.

```
// Crear tabla (ojo: usar comilla simple invertida en nombres)
     mysql> CREATE TABLE `favoritos` (

     `numero` INT NOT NULL PRIMARY KEY ,

     `url` VARCHAR( 50 )  ,

     `descripcion` VARCHAR( 50 )

     );

     Query OK, 0 row affected (0.02 sec)



// Insertamos un registro en la tabla para que no esté vacía:
     mysql> INSERT INTO `favoritos` ( `numero` , `url` , `descripcion` )

     VALUES ('1', 'http://www.pntic.mec.es', 'Web del cnice');

     Query OK, 1 row affected (0.00 sec)
```

Paso 3: Asignación de permisos a un usuario

Crearemos un usuario en mysql asignandole permisos sobre nuestra base de datos. Nuestro usuario se llamará 'cnice' y su contraseña será 'cnice'. Se podrá conectar desde el mismo equipo (localhost) y tendrá todos los privilegios sobre todas las tablas de nuestra base de datos 'favoritos':

```
// Conceder permisos
     mysql> GRANT ALL ON favoritos.* to cnice@localhost identified by
     "cnice";

     Query OK, 0 row affected (0.01 sec)

     mysql> flush privileges; // actualizar permisos

     Query OK, 0 row affected (0.01 sec)
```

Paso 4: Creación del script

Crearemos un script en php que accederá a mi base de datos y permitirá al usuario insertar registros y consultar el contenido de la tabla.

```
// Script PHP para el acceso a datos MySQL

    <?

    ////////////////////////////////////////////////////////////////////////
    ////////

    //

    //      MisFavoritos on-line. (C) 2007 - CNICE.

    //      Nombre del script: index.php

    //

    ////////////////////////////////////////////////////////////////////////
    ////////

    // Parámetros de conexión con la base de datos

    define( "DB_HOST",      "localhost" );

    define( "DB_USER",      "cnice" );

    define( "DB_PASSWD",    "cnice" );

    define( "DB_NAME",      "favoritos" );

    error_reporting( 0 ); //Para que no muestre warnings ni errores

    ?>

    <HTML>

        <HEAD><TITLE> Favoritos - mysql </TITLE></HEAD>

        <H1>Favoritos</H1>

        <STYLE type="text/css">

        <!--A {font-family: Arial; color: #00FF00}-->

        </STYLE></HEAD>

        <BODY>
```

```
    Favoritos on-line. Acceso a datos mysql desde páginas PHP.<HR>

    Elija la operación que desee efectuar:<BR>

    <TABLE BORDER>

           <TD><A HREF="index.php?Accion=Ayuda">Ayuda</A></TD>

           <TD><A HREF="index.php?Accion=Nuevo">Nuevo</A></TD>

           <TD><A HREF="index.php?Accion=Borrar">Borrar</A></TD>

    </TABLE>

    <HR>


<?

$idCon = mysql_connect( DB_HOST, DB_USER, DB_PASSWD ) or die( "Error
en la conexión: " . mysql_error());

mysql_select_db( DB_NAME, $idCon );

echo "<TABLE BORDER>";

echo "<TR><TD>Número</TD><TD>URL</TD><TD>Descripción</TD></TR>";

//Mostramos el contenido de la tabla

$cSql = "SELECT * FROM favoritos";

$idQry = mysql_query( $cSql, $idCon );

while ($idRec = mysql_fetch_array( $idQry )) {

printf('<TR><TD>%s</TD><TD><A HREF="%s">%s</A></TD><TD>%s</TD></TR>',
$idRec[0], $idRec[1], $idRec[1], $idRec[2]);

}

echo "</TABLE>";



$Accion= $_GET['Accion'];

$numero=$_GET['numero'];

$url=$_GET['url'];

$descripcion=$_GET['descripcion'];
```

```
switch ($Accion)

    {

//--------------------------------- Visualizar Ayuda -------

      case ('Ayuda'):

         echo "<HR>Las acciones disponibles son: <BR> <BR>";

         echo "<TABLE BORDER>";

         echo "<TR><TD>Ayuda: Muestra éste mensaje de ayuda
</TD></TR>";

         echo "<TR><TD>Nuevo: Crea un nuevo registro </TD></TR>";

         echo "<TR><TD>Borrar: Elimina un registro </TD></TR>";

         echo "</TABLE>";

         echo "<HR>";

         break;



//------------------------------------------ Nuevo -------

      case ('Nuevo'):

        //Si no enviamos una url, pintamos el formulario

        if (isset($url)==FALSE){

                echo '<FORM METHOD="GET" ACTION="index.php">';

                echo '<INPUT TYPE=HIDDEN NAME="Accion"
VALUE="Nuevo">';

                echo 'Num.: <INPUT TYPE="text" NAME="numero"><BR>';

                echo 'URL: <INPUT TYPE="text" NAME="url"><BR>';

                echo 'Descripción: <INPUT TYPE="text"
NAME="descripcion">';

                echo '<INPUT TYPE="Submit" VALUE="Insertar
Registro">';

                echo '</FORM>';
```

```
        }

        else{

                $cSql = "INSERT INTO favoritos values
($numero,'$url','$descripcion')";

                $idQry = mysql_query( $cSql, $idCon );



                if ($idQry==FALSE){

                        echo "Error al añadir un registro";

                }

                else{

                        echo "<BR>Registro $numero, $url, $descripcion
añadido satisfactoriamente.";

                        echo 'Clic <a href="index.php">aquí</a> para
refrescar.';

                }



      }

      break;

//----------------------------------------- Borrar -------

      case ('Borrar'):

        //Si no enviamos un NumReg, pintamos el formulario

        if (isset($numero)==FALSE){

                echo '<FORM METHOD="GET" ACTION="index.php">';

                echo '<INPUT TYPE=HIDDEN NAME="Accion"
VALUE="Borrar">';

                echo 'Introduzca Número de Registro a eliminar: <INPUT
TYPE="text" NAME="numero">';

                echo '<INPUT TYPE="Submit" VALUE="Aceptar">';

                echo '</FORM>';

        }
```

```
        else{

                //Primero comprobamos si existe dicho registro

                $cSql = "SELECT * FROM favoritos WHERE numero =
$numero";

                $idQry = mysql_query( $cSql, $idCon );

                $nfilas = mysql_num_rows($idQry);

                echo "Filas=$nfilas";

                //Si no existe, no podemos borrarlo

                if ($nfilas==0){

                        echo "No existe el registro $numero. Imposible
eliminar dicho registro.";

                }

                else{

                        $cSql = "DELETE FROM favoritos WHERE numero =
$numero";

                        $idQry = mysql_query( $cSql, $idCon );

                        if ($idQry==FALSE) echo "Error al eliminar el
registro $numero";

                        else{

                                echo "Registro $numero eliminado
satisfactoriamente.";

                                echo 'Clic <a
href="index.php">aquí</a> para refrescar.';

                        }

                }



        }

        break;



}

?>
```

```
    </BODY>

</HTML>
```

Paso 5: Prueba del script

Una vez creado el script, debemos subirlo a nuestro servidor dentro del 'Raíz de documentos' del servidor web y acceder desde el navegador. Ejemplo, si dentro de nuestra carpeta /var/www hemos creado una carpeta llamada 'favoritos' y hemos subido el script con el nombre 'index.php', para probarlo debemos poner en el navegador: http://ip-del-servidor/favoritos/index.php

Instalación y configuración de proftpd

Introducción

Proftpd es un servidor de ftp rápido, de fácil instalación y fléxible configuración con un esquema similar a la configuración de apache. Además permite diferentes posibilidades de autentificación: mediante usuarios del sistema unix, mediante base de datos mysql o autentificación mediante servidor LDAP.

Instalación de proftpd

Proftpd se puede instalar automáticamente mediante apt-get:

```
// Instalación de proftpd
     # apt-get install proftpd
```

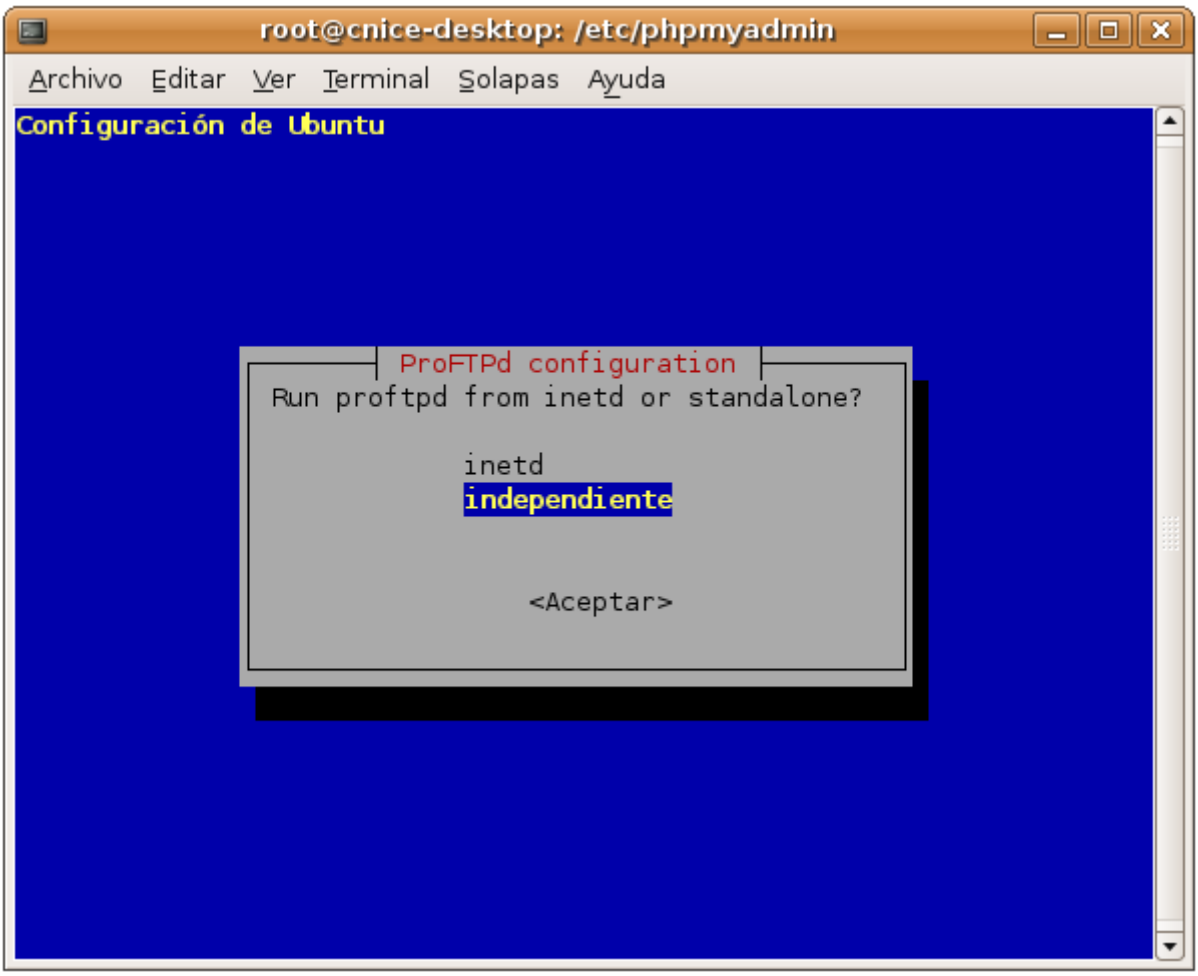
Si disponemos de un servidor LDAP, podemos instalar la versión apta para ldap 'proftpd-ldap'.

Configuración de proftpd

Al instalar el paquete proftpd-ldap se iniciará el asistente de configuración de proftpd. Si más adelante deseamos lanzar de nuevo el asistente, debemos ejecutar:

```
// Lanzar el asistente de configuración de proftpd
     # dpkg-reconfigure proftpd
```

Este asistente únicamente nos hace una pregunta que es si deseamos ejecutar el servidor desde inetd (solo se carga en memoria cuando existan peticiones) o como un servicio independiente (permanentemente en memoria). El funcionamiento como servicio independiente es más eficiente.



El archivo de configuración de proftpd es el archivo:

```
// Archivo de configuración de proftpd
     /etc/proftpd.conf
```

No es necesario modificar ningún parámetro del archivo /etc/proftpd.conf para un uso normal del servidor ftp en el centro educativo. Con solo arrancar el servidor ftp, debería funcionar.

```
// Arranque del servidor ftp
     # /etc/init.d/proftpd restart
```

Para que proftpd arranque automáticamente al iniciar el servidor, debemos crear los enlaces simbólicos correspondientes tal y como se indica en el apartado Arranque automático de servicios al iniciar el sistema.

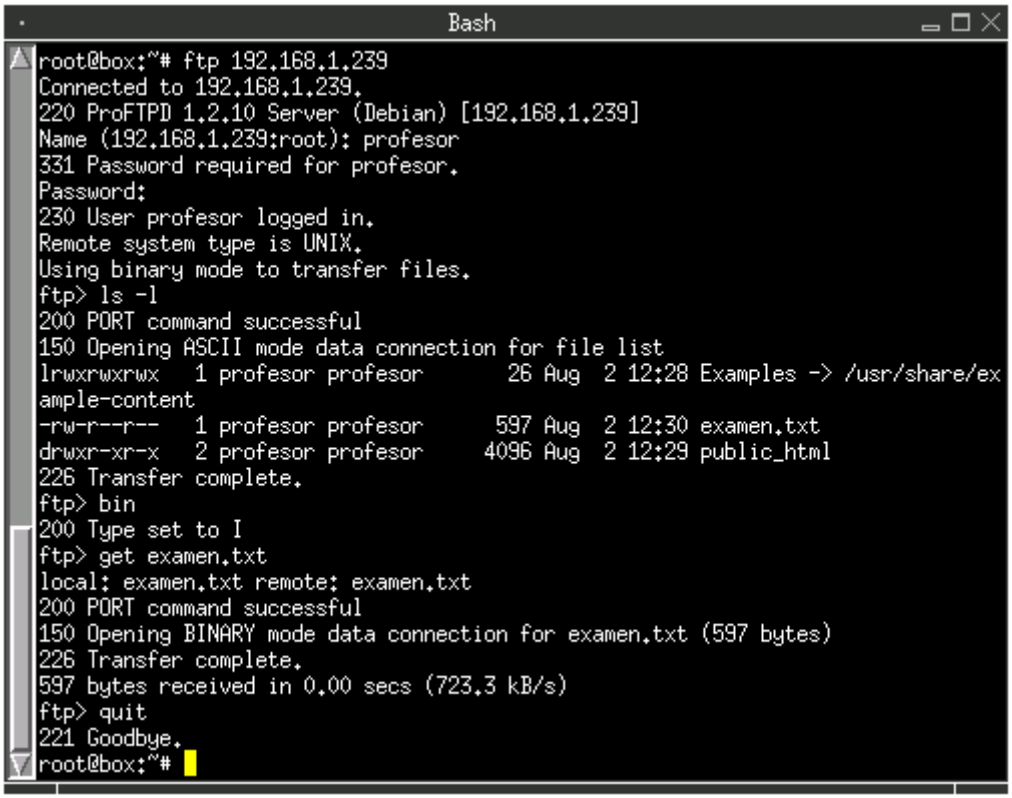
Configuración de proftpd-ldap

Si en la red disponemos de un servidor LDAP y nuestro sistema está configurado para autentificarse por LDAP (ver capítulo OpenLDAP > Autentificación del sistema con OpenLDAP), podemos instalar proftpd-ldap para que proftpd autentifique a los usuarios contra nuestro servidor LDAP. Para ello, es necesario configurar tres parámetros: quien es el servidor LDAP (LDAPServer), cual es el usuario administrador y la contraseña de LDAP (LDAPDNInfo) y qué unidad organizativa tiene la información de los usuarios (LDAPDoAuth). También configuraremos como máscara de creación de archivos y carpetas la máscara 002 porque utilizamos grupos privados de usuario:

```
// Parámetros destacables a configurar
     # Para que autentifique contra nuestro servidor LDAP

     AuthPAM on

     LDAPServer localhost

     LDAPDNInfo cn=admin,dc=ieslapaloma,dc=com ldapadmin // ldapadmin =
     contraseña

     LDAPDoAuth on "ou=users,dc=ieslapaloma,dc=com"



     # Permisos recomendados para UPG (grupos privados de usuario)

     Umask                           002  002
```

Práctica - Probar el servidor ftp

Para probar que está funcionando el servidor ftp intentaremos entrar con el usuario profesor y una vez dentro descargaremos un archivo desde el servidor ftp a nuestro PC:



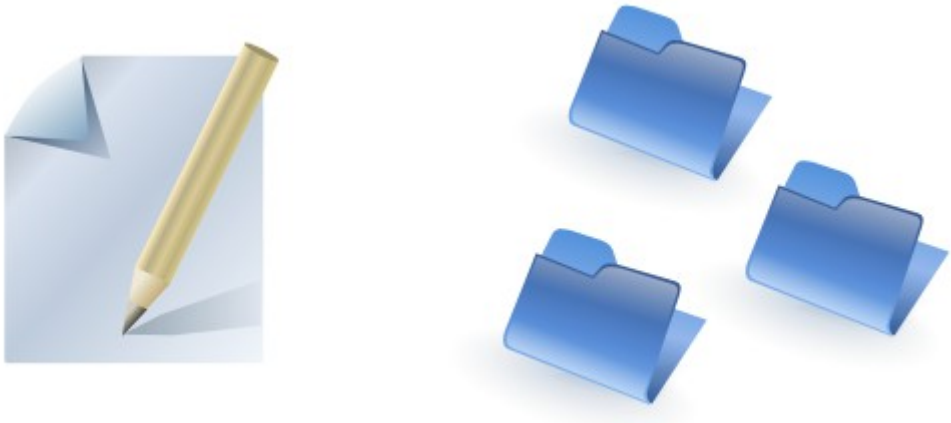
10.- Copias de seguridad

Las copias de seguridad son un elemento fundamental para que el trabajo que realizamos se pueda proteger de aquellos problemas o desastres que pueden ocurrir. El objetivo de las copias de seguridad no es evitar esos problemas, sino poder recuperar los datos en el caso de que ocurran, cosa que sin duda siempre sucede y además en el momento más inoportuno.

En ocasiones, los discos duros se rompen

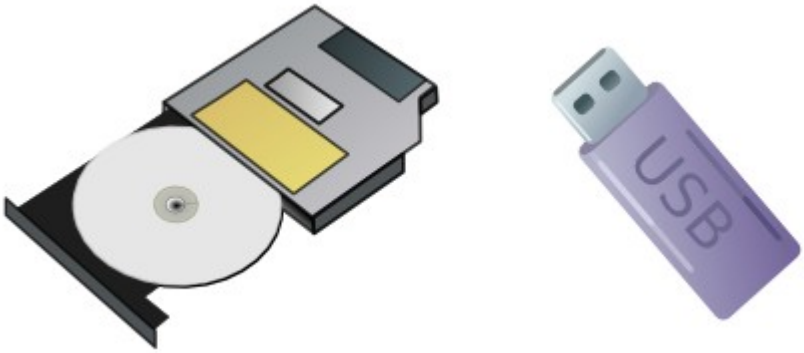
Las causas que pueden provocar la pérdida de información son muy variadas, desde el mal funcionamiento de una aplicación hasta una rotura de un disco duro, pasando por todo tipo de programas maliciosos. Es por lo tanto imprescindible, planificar y llevar a cabo las tareas de prevención correspondientes.

Para estar preparados ante cualquier desastre que elimine la información de los discos duros del servidor, debemos planificar una política de realización de copias de seguridad periódicas que salvaguarden tanto los datos de los usuarios como los archivos de la configuración del sistema y los servicios.



Conviene planificar una politica de copias de seguridad

Para realizar una copia de seguridad debemos decidir el tipo de soporte donde vamos a almacenar los datos. Lo ideal es utilizar un medio de almacenamiento extraible como cintas magnéticas, aunque es muy frecuente realizar las copias en discos duros. Actualmente están muy extendidos los discos extraibles usb (memorias flash de bolsillo) cuyas capacidades alcanzan los 2 GB aunque al ser un dispositivo facilmente manipulable, existe la posibilidad de un borrado fácil de la copia de seguridad en él almacenada.



Se recomienda realizar la copia en dispositivos extraíbles

La segunda decisión que tomaremos es la planificación de la forma en que realizaremos la copia de seguridad. En función de la cantidad de datos a salvaguardar, podemos elegir entre tres tipos de tareas de copia de seguridad. Es importante seleccionar la tarea apropiada puesto que ello nos permitirá minimizar el número de cintas (u otros medios) y el tiempo empleado en realizar dicha tarea.

Tipos de copia de seguridad

En función de la cantidad de archivos que se salvaguardan a la hora de realizar la copia de seguridad, podemos distinguir tres tipos de copia:

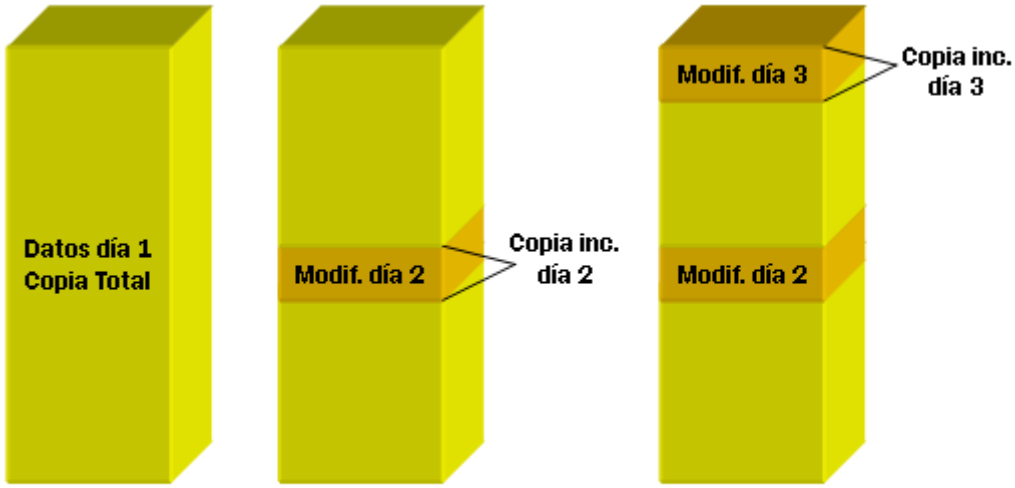
- Copia de seguridad total o íntegra
- Copia de seguridad incremental
- Copia de seguridad diferencial

Copia normal o copia total

Una copia de seguridad normal, es una copia de seguridad total de todos los archivos y directorios seleccionados.

Copia incremental

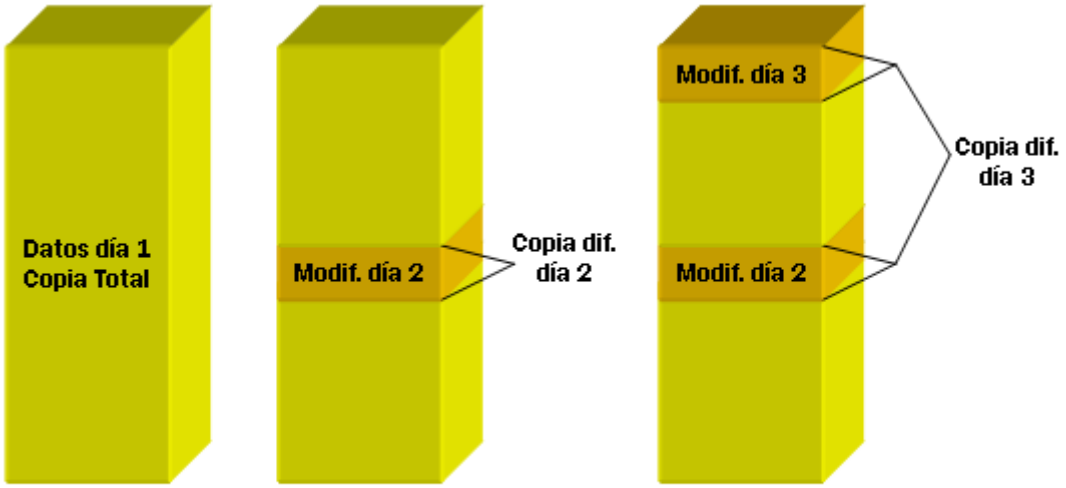
En un proceso de copia de seguridad incremental, se hace una copia de seguridad sólo de los archivos que han cambiado desde la última copia de seguridad realizada. Ejemplo, si hacemos copia de seguridad total el día 1 de cada mes y copia de seguridad incremental el resto de los días, cada copia incremental solo guardará los archivos que se hayan modificado ese día. Si tenemos que realizar la restauración de archivos **ante un desastre, debemos disponer de la copia total y de todas las copias incrementales que hayamos realizado desde la copia total.**



Las copias incrementales guardan solo los archivos modificados desde la última copia incremental

Copia diferencial

Una copia de seguridad diferencial es una copia de todos los archivos que han cambiado desde la última copia de seguridad total que hayamos hecho. Ejemplo, si hacemos copia de seguridad total el día 1 de cada mes y copia de seguridad diferencial el resto de los días, cada copia diferencial guardará los archivos que se hayan modificado desde el día 1. La ventaja es que se requiere menos espacio que la copia total y que **en el proceso de restauración únicamente necesitaremos la última copia total y la última copia diferencial.** Una copia diferencial anula a la copia diferencial anterior. Por el contrario, se consume más tiempo en realizar la copia y también más espacio que en el caso de copia incremental.



Las copias diferenciales guardan solo los archivos modificados desde la última copia total

Recomendación sobre el tipo de copia a efectuar

Si el volumen de datos de nuestra copia de seguridad no es muy elevado (menos de 4 GB), lo más práctico es realizar **siempre copias totales** ya que en caso de desastre, tan solo debemos recuperar la última copia.

Si el volumen de datos de nuestra copia de seguridad es muy elevado (mayor de 50 GB) pero el volumen de datos que se modifican no es elevado (sobre 4 GB), lo más práctico es realizar una primera copia total y posteriormente realizar **siempre copias diferenciales**. Así, en caso de desastre, tan solo debemos

recuperar la copia total y la última diferencial. Periódicamente debemos realizar una copia total y así empezar de nuevo.

Si el volumen de datos de nuestra copia de seguridad es muy elevado (mayor de 50 GB) y el volumen de datos que se modifican también lo es, las copias diferenciales ocuparán mucho espacio, por lo tanto en este caso lo más práctico será realizar una primera copia total y posteriormente realizar **siempre copias incrementales** ya que son las que menos espacio ocupan. El problema es que en caso de desastre debemos recuperar la última copia total y todas las incrementales realizadas desde que se hizo la última copia total. En estos casos, conviene hacer copias totales más a menudo para no tener que mantener un número muy elevado de copias incrementales.

En grandes compañías donde la realización de copias de seguridad está perfectamente planificada, se suelen utilizar sistemas mixtos. Por ejemplo en un caso típico se realizarían las siguientes tareas:

- Todos los días 1 de cada més, a las 23:00 horas: copia de seguridad total
- Todos los viernes a las 23:00 horas: copia de seguridad diferencial desde la copia de día 1
- Todos los días (excepto los viernes y el día 1) a las 23:00 horas: copia de seguridad incremental desde la copia del día anterior.

Con ésta planificación nos aseguramos disponer de copia de seguridad diaria. En caso de desastre deberíamos recuperar la copia total, la última diferencial y todas las incrementales desde la última diferencial.

En una política de este tipo se pueden utilizar por ejemplo 5 juegos diferentes de cintas de forma que se almacenen las copias de seguridad diarias de los últimos 3 meses. Luego se van reutilizando pero no más de 20 veces ya que las cintas se deterioran y la fiabilidad disminuye.

Creación de copias de seguridad

Elección de las carpetas a salvaguardar

Lo primero que debemos determinar son las carpetas que queremos salvaguardar en nuestro proceso de copias de seguridad.

En un sistema informático que da servicio a usuarios, la información más importante es precisamente la información de los usuarios, por lo tanto, la carpeta /home es una de las carpetas que debemos salvaguardar.

El objetivo de la realización de copias de seguridad es el reestablecimiento del servicio en el mínimo tiempo posible, por eso es conveniente realizar una copia de seguridad de los archivos de configuración del servidor, los cuales se encuentran en la carpeta /etc.

Otras carpetas de cierta importancia que se pueden salvaguardar son la carpeta /root y la carpeta /var/log. La primera es la carpeta personal del usuario root y la segunda es la carpeta donde se almacenan las incidencias del sistema (archivos de log del sistema). Resumiendo, deberíamos salvaguardar las siguientes carpetas:

- /home (Carpetas personales de los usuarios)
- /etc (Archivos de configuración del sistema)
- /root (Carpeta personal del usuario root)
- /var/log (Carpeta de logs del sistema)
- /var/www (Web de la intranet)

Por qué se debe comprimir la copia de seguridad

Cuando realizamos copias de seguridad, los datos deben comprimirse siempre por tres razones:

- La copia se realiza más rápidamente
- El tamaño de la copia es menor
- La compresión garantiza la integridad de los datos

Al quedar los datos reducidos, la cantidad de datos a copiar en el soporte de almacenamiento es mucho

menor que lo que ocupan los datos descomprimidos; eso unido al hecho de que los datos estén compactados en un único archivo, hace que el tiempo en trasmitir los datos desde el servidor al soporte, sea menor que si no se comprime.

La integridad de los datos queda garantizada porque el algoritmo de compresión añade un código de redundancia cíclica (CRC) que se consulta a la hora de descomprimir los datos de forma que tenemos seguridad si están correctos o no lo están.

Nomenclatura de los archivos de copia de seguridad

La creación manual de la copia de seguridad consiste en ejecutar manualmente el comando que genera la copia. El resultado es un único archivo comprimido que contiene todos los datos que se quieren salvaguardar. Normalmente, el nombre del archivo suele incluir el tipo de copia, las carpetas que contiene y la fecha (en el caso de copias totales) o fechas (en el caso de copias diferenciales e incrementales) de los datos.

Ejemplo, si hoy fuera 1 de febrero de 2012 y deseáramos crear una copia de seguridad total de las carpetas etc y home, lo normal es que el nombre del archivo fuera:

```
// Nombre de archivo copia de seguridad total
     CopiaTotal_etc-home_01feb12.tar.bz2
```

Si una semana después, el 8 de febrero de 2012 deseáramos crear una copia de seguridad diferencial desde la copia total del día 1 de las carpetas etc y home, lo normal es que el nombre del archivo fuera:

```
// Nombre de archivo copia de seguridad diferencial
     CopiaDiferencial_etc-home_01feb12-08feb12.tar.bz2
```

Si el día siguiente, 9 de febrero de 2012, deseáramos crear una copia de seguridad incremental desde la copia diferencial del día 8 de las carpetas etc y home, lo normal es que el nombre del archivo fuera:

```
// Nombre de archivo copia de seguridad incremental
     CopiaIncremental_etc-home_08feb12-12feb12.tar.bz2
```

Con ésta nomenclatura será más fácil identificar los datos que contienen los archivos de copia de seguridad ya que el nombre del archivo lleva implícito el tipo de copia, las carpetas de datos que contiene y la fecha o fechas de los archivos salvaguardados.

Creación manual de la copia de seguridad

Para crear copia de seguridad de una carpeta o carpetas, habitualmente se utiliza el comando tar que permite crear un único archivo que contenga todos los datos y además, permite comprimirlos en diferentes formatos.

Aquí utilizaremos la compresión bzip2 por ser una de las que más comprime. Los archivos tar comprimidos en bzip2 suelen llevar extensión '.tar.bz2'.

```
// Utilización de tar
     // Para crear copia de seguridad de varias carpetas

     tar -jcvf CopiaTotal.tar.bz2 carpeta1 carpeta2 carpeta3 ...
```

Opciones:

- j: Comprimir utilizando bzip2
- c: Crear nuevo archivo
- v: Mostrar los archivos añadidos
- f: Escribir hacia un archivo

```
// Para extraer los archivos que contiene el archivo tar.bz2
     tar -jxvf copia.tar.bz2
```

Opciones:

- j: Comprimir utilizando bzip2
- x: Extraer (descomprimir)
- v: Mostrar los archivos extraídos
- f: Extraer desde un archivo

```
// Para extraer solo un archivo del archivo tar.bz2
     tar -jxvf copia.tar.bz2 ruta-del-archivo/nombre-del-archivo


// Para ver una lista de los archivos que contiene el archivo tar.bz2
     tar -jtvf copia.tar.bz2
```

Opciones:

- t: Mostrar el contenido

```
// Para crear copia de seguridad de los archivos modificados tras una fecha dada
     tar -jcvf CopiaDiferencial.tar.bz2 -N 1feb2012
```

La opción -N en el comando tar significa **Newer** que traducido es **'más nuevo que'**. Si incluimos la opción -N 1feb12 significa que solamente va a añadir los archivos que se han modificado con posterioridad a dicha fecha, es decir, más nuevos que el 1 de febrero de 2012 a las 0 horas, 0 minutos.

Ejemplo, si hoy fuera 1 de febrero de 2012 y deseamos realizar una copia de seguridad total en la carpeta /tmp (temporal) de las carpetas /home y /etc, el nombre del archivo será CopiaTotal_etc-home_01feb12.tar.bz2 y el comando que debemos lanzar será:

```
// Crear copia total
     tar -jcvf /tmp/CopiaTotal_etc-home_01feb12.tar.bz2 /home /etc
```

Si utilizamos el comando 'date' podemos hacer que se ponga automáticamente la fecha actual en el nombre del archivo y nos servirá para cualquier día ya que tomará la fecha del sistema. El comando date muestra la fecha del sistema. Si queremos que muestre la fecha en un formato especial como por ejemplo 13sep12, debemos escribir date +%d%b%y.

Al escribir el comando date entre comillas simples inclinadas (`), la salida del comando date sustituirá al comando en su lugar, es decir, donde pone `date +%d%b%y` quedará sustituido por 14feb12 si hoy fuera esa fecha:

```
// Crear copia total poniendo la fecha de hoy en el nombre del archivo
     tar -jcvf /tmp/CopiaTotal_etc-home_`date +%d%b%y`.tar.bz2 /home /etc
```

Ejemplo, si hoy fuera 8 de febrero de 2012 y deseáramos realizar una copia de seguridad diferencial de los cambios producidos desde el día 1 de febrero de 2012 en la carpeta /tmp (temporal) de las carpetas /home y /etc, el nombre del archivo será CopiaDiferencial_etc-home_01feb12-08feb12.tar.bz2 y el comando que debemos lanzar será:

```
// Crear copia diferencial
     tar -jcvf /tmp/CopiaDiferencial_etc-home_01feb12-08feb12.tar.bz2 /home
     /etc -N 01-feb-12
```

Pero si en lugar de escribir directamente 01feb12-08feb12 escribimos 01`date %b%y`-`date +%d%b%y` nos servirá el mismo comando para todos los días.

Automatización

El proceso de creación de copias de seguridad **debe ser un proceso automático** que no requiera la intervención del usuario para realizarse ya que por un olvido o dejadez del mismo podría ocurrir que el día que necesitemos la copia de seguridad, ésta **no se haya hecho**.

Para lanzar la realización automática de copias utilizaremos **cron**. Cron es un servicio que nos permite lanzar comandos automáticamente **los días y a las horas que deseemos**. Cada usuario tiene su propio cron en el que puede configurar sus **tareas programadas** mediante el comando '**crontab -e**' o con alguna aplicación gráfica como **gnome-schedule**. En nuestro caso, como realizamos copia de seguridad de carpetas que solamente tiene acceso el usuario root, debemos programar la copia mediante el **cron de root**.

Supongamos que deseamos crear una copia de seguridad total los días 1 de cada mes y una copia de seguridad diferencial el resto de días en la carpeta /tmp (temporal), de las carpetas /home y /etc. El comando que ejecutaremos el día 1 de cada mes será:

```
// Comando a ejecutar los días 1 de cada mes
     tar -jcvf /tmp/CopiaTotal_etc-home_`date +%d%b%y`.tar.bz2 /home /etc
```

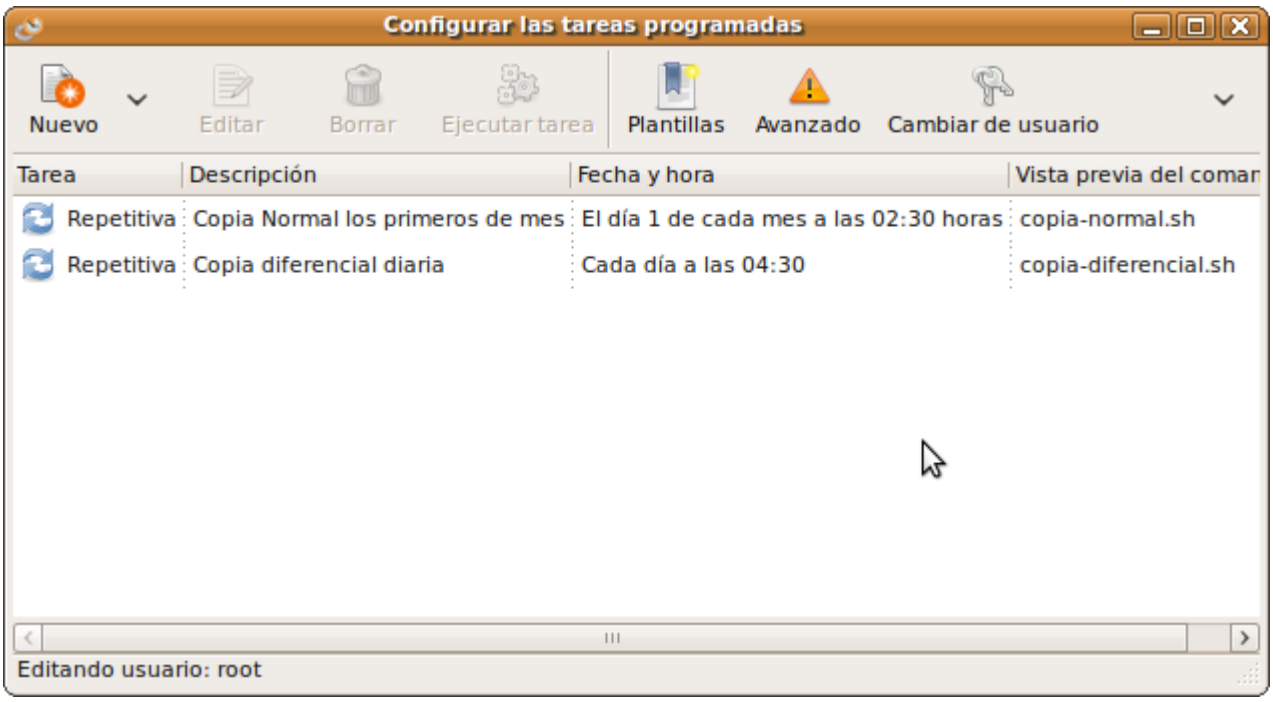
Como puede verse, utilizamos `date +%d%b%y` que si hoy es 1 de febrero de 2012 se sustituira por 1feb12. De ésta forma nos sirve el mismo comando para todos los meses.

El comando que ejecutaremos todos los días para realizar la copia diferencial, será:

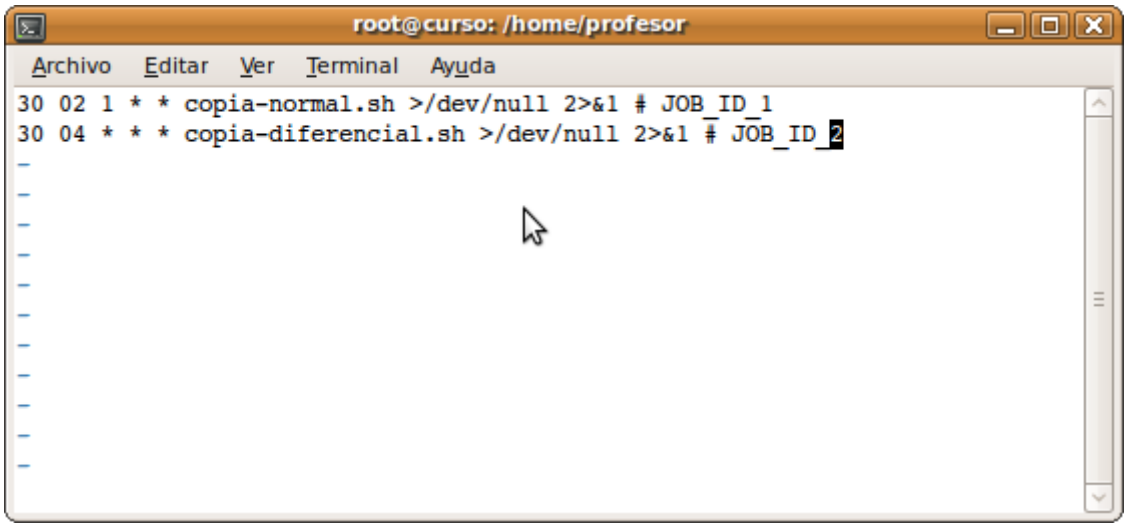
```
// Comando a ejecutar los días para hacer copia diferencial respecto al día 1
     tar -jcvf /tmp/CopiaDiferencial_etc-home_01`date +%b%y`-`date +%d%b
     %y`.tar.bz2 /home /etc -N 01`date +%b%y`
```

Como puede verse, utilizamos 01`date %b%y`-`date +%d%b%y` que si hoy es 13 de febrero de 2012 se sustituira por 01feb12-13feb12. También en la opción -N ponemos 01`date +%b%y` para que añada únicamente los archivos más nuevos que el día 1 del mes actual. De ésta forma nos sirve el mismo comando para todos los días. Podemos crear scripts para guardar los comandos, ejemplo: copia-normal.sh y copia-diferencial.sh.

Si deseamos programar para que automáticamente se ejecute la copia total el día 1 de cada més y la copia diferencial todos los días, debemos añadirlo en el cron del usuario root lo cual se puede realizar ejecutando el comando 'crontab -e' o bien utilizando **gnome-schedule**:



Al final nuestro archivo cron para que se ejecuten automáticamente los comandos que realizan las copias de seguridad quedará como el de la figura:



De ésta forma, los días 1 de cada mes a las 02:30 horas se realizará la copia total de las carpetas /etc y /home y todos los días a las 04:30 horas se realizará la copia diferencial respecto de la copia del día 1 del mes.

Las copias de seguridad se realizarán sobre la carpeta /tmp, pero lo recomendable es realizar la copia de seguridad sobre un dispositivo extraible, por ejemplo, un disco duro extero USB. Habrá que sustituir /tmp por la carpeta donde esté montado el dispositivo. Si se trata de un disco duro USB, lo más normal es que el dispositivo esté montado en la carpeta /mnt/sda1 o algo parecido ya que se monta como un dispositivo SCSI, lo que quiere decir que en lugar de poner /tmp en el comando tendríamos que poner /mnt/sda1.

Copias de seguridad en servidores remotos

Lo comentado anteriormente permite realizar copias de seguridad en un disco duro local. Una mejora añadida sería la creación de la copia en una carpeta remota. Al igual que se automatiza la creación de la copia, se podría ejecutar automáticamente un comando que, vía nfs, samba, ftp o ssh, vuelque los archivos en un servidor remoto para mayor seguridad. También existen herramientas para realizar directamente copias de seguridad remotas:

- rsync: permite realizar copias en carpetas remotas
- unison: permite mantener sincronizadas dos carpetas remotas

Aplicaciones para la realización de copias de seguridad

Existen aplicaciones tanto libres como de pago que facilitan la tarea de realización de copias de seguridad. Entre las aplicaciones libres destacamos:

- BackupPC: Herramienta para hacer copias de seguridad de PCs de la red
- Amanda: Herramienta para hacer copias de seguridad de PCs de la red
- afbackup: Herramienta para hacer copias de seguridad de PCs de la red

Estas aplicaciones tienen la ventaja de ser muy completas ya que disponen de un sinfín de posibilidades, pero son más complejas de manejar.

11.- Servidor de impresión

Introducción

En un sistema informático es muy frecuente la necesidad de imprimir documentos ya que es una de las aplicaciones principales de los ordenadores.

Hace unos años, cuando las redes locales no estaban muy extendidas, cada PC disponía de su propia impresora. A veces se compartía una impresora entre varios PCs mediante un conmutador de impresora que inicialmente eran manuales y posteriormente fueron electrónicos.

Con la generalización de las redes locales se fueron sofisticando los sistemas para compartir y optimizar el uso de impresoras. En la actualidad, esos sistemas están muy desarrollados gracias a los servidores de impresión.

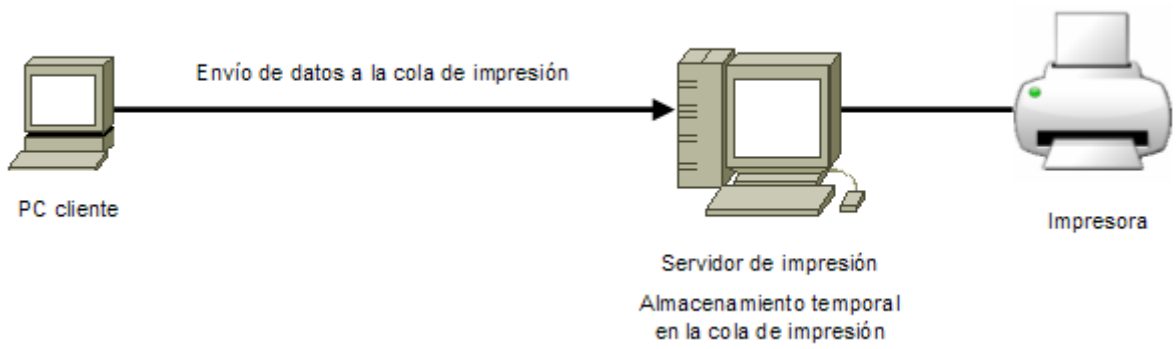
Un servidor de impresión es un software que permite que los PCs de una red local puedan hacer uso de las impresoras de la red de una forma eficaz ya que centraliza las tareas de impresión facilitando una gestión de las mismas.

Impresora y cola de impresión

Para poder imprimir documentos en papel, es evidente que necesitamos un periférico que comúnmente llamamos 'impresora' aunque a veces se le denomina 'dispositivo de impresión'. Las impresoras pueden utilizar diferentes tecnologías de impresión, aunque las más comunes son las impresoras de inyección de tinta y las impresoras láser.

Cuando distintos usuarios desean imprimir documentos, podrían enviarlos directamente hacia la impresora, pero eso consumiría recursos de sus PCs y mezclaría distintos trabajos. Una cola de impresión es un almacén temporal donde permanecen los documentos en espera de que puedan ser imprimidos según un orden secuencial.

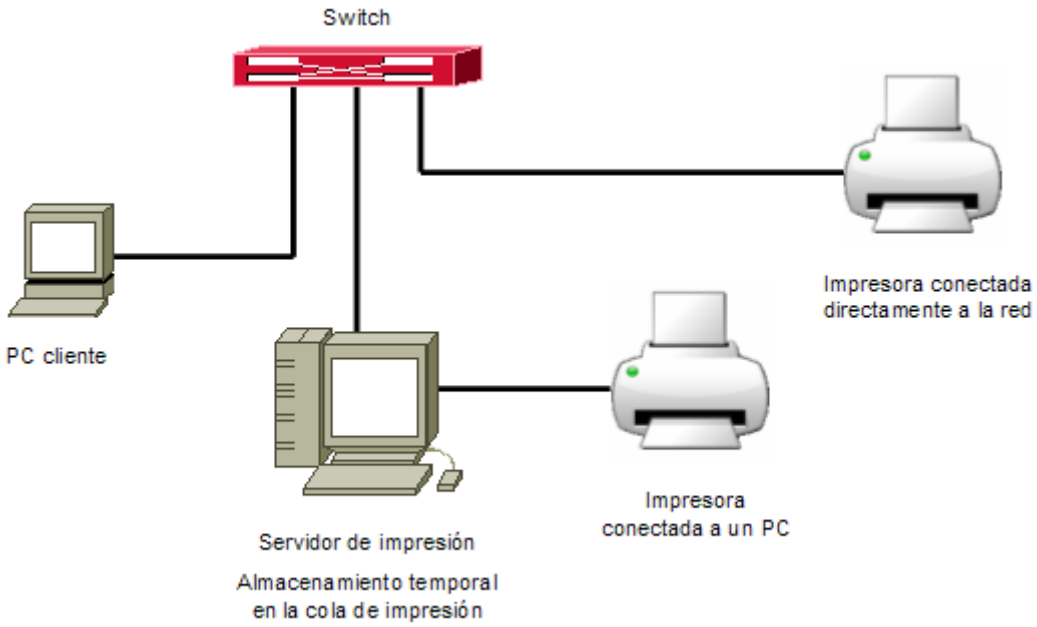
La cola de impresión (almacén temporal) puede estar en el propio PC del usuario, en un servidor de impresión o en la misma impresora de red. Lo mejor es que la cola esté en un servidor de impresión, de esa forma el PC del usuario queda menos cargado, los trabajos de impresión de distintos usuarios no se mezclan y existe la posibilidad de administrar los trabajos de impresión (establecer prioridades, límites, etc...)



Formas de conectar una impresora

Las impresoras pueden conectarse a un sistema básicamente de dos formas:

- Impresora conectada a un PC (por puerto paralelo o por USB)
- Impresora conectada directamente a la red



Cuando la impresora está conectada a un equipo, es necesario que dicho equipo esté encendido y que disponga de un software que comparta la impresora para que pueda ser utilizada por el resto de equipos de la red local. Habitualmente las impresoras conectadas a un equipo, suelen estar conectadas a un servidor ya que suelen estar siempre encendidos y además, como hemos comentado anteriormente, lo ideal es que la cola de impresión esté en el servidor.

Las impresoras conectadas directamente a la red son impresoras que disponen de una interfaz ethernet y tienen incorporado el protocolo TCP/IP que les permite integrarse perfectamente en nuestra red local. Suelen disponer de una pequeña pantalla con unos botones para poder configurar la dirección IP. Una vez hayamos configurado la dirección IP, desde un navegador podremos ir a **http://ip-de-la-impresora** para configurar el resto de parámetros y administrarla vía web. Cada vez es más frecuente ver impresoras con servidor de impresión propio aunque si no tienen esa funcionalidad, habrá que configurarla en un servidor de impresión quien administrará la cola de impresión.

Instalación y configuración del servidor de impresion

Como primera opción vamos a instalar una impresora local en el servidor de nuestra intranet educativa y vamos a compartirla para que los usuarios de la red puedan utilizarla independientemente del ordenador que estén utilizando. Esta impresora estará situada en la misma ubicación que el servidor y por ello, dado que hemos visto la conveniencia de aislar el servidor, su utilización deberá analizarse detalladamente. Insistimos en que el profesorado no utilizará el servidor para imprimir un documento, sino que utilizará la impresora conectada al servidor desde una estación remota.

Introducción

Aunque Linux dispone de otros sistemas de impresión, uno muy utilizado es el sistema CUPS (Common Unix Printer System - Sistema de impresión común en Unix) que será el que utilicemos en este curso. El software CUPS permite instalar, configurar, administrar y compartir impresoras en un servidor Linux de una forma bastante sencilla. Este software podrá satisfacer plenamente las necesidades de servidor de impresión que se puedan dar en un sistema informático mediano.

Instalación del servidor cups

Para instalar el servidor de impresión cups debemos instalar mediante apt-get el paquete cupsys que

contiene todas las aplicaciones necesarias que nos proporcionará un servidor de impresión.

```
// Instalación del servidor cupsys
     cnice@cnice-desktop:# apt-get install cupsys
```

Arranque y parada manual del servidor cups

El servidor cups, al igual que todos los servicios en Debian, dispone de un script de arranque y parada en la carpeta /etc/init.d.

```
// Iniciar o Reiniciar el servidor cups
     cnice@cnice-desktop:# /etc/init.d/cups restart
```

```
// Parar el servidor cups
     cnice@cnice-desktop:# /etc/init.d/cups stop
```

Arranque automático del servidor de impresión al iniciar el sistema.

Para un arranque automático del servicio al iniciar el servidor, debemos crear los enlaces simbólicos correspondientes tal y como se indica en el apartado Arranque automático de servicios al iniciar el sistema.

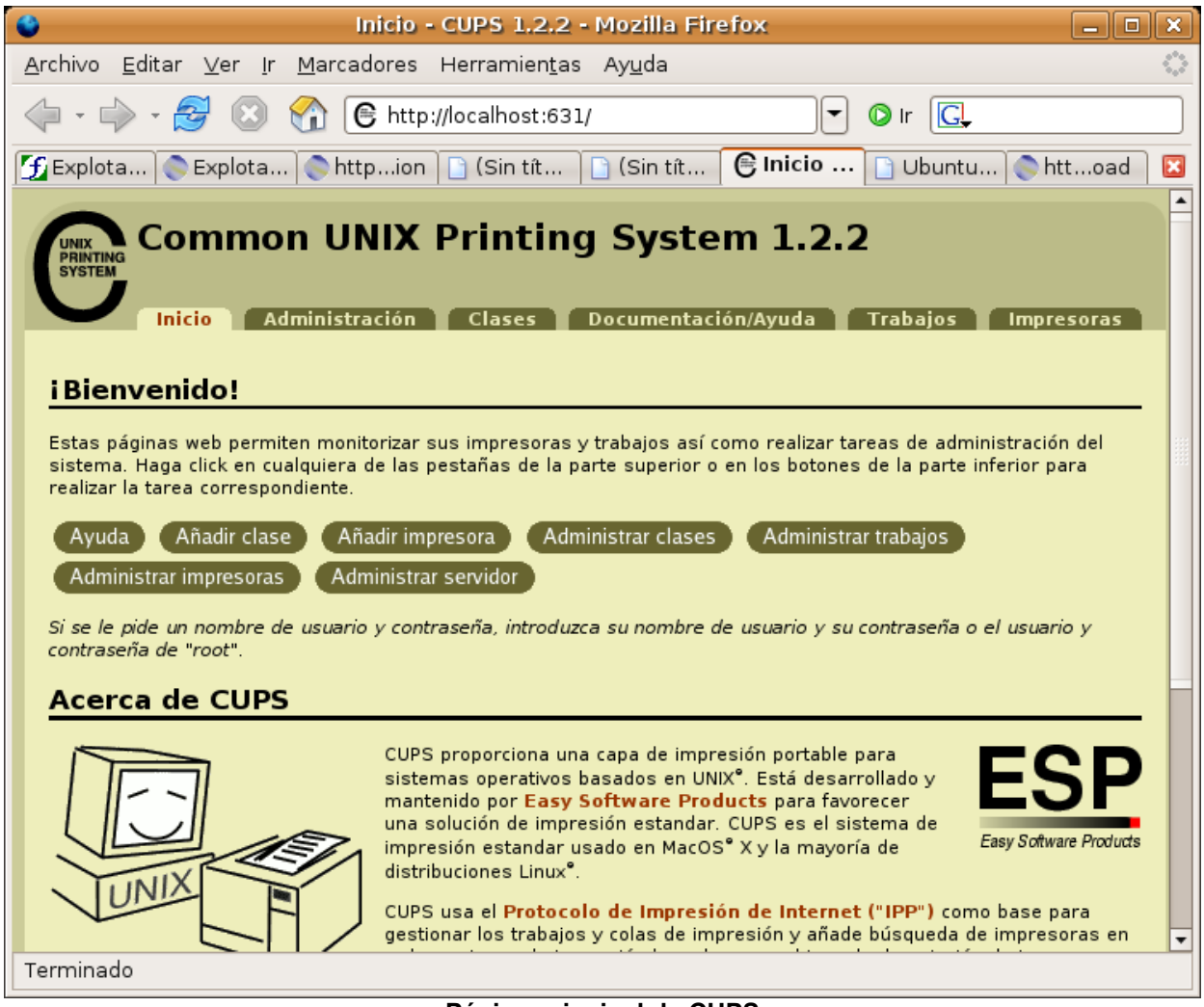
Configuración de cups

Todos los archivos de configuración de cups se encuentran en la carpeta /etc/cups. El archivo de configuración del servicio es el archivo /etc/cups/cupsd.conf pero apenas es necesario cambiar nada ya que la configuración del servicio se realiza via web.

Una vez que tenemos en marcha el servicio de impresión cups, podremos configurar impresoras y administrar tareas de impresión. Desde el servidor, debemos abrir un navegador e ir a la siguiente dirección:

```
// Configurar cups vía web
     http://localhost:631/
```

La página principal del administrador de servidor de impresión vía web es:



Página principal de CUPS

Para poder acceder a alguna de las opciones es necesario ser administrador, en tal caso habrá que poner nombre de usuario 'root' y su contraseña.

En la parte superior de la página principal, disponemos de seis menús que nos permitirán acceder a las distintas opciones de configuración de cups. A continuación comentamos brevemente las funciones de los distintos menús.

Inicio

Muestra la página de inicio de cups, desde la cual se puede acceder directamente a las opciones más habituales.

Administración

Desde éste menú se puede acceder a las tareas de administración de cups: administrar impresoras, trabajos de impresión, modificar archivos de configuración, ver errores, etc...

Clases

Permite crear grupos de impresoras para centralizar y gestionar grandes trabajos de impresión. No se utiliza en pequeños sistemas.

Documentación/ayuda

Permite acceder a la ayuda de cupsys. Los documentos están en inglés.

Trabajos

Permite gestionar los trabajos de impresión. Podemos acceder a la cola, ver el estado de la impresión y los trabajos pendientes de imprimir. Existe también la posibilidad de eliminar trabajos de la cola de impresión.

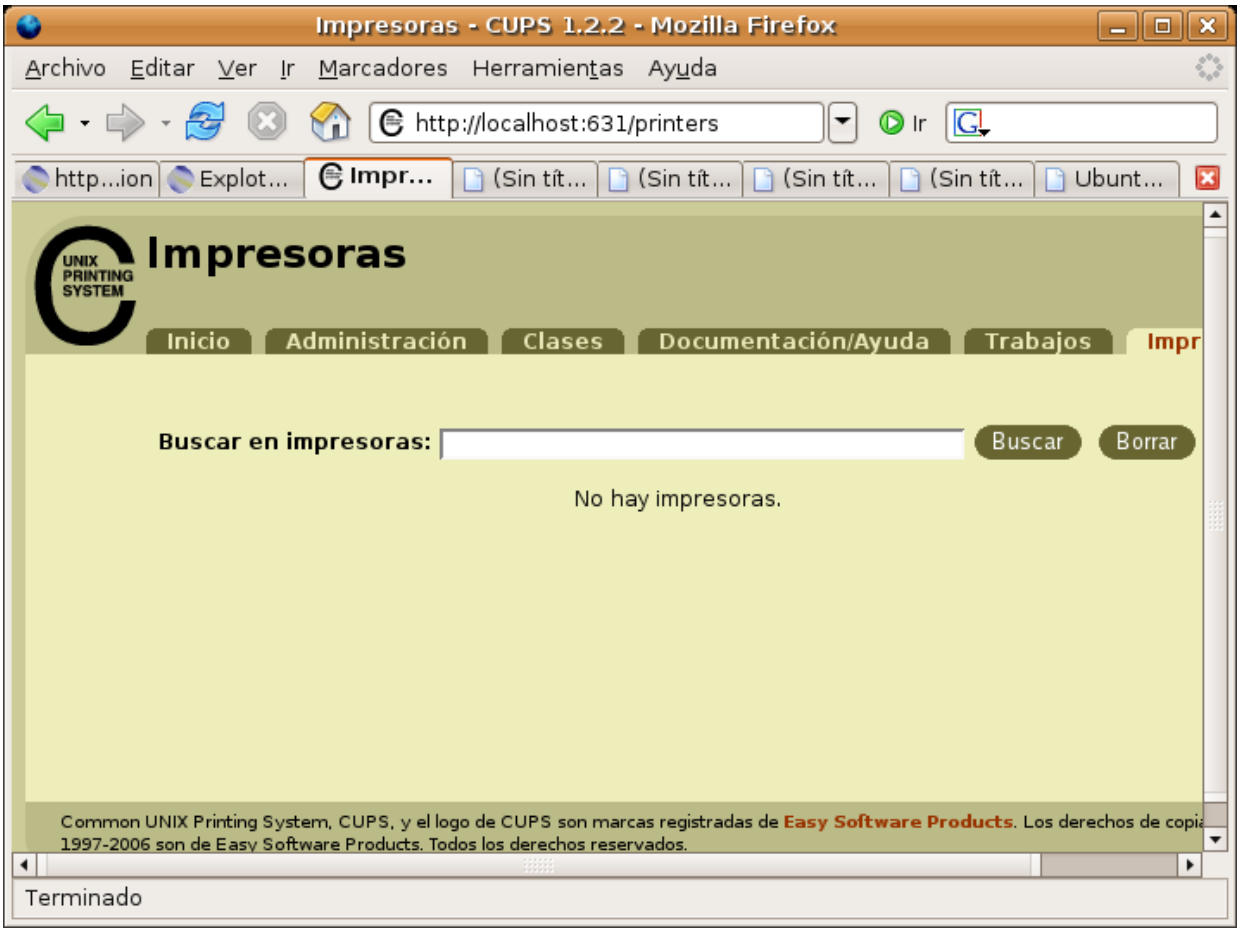
Impresoras

Desde aquí podremos agregar, configurar, eliminar, modificar y administrar impresoras.

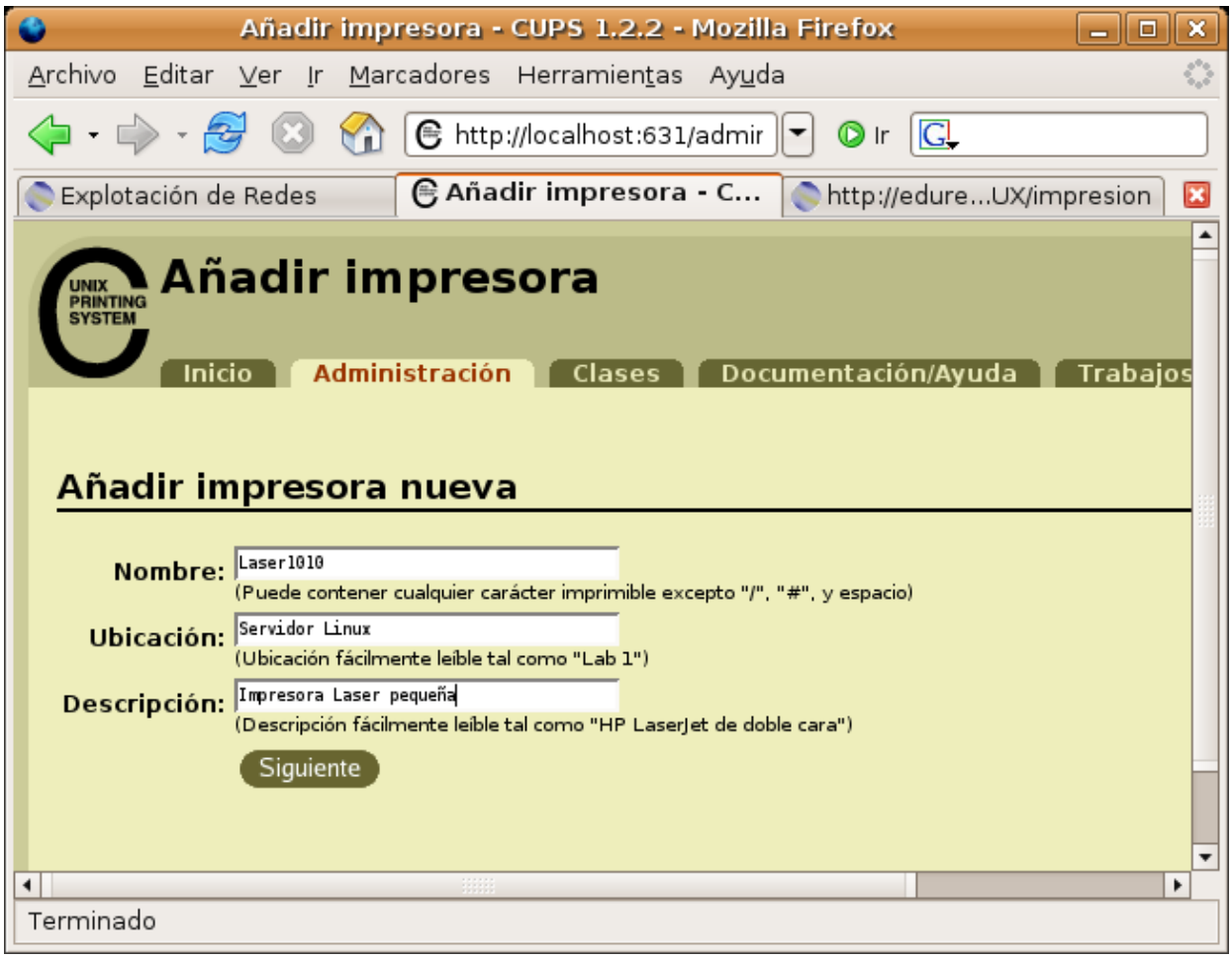
Añadir una impresora

Supongamos que disponemos de una impresora HP LaserJet 1010 conectada por USB a nuestro servidor Linux. Deberemos configurarla en cups para que la impresora pueda ser utilizada tanto desde el servidor como desde los distintos puestos de red.

Inicialmente, lo normal es que no haya ninguna impresora configurada en nuestro sistema, por lo tanto, si accedemos al menú 'Impresoras' lo que veremos será:

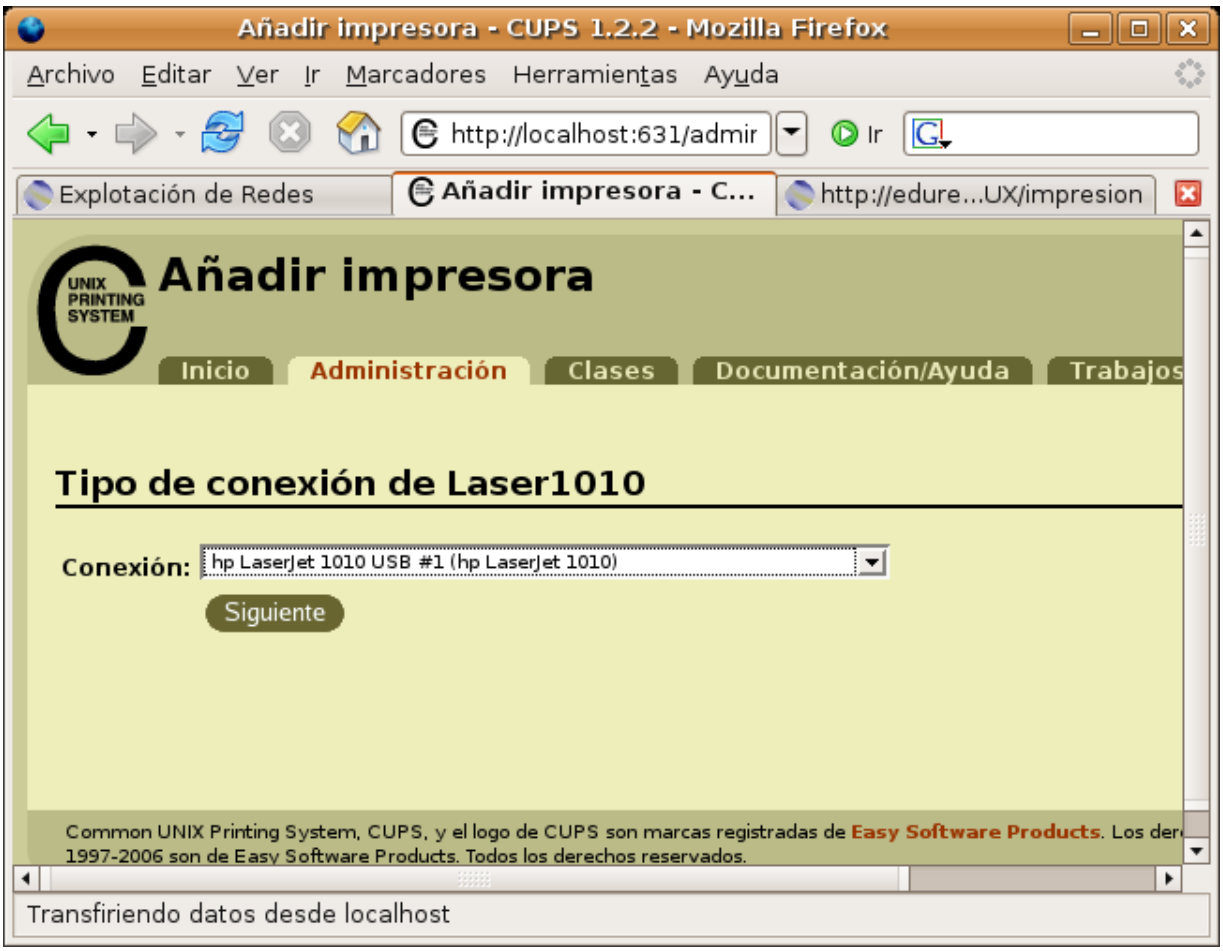


Para añadir una nueva impresora debemos ir al menú 'Administración' y pulsar el botón 'Añadir impresora'. Nos aparecerá un pequeño formulario con tres cajas: Nombre, Ubicación y Descripción donde deberemos poner el nombre que deseamos asignar a la impresora (conviene evitar el uso de espacios), la ubicación donde se encuentra, una descripción y pulsar 'Siguiente'. Ejemplo:

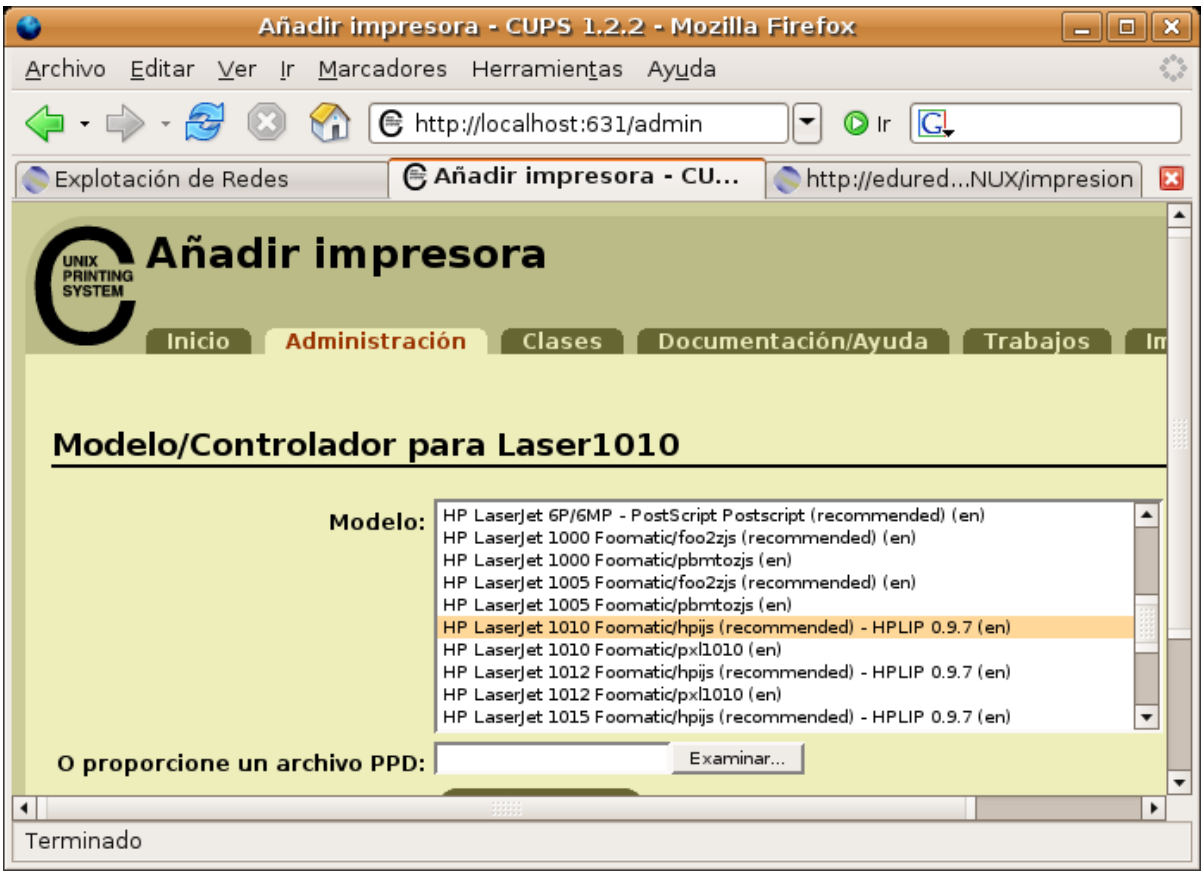


En el siguiente paso, debemos especificar el tipo de conexión con la impresora. Como es una impresora conectada al puerto USB, deberemos escoger 'USB #1'. Otras opciones son:

- AppSocket/HP JetDirect - Impresora conectada directamente a la red con protocolo HP
- Internet Printing Protocol (http) - Impresora accesible por http
- Internet Printing Protocol (ipp) - Impresora accesible por ipp
- LPD/LPR Host or Printer - Impresora conectada directamente a la red con protocolo LPD/LPR
- PDF Writing - Imprimir a PDF
- USB Printer #n - Impresora USB
- Windows Printer vía Samba - Impresora compartida en Windows o Linux con samba

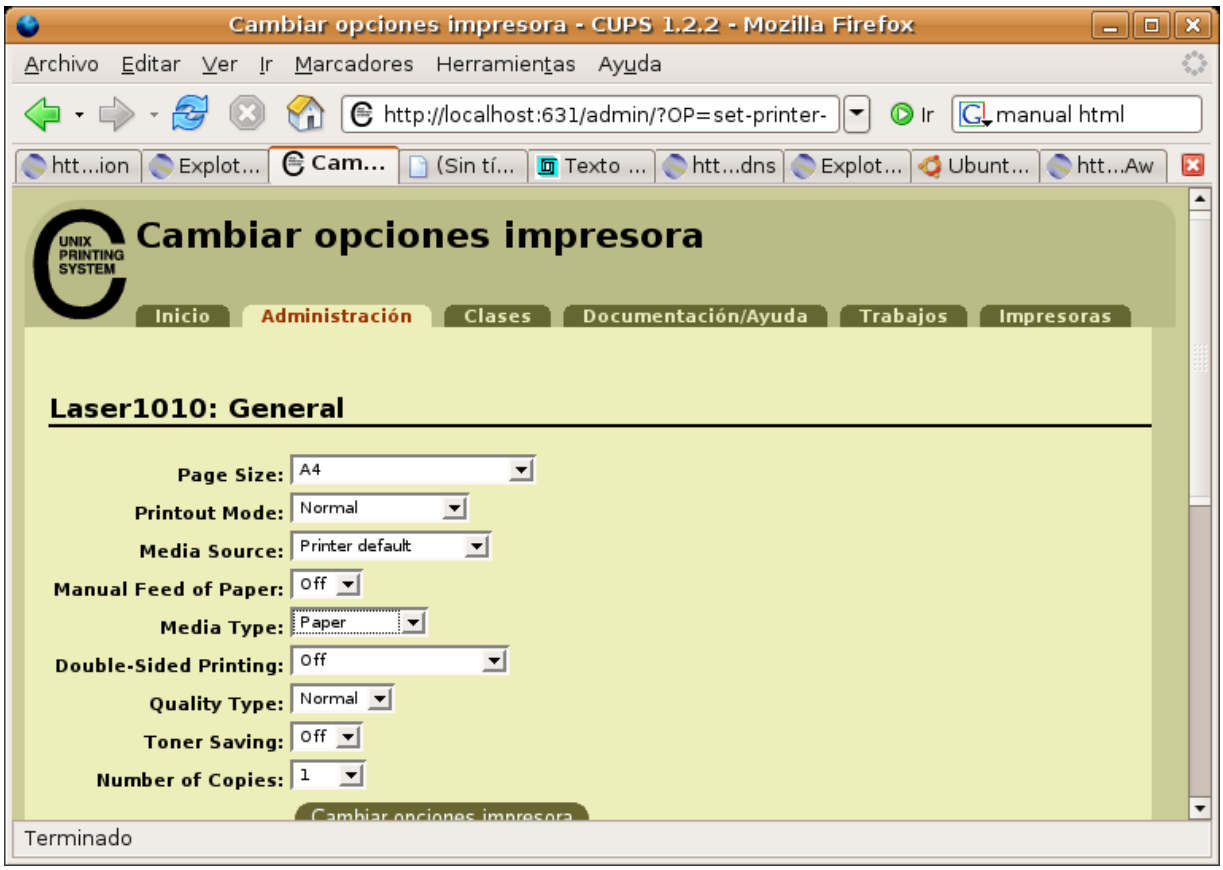


Posteriormente debemos elegir el driver de la impresora. Si no aparece nuestro modelo de impresora, deberemos averiguar si es compatible con otros modelos de la lista.



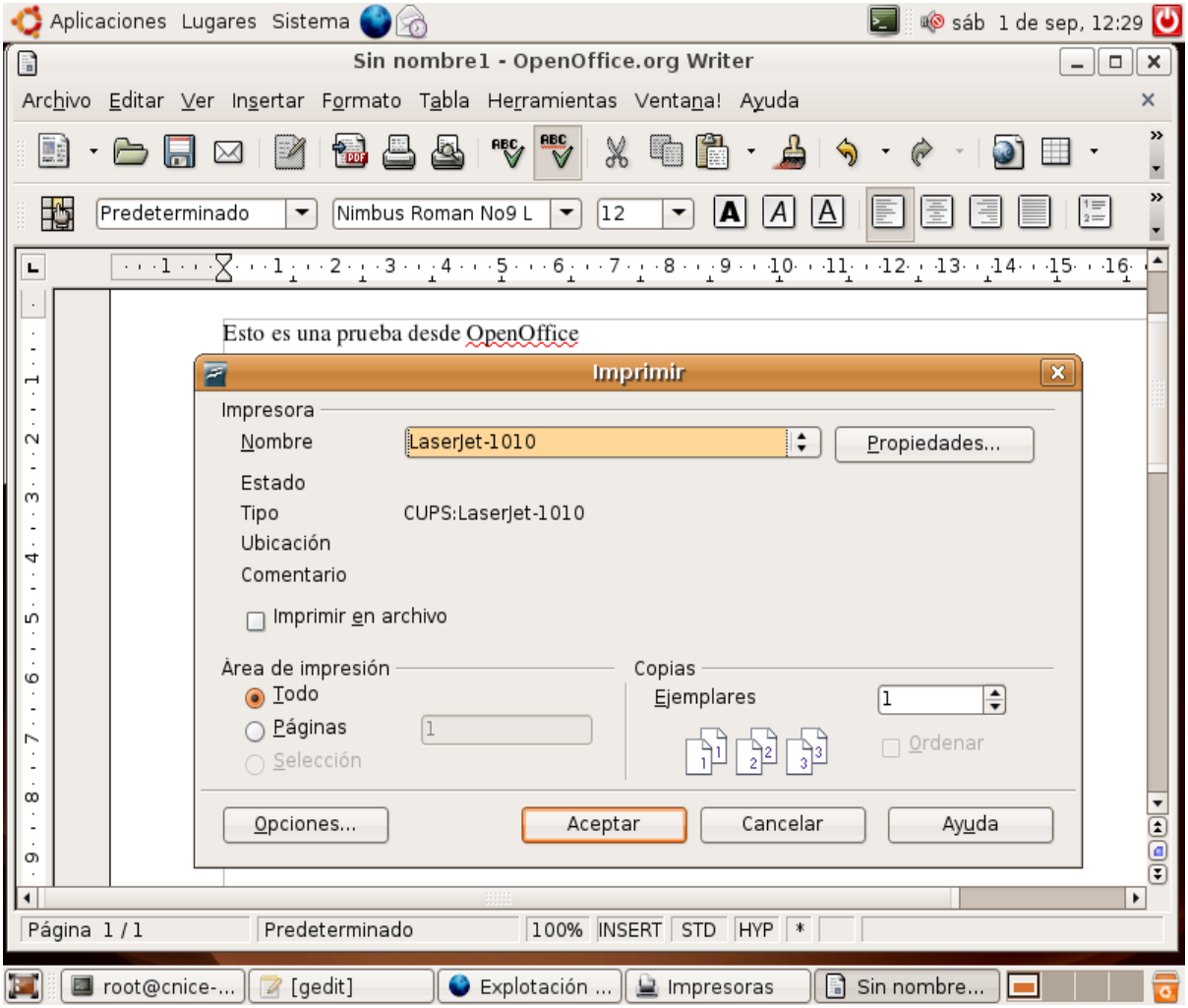
La impresora ha quedado configurada:

Acto seguido aparecerá la página de opciones de la impresora.



Si volvemos al menú 'Impresoras' ahora podremos observar que tenemos una impresora configurada.

Podemos utilizar el botón 'Imprimir página de prueba' para probar la impresora. Si la prueba resulta satisfactoria, desde éste momento ya podremos imprimir con cualquier aplicación que pueda utiliar cups como sistema de impresión, por ejemplo OpenOffice.org:



Ya tenemos nuestra impresora lista para ser utilizada desde el propio servidor. Para que la impresora pueda ser utilizada desde clientes por la red, es necesario ir a 'Administración' y activar la casilla 'Compartir impresoras públicas conectadas a este sistema'. De esta forma, CUPS compartirá la impresora utilizando el protocolo http.

Para utilizar esta impresora desde un cliente de la red, clic aqui

Software - Descargar software cups actualizado

Esta opción es un enlace a la web http://www.cups.org donde se puede descargar el software actualizado aunque en un sistema debian es mejor actualizar el software mediante apt-get.

Administración del servidor de impresión

La administración del servidor de impresión comprende las acciones relacionadas con la configuración de impresoras y gestión de usuarios y permisos para utilizar dichas impresoras. Para realizar la tarea de administración disponemos del comando 'lpadmin' que permite crear y eliminar impresoras (aunque es más sencillo hacerlo con la herramienta web) y establecer permisos a usuarios entre otras funciones.

Ejemplos de utilización del comando lpadmin:

Para permitir el uso de la impresora al usuario jessica y al grupo profesores:

```
// Permitir usuarios y grupos
     # lpadmin -p Laser1010 -u allow:jessica,@profesores
```

Para establecer límite de uso (páginas)

```
// Establecer límite de páginas en 5
     # lpadmin -p Laser1010 -o job-page-limit=5
```

Otros comandos cups

Aunque para administrar el servidor de impresión disponemos de la herramienta web de administración y de la herramienta cupsconfig, cups también dispone de comandos que nos permitirán realizar dichas funciones. Algunos de ellos son:

- **lp:** imprimir
- **cancel:** cancelar trabajos de impresión
- **lpinfo:** mostrar dispositivos o drivers de impresión
- **lppasswd:** establecer contraseñas de usuarios
- **lpstat:** estado de las colas de impresión
- **cupsenable/cupsdisable:** habilitar/deshabilitar cups

Configuración de la impresora en los clientes

Introducción

Una vez que ya tenemos una impresora configurada en el servidor de impresión, ya estamos en disposición de utilizarla tanto desde el propio servidor como desde el resto de los equipos de la red. Tan solo falta configurarla en los PCs clientes para poder utilizarla.

Instalación del cliente cups

Para poder utilizar el sistema cups en el resto de PCs de nuestra red, es necesario instalar y configurar el cliente cups. Para instalar el cliente de impresión cups debemos instalar mediante apt-get el paquete cupsys-client que contiene el software necesario para poder imprimir a través de un servidor de impresión cups.

```
// Instalación del cliente cupsys
     root@cnice-desktop:# apt-get install cupsys-client
```

Configuración del cliente cups

El archivo de configuración del cliente cups es el archivo /etc/cups/client.conf. Si dicho archivo no existe, **debemos crearlo con un editor de texto**. En dicho archivo tan solo hay que indicar quién es el servidor cups en el parámetro ServerName. En nuestro caso:

```
// Configuración del cliente cups. Crear archivo /etc/cups/client.conf
     ServerName 192.168.1.239
```

De ésta manera, todos los comandos de impresión funcionarán en nuestro sistema de la misma forma que lo hace en el propio servidor.

Probando la impresora

```
// Comprobar el estado del servidor de impresión
     root@cnice-desktop:# lpstat -t

     el planificador de tareas se está ejecutando

     no hay un destino predeterminado del sistema

     tipo de conexión para Laser1010: usb://HP/LaserJet%201010

     Laser1010 aceptando peticiones desde sáb 01 sep 2007 14:12:01 CEST

     la impresora Laser1010 está inactiva.  activada desde sáb 01 sep 2007
     14:12:01 CEST



     root@cnice-desktop:#



// Mostrar todos los dispositivos del servidor de impresión
     root@cnice-desktop:# lpinfo -v

     network socket

     network beh

     direct usb://HP/LaserJet%201010

     network http

     network ipp

     network lpd

     direct parallel:/dev/lp0

     network smb

     root@cnice-desktop:#
```

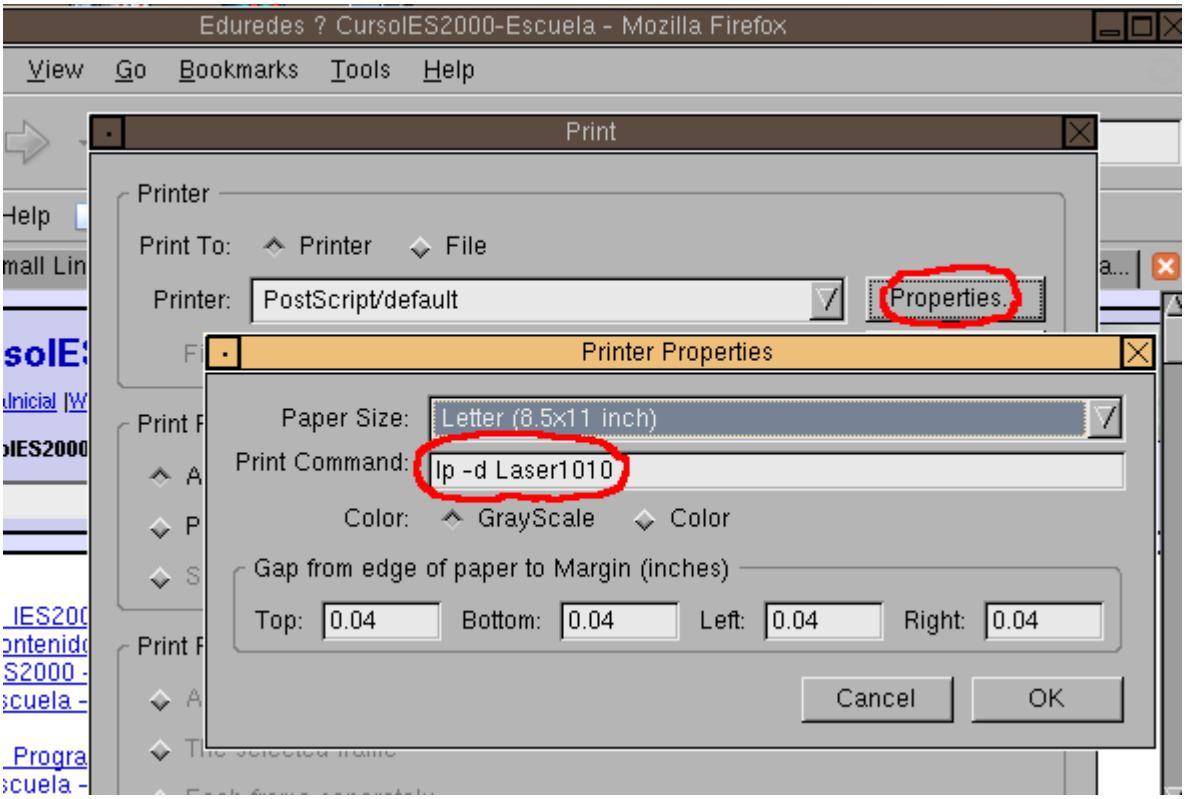
Imprimiendo desde las aplicaciones

Las aplicaciones que utilicen el sistema cups podrán imprimir directamente en las impresoras disponibles en el servidor de impresión.

Para aquellas impresoras que no utilizan el sistema cups, a veces permiten la configuración del comando de impresión que deben lanzar para poder imprimir. El comando para imprimir en cups es el comando 'lp'. Con la opción -d indicamos la impresora de destino. El archivo a imprimir puede ser un archivo de texto o un archivo postscript.

Ejemplo, si queremos utilizar nuestra impresora desde las versiones antiguas del navegador Mozilla Firefox y no nos ha detectado la impresora, podemos hacer clic en 'Imprimir' y en el diálogo de la impresora que nos

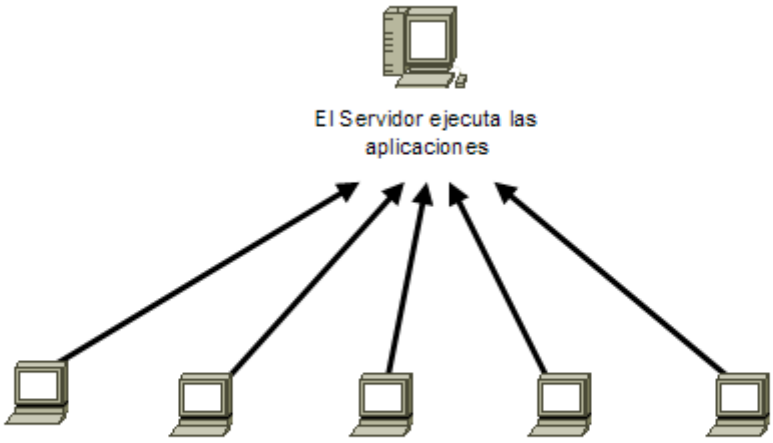
aparece, elegir la impresora 'Printer: Postscritp/default'. Después haremos clic en 'Propiedades' y en la casilla Print Command escribiremos 'lp -d Laser1010' de forma que cuando Firefox deba imprimir algo, lo haga utilizando cups sobre nuestra impresora que hemos denominado Laser1010.



12.- Servidor de terminales

Introducción y antecedentes

Un servidor de terminales es un servidor que ejecuta un software que permite a los usuarios acceder al mismo remotamente desde otros PCs (que hacen de terminales) y manejarlo como si estuvieran sentados frente al servidor.



En los sistemas Unix esto ha existido prácticamente desde siempre ya que los usuarios se conectaban al servidor mediante **telnet** y lanzaban procesos de forma remota. El telnet es una aplicación cliente de terminal que permite desde cualquier PC de la red conectarse a un servidor. Para que la conexión remota sea posible, el servidor deberá tener instalado el software de servidor de telnet que en Debian es el paquete telnetd. Al comenzar la conexión el usuario debía identificarse con nombre (login) y contraseña (password) para poder utilizar el sistema, de la misma forma que lo haría si se sienta en la consola principal del servidor. El telnet está prácticamente en desuso ya que la información que se envía desde el cliente al servidor y viceversa está sin encriptar y cualquier usuario que pinche la red podrá averigurar el nombre del usuario y su contraseña fácilmente.

El sustituto del telnet es el **ssh** (Secure SHell) que permite conectarse a un servidor remoto pero de forma segura ya que las comunicaciones en todo momento van encriptadas con algoritmos muy seguros de forma que es prácticamente imposible descifrar la información. Para más información sobre ssh, haga clic aquí.

En todo momento estamos hablando de accesos remotos en modo texto, es decir, mediante un símbolo del sistema introduciendo comandos como si se tratara de una ventana de ms-dos o un terminal en modo texto de unix. En los años 80-90 era impensable que múltiples usuarios pudieran conectarse a un sistema remoto con terminales gráficos ya que requieren de una gran cantidad de memoria.

En la actualidad, debido al abaratamiento de la memoria RAM, esto se ha convertido en una realidad que ha llegado a los centros educativos y a las pequeñas y medianas empresas.

Servidor de terminales en Linux

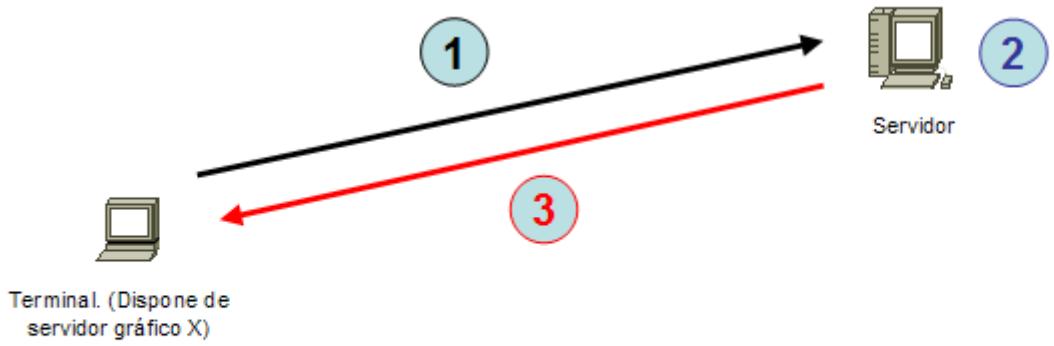
Servidor de terminales mediante X-Window

Linux por sí solo es un servidor de terminales ya que es un sistema operativo multiusuario (varios usuarios pueden ejecutar aplicaciones simultaneamente en el servidor) y utiliza para las aplicaciones gráficas el sistema X-Windows.

El sistema X-Window es un sistema gráfico cliente-servidor orientado a red que se compone de servidor gráfico X (que es quien dibuja las ventanas) y clientes X (que son las aplicaciones gráficas), con lo cual

resulta ideal si se quiere utilizar de forma remota. Cuando ejecutamos aplicaciones gráficas directamente sentados en la consola principal del servidor, las aplicaciones dirigen su salida hacia el servidor grafico local cuya dirección IP es 127.0.0.1, pero cuando ejecutamos aplicaciones gráficas desde un terminal, la salida se dirigirá hacia el servidor gráfico del terminal.

Para disfrutar de un terminal remoto en modo gráfico con X-Window, debemos disponer en nuestro terminal remoto de un servidor grafico X. Los clientes de nuestro servidor gráfico X serán las aplicaciones que lancemos en el servidor y que redigirán la salida hacia nuestro servidor gráfico para que podamos visualizar en nuestro terminal las ventanas generadas por las aplicaciones. Las aplicaciones se ejecutan en el servidor pero las ventanas se visualizan en el terminal.



1.- Desde el cliente se lanza la aplicación. Las pulsaciones de teclado y los movimientos de ratón son enviados al servidor

2.- El Servidor ejecuta la aplicación

3.- El Servidor redirecciona la salida gráfica hacia la IP del terminal en lugar de hacerlo a sí mismo (127.0.0.1)

Para realizar esto de forma sencilla y segura, podemos utilizar ssh. Si en el servidor disponemos de un servidor ssh con la opción de 'redirección X' activada, desde el cliente podemos conectarnos al servidor con un cliente ssh y ejecutar aplicaciones gráficas ya que el servidor ssh se encarga de redireccionar la salida gráfica de las aplicaciones que ejecutemos, hacia nuestro terminal, y además las comunicaciones irán encriptadas. También es posible conectarse por telnet y redirigir la salida gráfica hacia el servidor X de nuestro terminal pero en este caso las comunicaciones viajarán sin encriptar.

Servidor de terminales freenx server

El sistema **freenx** server utiliza el sistema X-Window pero añadiéndo algunas mejoras como la compresión de los datos. Para su funcionamiento es necesario ejecutar en el servidor un componente (nxserver) y en el terminal un cliente de nx (nxclient). El servidor y el cliente para linux se puede descargar de http://freenx.berlios.de

Ventajas de la utilización de un servidor de terminales en centros educativos

En centros educativos, disponer de algún aula con un sistema de terminales nos ofrece algunas ventajas:

- **Aula de bajo coste:** Con un PC moderno para el servidor con suficiente memoria RAM (por ejemplo 4 GB) y 12 PCs obsoletos (Pentium I, Pentium II), podemos tener un aula de informática de 12 PCs por poco más que el precio de uno. El puesto de trabajo del profesor podría ser el propio servidor.
- **Aula de bajo mantenimiento a nivel software:** Como todos los PCs se conectan al servidor y ejecutan sus aplicaciones, solo es necesario instalar y configurar aplicaciones en el servidor. Lo mismo ocurre con las impresoras y el acceso a Internet, solo hay que configurarlo en el servidor. Solamente hay que mantener un PC para que funcionen los 12.
- **Aula de bajo coste de actualización:** A medida que pasa el tiempo, los PCs se van quedando obsoletos y es necesario ir ampliando el disco duro, la memoria, y cuando se quedan pequeños, hay que cambiar el PC completo. En éste caso solo será necesario aumentar las prestaciones del servidor.
- **Datos más seguros:** Al quedar todos los documentos de los usuarios almacenados en el servidor, el acceso no autorizado a datos de otros usuarios es más dificil ya que requiere el acceso físico al servidor. La realización de copias de seguridad es más sencilla ya que todos los documentos de los

usuarios se encuentran en el servidor.

También tiene **algún inconveniente**, por ejemplo la utilización de los medios de almacenamiento locales (disquetera, discos usb) es compleja, de hecho se desaconseja. Como las aplicaciones se ejecutan en el servidor, si introducimos un disquete en el terminal, el servidor no va a poder acceder a su contenido. Lo mismo ocurre con los pendrives, además los pentium I no disponen de puerto USB.

La solución que se suele adoptar en un sistema de éste tipo para el acceso físico a los documentos, es utilizar una herramienta web tipo blog, wiki o portal, de forma que los usuarios tienen acceso a sus documentos vía web. Con una herramienta web adecuada, los usuarios podrán subir y bajar archivos, así como crear o eliminar carpetas de una forma sencilla. Para almacenar sus documentos en un pendrive, los usuarios deberán ir a un terminal con puerto USB y descargar sus documentos desde la web.

Hay quienes piensan que esta forma de trabajo es ventajosa ya que podremos acceder a nuestros documentos desde cualquier PC de la intranet, y si el servidor web es accesible desde fuera, desde cualquier PC de Internet.

Si el servidor de terminales es un potente servidor con una gran cantidad de memoria, podrá satisfacer las necesidades de un gran número de usuarios de forma simultánea. En algunos centros educativos están proliferando este tipo de sistemas ya que con un servidor que disponga de 8 GB de memoria RAM y discos duros rápidos, se pueden conectar unos 30 usuarios simultáneos desde terminales (que pueden ser PCs obsoletos como Pentium I) y disfrutar de las prestaciones de un PC actual y ejecutar cualquier aplicación que haya instalada en el servidor, además el único equipo que hay que mantener es el servidor con lo cual ahorramos costes de mantenimiento. El proyecto LTSP (http://www.ltsp.org en inglés) están enfocados a implantar este tipo de sistemas en centros educativos.

El único requisito que deben cumplir los PCs que hacen de terminales es disponer de tarjeta de red y disponer de servidor gráfico X. Todos los Linux disponen de servidor gráfico X. Una distribución de Linux ligera apta para ser usada en los terminales es Damn Small Linux (http://www.damnsmalllinux.org). También existen servidores gráficos X para sistemas operativos Microsoft Windows, algunos de pago como X-win32 y libres como Xming o como cygwin (http://www.cygwin.com) que mas que un servidor X para Windows es casi un Unix para Windows.

Conectando al servidor de terminales

Uno de los problemas que nos encontramos en el centro es que tanto el profesorado como el personal de administración pretende utilizar el servidor como un cliente más. Es muy frecuente que nos pregunten el motivo de tener un equipo con una contraseña que únicamente conoce una persona y por lo tanto no puede ser utilizado ni para imprimir un documento. El personal de administración, escaso de recursos informáticos, considera un derroche económico tener un equipo parado. La realidad es que el servidor es el equipo más importante de la estructura que estamos creando y por ello debe estar ubicado en un lugar de difícil acceso, aunque ello dificultará nuestras tareas de configuración y mantenimiento. Pero eso no es problema ya que de forma remota disponemos de pleno acceso al servidor aunque no estemos situados físicamente sobre la consola del mismo.

Conexión remota mediante ssh

SSH es un protocolo que, entre otras cosas, permite establecer una conexión en modo texto desde un terminal hacia un servidor, pero si en el terminal disponemos de un servidor gráfico X, sería posible incluso ejecutar aplicaciones gráficas en el servidor y redirigir la salida hacia el servidor grafico de nuestro terminal, logrando una conexión gráfica remota.

Para realizar una conexión gráfica remota mediante ssh necesitamos tan solo dos elementos:

- Un servidor con el servicio ssh corriendo.
- Un terminal que disponga de servidor grafico X y de cliente ssh.

Como terminal gráfico X sirve un PC que tenga cualquier distribución de linux en modo gráfico. Incluso existe la posibilidad de conectar desde sistemas operativos Microsoft Windows si instalamos cygwin. Los pasos a realizar son dos:

- Establecer la conexión con SSH desde el terminal al servidor.

- Ejecutar una aplicación gráfica.

El servidor ssh deberá tener activada la redirección del protocolo X (lo está por defecto), es decir, deberá tener el siguiente parámetro en el archivo de configuración /etc/ssh/sshd_config:

```
// Habilitar la redirección X en /etc/ssh/sshd_config
     X11Forwarding yes
```

Para arrancar el servidor ssh debemos ejecutar:

```
// Arrancar el servidor ssh
     # /etc/init.d/ssh start
```

Nota: Para más información sobre el servicio ssh, consultar el apartado 'Otros servicios'

Una vez dispongamos de un servidor ssh funcionando, desde el cliente podremos iniciar sesión en el servidor mediante el comando ssh (cliente ssh) y una vez iniciada sesión en el servidor, podemos lanzar cualquier aplicación gráfica que se visualizará en la pantalla del cliente.

Por ejemplo, supongamos que en nuestro terminal tenemos una versión reducida de Linux como Damn Small Linux (http://www.damnsmalllinux.org) y deseamos conectarnos a otro PC que tiene instalado Knoppix y ejecutar el editor gráfico gimp, los pasos que haremos serán:

```
// Conexión gráfica remota por ssh
     [dsl]$ ssh -X pepe@192.168.0.50 // Nos conectamos por ssh como
     usuario...

         ...pepe y añadimos la opción -X para redirigir Xwindows.

     [knoppix]$ gimp // Ya estamos conectados. Ejecutamos el gimp
```

El resultado será que desde el terminal podemos manejar la aplicación gimp que realmente se está ejecutando en el servidor. Podemos verlo en la siguiente imagen:

Ejecución de una aplicación gráfica remotamente

Si deseamos disponer del escritorio completo, podemos ejecutar:

```
// Ejecutar escritorio gnome
     $ gnome-session
```

De esta forma tendremos en nuestro terminal un escritorio gnome del servidor.

Los terminales podrían disponer de un sistema linux mínimo configurado de manera que al arrancar se conecten automáticamente al servidor de terminales. De ésta forma los usuarios creerán que están manejando el PC en el que están sentandos aunque realmente están manejando las aplicaciones del servidor.

Se podrían conectar simultaneamente tantos usuarios como permita la memoria del servidor aunque lógicamente, cuantos más usuarios se conecten de forma simultánea, mayor será la carga del servidor y más lenta será su respuesta. Para que el servidor vaya un poco suelto, debe disponer de unos 256 MB por cada cliente. Conviene que disponga de discos duros rápidos e incluso en sistema RAID 1 (espejo) para mayor seguridad y rapidez.

13.- VNC

VNC es un servicio que crea servidores gráficos sobre pantallas o displays virtuales y permite establecer conexiones remotas desde otros PCs de la red al servidor, de forma gráfica de manera similar a si fuera un servidor de terminales. La diferencia más significativa con respecto a un servidor de terminales Xwindow como el que hemos visto en el punto anterior es que mientras cuando hacemos una conexión Xwindow el cliente debe disponer de un servidor gráfico, cuando hacemos la conexión con VNCServer, la imagen gráfica se genera en el servidor y básicamente lo que fluye por la red son pantallazos jpg, de esa forma el cliente puede ser más ligero pero la carga del servidor es mucho mayor.

Para que pueda funcionar es necesario instalar y ejecutar el servidor VNC. Este servidor atenderá las peticiones de los clientes. El terminal deberá disponer del cliente de VNC llamado **vncviewer** del que hay versiones para todos los sistemas operativos incluidos MS-DOS, Linux y Microsoft Windows. En PCs obsoletos que se deseen utilizar como terminales, se podría instalar la versión para MS-DOS del cliente VNC. En http://www.veder.com/nwdsk/index.html existen imágenes de disquetes basadas en Free-DOS que configuran la tarjeta de red y dispone de un cliente VNC para DOS. También se podría instalar una versión de linux reducida como DSL.

Cuando ejecutamos el servidor de VNC, se crea un nuevo escritorio (nuevo display X) al cual se puede acceder de forma remota con el cliente de VNC. Se pueden ejecutar tantos servidores VNC como permita la memoria del sistema, pudiendo varios usuarios acceder de forma simultánea, cada uno a su escritorio independiente, al contrario que la versión del servidor VNC para Windows que sólo permite acceder al escritorio principal. Podemos forzar la introducción de una contraseña para permitir el acceso vía VNC al servidor.

En la estación de trabajo donde se ejecute el visor de VNC, éste aparece como una ventana en el entorno de escritorio local, presentando la interfaz de usuario; todas las funciones del S.O., así como las aplicaciones, se ejecutan en el servidor.

Instalación y configuración del servidor VNC

A pesar de que disponemos de otras aplicaciones de acceso remoto al servidor (ssh, free nx server), nos han comentado las bondades del programa VNC, que puede ser ejecutado en sistemas Windows y Linux. Sabemos que para el servicio que necesitamos sus funcionalidades son similares; más aun, pues con VNC podemos conectarnos al servidor mediante el cliente VNC o mediante el navegador de nuestro sistema operativo.

Instalación del servidor VNC

Para disponer de servidor VNC, instalaremos el paquete **tightvncserver**. Dicho paquete se encuentra en el repositorio 'universe' de Ubuntu. Haz clic aquí para saber cómo activar dicho repositorio. Una vez activado el repositorio 'universe', para instalar la última versión del servidor vnc debemos ejecutar desde una consola de root el siguiente comando:

```
// Instalación de vncserver
     # apt-get install tightvncserver
```

Puesta en marcha del servidor VNC

Para que se pueda acceder al servidor de forma remota mediante un cliente VNC, primero es necesario que en el servidor se esté ejecutando tightvncserver.

Al ejecutar tightvncserver, se crea un servidor gráfico en un display virtual al que se puede acceder remotamente desde otros PCs de la red que dispongan del cliente VNC.

La primera vez que ejecutemos tightvncserver en el servidor, nos pedirá que proporcionemos una

contraseña que será la contraseña que deberán utilizar los clientes para conectarse. Ésta contraseña se puede cambiar en cualquier momento ejecutando el comando 'vncpasswd' en el servidor.

Vamos a crear un servidor gráfico, para ello podríamos ejecutar por ejemplo:

```
// Creación de un servidor grafico
     # tightvncserver :1 -geometry 800x600 -depth 24
```

Con el comando anterior estaríamos creando un nuevo servidor gráfico en un display virtual cuyo número de display será el :1, su tamaño será de 800 x 600 píxels y una profundidad de color de 24 bits/pixel (true color).

Si hemos lanzado el comando tightvncserver con el usuario root, cuando alguien se conecte de forma remota, accederá como root. Si hubiéramos lanzado el comando con el usuario pepe (por ejemplo), cuando alguien se conecte de forma remota, lo hará como usuario pepe.

Destrucción de un servidor gráfico VNC

Cada vez que ejecutamos el comando tightvncserver, se crea un nuevo escritorio que puede ser accedido remotamente. Dichos escritorios consumen una cantidad considerable de memoria en el servidor, por lo que solo debemos crear los que necesitemos. Si hemos creado más de los necesarios, podemos destruirlos mediante el comando tightvncserver indicando el número del servidor a destruir, precedido por dos puntos:

```
// Destrucción de un servidor gráfico VNC
     # tightvncserver -kill :1
```

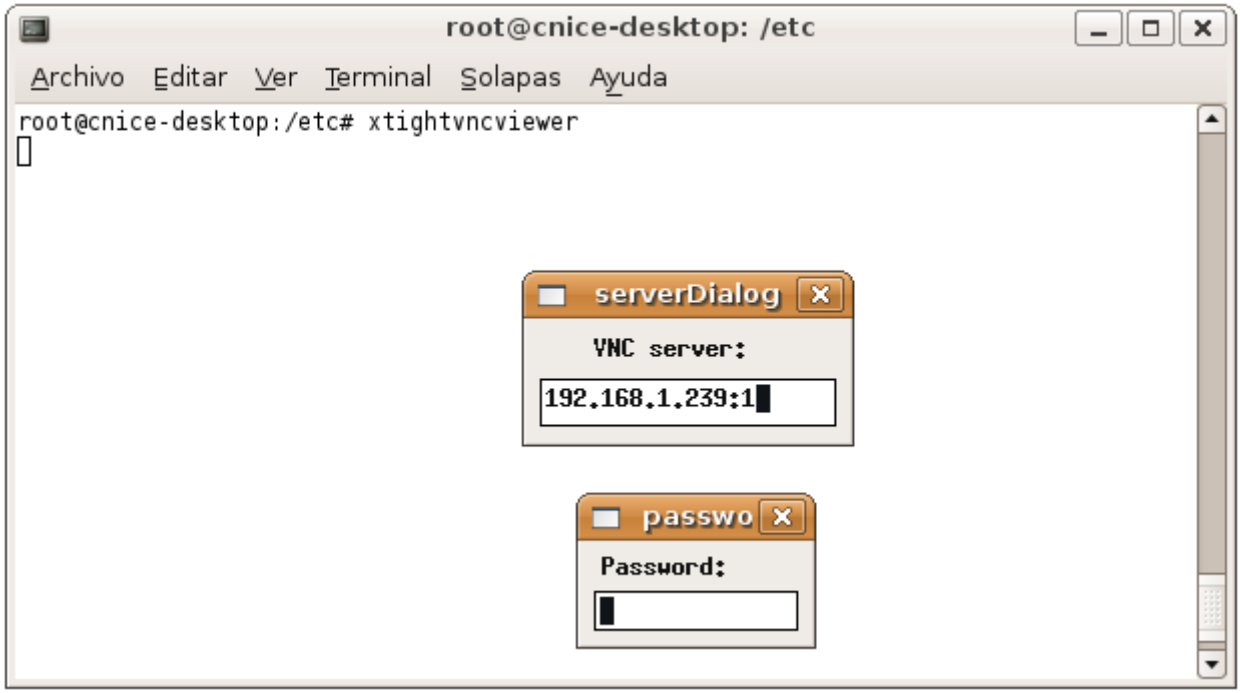
Conectando al servidor VNC

Conexión con cliente vnc

Para conectar al servidor VNC necesitamos un cliente VNC, como por ejemplo **vncviewer**. En debian podemos instalarlo directamente con apt-get ejecutando:

```
// Instalación del cliente VNC
     # apt-get install xtightvncviewer
```

Una vez instalado el cliente, tan solo debemos ejecutarle y proporcionarle la IP del servidor, seguido de dos puntos ':' y seguido del número de display, ejemplo 192.168.1.239:1 si la dirección IP del servidor fuera la 192.168.1.239 y el número de display fuera 1. Acto seguido nos pedirá la contraseña de acceso que pusimos al instalar el servidor. Dicha contraseña se puede especificar ejecutando el comando 'vncpasswd' en el servidor.



Ejecución del cliente de VNC

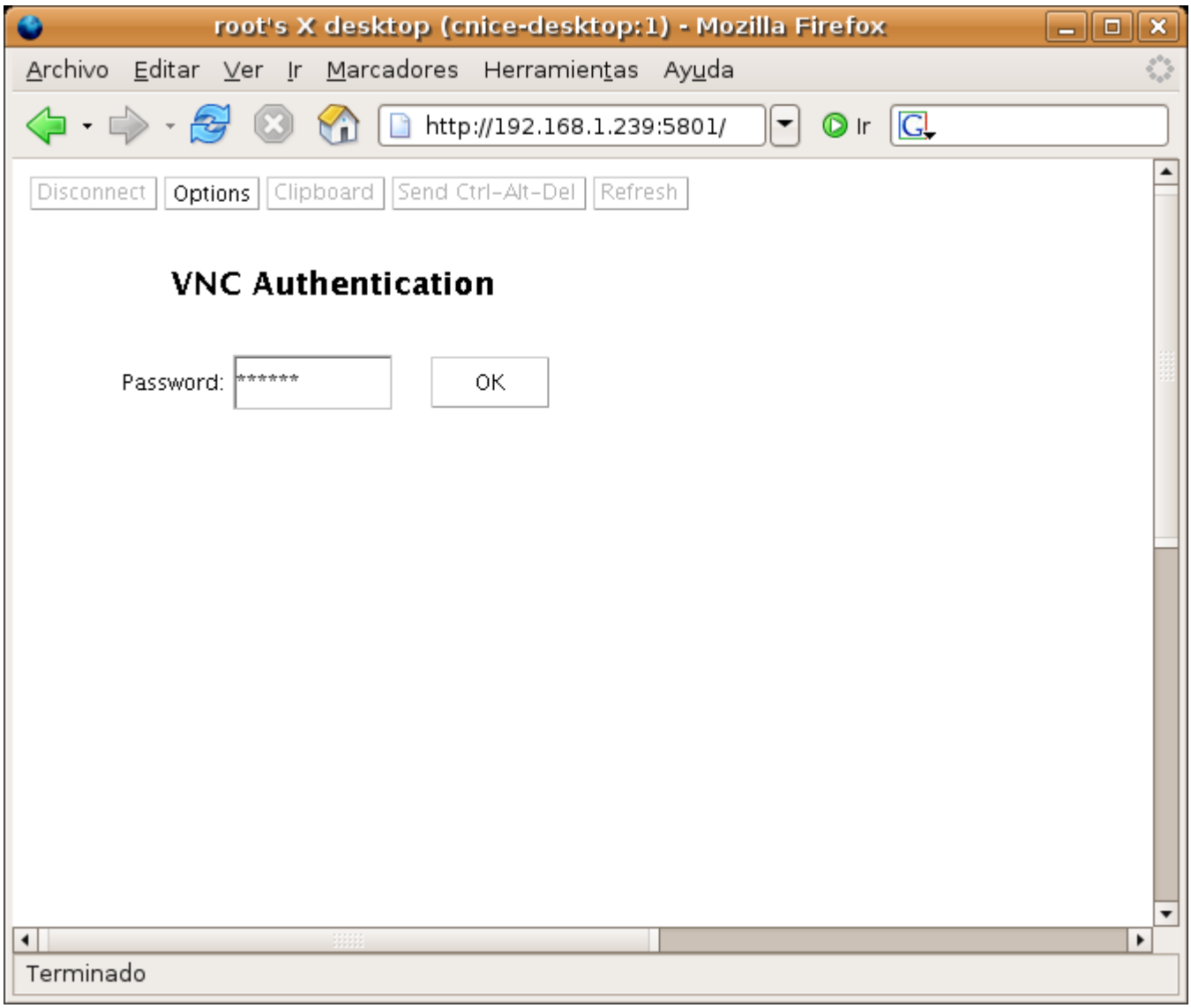
Conexión vía web

Otra forma más sencilla de conectar a un servidor vnc es utilizando un navegador web que disponga de máquina virtual java. Con éste método de conexión no es necesaria la instalación del cliente vnc ya que yendo a la dirección http://ip_del_servidor:580x (x = display) podremos acceder al display desde el navegador.

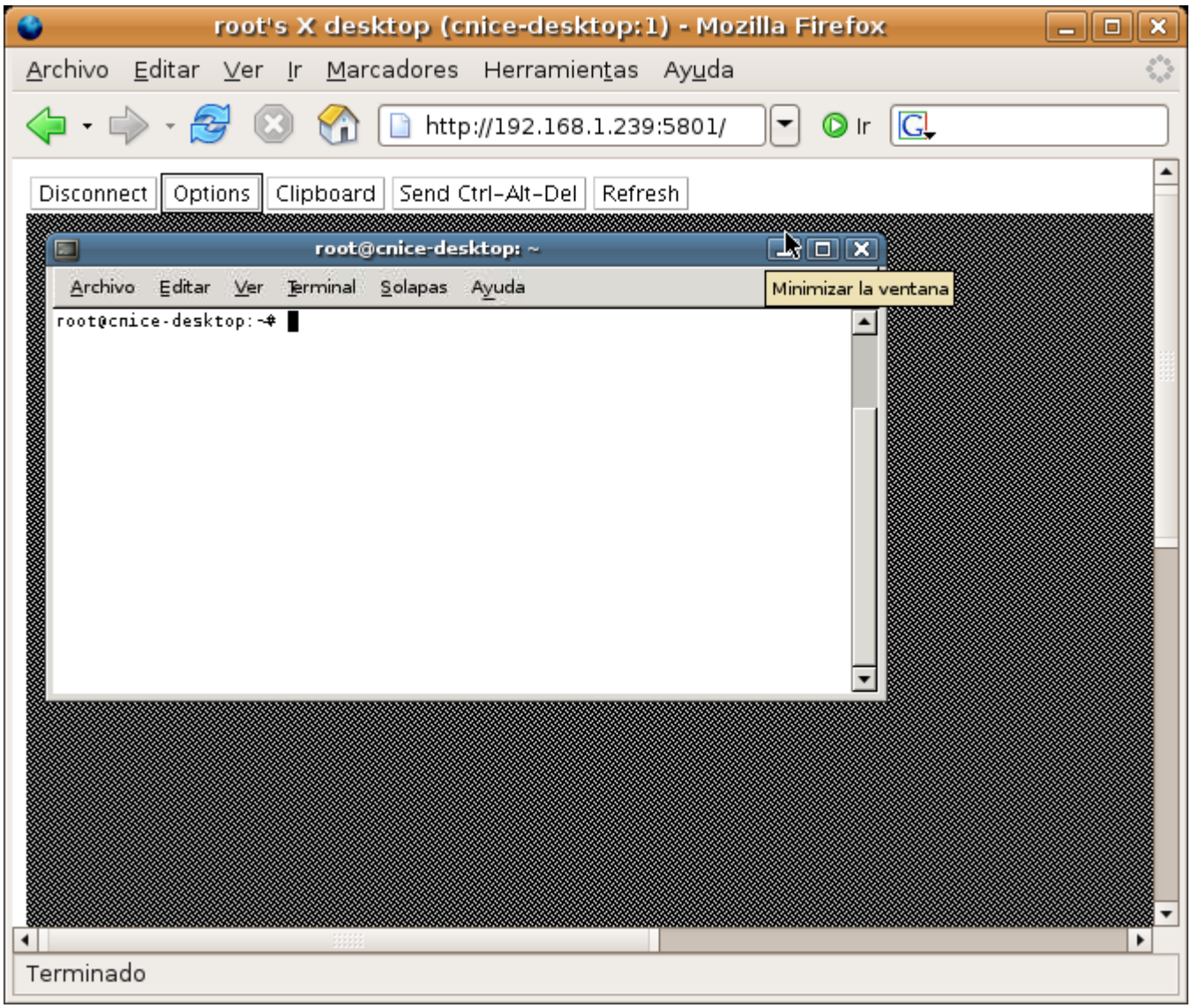
Para que sea posible acceder al servidor vnc por web es necesario instalar el componente java en el servidor ejecutando el siguiente comando:

```
// Instalación de tightvnc-java
     # apt-get install tightvnc-java
```

Ejemplo, supongamos que hemos creado el display nº 1. Si vamos a http://ip_del_servidor:5801 podremos acceder. Primero deberemos introducir la contraseña.



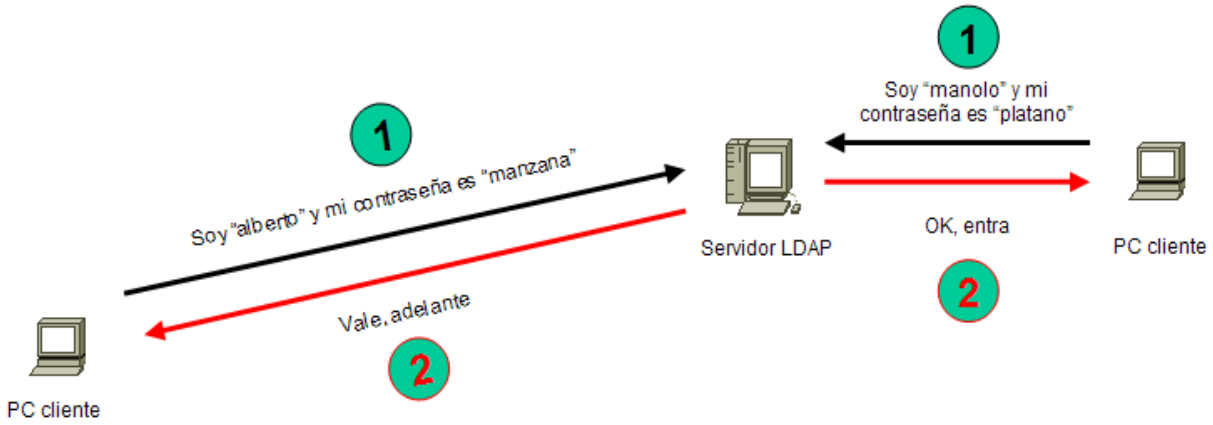
Acto seguido accederemos al escritorio de igual forma que si utilizáramos el cliente vnc.


root's X desktop (cnice-desktop:1) - Mozilla Firefox
Archivo Editar Ver Ir Marcadores Herramientas Ayuda
http://192.168.1.239:5801/
Ir
Disconnect
Options
Clipboard
Send Ctrl-Alt-Del
Refresh
root@cnice-desktop: ~
Archivo Editar Ver Terminal Solapas Ayuda
Minimizar la ventana
root@cnice-desktop:~#
Terminado

14.- Servidor LDAP

Un servidor LDAP es un servidor de datos optimizado para la realización rápida de consultas de lectura y orientado al almacenamiento de datos de usuarios a modo de directorio.

La principal utilidad de un directorio LDAP es como servidor de autentificación para los distintos servicios de un sistema informático como puedan ser: autentificación para entrar en un PC, para entrar en una aplicación web, para acceder a un servidor ftp, para acceder a servidores de correo entrante POP3 y saliente SMTP, etc...



Si en nuestra red disponemos de un servidor LDAP y configuramos todos los PCs y todos los servicios de la red para que se autentifiquen en él, bastará con crear las cuentas de usuario y grupos de usuarios en nuestro servidor LDAP para que los usuarios puedan hacer uso del sistema y de sus servicios desde cualquier puesto de la red. Es un sistema ideal para centralizar la administración de usuarios en un único lugar.

En el curso veremos cómo poner en marcha un servidor LDAP y cómo configurar el resto de PCs clientes de la red para que se autentifiquen en él. También utilizaremos OpenSSL para que durante el proceso de autentificación los datos viajen encriptados por la red, así ningún curioso podrá averiguar nuestras contraseñas. Además utilizaremos LDAP para que autentifique el acceso al servidor ftp y el acceso a páginas restringidas en el servidor web.

Instalación y configuración de OpenLDAP

Para simplificar la administración de los usuarios del sistema es ideal utilizar una base de datos accesible mediante LDAP. Almacenar las cuentas de usuario de forma centralizada en un único repositorio facilitará la creación, modificación y eliminación de cuentas de usuario y grupos de usuarios. Será necesario configurar los PCs de la red para que utilicen el servidor LDAP como servidor de autentificación.

Instalación de OpenLDAP

El servidor OpenLDAP está disponible en el paquete **slapd** por tanto, lo instalaremos utilizando apt-get. También nos conviene instalar el paquete **db4.2-util** que son un conjunto de utilidades para la base de datos dbd que es la que utilizaremos para nuestro servidor ldap y el paquete **ldap-utils** que contiene utilidades adicionales:

```
// Instalación del servidor LDAP
     # apt-get install slapd db4.2-util ldap-utils
```

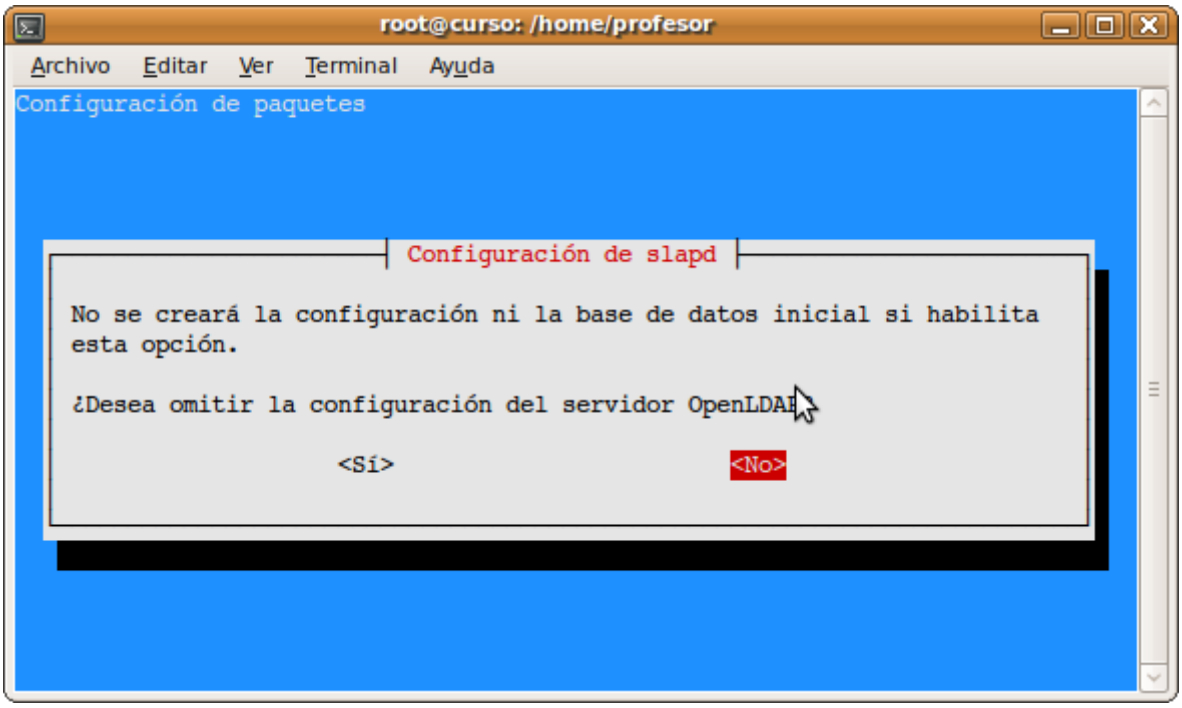
Durante la instalación, nos pedirá que introduzcamos la contraseña de administrador del servidor ldap. Podemos configurar cualquier contraseña, como por ejemplo 'ldapadmin'

Configuración de OpenLDAP

La configuración del servidor LDAP se almacena en la carpeta /etc/ldap/slapd.d/cn=config/. En lugar de editar manualmente los archivos de configuración, es mejor lanzar el asistente de configuración de slapd. Para ello debemos ejecutar el siguiente comando:

```
//Lanzar el asistente de configuración de slapd
     # dpkg-reconfigure slapd
```

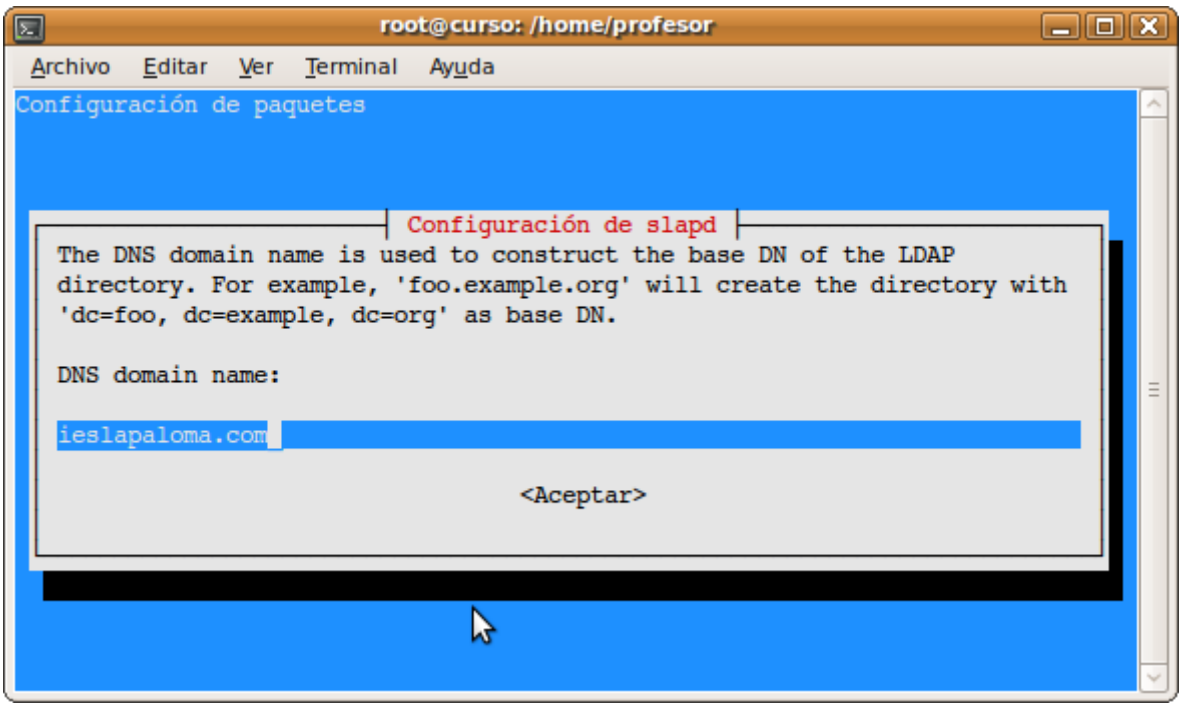
Lo primero que nos pregunta el asistente es si deseamos omitir la configuración del servidor LDAP:



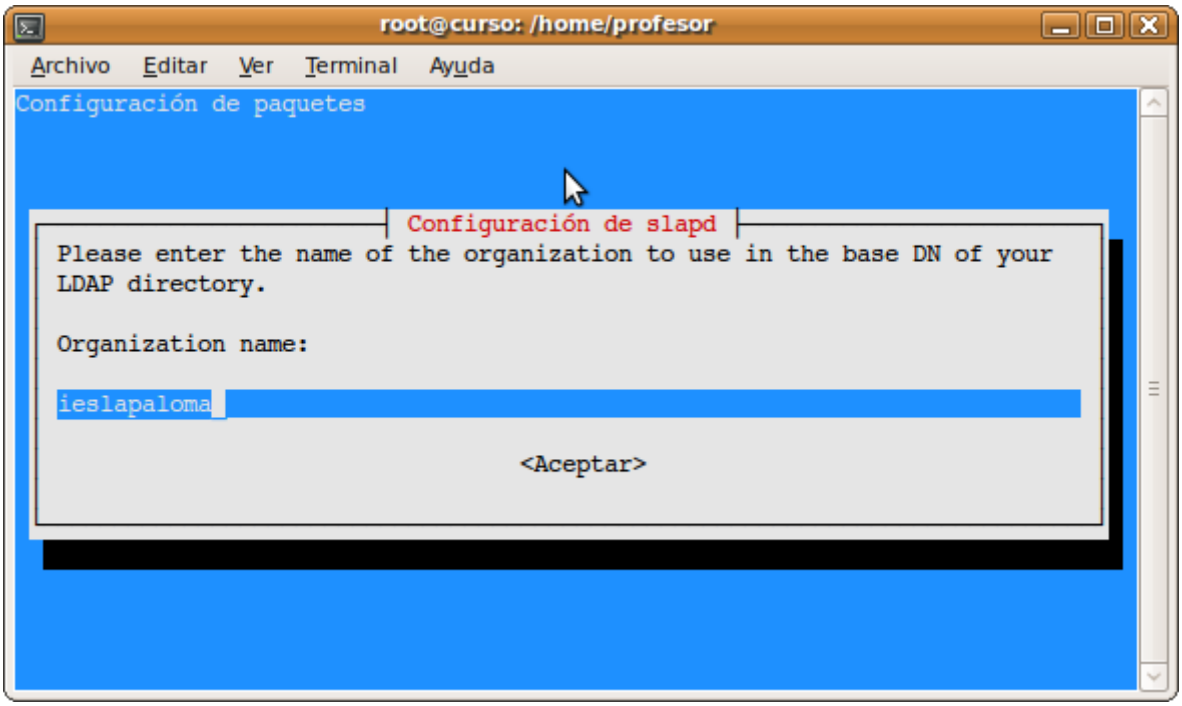
Obviamente responderemos que no, ya que precisamente lo que queremos es configurar el servidor LDAP.

Nuestro directorio LDAP debe tener una base, a partir de la cual cuelgan el resto de elementos. Como nombre de la base, habitualmente se utiliza el nombre del dominio. Ejemplo, si nuestro dominio es ieslapaloma.com, lo normal es que la base para nuestro directorio LDAP sea: dc=ieslapaloma,dc=com.

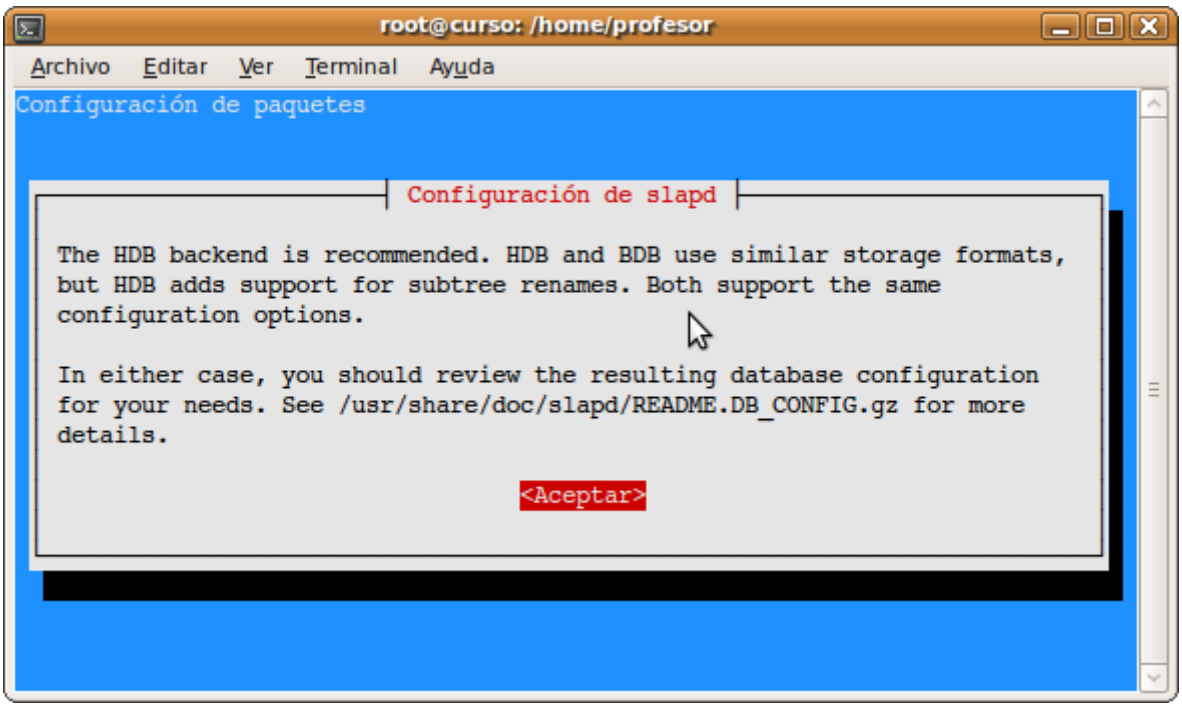
La siguiente pregunta que nos hace el asistente es el nombre de nuestro dominio. Éste nombre lo utilizará para crear el nombre distinguido (DN) o dicho más claramente, nombre identificativo de la base de nuestro directorio LDAP.



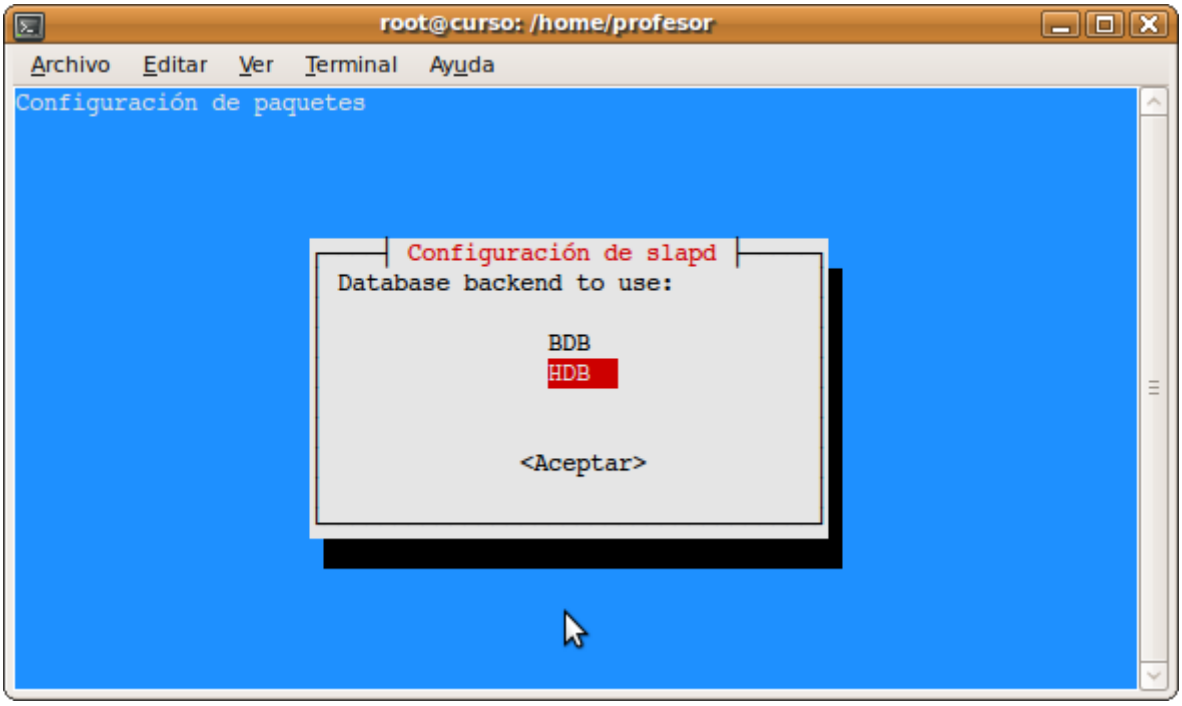
Posteriormente nos preguntará por el nombre de nuestra organización.



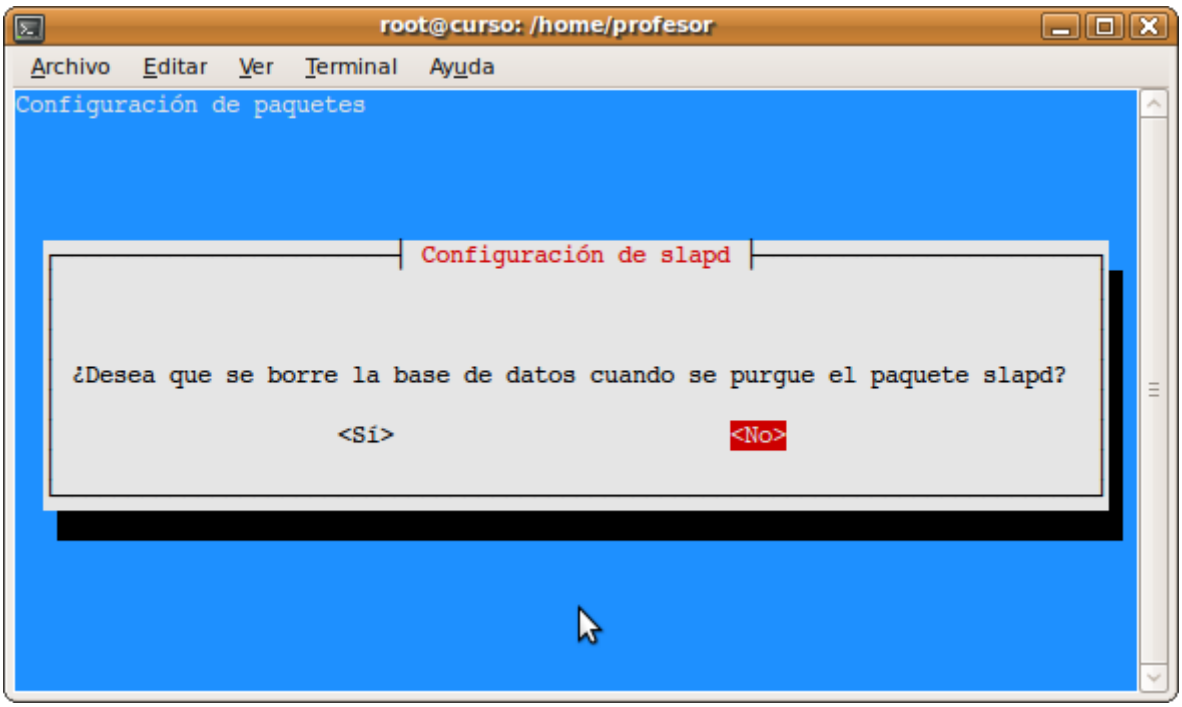
Tras poner el nombre de nuestra organización, nos recomendará la utilización del formato de base de datos HDB en lugar del sistema antiguo BDB:



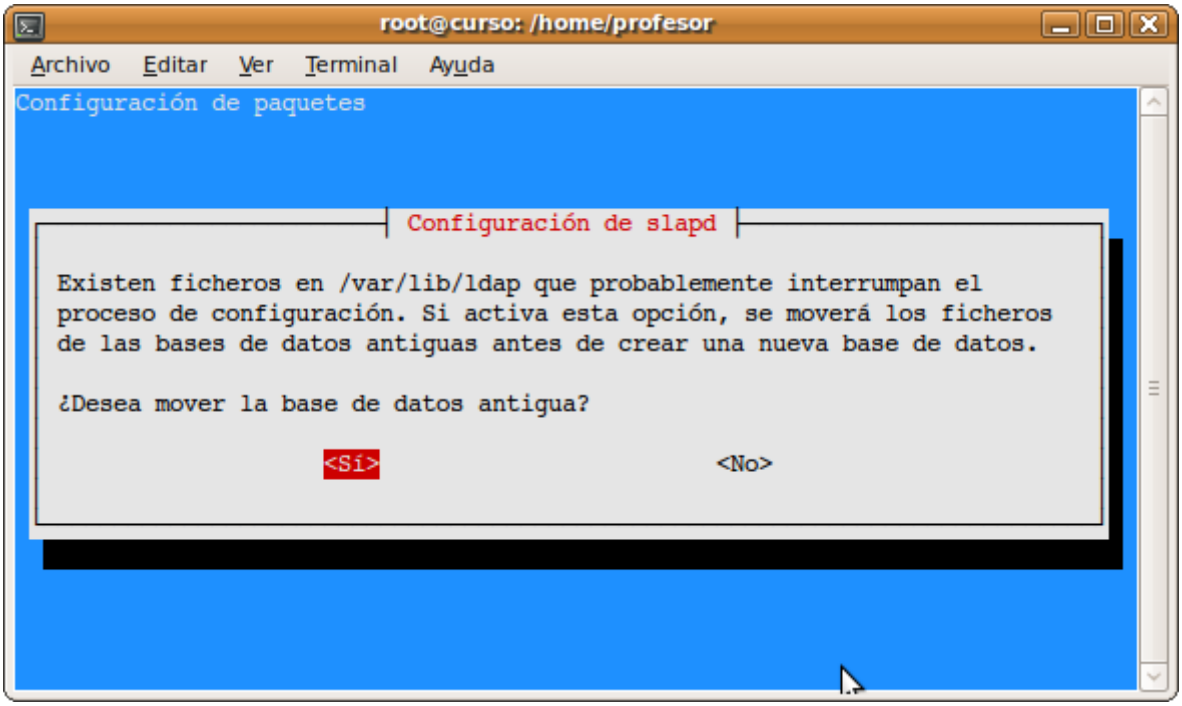
Acto seguido nos informará sobre los posibles gestores de datos para almacenar el directorio y en la siguiente ventana nos preguntará qué sistema utilizar. Lo recomendable es utilizar el sistema HDB.



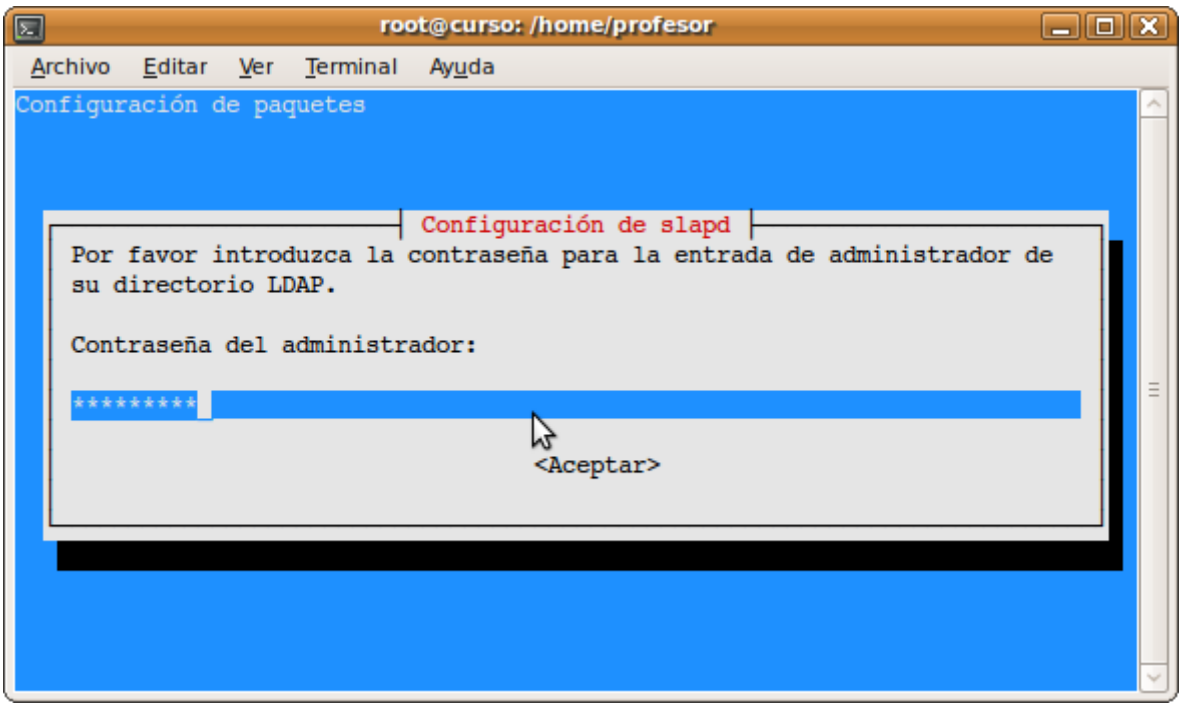
Después nos preguntará si queremos que se elimine la base de datos cuando quitemos slapd. Por si acaso, lo mejor es responder que no:



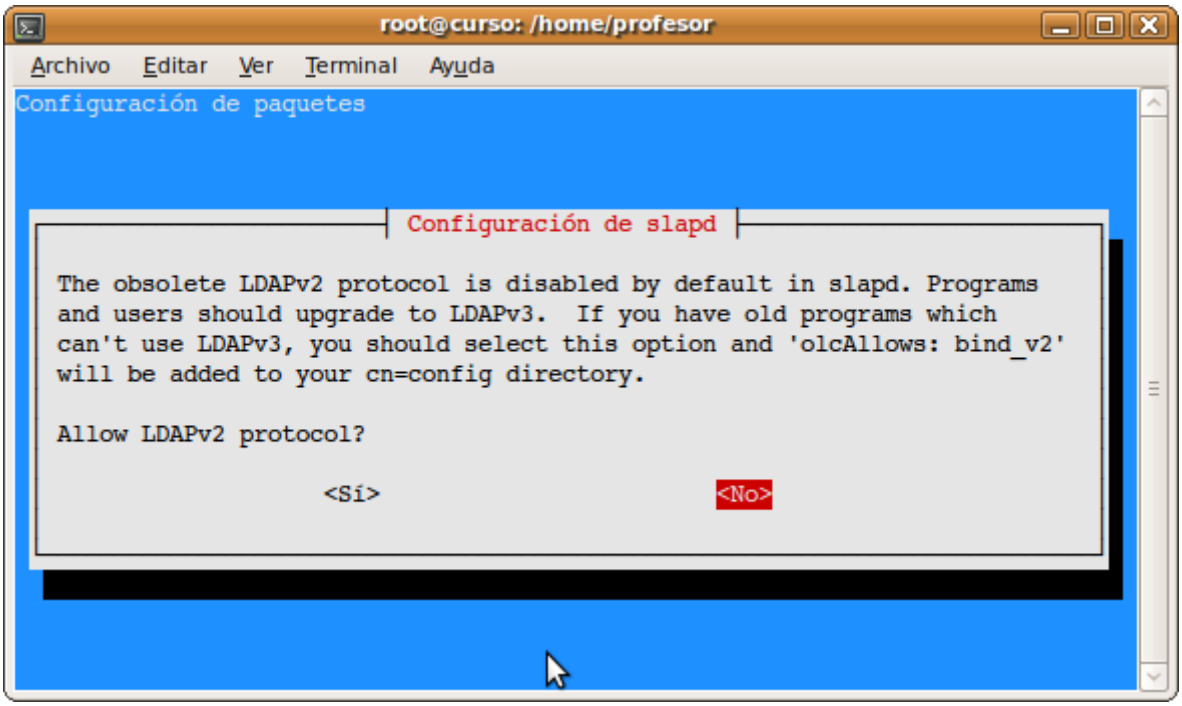
En el caso de que exista una base de datos LDAP previa, nos preguntará si deseamos moverla. Lo mejor es responder Sí, para evitar que interfiera en la nueva base de datos:



Después nos preguntará por la contraseña que deseamos poner al usuario admin (administrador) del servidor LDAP. Dicha contraseña nos la pedirá dos veces para evitar errores de tecleo. Podemos poner cualquier contraseña, por ejemplo 'ldapadmin'.



Luego nos preguntará si deseamos utilizar LDAP versión 2, respondemos que no ya que apenas se utiliza.



Y ya tendríamos nuestro servidor LDAP listo para trabajar con él.

Arranque y parada manual del servidor LDAP

El servidor LDAP, al igual que todos los servicios en Debian, dispone de un script de arranque y parada en la carpeta /etc/init.d.

```
// Arrancar o reiniciar el servidor LDAP
```

```
     root@cnice-desktop:# /etc/init.d/slapd restart


// Parar el servidor LDAP
     root@cnice-desktop:# /etc/init.d/slapd stop
```

Arranque automático del servidor LDAP al iniciar el sistema.

Para un arranque automático del servicio al iniciar el servidor, debemos crear los enlaces simbólicos correspondientes tal y como se indica en el apartado Arranque automático de servicios al iniciar el sistema.

Administración de OpenLDAP

Introducción

Una vez instalado y configurado el servidor LDAP, la siguiente tarea es la del diseño de la estructura y la introducción de datos en el directorio.

Puesto que la finalidad de nuestro servidor LDAP es que sirva de almacen de usuarios y grupos para autentificar sistemas linux y servicios como ftp y web, deberemos crear una estructura que parta de la base de nuestro directorio, para almacenar dicha información. Tal y como se explica más abajo, crearemos una unidad organizativa (ou) llamada **groups**, para almacenar los grupos de usuarios y crearemos otra unidad organizativa llamada **users** para almacenar a los usuarios.

Explorador de directorios LDAP

Para acceder al directorio LDAP y poder crear y modificar elementos en dicho directorio, es necesario disponer de un explorador de directorios LDAP (LDAP browser). Existen muchos exploradores LDAP tanto de pago como libres. Entre las aplicaciones libres destacamos gq, phpldapadmin (aplicación web) y JXplorer.

Para instalar gq, podemos utilizar apt-get. Una vez instalada, para ejecutar gq tan solo debemos pulsar alt+f2 y escribir gq.

Para instalar phpldapadmin, al igual que otras aplicaciones web, deberemos descargarla desde http://phpldapadmin.sourceforge.net/ y descomprimirla dentro del DocumentRoot de apache, es decir, dentro de la carpeta /var/www, por ejemplo en /var/www/phpldapadmin. Para ejecutarla, si la hemos descomprimido en la carpeta anterior, debemos ir a http://ip_del_servidor_web/phpldapadmin/ con el navegador y veremos la página principal de la aplicación:

JXplorer - Explorador LDAP en java.

Por su calidad superior, en este curso utilizaremos JXplorer para administrar el directorio LDAP.

Instalación de JXplorer

Previo a instalar jxplorer, es necesario instalar la máquina virtual java de Sun, para lo cual utilizaremos apt-get:

```
// Instalación de Java
     # apt-get install sun-java6-bin sun-java6-jre sun-java6-plugin
```

El comando anterior instalará java en la carpeta /usr/lib/jvm/java-6-sun/jre/bin/. Posteriormente tendremos que editar el archivo /root/.bashrc y añadir las variables que permitan al shell encontrar los binarios del JRE:

```
// Añadir en /root/.bashrc
     CLASSPATH=/usr/lib/jvm/java-6-sun/jre/bin/

     JAVA_HOME=/usr/lib/jvm/java-6-sun/jre/bin/

     PATH=/usr/lib/jvm/java-6-sun/jre/bin:/sbin:/bin:/usr/sbin
     :/usr/bin:/usr/bin/X11:/usr/local/sbin:/usr/local/bin
```

Una vez instalado el java y establecidas las variables CLASSPATH, JAVA_HOME y PATH en el archivo

/root/.bashrc, debes cerrar el terminal y volver a abrirlo, para que cargue las variables de entorno. Si ejecutas el comando set en el terminal, podrás comprobar que ha cargado las variables de entorno y podrás instalar JXplorer. JXplorer no está disponible en los repositorios de paquetes de debian, pero está incluido en el DVD del curso, en la carpeta /REDES_LINUX/software/jxplorer3.2_linux.bin. Debemos copiar el archivo en la carpeta /tmp de nuestro sistema y ejecutar:

```
// Instalar JXplorer
     # sh /tmp/jxplorer3.2_linux.bin
```

Se iniciará un sencillo asistente de instalación que al finalizar habrá creado un enlace en nuestra carpeta home, por lo tanto para ejecutarlo debemos escribir:

```
// Ejecutar JXplorer: Entran en la carpeta de instalación y ejecutar:
     # ./jxplorer.sh
```

Veremos la pantalla principal de JXplorer:

Conexión con el servidor LDAP

La conexión con el servidor LDAP podemos hacerla como usuario anónimo o como usuario administrador. Si conectamos de forma anónima solo podremos visualizar los elementos pero no podremos hacer cambios. Si conectamos como administrador, podremos crear, modificar y eliminar elementos de cualquier tipo.

Para conectar al servidor LDAP como administrador necesitamos la siguiente información:

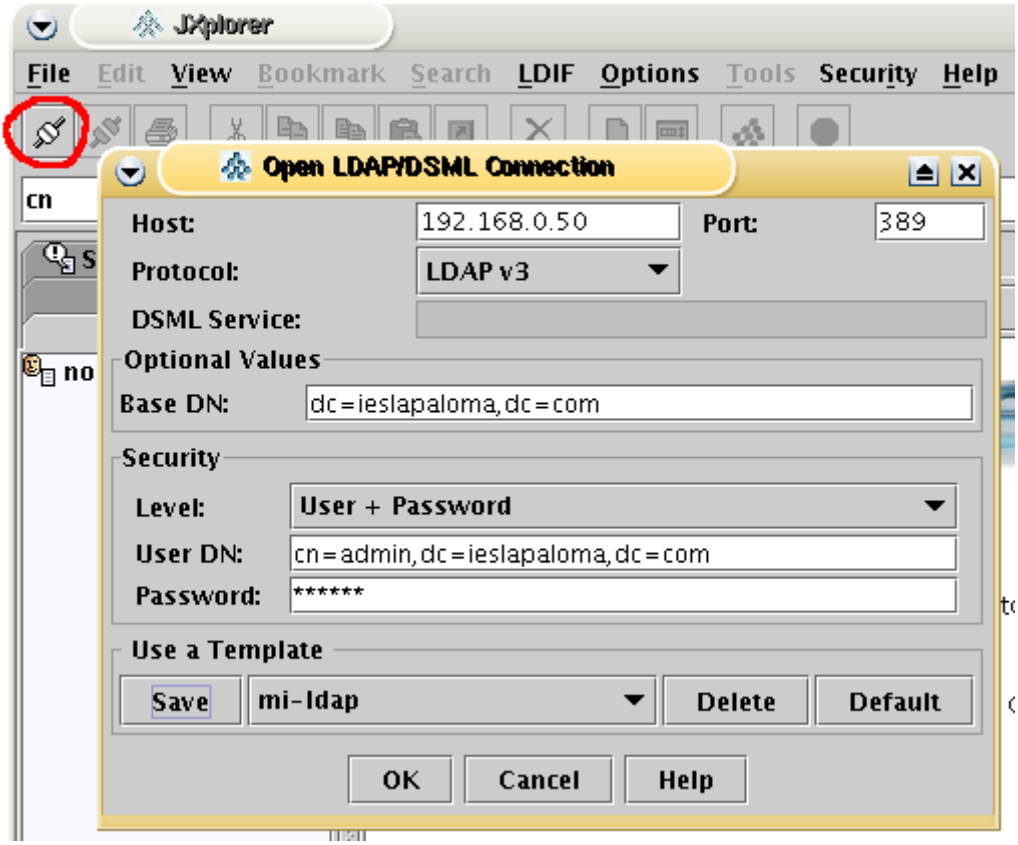
- Dirección IP del servidor LDAP
- Protocolo del servidor (LDAP v3 en nuestro caso)
- Base del directorio (dc=ieslapaloma,dc=com en nuestro caso)
- Nombre de usuario administrador (cn=admin,dc=ieslapaloma,dc=com en nuestro caso)
- Contraseña (ldapadmin en nuestro caso)

La base del directorio se suele denominar en inglés 'base DN' o 'Nombre Distinguido de la base del directorio'. Se corresponde con el parámetro 'suffix' del archivo de configuración del servidor LDAP /etc/ldap/slapd.conf.

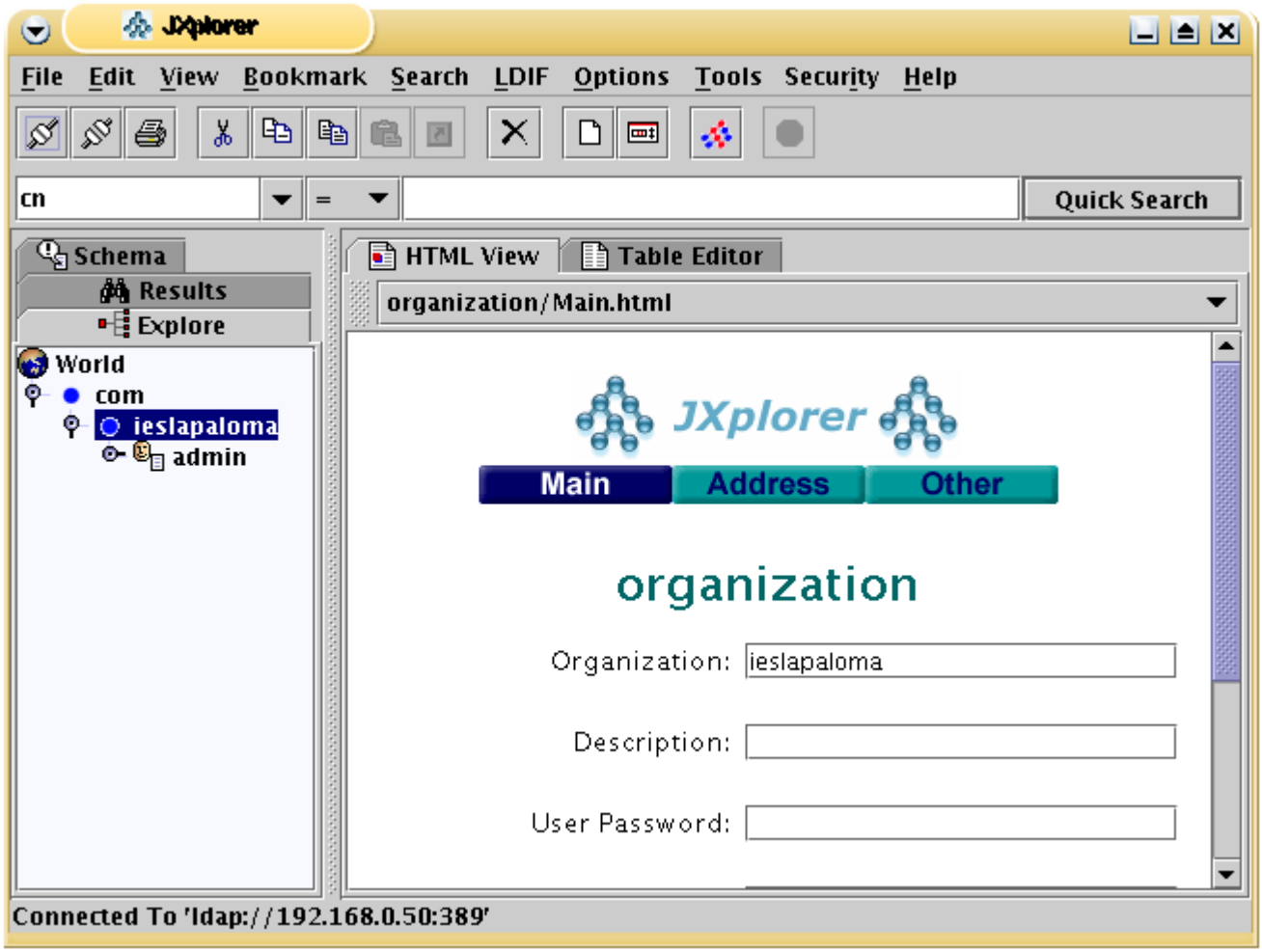
El nombre del usuario con el que nos conectamos se suele denominar en inglés 'user DN' o también 'bind DN'

El nombre de usuario administrador por defecto suele ser admin y a menudo hay que proporcionar nombre y base del directorio: cn=admin,dc=ieslapaloma,dc=com

Al hacer clic en el botón 'conectar' (marcado con círculo rojo en la figura) nos aparecerá el diálogo de conexión para que introduzcamos los datos de la conexión. Para no tener que introducir dicha información cada vez que conectemos, podemos grabar los datos pulsando 'Save'.



Si pulsamos OK, JXplorer conectará con el servidor LDAP y mostrará el directorio:



Vemos que en nuestro directorio solamente hay dos elementos: una organización llamada 'ieslapaloma' y el usuario administrador llamado 'admin'.

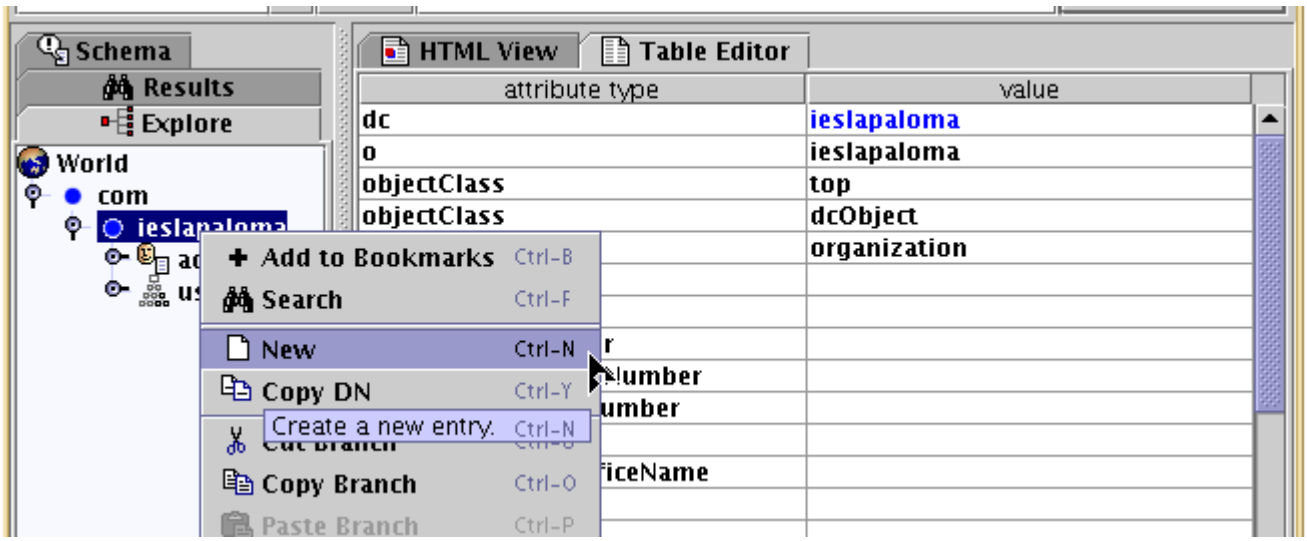
Organización del directorio LDAP

Creación de las unidades organizativas

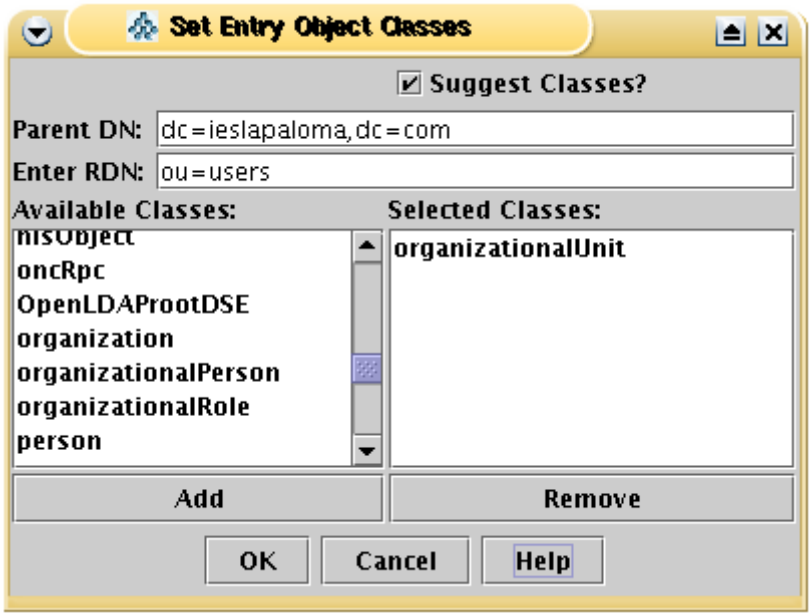
Puesto que nuestro directorio va a almacenar usuarios y grupos, vamos a crear sendas unidades organizativas (en inglés organizational unit - ou) llamadas 'users' y 'groups' que nos servirán para organizar los usuarios y los grupos por separado.

Dentro de la unidad organizativa 'users' crearemos todos los usuarios del sistema. Dentro de la unidad organizativa 'groups' crearemos todos los grupos del sistema.

Para crear una unidad organizativa dentro de nuestra organización, haremos clic con el derecho sobre la organización 'ieslapaloma' y en el menú contextual elegiremos 'New':



Nos aparecerá la ventana 'Set Entry Object Classes' que podríamos traducir por 'Seleccione las clases objeto de la nueva entrada' o mejor, 'Seleccione las tipologías'. En ella podremos elegir los 'tipos' que tendrá nuestro nuevo elemento. Como se trata de una unidad organizativa (en inglés organizational unit - ou) debemos seleccionar el tipo organizationalUnit en la lista de la izquierda y pulsar el botón añadir (Add). Los otros dos tipos que aparecen por defecto (organizationalRole y simpleSecurityObjet) no los necesitaremos, por lo tanto podemos seleccionarlos de la lista de la derecha y pulsar el botón quitar (remove). En la casilla 'Enter RDN' (introducir Nombre Distinguido Relativo) debemos poner el nombre de nuestro elemento. Escribiremos ou=users. Estaremos en la situación de la siguiente figura:



Tan solo debemos pulsar el botón OK y ya se habrá creado nuestra unidad organizativa 'users'. Repetiremos los pasos para crear otra unidad organizativa llamada 'groups'. El resultado que obtendremos será:

Usuarios y grupos

Ahora solamente nos queda crear los usuarios, crear los grupos y asignar los usuarios a sus grupos. Dentro de nuestra unidad organizativa 'groups' crearemos los siguientes grupos:

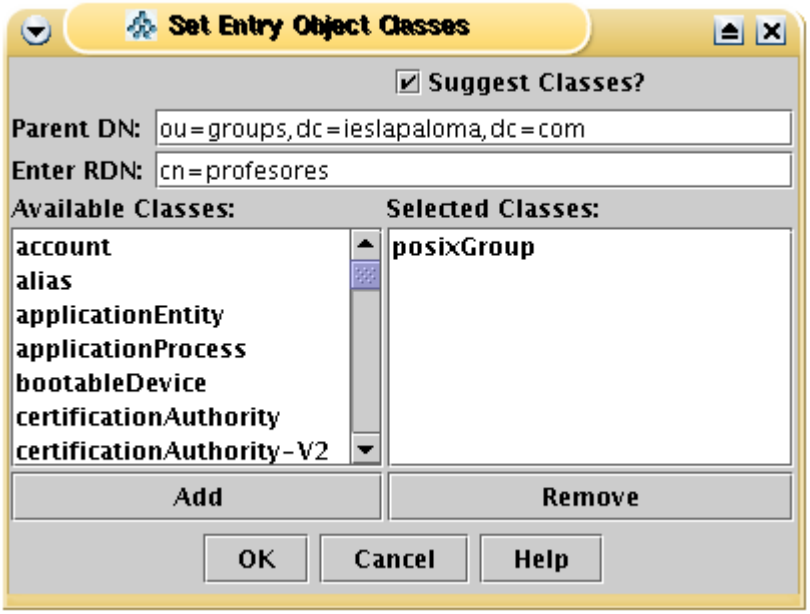
- profesores (gid=1001)
- alumnos (gid=1002)

Dentro de nuestra unidad organizativa 'users' crearemos los siguientes usuarios:

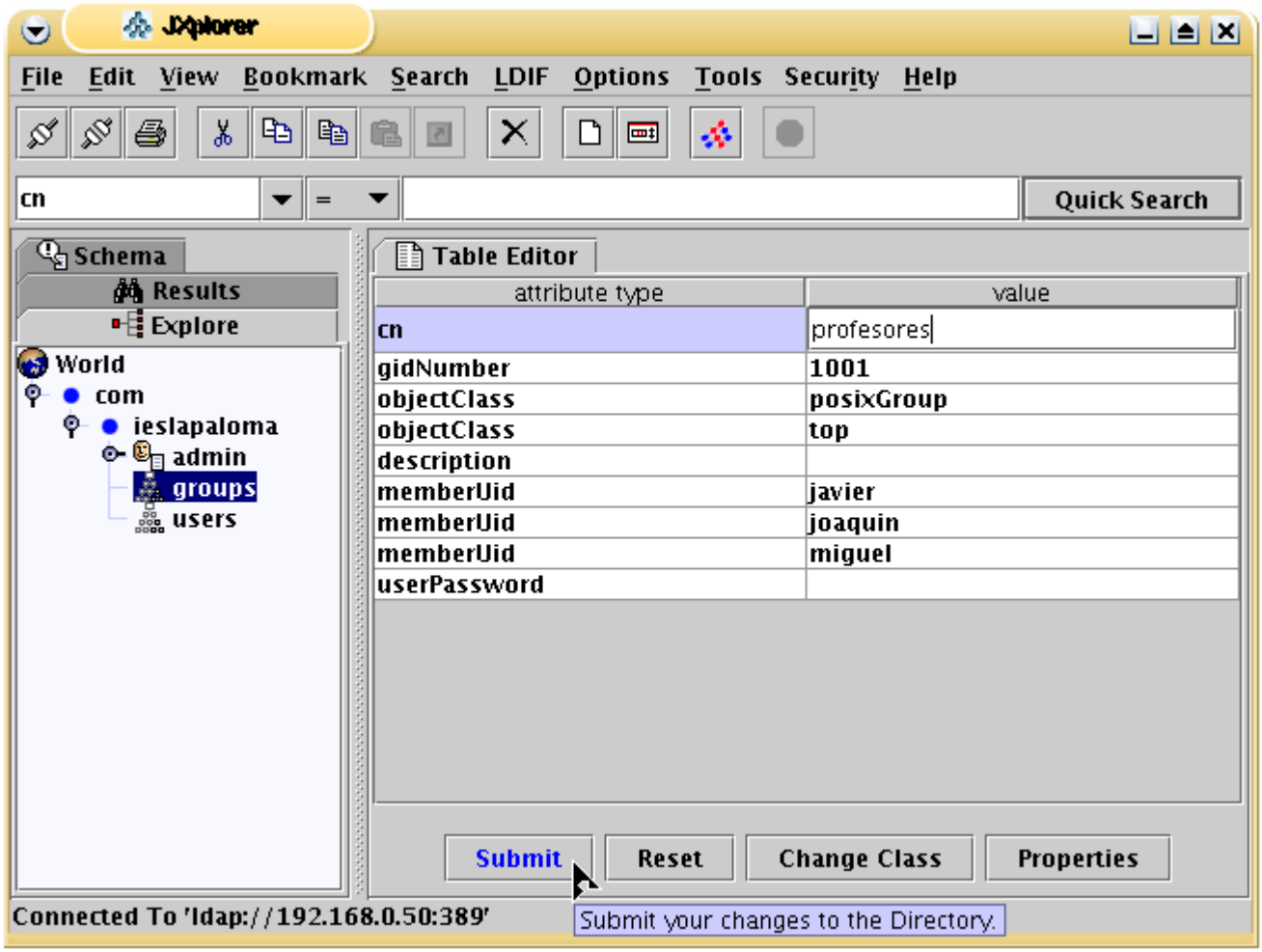
- javier (uid=1001, profesor)
- joaquin (uid=1002, profesor)
- miguel (uid=1003, profesor)
- jessica (uid=1004, alumno)
- joel (uid=1005, alumno)

Creación de grupos

Para crear los grupos, haremos clic con el derecho en la unidad organizativa 'groups' e igual que antes haremos clic en 'New'. Nuestro nuevo elemento será un nuevo grupo posix, por lo tanto debemos agregar el tipo 'posixGroup' de la lista de la izquierda. El nombre (RDN) será profesores, por tanto debemos escribir 'cn=profesores' (cn= Common Name - Nombre Común):

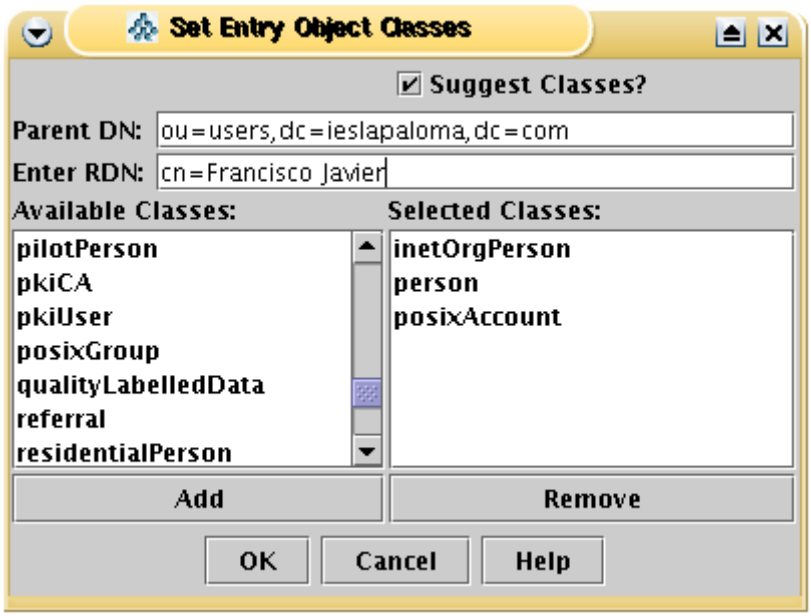


Al pulsar OK nos apacererá la siguente figura, en la cual observamos los atributos clásicos de un grupo posix. Debemos rellenar al menos el campo gidNumber. También podemos introducir miembros al grupo. En el parámetro memberUid añadimos javier. Luego, haciendo clic con el derecho en javier > Add another value, podemos añadir otro valor: joaquin. De igual manera añadiremos a miguel. No importa que todavía no hayamos creado a dichos usuarios:

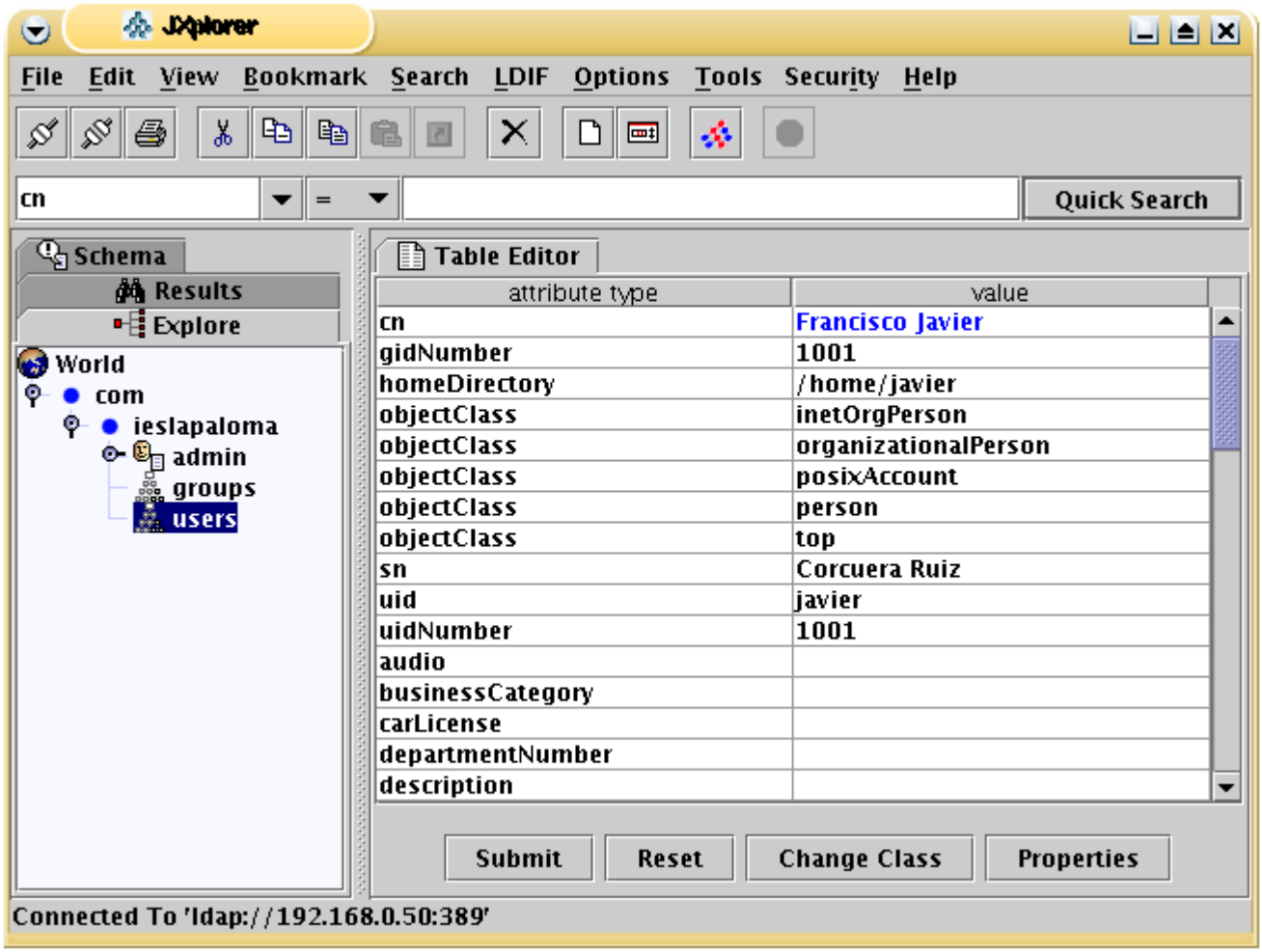


Creación de usuarios

Para crear los usuarios, haremos clic con el derecho en la unidad organizativa 'users' e igual que antes haremos clic en 'New'. Nuestro nuevo elemento será un nuevo usuario posix, por lo tanto debemos agregar el tipo 'posixAccount' de la lista de la izquierda. Pero nuestro usuario también será una persona, por eso nos interesará agregar el tipo 'person' para disponer de los atributos de dicho tipo (nombre, apellidos, ...), además como será usuario de Internet nos interesará agregar también el tipo 'inetOrgPerson' para poder almacerar el e-mail y otros valores. Si su nombre es Francisco Javier, podemos escribir en la casilla RDN 'cn=Francisco Javier' (cn= Common Name - Nombre Común):



Al pulsar OK nos apacererá la siguente figura, en la cual observamos los atributos de las tres tipologías de nuestro elemento: persona, usuario de internet y cuenta posix. Debemos rellenar al menos los campos gidNumber (grupo primario que será el 1001), homeDirectory, uid (identificador), uidNumber, loginShell y sn (surname - apellidos). También añadiremos el e-mail aunque en la figura no se vea ya que está más abajo:



Lo mismo haremos con el resto hasta que tengamos creados los cinco usuarios. Al final nuestro servidor LDAP tendrá la siguiente información:

Ya tendríamos creada la estructura, los grupos y los usuarios que necesitamos para nuestro sistema.

Autenticación basada en LDAP

Introducción

Como ya hemos comentado anteriormente, una de las utilidades más importantes de un servidor LDAP es como servidor de autentificación. Autentificarse es necesario para entrar en un sistema linux. También para acceder a algunos servicios como un servidor FTP o a páginas privadas en un servidor web. En otros apartados veremos como utilizar un servidor LDAP para permitir el acceso a páginas web privadas y para autentificar a usuarios del servidor de ftp Proftpd. Aquí veremos las modificaciones que hay que realizar en un sistema Linux para que autentifique a los usuarios en un servidor LDAP en lugar de utilizar los clásicos archivos **/etc/passwd, /etc/group y /etc/shadow**. Para ello es necesario instalar y configurar los paquetes libpam-ldap y libnss-ldap.

Librerías de autentificación pam-ldap y nss-ldap

La librería **pam-ldap** permite que las aplicaciones que utilizan PAM para autentificarse, puedan hacerlo mediante un servidor LDAP. Para que el sistema linux se autentifique mediante un servidor LDAP es necesario instalar esta librería ya que utiliza PAM. El archivo de configuración de ésta librería es **/etc/ldap.conf**. Hay otras aplicaciones o servicios que utilizan PAM para la autentificación y por tanto podrían, gracias a la librería pam-ldap, autentificarse ante un servidor LDAP.

Para especificar el modo de autentificación de cada servicio es necesario configurar los archivos que se encuentran en la carpeta **/etc/pam.d/**.

La librería **nss-ldap** permite que un servidor LDAP suplante a los archivos **/etc/passwd, /etc/group y /etc/shadow** como bases de datos del sistema. Toma la configuración del archivo anterior **/etc/ldap.conf**. También habría que configurar el arhivo **/etc/nsswitch.conf** para que se utilice LDAP como base de datos del sistema en lugar de los archivos passwd, group y shadow. El asistente que veremos a continuación, configurará automáticamente todos los archivos necesarios para permitir la autentificación del sistema por ldap.

La instalación de ambas librerías se puede realizar mediante apt-get.

Instalación de libpam-ldap y libnss-ldap

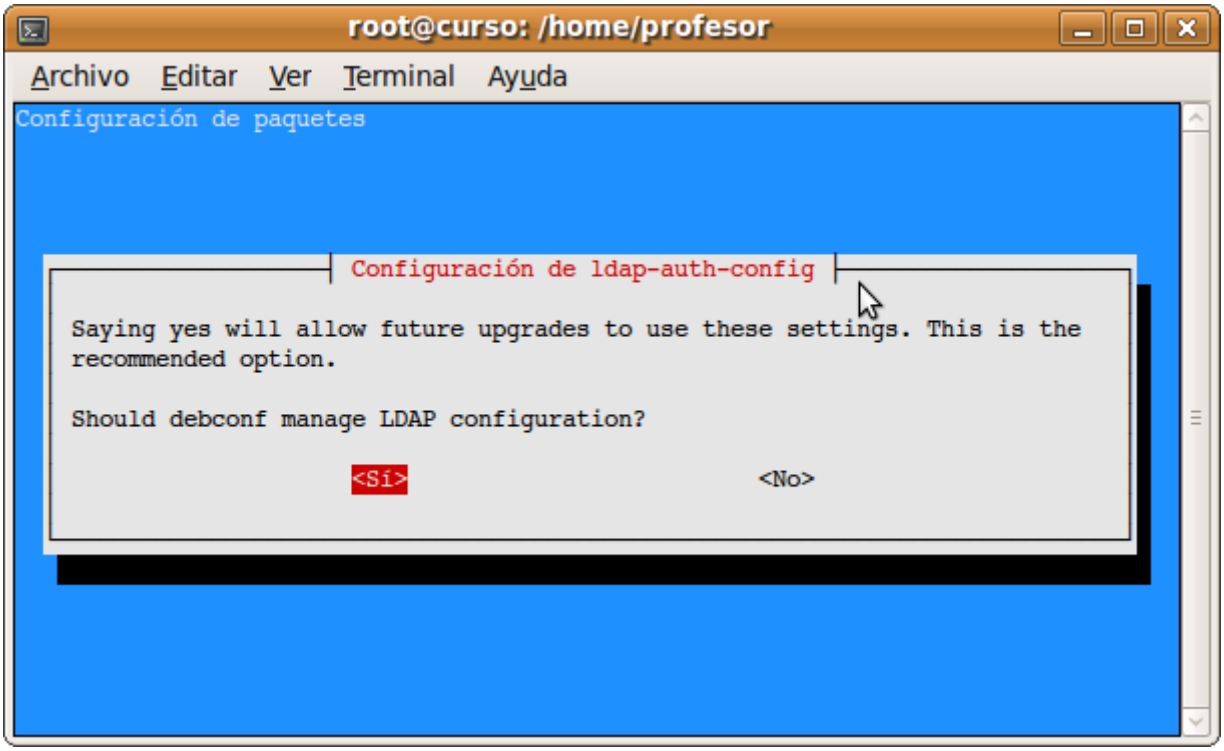
La instalación de las librerías libpam-ldap y libnss-ldap se puede realizar ejecutando el comando:

```
// Instalación de las librerías libpam-ldap y libnss-ldap
     # apt-get install libpam-ldap libnss-ldap
```

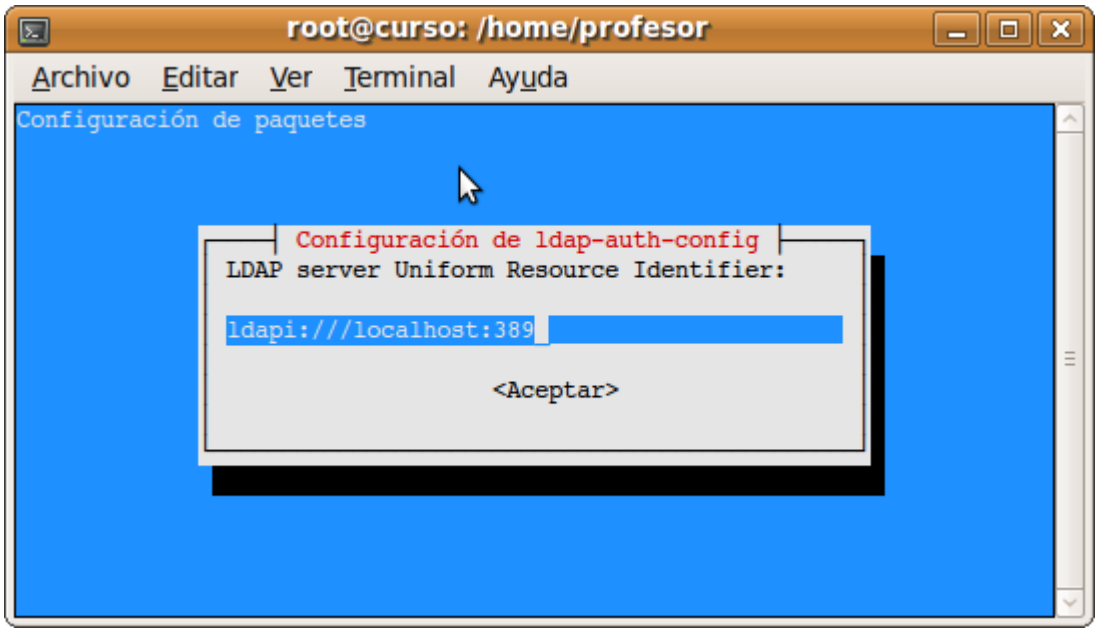
La configuración de las librerías libpam-ldap y libnss-ldap se hará con el comando:

```
// Configuración de las librerías libpam-ldap y libnss-ldap
     # dpkg-reconfigure ldap-auth-config
```

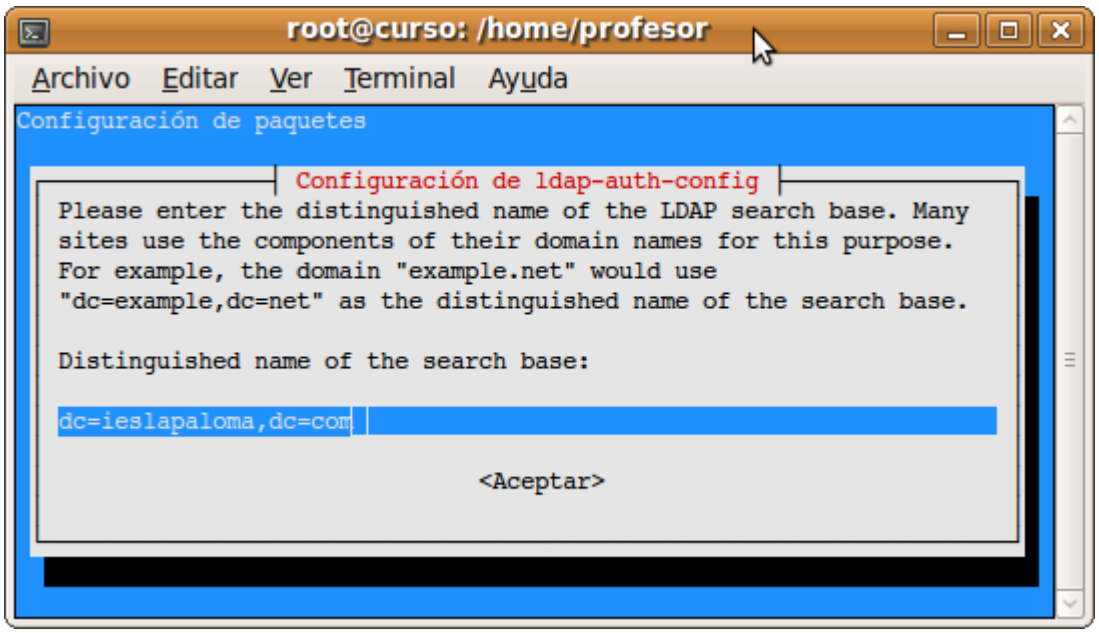
La primera pregunta que nos hace el asistente es si queremos que el asistente DebianConf maneje los archivos de configuración, lo recomendable es responder que sí.



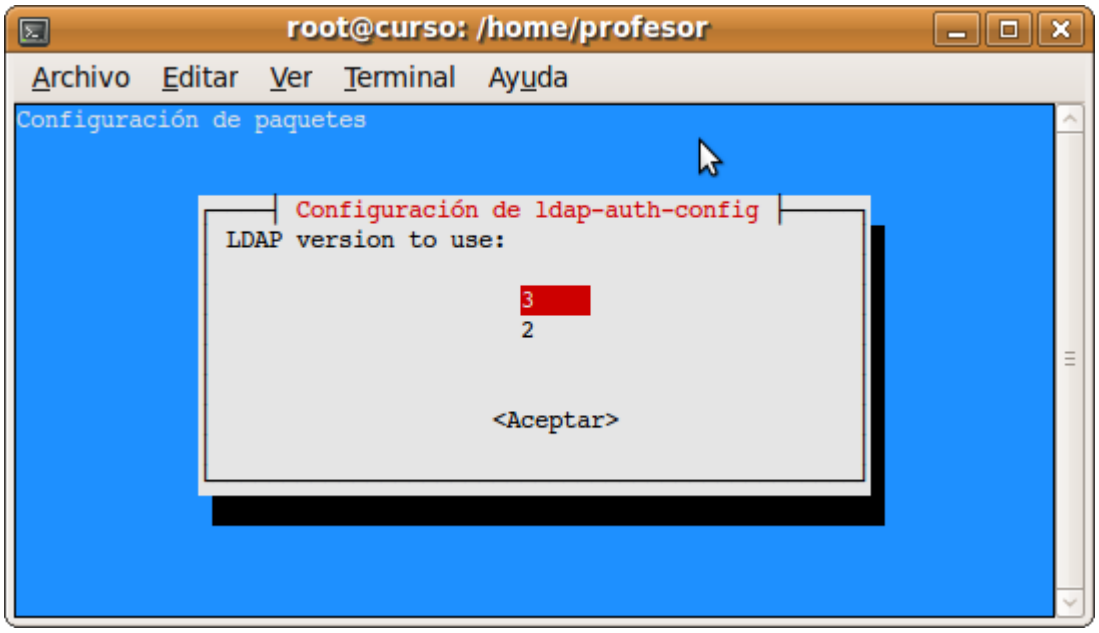
Después nos preguntará quién es el servidor LDAP. Podemos poner la IP o el nombre:



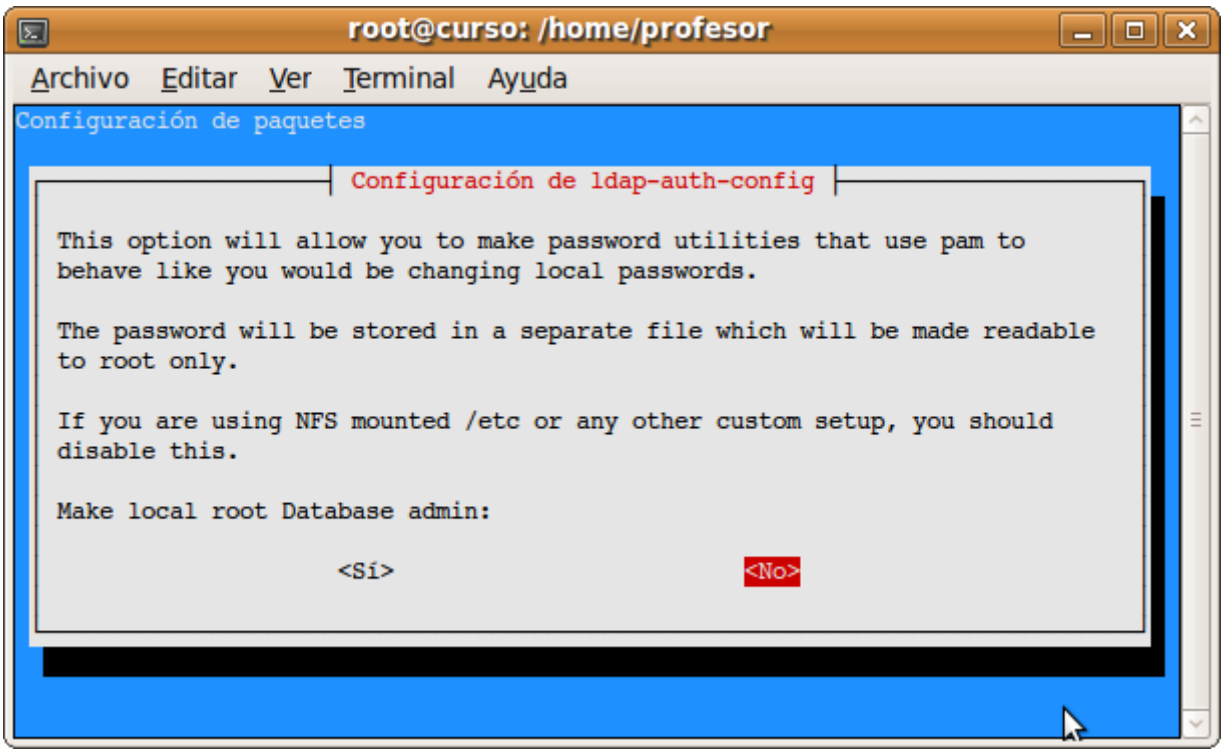
Luego nos preguntará por la base del directorio LDAP (base DN):



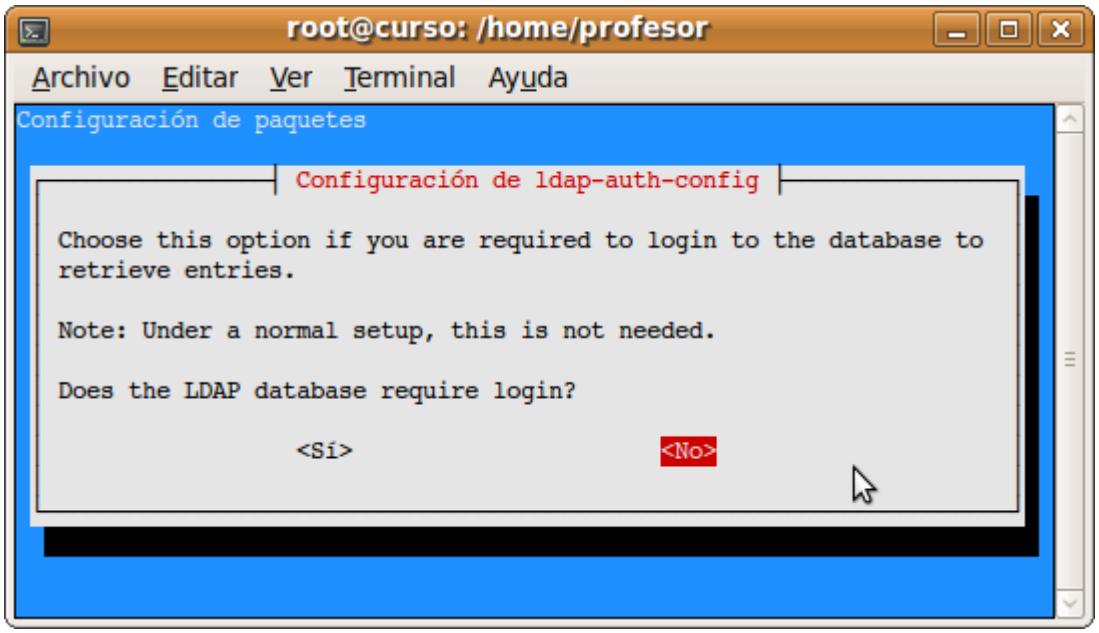
Acto seguido tendremos que indicar la versión de LDAP a utilizar:



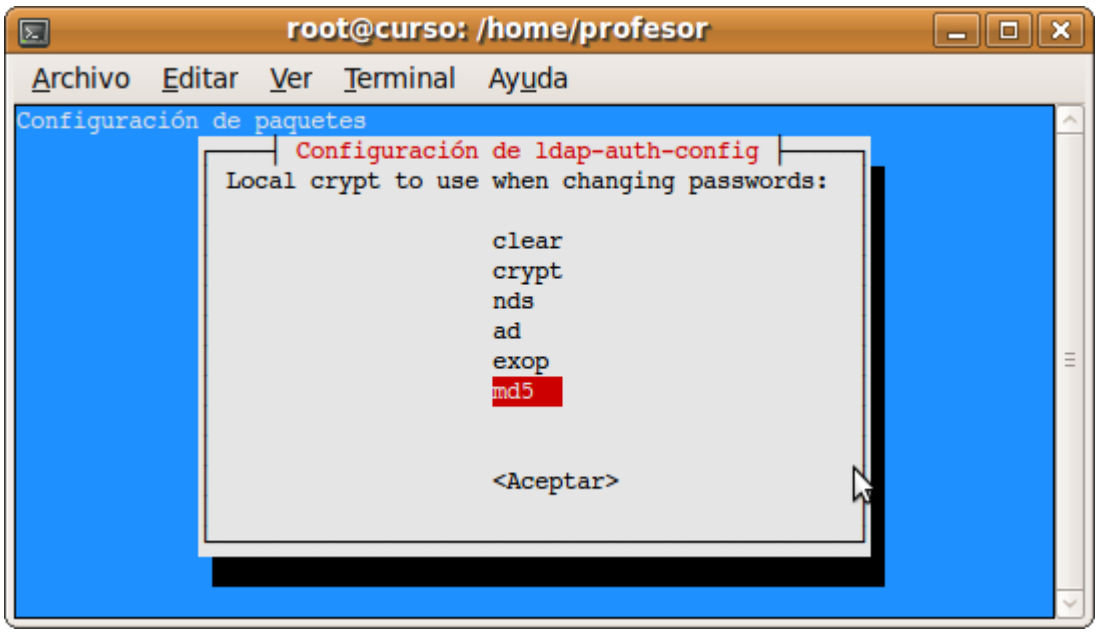
Luego nos preguntará si deseamos que las contraseñas se almacenen en un archivo a parte. Responderemos que no.



En el siguiente paso nos pregunta si necesitamos autentificarnos en el servidor LDAP o no. Como la librería únicamente va a realizar consultas, no es necesario autentificarse por lo tanto debemos responder 'No':



Finalmente nos preguntará qué sistema de cifrado queremos utilizar para las contraseñas almacenadas en caché. Elegiremos md5:



Ahora solamente nos quedaría indicar al sistema para que se autentifique por ldap. Podríamos hacerlo editando manualmente el archivo /etc/nsswitch.conf pero lo haremos mediante el comando auth-client-config

```
// Configuración del sistema para que se autentifique por ldap
     # auth-client-config -t nss -p lac_ldap
```

Probar la autentificación

Nuestro servidor LDAP ya debería autentificar correctamente . Podemos probar la autentificación de los servicios mediante el comando pamtest que se encuentra en el paquete libpam-dotfile, por lo tanto debemos instalarlo:

```
// Instalación del comando pamtest
     # apt-get install libpam-dotfile
```

Si deseamos probar que funciona el servicio passwd (cambiar contraseña) sobre un usuario del directorio LDAP (ejemplo jessica) , podemos ejecutar:

```
// Probando el cambio de contraseña
     root@curso:/etc/pam.d# pamtest passwd jessica

     Trying to authenticate  for service .

     Password:                    // Introducimos el password de jessica

     Authentication successful. // La autentificación ha sido satisfactoria
```

También podemos utilizar el comando finger sobre usuarios que estén solamente en el directorio LDAP, por ejemplo joel:

```
// Probando finger
     root@curso:/etc/pam.d# finger joel
```

```
Login: joel                             Name: Joel Javier

Directory: /home/www/alumnos            Shell: /bin/sh

Last login Tue Sep 27 18:02 (CEST) on pts/3 from 192.168.0.213

No mail.

No Plan.
```

Podemos por ejemplo, desde una consola de root, cambiar mediante el comando 'su' (su=Switch User - cambiar de usuario) a un usuario que esté en el directorio LDAP, para lo cuál no nos pedirá contraseña ya que root tiene permiso para cambiar a cualquier usuario. Si posteriormente cambiamos a otro usuario del directorio, ahora sí que nos pedirá contraseña. Deberemos introducir la contraseña que esté almacenada en el directorio LDAP para dicho usuario:

```
// Cambiando de usuario
root@curso:/etc/pam.d# su joel     // Somos root y cambiamos a joel
joel@curso:                         // No nos pide password
joel@curso:/etc/pam.d$ su jessica  // Somos joel, y cambiamos a jessica
Password:                               // Nos pide password, le introducimos
jessica@curso:/etc/pam.d$           // Ha cambiado correctamente
```

Las opciones de configuración de PAM son muy variadas. Para obtener más información se puede instalar el paquete libpam-doc que instala bastante documentación al respecto bajo la carpeta /usr/share/doc/libpam-doc/

Acceso a carpetas privadas con autentificación por LDAP

Otra posibilidad muy interesante es que los profesores e incluso el sitio web de la Intranet de nuestro centro, puedan disponer de carpetas privadas accesibles mediante el navegador pero no por cualquier usuario; por ejemplo los profesores podrían disponer de una carpeta donde almacenar información confidencial accesible desde la web -notas, por ejemplo-. Así mismo puede ocurrir que queremos tener en el servidor web de nuestra intranet páginas a las que sólo puedan tener acceso de lectura los profesores del centro. Vamos a ver cómo conseguir todo esto.

Lo primero que hemos de tener en cuenta es que para que podamos autenticar a los usuarios en apache mediante LDAP, hemos de habilitar un módulo especial en nuestro servidor web para que apache pueda validar el acceso a las carpetas deseadas a través de la base de usuarios del servidor LDAP. Dicho módulo se habilita ejecutando el siguiente comando:

```
// Habilitar módulo de autentificación de apache con ldap
     # a2enmod ldap
```

El siguiente paso es crear una carpeta de nombre "privada" colgando de "/var/www", lugar donde ubicaremos las páginas privadas de nuestro servidor web. Dicha carpeta tendrá como grupo propietario el grupo profesores.

Tras ello debemos editar de nuevo el archivo "/etc/apache/modules.conf" (el cual es referenciado por el archivo httpd.conf mediante la orden include) e incluir las siguientes entradas en los correspondientes apartados "LoadModule" y "AddModule":

Antes de modificar archivos de configuración conviene hacer copia de seguridad de los mismos. En caso de que se produzcan errores en el reinicio de apache, se puede recurrir a los archivos de log dentro de la carpeta /var/log/apache para analizar la causa.

```
Añadir en /etc/apache/modules.conf
     LoadModule auth_ldap_module /usr/lib/apache/1.3/auth_ldap.so

     AddModule auth_ldap.c
```

Posteriormente introducimos en /etc/apache/httpd.conf textualmente las siguientes líneas, mediante las cuales logramos definir la carpeta "privada" como aquella a partir de la cual el contenido allí contenido será privado y sólo accesible por los profesores de nuestro centro y por el administrador.

```
// Carpeta privada con acceso a profesores. Añadir en /etc/apache/httpd.conf
     Alias privada "/var/www/webprivada/"

     <Directory "/var/www/webprivada">

             Options Indexes FollowSymLinks

             AllowOverride None

             Order allow,deny

             Allow from all

             AuthType basic

             AuthName "Identificacion LDAP ieslapaloma.com"

             AuthLDAPUrl ldap://ip-servidor-ldap:389/dc=ieslapaloma,dc=com?
     uid

             AuthLDAPBindDN "cn=admin,dc=ieslapaloma,dc=com"

             AuthLDAPBindPassword xxxxxx

             AuthLDAPGroupAttributeIsDN off

             AuthLDAPGroupAttribute memberUid

             require group cn=profesores,ou=groups,dc=ieslapaloma,dc=com

     </Directory>
```

En el parámetro AuthLDAPUrl sustituiremos la cadena 'ip-servidor-ldap' por la dirección IP o el nombre del servidor LDAP y en el parámetro "AuthLDAPBindPassword" la cadena "xxxxxx" por la contraseña que hayamos asignado al usuario "administrador (admin) " del servidor LDAP.

En el parámetro AuthLDAPUrl vemos que al final termina con '?uid'. Significa que lo que debe de introducir el usuario es su uid (login del usuario). Podemos filtrar la entrada del usuario y poner condiciones si terminamos la url con '?uid??(atributo=valor)'. De ésta forma solamente serían válidos aquellos usuarios que

tengan un atributo con un valor determinado, ejemplo '?uid??(gidNumber=1001)' solo admitiría usuarios cuyo grupo primario sea 1001.

El parámetro AuthLDAPGroupAttributeIsDN debe estar a off para que no utilice el cn (nombre común) del usuario sino el uid a la hora de comprobar la pertenencia a un grupo.

En el parámetro AuthLDAPGroupAttribute debemos indicar el campo que se analizará para comprobar la pertenencia a un grupo.

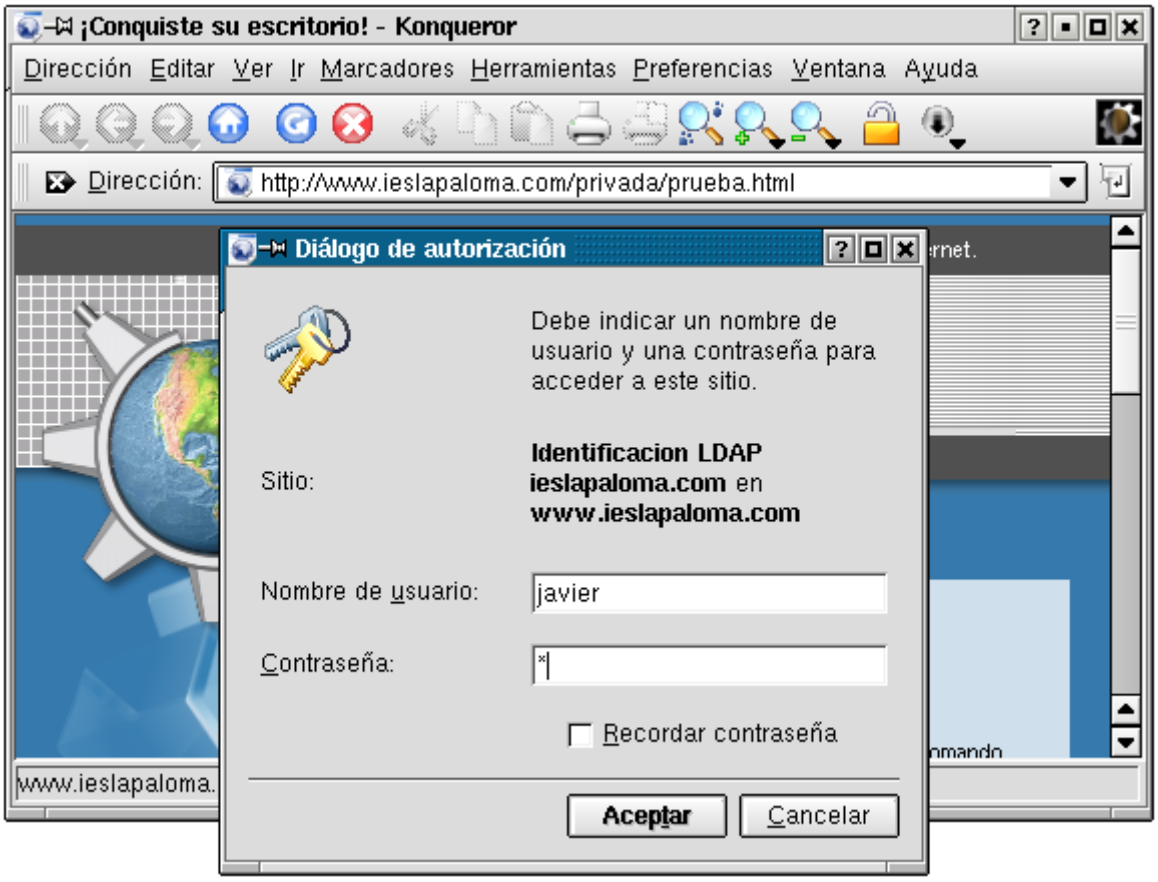
En el parámetro 'require', exigimos que pertenezca a un grupo. Otras opciones son 'require user' seguido de una lista de usuarios permitidos, ejemplo 'require user miguel joaquin jessica'. Para permitir a cualquier usuario que exista en el servidor LDAP, podemos usar 'requiere valid-user'.

Más información en: http://httpd.apache.org/docs/2.0/mod/mod_auth_ldap.html

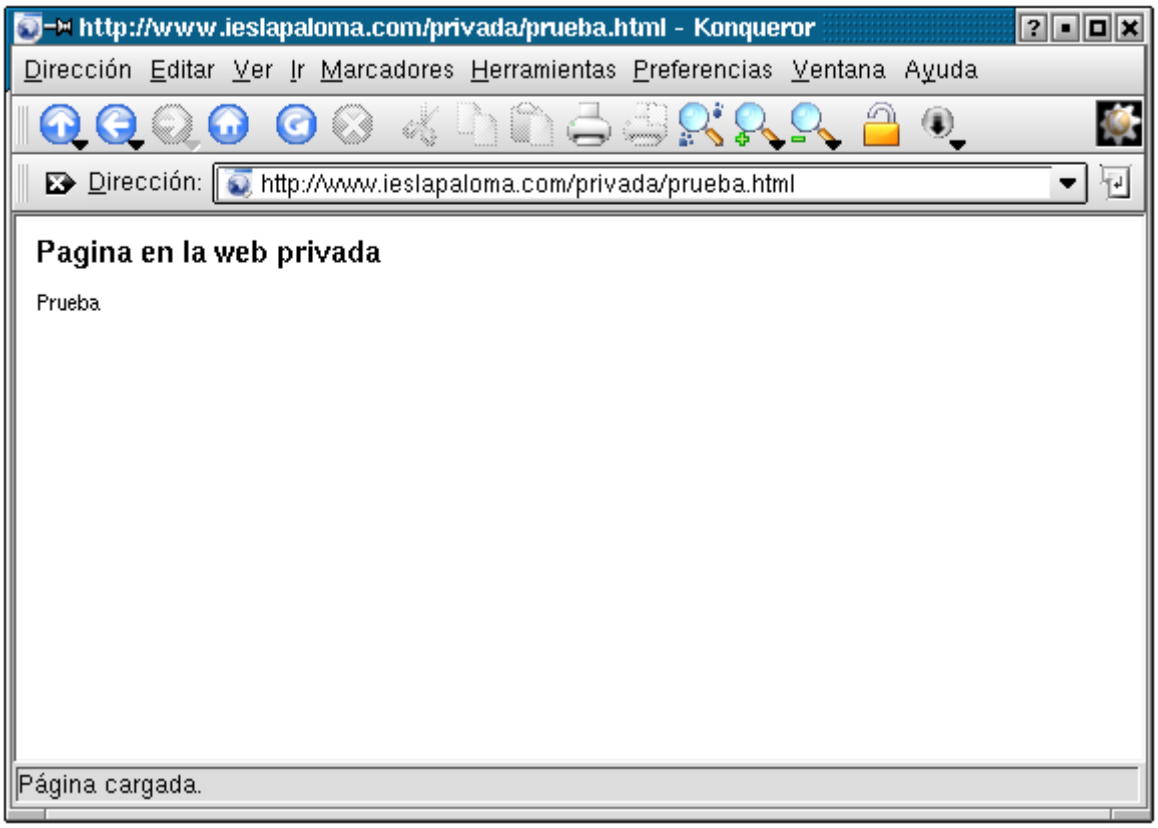
Guardamos los cambios realizados y para completar el proceso reiniciaremos el servidor "apache"

```
// Reiniciar apache
     # /etc/init.d/apache restart
```

Si ubicamos un fichero de nombre "prueba.html" en dicha carpeta ("/var/www/privada"), podremos acceder a ella mediante la URL "http://www.ieslapaloma.com/privada/prueba.html", mostrándose la siguiente pantalla en la cual se nos pedirá autenticación, y en la cual serán válidas las credenciales de algún profesor.



Una vez validado adecuadamente algún usuario con permisos de acceso a los contenidos privados se mostrará la página solicitada.



Además podemos crear una carpeta privada para cada profesor, de modo que el contenido allí existente sólo fuera accesible por él mismo previa autenticación; para ello crearemos una carpeta de nombre 'privada' colgando de la carpeta personal de cada profesor (por ejemplo en el caso del profesor Javier, en '/home/javier/public-html/'). Además de la creación de dicha carpeta 'privada' en la ruta correspondiente, hemos de editar el fichero 'httpd.conf' e incluir la siguiente entrada en el apartado correspondiente a los directorios:

```
// Carpeta privada de javier. Añadir en /etc/apache/httpd.conf
     Alias javier-p "/home/javier/public_html/privada"

     <Directory "/home/javier/public_html/privada">

             Options Indexes FollowSymLinks

             AllowOverride None

             Order allow,deny

             Allow from all

             AuthType basic

             AuthName "Identificacion LDAP ieslapaloma.com"

             AuthLDAPUrl ldap://ip-servidor-ldap:389/dc=ieslapaloma,dc=com?
     uid

             AuthLDAPBindDN "cn=admin,dc=ieslapaloma,dc=com"

             AuthLDAPBindPassword xxxxxx
```

```
        require user javier

</Directory>
```

Igual que antes, sustituiremos las cadenas 'ip-servidor-ldap' y "xxxxxx" por sus valores correctos. Además hemos de introducir esta entrada para cada uno de los profesores del centro, sustituyendo en las rutas de las dos primeras líneas el valor "javier" por el del profesor que deseamos que tenga el acceso seguro, así como dicho valor también en la penúltima línea.

Tras almacenar los cambios en el fichero de configuración y reiniciar el servicio apache, para acceder a un fichero de nombre "prueba.html" ubicado en la carpeta privada del profesor Javier teclearemos la URL: 'http://www.ieslapaloma.com/~javier/privada/prueba.html'

Es posible hacer, y de hecho es recomendable, que las carpetas privadas sean además seguras, es decir, utilicen un canal SSL, con lo cual el acceso a las carpetas seguras sería 'https' en el puerto '443', el resto de las rutas de las URL de acceso se mantendrían estables. Para lograrlo hemos de introducir en cada una de las entradas '<Directory>' la instrucción 'SSLRequireSSL'.

15.- Enrutamiento y Proxy

Se puede definir el enrutamiento como la capacidad de transmitir datos entre redes interconectadas. Al agente encargado de realizar este encaminamiento de información entre redes se conoce como **enrutador o router** pudiendo ser de tipo hardware si es un dispositivo físico dedicado al encaminamiento y de tipo software en caso de ser un PC que ejecuta una aplicación que realice las funciones propias del enrutamiento.

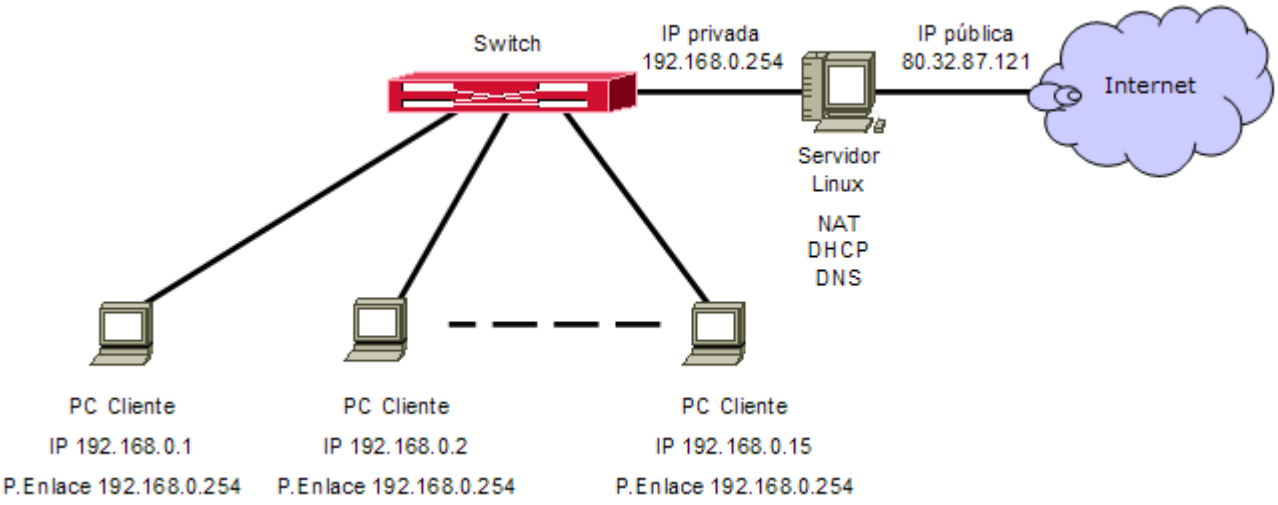
Con el software adecuado, nuestro servidor Linux podrá actuar de enrutador en nuestra red de manera que permitirá que los equipos de la red local se conecten a Internet como si lo hicieran a través de un router.

La tecnología empleada para permitir que los equipos de la red local se conecten a Internet a través de nuestro servidor Linux se denomina NAT - Network Address Traslation (Traducción de Direcciones de Red). El software NAT que se ejecuta en nuestro servidor permite, que con una única dirección IP pública en el servidor, tengan acceso a Internet el resto de PCs de la red.

En los PCs de la red local se deberá configurar como puerta de enlace (gateway) la dirección IP interna del servidor para que sea éste quien reciba y procese los paquetes provenientes de la red interna y con destino hacia Internet.

Cuando desde un PC de la red local se quiere acceder a Internet, el paquete de datos se enviará al servidor linux ya que es la puerta de enlace. El software NAT del servidor cambiará en el paquete de datos la dirección IP de origen del PC de la red local por la dirección IP pública del servidor y lanzará el paquete de datos hacia Internet. En una tabla interna almacenará el puerto de salida del paquete junto con la IP del PC de la red local con la finalidad de que cuando llegue la respuesta desde Internet, realizar el proceso inverso y poder redirigirlo hacia el PC que lanzó la petición.

Si nuestro servidor Linux, dispone además de servidor DHCP, la configuración de las direcciones IP, la puerta de enlace y el servidor DNS de nuestros PCs, podrá ser establecida automáticamente por el servidor DHCP.



Una alternativa podría ser instalar en el servidor un proxy como **squid**, de esa forma las páginas accedidas por los clientes serían cacheadas en el servidor con lo cual se aceleraría la conexión a Internet, especialmente cuando son muchos los clientes que acceden a los mismos sitios. Un proxy facilita también el control de la conexión impidiéndola o restringiéndola a medida de nuestras necesidades. El inconveniente de compartir una conexión a Internet con un proxy es que trabaja a nivel de aplicación y por tanto del protocolo de cada aplicación (HTTP, FTP, SMTP, etc...). Esto obliga a configurar las aplicaciones (navegador, clientes de correo, clientes ftp, etc...) para que utilicen el proxy, cosa que no es necesario hacer cuando se dispone de un router ya que el router NAT trabaja a nivel de red TCP/IP y es totalmente trasparente a las aplicaciones.

Otro servicio que se podría disponer en el servidor es un cortafuegos como **iptables** que permite filtrar qué paquetes de datos pueden entrar y qué paquetes de datos pueden salir, con la finalidad de controlar el acceso a Internet y ganar en seguridad frente a ataques externos.

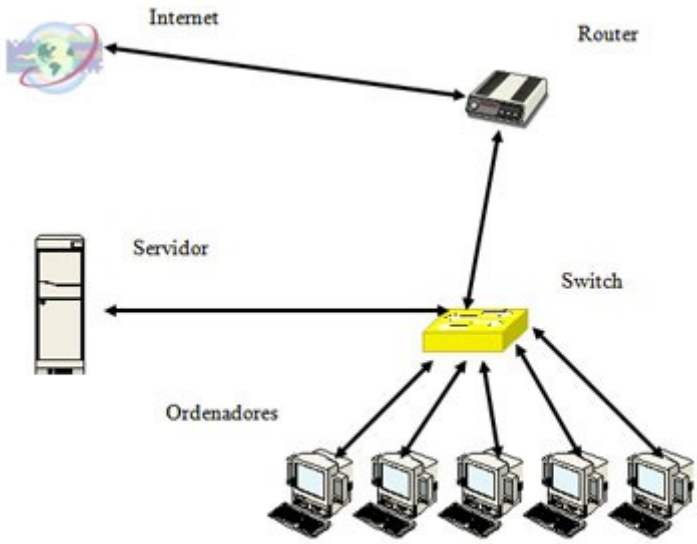
Más adelante veremos una configuración básica de **iptables** que nos permitirá permitir o denegar las

conexiones a diferentes redes y puertos, así como una configuración básica de **squid** para poder compartir y controlar la conexión a Internet mediante el proxy.

Enrutamiento en Linux

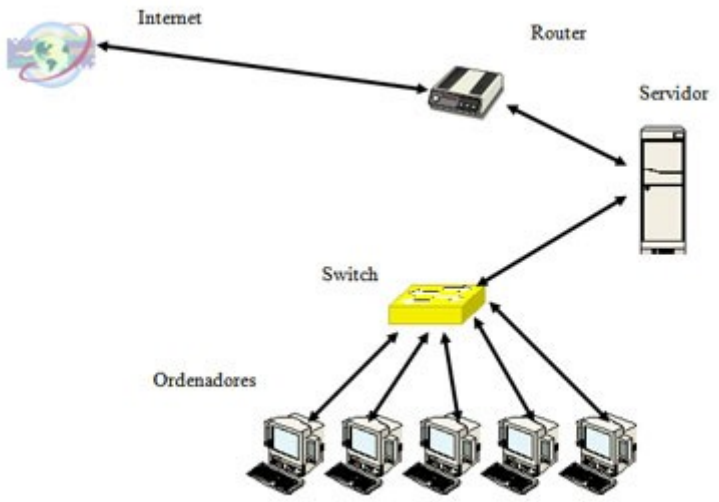
En nuestro Centro Educativo hemos venido detectando problemas de saturación de la línea de conexión a Internet sin motivo justificado. Hemos detectado que en algún ordenador de la sala de profesorado y de algún departamento hay instalados programas de P2P (descarga masiva) y somos conscientes de que estos programas saturan el canal de salida a Internet del centro, además sospechamos que el alumnado también utiliza este tipo de programas.

El router ADSL está conectado a un switch y por lo tanto a través de múltiples utilidades es fácil conocer su dirección IP y configurar nuestro equipo como puerta de enlace, con el consiguiente acceso libre a Internet y a la descarga masiva. Nos encontramos con un esquema del tipo:



Este esquema no permite controlar el tráfico de red puesto que los PCs tienen acceso directo al router.

Situando el servidor entre la red y el router, todo el tráfico hacia Internet pasa por el servidor lo que nos permitirá analizarlo, generar estadísticas, filtrar accesos, instalar un proxy-caché, etc., de forma sencilla y centralizada.



Activación del enrutamiento en Linux

Las funciones de enrutamiento mediante NAT son realizadas por el cortafuegos que analizará los paquetes provenientes de la red local interna cuyo destino sea Internet y los modificará convenientemente para que salgan hacia Internet como si fueran emitidos por el servidor. A partir del nucleo 2.4 de Linux, el cortafuegos empleado es **iptables**.

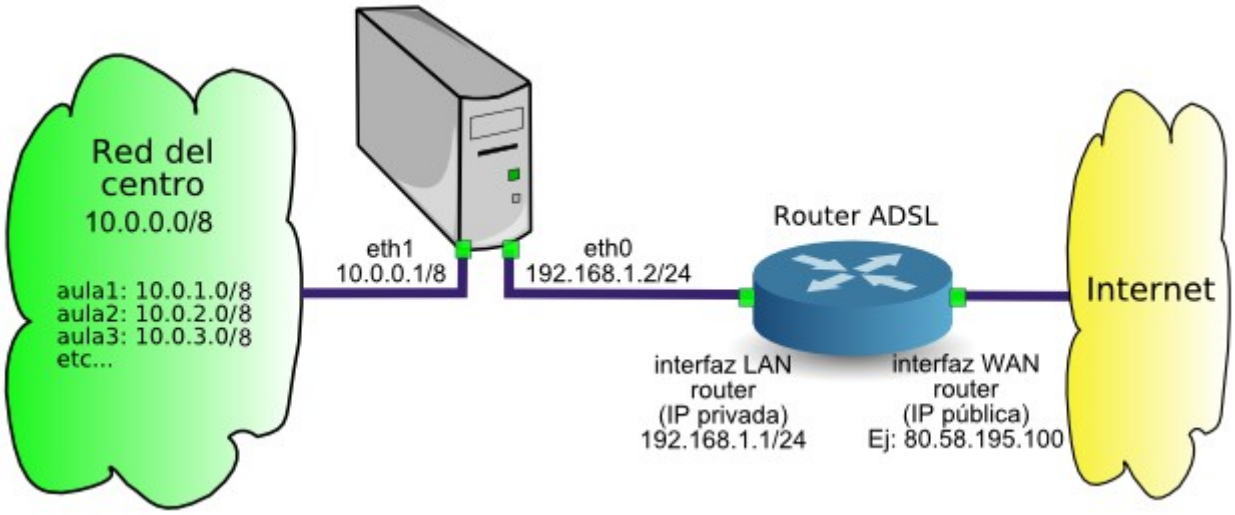
Para posibilitar que nuestro servidor Linux sea capaz de comportarse como un router y hacer de puerta de enlace para los PCs de nuestra red local, será necesario crear un script que configure el cortafuegos iptables para que realice NAT desde dentro de la red local hacia Internet.

Creación del script para activar enrutamiento

Para activar el enrutamiento en un sistema Linux, tan solo basta con poner a '1' la variable ip_forward del sistema, es decir, basta con ejecutar desde una consola de root:

```
// Activar el enrutamiento en un sistema Linux
     # echo "1" > /proc/sys/net/ipv4/ip_forward
```

Posteriormente tendríamos que configurar el filtrado para que acepte el redireccionamiento de paquetes desde dentro hacia fuera de nuestra red y mediante NAT permita que los PCs de la red interna naveguen con la dirección IP 'publica' del servidor. Supongamos que el router Linux tiene una tarjeta (eth0) configurada con la IP 192.168.1.2/24 y conectada al router, cuya IP es 192.168.1.1/24, y por otro lado, tenemos otra tarjeta (eth1) configurada con la ip 10.0.0.1/8 y conectada al switch para dar servicio a nuestra red interna que utiliza el rango 10.0.0.0/8. Nuestro esquema sería como el que vemos en la siguiente figura:



Router Linux

Tendríamos que indicar que se acepten todos los paquetes que son para reenviar, es decir, aquellos que llegan a nuestra máquina pero que no es ella la destinataria. Para ello, tendríamos que aceptar los paquetes de tipo FORWARD, como veremos en la siguiente sección. Por otro lado, tendríamos que indicar que los paquetes que llegan desde nuestra red interna (-s 10.0.0.0/8) y que salgan por la interfaz eth0 hacia el router (-o eth0), después de enrutarlos en nuestra máquina (POSTROUTING), debemos enmascararlos (MASQUERADE), es decir, hacer NAT. Los comandos a ejecutar serían:

```
// Haciendo NAT en el servidor
     # iptables -A FORWARD -j ACCEPT

     # iptables -t nat -A POSTROUTING -s 10.0.0.0/8 -o eth0 -j MASQUERADE
```

Podríamos realizar un script que activara el enrutamiento y el NAT y otro para desactivarlo:

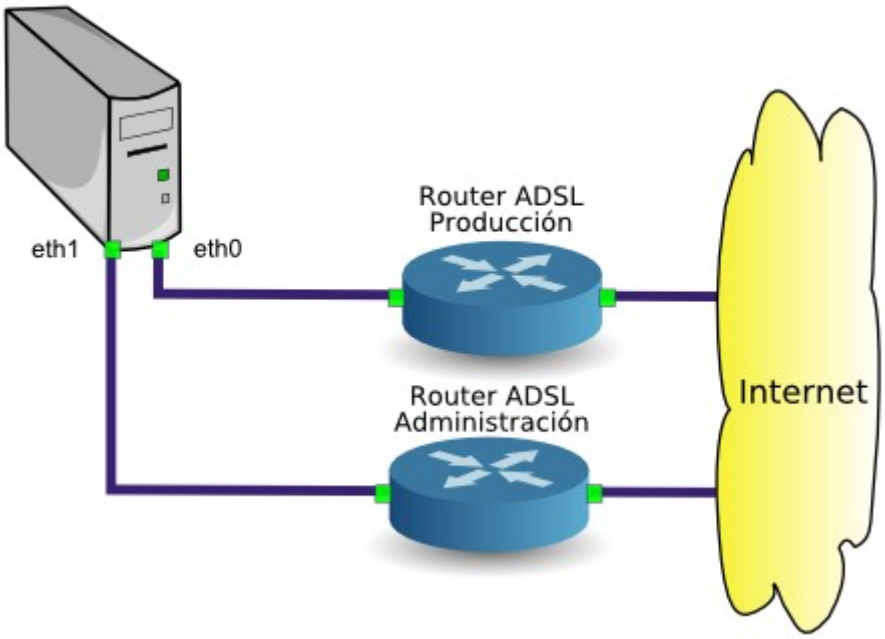
```
// activar-enrutamiento.sh
     echo "1" > /proc/sys/net/ipv4/ip_forward

     iptables -A FORWARD -j ACCEPT

     iptables -t nat -A POSTROUTING -s 10.0.0.0/8 -o eth0 -j MASQUERADE



// desactivar-enrutamiento.sh
     echo "0" > /proc/sys/net/ipv4/ip_forward
```

Así, nuestro servidor se convertiría en un router. Si todas las comunicaciones de la red pasan por nuestro servidor, podremos tenerlas controladas, como veremos en las siguientes secciones.

Crear y eliminar rutas fijas

Cuando activamos el enrutamiento en Linux, nuestra máquina se convierte en un router automático, de forma que todo lo que entre por la interfaz eth0 con destino a una red diferente de la definida en eth0, lo reenviará por la interfaz eth1 y de igual forma, todo lo que entre por la interfaz eth1 con destino a una red diferente de la definida en eth1, lo reenviará por la interfaz eth0. Es el funcionamiento normal de un router, enrutar todo.

En algunos casos, puede que nos interese que ciertos paquetes salgan por una interfaz concreta. Por ejemplo, supongamos que en nuestra red disponemos de dos conexiones ADSL independientes, una para dar servicio de conexión a Internet al servidor (interfaz de producción) y otra, para conectarnos desde nuestra casa al servidor, para realizar tareas de administración (interfaz de administración).Supongamos que la interfaz **eth0** está conectada al router ADSL de **producción** y la interfaz **eth1** está conectada al router ADSL para realizar tareas de **administración**.



Rutas fijas

Lo normal es que la interfaz eth0 tenga configurada como puerta de enlace la IP del router de conexión a Internet, pero la interfaz eth1 no debería tener configurada la puerta de enlace, para que no exista tráfico hacia Internet por dicha interfaz. Si en el ADSL de nuestra casa tenemos IP fija, podemos crear una ruta para que cuando la IP destino sea la **IP fija** de nuestra casa, los paquetes se enruten por eth1 en lugar de hacerlo por eth0. Ejemplo, si nuestra IP de casa es 80.58.12.27, el comando a ejecutar será:

```
//Crear una ruta para una IP concreta
     # route add 80.58.12.27 eth1
```

En lugar de una IP concreta, quizás nos interese crear una ruta para toda una **red**. Supongamos que queremos que cuando la IP destino sea una IP del CNICE, salga por la interfaz eth1. Teniendo en cuenta que el rango de IPs públicas del CNICE es 192.144.238.0/24, el comando a ejecutar sería:

```
//Crear una ruta para una red concreta
     # route add -net 193.144.238.0/24 eth1
```

Si queremos **eliminar** una ruta, utilizaremos el parámetro 'del' seguido de la IP o la red destinataria. Ejecutaríamos el siguiente comando:

```
//Eliminar una ruta
     # route del -net 193.144.238.0/24
```

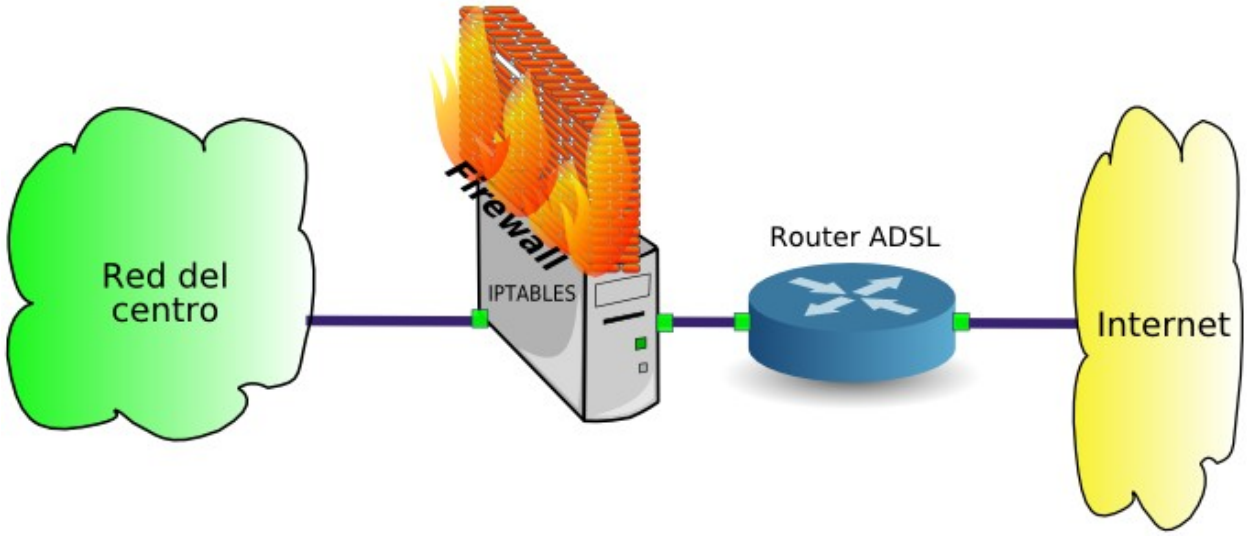
Si queremos **ver** la configuración de la tabla de rutas, debemos ejecutar el comando route sin parámetros:

```
//Ver rutas
     # route
```

Establecer rutas puede ser muy interesante cuando queremos dividir nuestra red en diferentes subredes y disponemos de un servidor con varias tarjetas de red.

Cortafuegos iptables

Desde la versión 2.4 del núcleo de linux, el cortafuegos utilizado para gestionar las conexiones es **iptables**. Las posibilidades de iptables son prácticamente infinitas y un administrador que quiera sacarle el máximo provecho, puede realizar configuraciones extremadamente complejas. Para simplificar, diremos que básicamente, iptables permite crear reglas que analizarán los paquetes de datos que entran, salen o pasan por nuestra máquina, y en función de las condiciones que establezcamos, tomaremos una decisión que normalmente será permitir o denegar que dicho paquete siga su curso.



El cortafuegos controla las comunicaciones entre la red y el exterior

Para crear las reglas, podemos analizar muchos aspectos de los paquetes de datos. Podemos filtrar paquetes en función de:

Tipo de paquete de datos:
- Tipo INPUT: paquetes que llegan a nuestra máquina
- Tipo OUTPUT: paquetes que salen de nuestra máquina
- Tipo FORWARD: paquetes que pasan por nuestra máquina

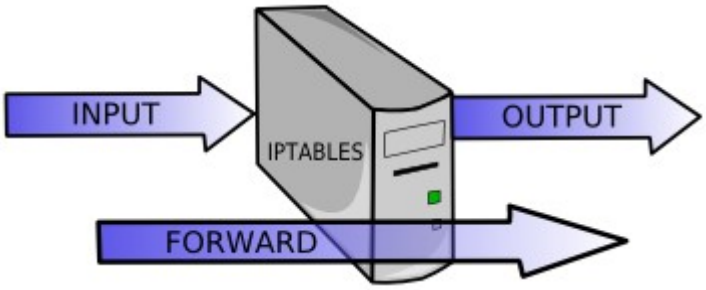
Interfaz por la que entran (-i = input) o salen (-o = output) los paquetes
- eth0, eth1, wlan0, ppp0, ...

IP origen de los paquetes (-s = source)
- IP concreta, ej: 10.0.1.3
- Rango de red, ej: 10.0.1.0/8

IP destino de los paquetes (-d = destination)
- IP concreta, ej: 10.0.1.3
- Rango de red, ej: 10.0.1.0/8

Protocolo de los paquetes (-p = protocol)
- tcp, udp, icmp...

Hacer NAT (modificar IP origen y destino para conectar nuestra red a otra red o a Internet) y...
- Filtrar antes de enrutar: PREROUTING
- Filtrar después de enrutar: POSTROUTING



Los paquetes pueden entrar, salir o pasar

Una forma sencilla de trabajar con iptables es permitir las comunicaciones que nos interesen y luego denegar el resto de las comunicaciones. Lo que se suele hacer es definir la política por defecto aceptar (ACCEPT), después crear reglas concretas para permitir las comunicaciones que nos interesen y finalmente, denegar el resto de comunicaciones. Lo mejor será crear un script en el que dispondremos la secuencia de reglas que queremos aplicar en nuestro sistema. Un ejemplo típico podría ser el siguiente:

```
#!/bin/sh

# Script cortafuegos.sh para la configuración de iptables

#

# Primero borramos todas las reglas previas que puedan existir

iptables -F

iptables -X

iptables -Z

iptables -t nat -F


# Después definimos que la politica por defecto sea ACEPTAR

iptables -P INPUT ACCEPT

iptables -P OUTPUT ACCEPT

iptables -P FORWARD ACCEPT

iptables -t nat -P PREROUTING ACCEPT

iptables -t nat -P POSTROUTING ACCEPT
```

```
# Para evitar errores en el sistema, debemos aceptar

# todas las comunicaciones por la interfaz lo (localhost)

iptables -A INPUT -i lo -j ACCEPT


# Aceptamos las comunicaciones que nos interesan y luego denegamos el
resto.


# Ejemplo: Denegamos acceso al aula 1

iptables -A FORWARD -s 10.0.1.0/24 -j DROP


# Aceptamos SMTP, POP3 y FTP (correo electrónico y ftp)

iptables -A FORWARD -s 10.0.0.0/8 -p tcp --dport 25 -j ACCEPT

iptables -A FORWARD -s 10.0.0.0/8 -p tcp --dport 110 -j ACCEPT

iptables -A FORWARD -s 10.0.0.0/8 -p tcp --dport 20 -j ACCEPT

iptables -A FORWARD -s 10.0.0.0/8 -p tcp --dport 21 -j ACCEPT


# HTTP y HTTPS no es necesario porque nuestro servidor será servidor
proxy

# Dejamos comentadas las líneas, por si algún día las necesitamos

#iptables -A FORWARD -s 10.0.0.0/8 -p tcp --dport 80 -j ACCEPT

#iptables -A FORWARD -s 10.0.0.0/8 -p tcp --dport 443 -j ACCEPT


# DNS no es necesario porque nuestro servidor será servidor DNS

# Dejamos comentadas las líneas (tcp y udp), por si algún día las
necesitamos

#iptables -A FORWARD -s 10.0.0.0/8 -p tcp --dport 53 -j ACCEPT

#iptables -A FORWARD -s 10.0.0.0/8 -p udp --dport 53 -j ACCEPT


# Al PC del Director le damos acceso a todo (cliente VIP)
```

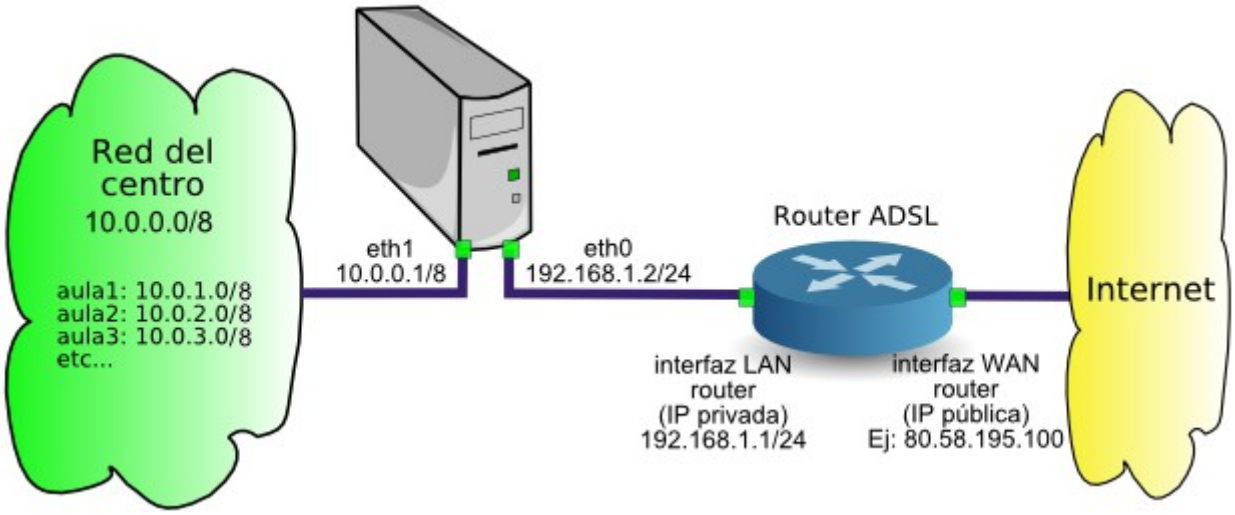
```
iptables -A FORWARD -s 10.0.0.7 -j ACCEPT


# Denegamos resto de comunicaciones (no funcionará el p2p)

iptables -A FORWARD -s 10.0.0.0/8 -j DROP


# Hacemos NAT si IP origen 10.0.0.0/8 y salen por eth0

iptables -t nat -A POSTROUTING -s 10.0.0.0/8 -o eth0 -j MASQUERADE


# Activamos el enrutamiento

echo 1 > /proc/sys/net/ipv4/ip_forward


# Comprobamos cómo quedan las reglas

iptables -L -n
```

En el script anterior vemos una serie de reglas que se van a ir ejecutando secuencialmente conformando la configuración del cortafuegos iptables. Cuando indicamos "-A FORWARD" nos referimos a paquetes que van a pasar por nuestro servidor. Otras opciones son "-A INPUT" y "-A OUTPUT". Acto seguido ponemos las condiciones. Si ponemos "-s 10.0.0.0/8" nos referimos a paquetes cuya IP origen está en el rango 10.0.0.0/8. Cuando ponemos "-p tcp" nos referimos a paquetes que utilizan el protocolo tcp. Cuando indicamos "--dport 25" nos referimos a paquetes cuyo puerto de destino es el 25, es decir, protocolo SMTP (correo saliente). Si en una regla no ponemos la condición -p ni la condición --dport, significa que no nos importa el protocolo (cualquier protocolo) ni nos importa el puerto destino (cualquier puerto destino). Por ejemplo, en la regla donde damos acceso al PC del Director, no hemos indicado ni protocolo ni puerto, por lo que dejará pasar todos los protocolos y todos los puertos.

Proxy Squid

Introducción

Un proxy de conexión a Internet es un servidor que hace de **intermediario** entre los PCs de la red y el router de conexión a Internet, de forma que cuando un usuario quiere acceder a Internet, su PC realiza la petición al servidor Proxy y es el Proxy quien realmente accede a Internet. Posteriormente, el Proxy enviará los datos al PC del usuario para que los muestre en su pantalla. El PC del usuario no tendrá conexión directa con el router, sino que accederá a Internet por medio del proxy.



El proxy es un intermediario

Ventajas de disponer de un proxy:

-Los PCs de los usuarios **no tienen acceso al router**, todas las comunicaciones exteriores pasarán por el Proxy, lo que nos permitirá tener las comunicaciones bajo control. Podemos permitir o denegar el acceso web, ftp, email, messenguer, p2p, etc...

-Las páginas se **cachean** en la memoria temporal del proxy lo cual acelera la descarga cuando varios usuarios acceden a las mismas páginas a la vez. Esta circunstancia se da mucho en los centros educativos cuando el profesor está explicando un tema y todos los alumnos acceden a la vez a la misma página.

-Es fácil crear una lista de **urls prohibidas** a las que el proxy denegará el acceso.

-Es fácil permitir o denegar el acceso a **subredes** o a PCs concretos. Si diseñamos la red de forma que cada aula del centro tenga un rango determinado, por ejemplo 10.0.X.Y donde X es el número de aula e Y el número de PC, sería posible permitir o denegar la conexión a Internet aula por aula.

-El proxy guarda **informes** de todas las conexiones que hacen los usuarios. Al principio puede ser interesante ver a qué páginas de contenido inadecuado acceden nuestros alumnos, para agregarlas a la lista de urls prohibidas.

-Los PCs de nuestra red están más **seguros** de ataques externos ya que el proxy hace de barrera cortafuegos.

Inconvenientes de la utilización de un Proxy:

No todo son ventajas, también hay algún inconveniente en la utilización de un Proxy:

-Para que las aplicaciones accedan a Internet a través del proxy, es **necesario configurar** cada aplicación: navegador web, cliente ftp, cliente de correo, etc...

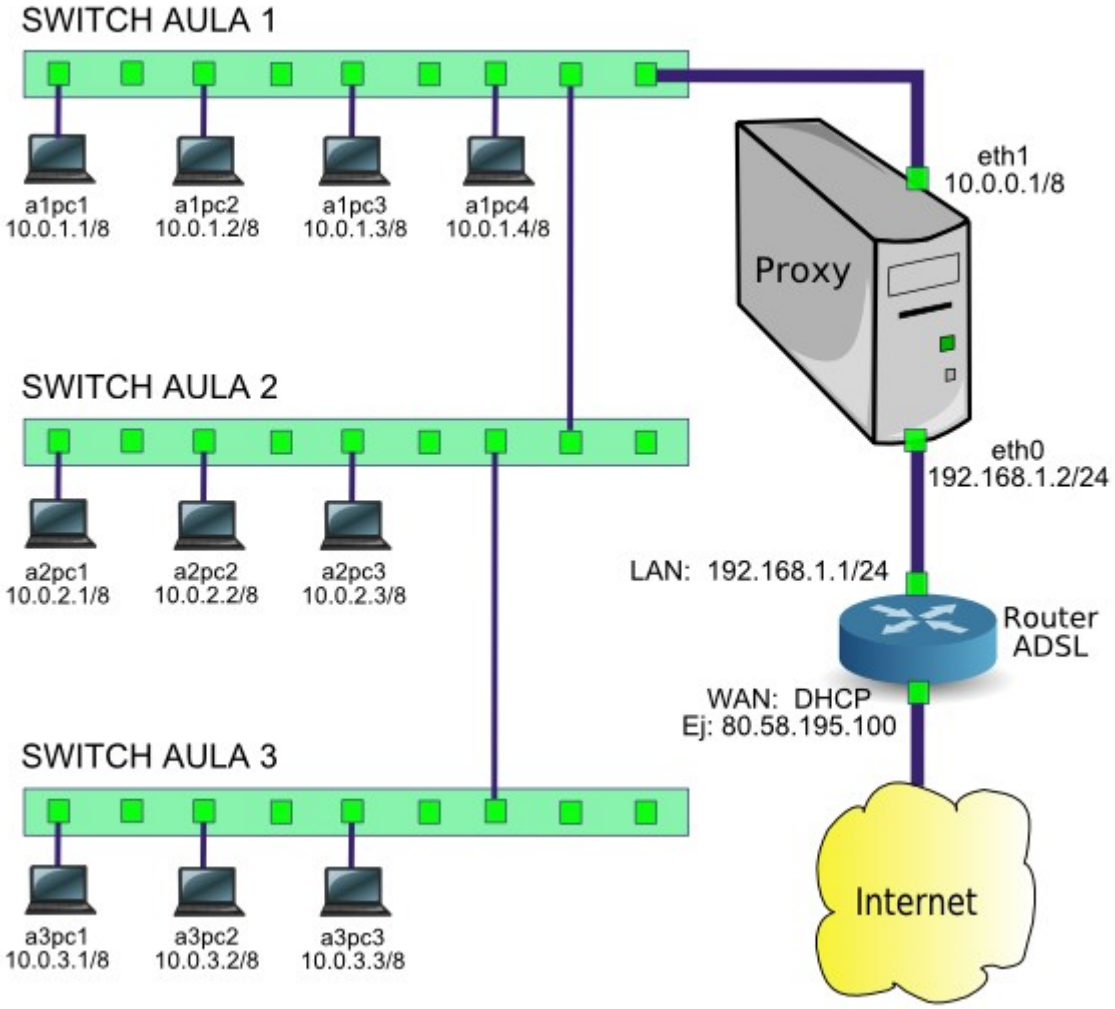
-Todas las comunicaciones con el exterior pasarán por el servidor. Si el proxy **falla**, la red se quedará sin conexión a Internet. Para subsanar lo más rápidamente posible el problema ante un fallo del Proxy, será conveniente disponer de un proxy de repuesto.

-El proxy requiere **mantenimiento**. Para que todo funcione, es necesario que exista un administrador de la red que se encargue de actualizar, revisar, mantener y reparar el proxy cuando deje de funcionar.

Diseño recomendado de la red del centro

Para facilitar la gestión del acceso a Internet en el centro, se recomienda diseñar la red de forma que cada aula tenga un rango de IPs determinado. Para no quedarnos cortos, lo mejor es utilizar el rango 10.0.0.0/8 siguiendo el esquema 10.W.X.Y donde W sería el número de edificio, X el número de aula e Y el número de

PC, que nos permitiría tener un máximo de 254 edificios con 254 aulas cada uno y 254 PCs por aula. Si disponemos de un único edificio con tres aulas, un sencillo esquema de direccionamiento IP podría ser el siguiente:



Direccionamiento de nuestra red

Direccionamiento IP recomendado para nuestra red:
-Utilizar el rango **10.0.0.0/8** para el direccionamiento de red del centro educativo.
-Utilizar la IP **10.0.0.1** para el servidor proxy. Conviene que dicho servidor sea también servidor DNS.
-Las aulas usarán la dirección **10.0.X.Y** donde X sea el número de aula e Y sea el número de PC. Ejemplo, si en la aula 1 hay 4 PCs, en el aula 2 hay 3 y en el aula 3 hay 3, el direccionamiento sería:

Aula	PC	Nom.	IP	Máscara	P.Enlace	DNS
1	1	a1pc1	10.0.1.1	255.0.0.0	sin configurar	10.0.0.1
1	2	a1pc2	10.0.1.2	255.0.0.0	sin configurar	10.0.0.1
1	3	a1pc3	10.0.1.3	255.0.0.0	sin configurar	10.0.0.1
1	4	a1pc4	10.0.1.4	255.0.0.0	sin configurar	10.0.0.1
2	1	a2pc1	10.0.2.1	255.0.0.0	sin configurar	10.0.0.1
2	2	a2pc2	10.0.2.2	255.0.0.0	sin configurar	10.0.0.1
2	3	a2pc3	10.0.2.3	255.0.0.0	sin configurar	10.0.0.1
3	1	a3pc1	10.0.3.1	255.0.0.0	sin configurar	10.0.0.1
3	2	a3pc2	10.0.3.2	255.0.0.0	sin configurar	10.0.0.1
3	3	a3pc3	10.0.3.3	255.0.0.0	sin configurar	10.0.0.1

Instalación del Proxy squid

Linux dispone del Proxy **squid**. Se trata de una aplicación de gran éxito que se lleva utilizando muchos años y dispone de cientos de posibilidades para personalizar su funcionamiento a nuestras necesidades. Para instalar la última versión de squid, podemos hacerlo con apt-get desde una consola de root:

```
// Instalación del servidor Proxy squid
     # apt-get install squid
```

De esta forma instalaríamos los programas necesarios para disponer de un completo servidor Proxy en nuestra red. Tan solo será necesario configurarlo y ponerlo en marcha.

Arranque y parada del proxy squid

El servicio squid, al igual que todos los servicios, dispone de scripts de arranque y parada en la carpeta /etc/init.d. Debemos ejecutarlos desde una consola de root.

```
// Arrancar o reiniciar el servidor squid
     # /etc/init.d/squid restart



// Parar el servidor squid
     # /etc/init.d/squid stop



// Recargar configuración del servidor squid
     # /etc/init.d/squid reload
```

Para un arranque automático del servicio al iniciar el servidor, debemos crear los enlaces simbólicos correspondientes tal y como se indica en el apartado Arranque automático de servicios al iniciar el sistema.

Configuración básica del proxy squid

El archivo de configuración del proxy es el archivo **/etc/squid/squid.conf**. Si observamos dicho archivo, veremos que es un archivo muy extenso en el que hay cientos de parámetros que podemos establecer, pero para una utilización básica, son unos pocos los parámetros que debemos configurar. De todos los apartados que dispone el archivo /etc/squid/squid.conf, solo destacaremos los siguientes:

OPTIONS FOR AUTHENTICATION (Opciones de autentificación)
Aquí se establecen las opciones de autentificación del Proxy. Aunque aquí no vamos a hablar de ello, existe la posibilidad de configurar squid para que solicite usuario y contraseña para poder navegar por Internet. Si se quiere hacer uso de esta funcionalidad, lo normal sería tener almacenados los usuarios y las contraseñas en un servidor LDAP y en función de los grupos a los que pertenezcan los usuarios, podríamos habilitar o deshabilitar el acceso. Esto puede ser interesante en empresas, donde el administrador de red da acceso a Internet solo a los usuarios que lo necesitan. En un centro educativo supondría bastante trabajo llevar una administración de este tipo ya que habría que crear y gestionar un usuario para cada alumno y para cada profesor. Es más fácil administrar por redes y por aulas.

ACCESS CONTROL (Control de Acceso)
En esta sección estableceremos los permisos de acceso, es decir, quien puede navegar y quien no. Lo primero que tendremos que hacer es crear listas de control de acceso (Access Control List - ACL) y luego dar permisos a dichas listas.

Una lista de control de acceso (acl) se crea utilizando la palabra acl seguido del nombre que queramos dar a la lista y seguido de una condición que cumplirán los miembros de la lista. Entre las condiciones más utilizadas destacamos: src (IPs o URLs origen), dst (IPs o URLs destino), port (puertos) y proto (protocolos). Ejemplos:

Si en mi red local utilizo el direccionamiento 10.0.0.0/8, puedo crear una lista para definir a toda mi red:

```
//acl para definir toda mi red
     acl todos src 10.0.0.0/8
```

Si en mi red local utilizo el direccionamiento 10.0.X.0/24, para el aula X, puedo crear una lista para cada aula:

```
//Una acl para cada aula
     acl aula1 src 10.0.1.0/24

     acl aula2 src 10.0.2.0/24

     acl aula3 src 10.0.3.0/24

     acl aula4 src 10.0.4.0/24

     acl aula5 src 10.0.5.0/24
```

Luego tendría que dar permiso a las listas. Para ello se utiliza la palabra clave **http_access** seguido del permiso allow (permitir) o deny (denegar) y seguido del nombre de la lista. Ejemplos:

Si quiero dar permiso a toda mi red para que navegue por Internet:

```
//Permiso para que navegue toda mi red
     http_access allow todos
```

Si quiero dar permiso a las aulas 1, 2 y 3 para que navegue por Internet pero no quiero que naveguen las aulas 4 y 5:

```
//Permiso para que naveguen las aulas 1, 2 y 3 y no naveguen las aulas 4 y 5
     http_access allow aula1

     http_access allow aula2

     http_access allow aula3

     http_access deny aula4

     http_access deny aula5
```

Por defecto, squid viene configurado para actuar como **caché** de acceso a Internet, pero no tiene creadas listas de control de acceso. Si configuramos el navegador de Internet de los PCs cliente para que utilicen el Proxy, veremos que tenemos denegado el acceso al Proxy. Para empezar a disfrutar del Proxy, tendremos que crear una lista de control de acceso con el rango de nuestra red y darla permiso. Si en nuestra red utilizamos el rango 10.0.0.0/8, deberíamos añadir en /etc/squid/squid.conf:

```
//Permiso para que navegue toda mi red.
```

```
acl todos src 10.0.0.0/8

http_access allow todos
```

Cuando creamos acls, podemos sustituir el rango de IPs por el nombre de un archivo externo, y de esa manera podemos indicar el en archivo externo el rango o los rangos de IPs a los que queremos referirnos, sin necesidad de estar continuamente modificando el archivo squid.conf. Más adelante veremos un ejemplo cómo tener un archivo externo con las urls prohibidas a las que no podrán navegar nuestros alumnos.

NETWORK OPTIONS (Opciones de red)
En esta sección estableceremos con el parámetro http_port, el puerto en el que escucha el Proxy. Lo mejor es dejar el valor por defecto que es el puerto 3128:

```
//Configurar squid en el puerto 3128
     http_proxy 3128
```

Squid puede trabajar en modo **transparente**. La ventaja de configurar squid en dicho modo de trabajo, es que no sería necesario configurar el navegador de los PCs clientes para trabajar con el proxy, sino que simplemente configuramos la puerta de enlace del PC cliente con la IP del servidor proxy. Posteriormente tendremos que configurar el cortafuegos del servidor para que redirija las peticiones al puerto 80 hacia el puerto 3128 y así las reciba squid. Si deseamos poner el Proxy en modo transparente, deberemos indicarlo después del puerto. En tal caso, el parámetro http_port quedaría así:

```
//Configurar squid en el puerto 3128, en modo transparente
     http_proxy 3128 transparent
```

```
//Redirigir las peticiones al puerto 80 hacia el puerto 3128. Ejecutar como
root:
     # iptables -t nat -A PREROUTING -i eth1 -p tcp --dport 80 -j REDIRECT
     --to-port 3128
```

MEMORY CACHE OPTIONS
En esta sección estableceremos la memoria RAM utilizada para la caché. Una buena opción es utilizar sobre un tercio de la memoria RAM del sistema. Ejemplo, si nuestro sistema tiene 512 MB de memoria RAM, una buena opción sería:

```
//RAM utilizada por squid
     cache_mem 192 MB
```

DISK CACHE OPTIONS
En esta sección estableceremos el espacio de disco duro utilizado para la caché. Una buena opción es utilizar el 50% de la capacidad total del disco duro. Ejemplo, si nuestro disco tiene sistema tiene 80 GB de memoria RAM, una buena opción sería utilizar 40 GB. Deberemos utilizar la palabra clave cache_dir seguida de la palabra ufs que es el formato utilizado por squid, de la carpeta donde queremos que se almacene la cache, el tamaño de la caché en MB, el número de subdirectorios de primer nivel y el número de subdirectorios de segundo nivel. Ejemplo, si queremos que la caché se guarde en /var/spool/squid, que utilice 40 GB y que cachee hasta 16 subdirectorios de primer nivel y hasta 256 subdirectorios de segundo nivel, escribiremos:

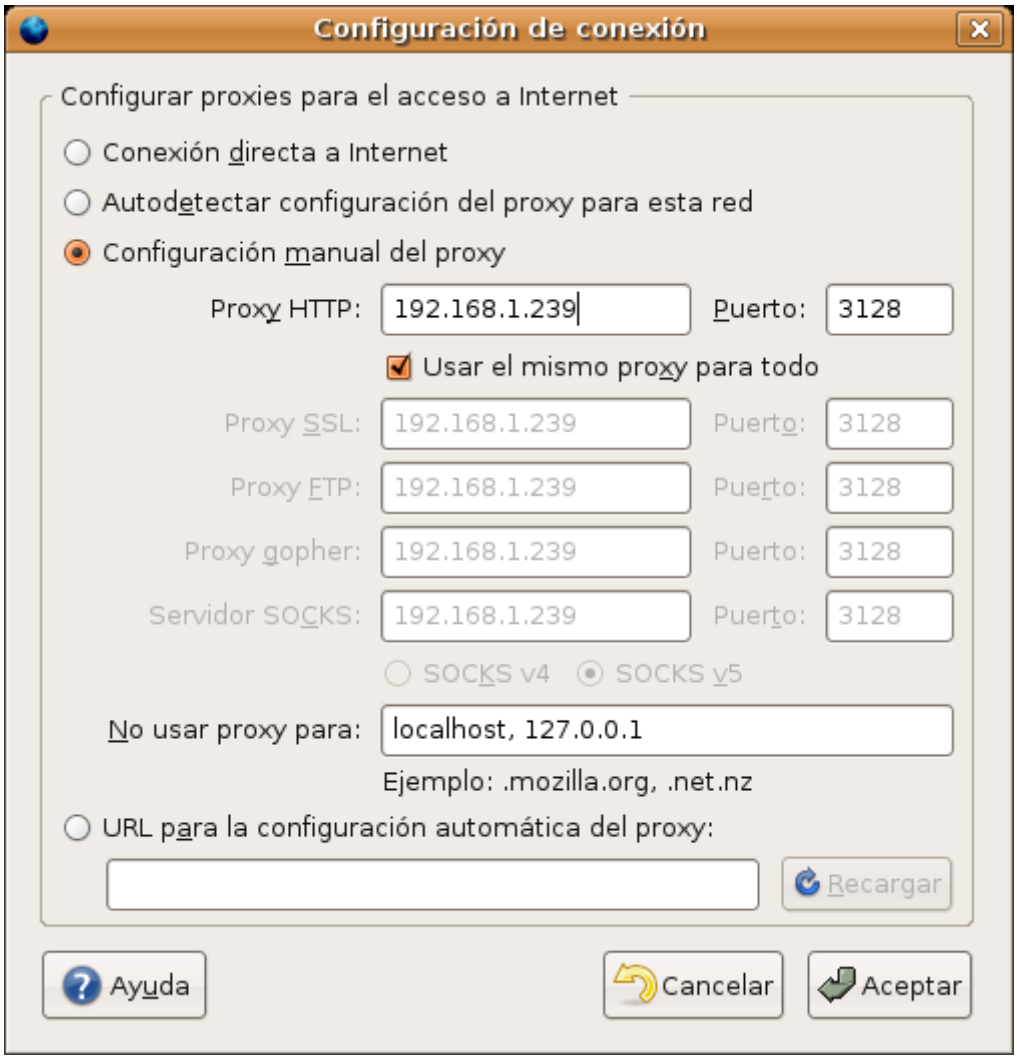
```
//Espacio en disco utilizado por squid
     cache_dir ufs /var/spool/squid 40000 16 256
```

Configuración del navegador de los PCs clientes, para que utilicen el Proxy

Supongamos que nuestro servidor Proxy tiene la IP 192.168.1.239 y el servidor squid está escuchando en el puerto 3128 que es el puerto que utiliza por defecto. Con estos dos datos, la IP y el puerto, ya podemos configurar el navegador de Internet de los PCs clientes.

Mozilla Firefox

Para que Firefox utilice nuestro Proxy en sus conexiones, debemos ir a Herramientas > Opciones > Avanzado > Red y en el apartado Conexión, hacer clic en el botón Configuración. En la ventana que aparece, debemos configurar la IP y el puerto de nuestro servidor Proxy:



Configuración del Proxy en Firefox

A partir de este momento, Firefox enviará a nuestro Proxy cualquier consulta web que realice, y será nuestro Proxy quien realizará la conexión en caso necesario.

Internet Explorer

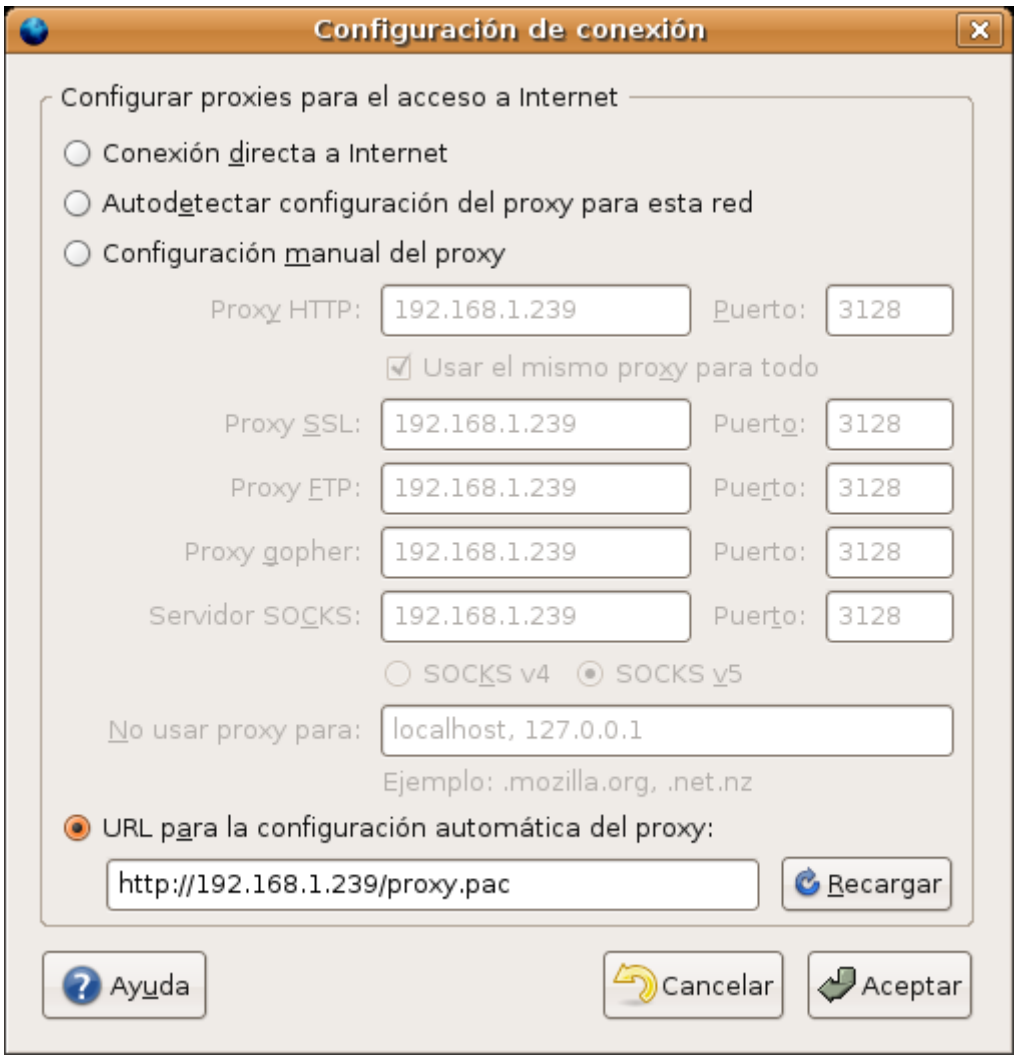
Para indicar a Internet Explorer que debe utilizar un Proxy para realizar conexiones, debemos ir a Herramientas > Opciones de Internet > Conexiones > Configuración de LAN y activar la casilla 'Usar un servidor proxy para la LAN'. En la casilla 'Dirección' pondremos la IP de nuestro Proxy y el 'Puerto' el puerto, tal y como se muestra en la siguiente ventana:

Configuración del Proxy en Internet Explorer

Archivo de configuración automática del proxy

Para no tener que recordar la dirección del proxy y facilitar la tarea a la hora de configurar el proxy en los PCs clientes, existe la posibilidad de crear un archivo de configuración automática del proxy. Dicho archivo indicará al navegador, en función de la url a la que quiera conectarse, si debe hacerlo directamente o debe hacerlo a través del proxy. En un direccionamiento como el que tenemos en nuestro centro, cuando accedemos a nuestra red 10.0.0.0/8 o a la dirección de localhost 127.0.0.1, la conexión debe ser directa, en cambio, cuando accedemos a cualquier otra dirección, deberá ser a través de proxy.

```
//Archivo de configuración automática del proxy
//Archivo /var/www/proxy.pac
     function FindProxyForURL(url,host){

             if (isInNet(host, "10.0.0.0", "255.0.0.0"))

                     return "DIRECT";

             else if (isInNet(host, "127.0.0.1", "255.255.255.255"))

                     return "DIRECT";

             else return "PROXY 192.168.1.239:3128";

     }
```



Configuración del Proxy a través de un archivo de configuración

Permitir o denegar el acceso desde ciertos rangos de IPs

Tal y como se ha comentado anteriormente, con squid es sencillo permitir o denegar el acceso a Internet por rangos de IPs. Si tenemos nuestra red diseñada de forma que cada aula utiliza un rango concreto, podremos permitir o denegar el acceso a un aula de forma sencilla.

Para no tener que tocar el archivo squid.conf, lo mejor es crear una acl que cargue las aulas desde un archivo externo. Podemos crear con un editor de texto el archivo /etc/squid/aulas-prohibidas.txt en el que indicaremos los rangos de IPs que no queremos que naveguen. Por ejemplo, si no queremos que naveguen las aulas 2 y 3, el contenido del archivo /etc/squid/denegar-aulas.txt deberá ser:

```
//Archivo /etc/squid/aulas-prohibidas.txt
     10.0.2.0/24

     10.0.3.0/24
```

Después tendremos que editar squid.conf para crear una acl que cargue los rangos desde el archivo /etc/squid/daulas-prohibidas.txt y deniegue el acceso a dichos rangos.

```
//Archivo externo para indicar las aulas a las que no las permitimos navegar
```

```
//Editar squid.conf e introducir estas dos líneas:
     acl aulas-prohibidas src "/etc/squid/aulas-prohibidas.txt"

     http_access deny aulas-prohibidas
```

Por último, tan solo tenemos que recargar la configuración de squid para que entre en funcionamiento la nueva configuración:

```
//Recargar la configuración de squid
     # /etc/init.d/squid reload
```

Igualmente podemos crear una acl para indicar las urls prohibidas desde un archivo externo:

```
//Archivo externo para indicar las urls prohibidas
//Editar squid.conf e introducir estas dos líneas:
     acl urls-prohibidas dst "/etc/squid/urls-prohibidas.txt"

     http_access deny urls-prohibidas
```

Si no queremos que nuestros alumnos accedan a www.sex.com ni a www.misvecinitas.com, el contenido del archivo /etc/squid/urls-prohibidas.txt debería ser:

```
//Archivo /etc/squid/urls-prohibidas.txt
     www.sex.com

     www.misvecinitas.com
```

La filosofía sería denegar las aulas prohibidas, denegar las urls prohibidas y luego permitir todo lo demás. Resumiendo, nuestro archivo squid.conf será como el original con las siguientes modificaciones, justo después de la línea # INSERT YOUR OWN RULE(S) HERE TO ALLOW ACCESS FROM YOUR CLIENTS que podríamos traducir como: Inserte sus propias reglas para permitir acceso a sus clientes:

```
//Resumen de modificaciones en squid.conf
     # INSERT YOUR OWN RULE(S) HERE TO ALLOW ACCESS FROM YOUR CLIENTS

     acl aulas-prohibidas src "/etc/squid/aulas-prohibidas.txt"

     http_access deny aulas-prohibidas

     acl urls-prohibidas dst "/etc/squid/urls-prohibidas.txt"

     http_access deny urls-prohibidas

     http_access allow all
```

Así, editando los archivos /etc/squid/aulas-prohibidas.txt y /etc/squid/urls-prohibidas.txt y recargando la configuración de squid ejecutando /etc/init.d/squid reload, podemos reconfigurar squid sin necesidad de tocar el archivo de configuración squid.conf.

El inconveniente es que cada vez que queremos permitir o denegar el acceso a Internet a un aula, tenemos que andar editando el archivo aulas-prohibidas.txt lo que puede resultar un poco engorroso. Podemos crear dos scripts de unix que hagan el trabajo por nosotros y solamente tengamos que ejecutar los scripts

indicando el número de aula que queremos prohibir o permitir:

```
Nombre del script: prohibir-aula.sh
     #/bin/bash

     #

     # Script para prohibir la navegación de un aula

     # Se creará el rango del aula en /etc/squid/aulas-prohibidas.txt

     # Indicar el número de aula al ejecutar el script



     if [ $# -ne 1 ]; then

             echo "Es necesario introducir el numero de aula a prohibir"

             exit -1

     fi

     echo Prohibir navegar aula $1, subred 10.0.$1.0/24

     echo 10.0.$1.0/24 >> /etc/squid/aulas-prohibidas.txt

     /etc/init.d/squid reload

     echo subredes denegadas:

     cat /etc/squid/aulas-prohibidas.txt

//Nombre del script: permitir-aula.sh
     #/bin/bash

     #

     # Script para permitir la navegación de un aula

     # Se eliminará el rango del aula de /etc/squid/aulas-prohibidas.txt

     # Indicar el número de aula al ejecutar el script



     if [ $# -ne 1 ]; then

             echo "Es necesario introducir el numero de aula"

             exit -1

     fi

     subred=10.0.$1.0/24
```

```
echo Permitir navegar aula $1, subred $subred

patron=`echo /10.0.$1.0/d`

cat /etc/squid/aulas-prohibidas.txt | sed -e $patron > /tmp/temp.txt

cat /tmp/temp.txt >  /etc/squid/aulas-prohibidas.txt

/etc/init.d/squid reload

echo Subredes denegadas:

cat /etc/squid/aulas-prohibidas.txt
```

Si deseamos que el aula 1 no navegue, deberíamos ejecutar: prohibir-aula 1. Si luego deseamos permitir que el aula 1 navegue, tendríamos que ejecutar: permitir-aula 1.

Aún con los scripts prohibir-aula.sh y permitir-aula.sh, sigue siendo engorroso realizar cambios ya que el profesor tendría que iniciar sesión en el servidor por ssh y lanzar el script. Lo mejor será crear una página en PHP con botones de comando, donde con un simple clic podamos ejecutar los scripts cómodamente desde el navegador.

Análisis de conexiones

Una de las funcionalidades principales que nos ofrece squid es que registra todos los accesos a Internet. Cada vez que un PCs accede a Internet, squid registrará en el archivo /var/log/squid/access.log la fecha y hora, el PC y la url a la que ha accedido.

```
//Archivo de registro de squid
     /var/log/squid/access.log
```

Analizar el archivo /var/log/squid/access.log nos va a resultar de gran ayuda ya que podemos ver a qué páginas web no permitidas se están conectando los alumnos, lo que nos permitirá ir recopilándolas en nuestro archivo urls-prohibidas.txt.

16.- Varios

Arranque automático de servicios al iniciar el sistema.

Cuando Linux arranca, puede hacerlo de 7 modos distintos, numerados del 0 al 6. A estos modos se les denomina **niveles de ejecución** y son los siguientes:

- **Nivel 0 (Halt)**: Detiene el sistema
- **Nivel 1 (Monousuario)**: Permite entrar en el sistema como root sin contraseña y en modo texto.
- **Nivel 2 (Multiusuario sin red)**: Modo multiusuario en modo texto y sin red.
- **Nivel 3 (Multiusuario con red)**: Modo multiusuario en modo texto y con red. Así arrancan los servidores.
- **Nivel 4 (Pruebas)**: No utilizado
- **Nivel 5 (Multiusuario con red y modo gráfico)**: Multiusuario en entorno gráfico. Así arrancan los PCs de usuario.
- **Nivel 6 (Reboot)**: Reinicia el sistema.

Normalmente Linux arranca en modo 5, aunque los servidores a veces se les configura para arrancar en modo 3. El modo de arranque del sistema se configura en el archivo /etc/sysinit.

En función del nivel de ejecución, existe la posibilidad de configurar qué servicios deben iniciarse de forma automática, para ello es necesario crear unos enlaces simbólicos en las carpetas /etc/rcN.d (donde N es un número de 0 a 6 que indica el nivel de ejecución de linux) que apunten al script de inicio del servicio que se encuentra en /etc/init.d/. Dichos enlaces deberán tener un nombre un poco especial ya que deberán comenzar con la letra 'S' de Start (arrancar) seguida de un número de dos cifras (para establecer el orden de arranque de los servicios) y del nombre del servicio, ejemplo: S20samba ó S30nfs. Si lo que nos interesa es que el servicio no arranque, la primera letra deberá ser una K de Kill (detener) en lugar de una S, ejemplo: K20samba ó K30nfs.

Estos enlaces se pueden crear con el comando **update-rc.d**. Ejemplo, si deseamos que el servicio samba se arranque cuando el servidor inicia en los niveles 3, 4 y 5 y no arranque cuando inicia en los niveles 1, 2 y 6, ejecutaremos el siguiente comando **(Ojo, no olvidar el punto del final (.) al escribir el comando)**:

```
// Crear enlaces para inicio automático del servicio
     # update-rc.d samba start 20 3 4 5 . stop 20 1 2 6 .
```

De esta forma se crearán enlaces simbólicos de arranque con nombre S20samba en las carpetas /etc/rc3.d, /etc/rc4.d y /etc/rc5.d y de parada con nombre K20samba en las carpetas /etc/rc1.d,/etc/rc2.d y /etc/rc6.d.

El número 20 indica la prioridad. Sirve para arrancar o parar antes unos servicios que otros ya que los scripts se procesan por orden alfabético. Se puede utilizar cualquier número entre 10 y 99.

Si por alguna razón el comando update-rc.d no crea los enlaces porque ya están creados, existe la posibilidad de eliminarlos con la opción '-f' (forzado) y acto seguido volver a crearlos:

```
// Eliminación forzosa de enlaces para inicio automático del servicio
     # update-rc.d -f samba remove


// Volver a crear enlaces para inicio automático del servicio
     # update-rc.d samba start 20 3 4 5 . stop 20 1 2 6 .
```

Acceso a entorno gráfico como root

Aunque en un entorno de producción nunca necesitaremos acceder como root al entorno gráfico, durante el curso será muy cómodo poder autentificarnos como root y disfrutar de las herramientas gráficas. Por defecto viene deshabilitado el acceso como root al entorno gráfico. Para habilitarlo, previamente tenemos que establecer la contraseña de root con el comando 'sudo passwd root'. Luego, en el entorno de ventanas GNOME, debemos ejecutar 'sudo gdmsetup' y en la pestaña 'Seguridad' activar la opción 'Permitir entrada local al administrador del sistema'. En el entorno de ventanas KDE, debemos establecer a 'True' el parámetro 'AllowRootLogin' en el archivo '/etc/kde3/kdm/kdmrc'.

Resolución local de nombres de dominio

Si no disponemos de servidor DNS pero queremos resolver nombres de dominio de nuestra red local por sus respectivas IPs, una opción es editar el archivo **/etc/hosts** y añadir en él tantas líneas como nombres queramos resolver. Ejemplo, si quieremos que nuestro PC resuelva el nombre de nuestro servidor 'www.ieslapaloma.com' por la IP local 192.168.1.239, tendremos que añadirlo en el archivo /etc/hosts:

```
//Añadir en /etc/hosts
192.168.1.239    www.ieslapaloma.com
```

Así, cada vez que accedamos a http://www.ieslapaloma.com, se cargará la página de nuestro servidor.

Configuración de la red

En sistemas Debian, la red se configura en el archivo:

```
// Archivo de configuración de red
     /etc/network/interfaces
```

En dicho archivo se configuran los parámetros de todas las interfaces de red como la dirección IP, la máscara de subred, la dirección de red, la dirección de broadcast y la puerta de enlace. A continuación mostramos un ejemplo de dicho archivo:

```
 //Ejemplo de archivo /etc/network/interfaces
     auto lo

     iface lo inet loopback  #Interface lazo localhost. Necesaria



     auto eth0               #Primera tarjeta de red

     iface eth0 inet dhcp    #Configuramos por DHCP



     auto eth1               #Segunda tarjeta de red

     iface eth1 inet static  #Configuramos manualmente

     address 192.168.1.239 #Dirección IP

     netmask 255.255.255.0 #Máscara de subred

     gateway 192.168.1.1 #Puerta de enlace



     auto wlan0              #Tarjeta de red inalámbrica

     iface wlan0 inet dhcp   #Configuramos por DHCP
```

Después de editar el archivo de configuración /etc/network/interfaces, para que la nueva IP tome efecto debo reiniciar los servicios de red con el siguiente comando:

```
 //Aplicar la nueva configuración de red
      /etc/init.d/networking restart
```

Ahora tan solo nos quedaría la configuración de los servidores DNS. Los DNS se configuran en el archivo /etc/resolv.conf. Se pueden añadir tantas líneas como servidores queramos configurar. Si queremos configurar solamente dos de telefónica, el archivo quedaría así:

```
 //Archivo  /etc/resolv.conf con los DNS de telefónica
     nameserver 80.58.0.33
```

```
nameserver 80.58.32.97
```

Comandos útiles

su //Cambia el usuario actual a root o al usuario que indiquemos. Pide contraseña excepto a root.
sudo comando //Ejecuta comando como root
locate archivo //Localiza un archivo; updatedb actualiza base de datos de locate.
cat, more, less //Muestran el contenido de un archivo
poweroff, reboot //Apaga, reinicia el sistema
grep patrón archivo //Busca líneas que contengan un patrón
scp archivo usuario@pc-destino:/archivo //Copia archivo de un PC a otro, por ssh
ifconfig //Muestra o establece la configuración IP
mv archivo destino //Mueve un archivo de una carpeta a otra
rm archivo //Borra un archivo. Con opción -rf, borra una carpeta
tar xzpf archivo.tar.gz //Descomprime archivo tar.gz
set //Muestra variables de entorno
df -h //Muestra las particiones del disco y su ocupación
du carpeta -sh //Muestra el tamaño de una carpeta
du * -sh //Muestra el tamaño de todas las carpetas
setterm -blenght 0 //Anula el pitido en modo texto. Si estamos en entorno X, ejecutar xset b off
date -s "27 jan 09 18:38" //Establece la fecha y hora del sistema a 27 de enero de 2009 a las 18:38
dpkg --list //Lista todos los paquetes instalados
dpkg -L nombre-paquete //Muestra los archivos que conforman el paquete
dpkg -S texto //Muestra los paquetes con algún archivo que contenga 'texto'

Archivo /etc/apt/sources.list

El archivo /etc/apt/sources.list permite establecer los repositorios a los cuales accederá el comando apt-get cuando tenga que instalar una aplicación.

Durante el curso utilizaremos el repositorio principal (main) de ubuntu, por lo tanto, nuestro archivo /etc/apt/sources.list deberá contener las líneas:

```
// Repositorio 'main' de ubuntu


    deb http://es.archive.ubuntu.com/ubuntu/ jaunty main restricted


    deb-src http://es.archive.ubuntu.com/ubuntu/ jaunty main restricted


    deb http://es.archive.ubuntu.com/ubuntu/ jaunty-updates main
    restricted


    deb-src http://es.archive.ubuntu.com/ubuntu/ jaunty-updates main
    restricted
```

En alguna ocasión utilizaremos algún paquete que no se encuentra en el repositorio 'main' sino en otros repositorios como el 'universe' o el 'multiverse' de ubuntu. En tal caso tendremos que añadir a nuestro

archivo /etc/apt/sources.list las siguientes líneas:

```
// Repositorio 'universe' de ubuntu

     deb http://es.archive.ubuntu.com/ubuntu/ jaunty universe

     deb-src http://es.archive.ubuntu.com/ubuntu/ jaunty universe

     deb http://es.archive.ubuntu.com/ubuntu/ jaunty-updates universe

     deb-src http://es.archive.ubuntu.com/ubuntu/ jaunty-updates universe

// Repositorio 'multiverse' de ubuntu

     deb http://es.archive.ubuntu.com/ubuntu/ jaunty multiverse

     deb-src http://es.archive.ubuntu.com/ubuntu/ jaunty multiverse

     deb http://es.archive.ubuntu.com/ubuntu/ jaunty-updates multiverse

     deb-src http://es.archive.ubuntu.com/ubuntu/ jaunty-updates multiverse
```

Es posible que las líneas anteriores ya estén en nuestro archivo /etc/apt/sources.list pero estén anuladas por una almohadilla. En tal caso debemos eliminar la almohadilla para descomentar las líneas.

Cada vez que se haga una modificación del archivo /etc/apt/sources.list, es necesario ejecutar el comando 'apt-get update' para que nuestro sistema acceda al listado de paquetes de los nuevos repositorios y actualice nuestra base de datos con la información de los paquetes.

Esta configuración es válida para la versión 9.04 de Ubuntu (Jaunty Jackalope). Si utilizamos otra versión de Ubuntu, tendremos que sustituir dapper por su nombre corto:

Versiones de Ubuntu
Versión / Nombre largo / Nombre corto
6.06 / Dapper Drake / dapper
7.04 / Feisty Fawn / feisty
7.10 / Gutsy Gibbon / gutsy
8.04 / Hardy Heron / hardy
8.10 / Intrepid Ibex / intrepid
9.04 / Jaunty Jackalope / jaunty
9.10 / Karmic Koala / karmic

Si deseamos instalar un paquete que no está en los repositorios que tenemos configurados en /etc/apt/sources.list, podemos buscar repositorios para un paquete dado en: »http://www.apt-get.org

```
// Ejemplos de utilización de apt-get


     apt-get update // refrescar actualizaciones disponibles

     apt-get upgrade // actualizar todos los paquetes

     apt-get dist-upgrade // actualizar versión

     apt-get install paquete // instalar paquete

     apt-get remove paquete // desinstalar paquete

     apt-get --purge remove paquete // desinstalar paquete y eliminar
     configuración

     apt-get autoremove // eliminar paquetes obsoletos

     apt-get -f install // intentar arreglar paquetes

     apt-get -d paquete // Para bajar el paquete y sus dependencias sin
     instalar.

     dpkg --configure -a // intentar arreglar paquetes rotos

     dpkg -i paquete.deb // instalar archivo paquete.deb

     /etc/apt/sources.list // lista de repositorios APT

     http://packages.ubuntu.com/paquete // busca paquete en los
     repositorios Ubuntu
```

Teclado español en DSL Linux

Para que al arrancar DSL Linux utilice el teclado español, debemos agregar el parámetro '-k es' en la línea boot: del arranque del sistema.

Activar history-completion

La función history-completion consiste en recuperar los últimos comandos ejecutados en la consola mediante la flecha hacia arriba de los cursores, **indicando las iniciales del comando**. Se trata de una utilidad muy interesante ya que permite recuperar comandos anteriores indicando sus iniciales y pulsando la flecha arriba de los cursores, lo cual ahorra mucho tiempo al encontrar los comandos rápidamente evitando tener que pulsar la flecha arriba demasiadas veces. Para activarlo, hay que añadir las siguientes líneas en /etc/inputrc

```
 "\e[B":         history-search-forward #Up-Arrow
 "\e[A":         history-search-backward #Down-Arrow
```

Bash alias

Si deseamos crear alias de bash para nuestro usuario, debemos hacerlo en el archivo ~/.bashrc (el gusanillo significa 'nuestro home'). El archivo comienza por un punto '.' lo que le hace oculto. Para mostrar los archivos ocultos con ls debemos añadir la opción -a.

Si deseamos crear alias de bash para todos los usuarios, debemos hacerlo como root en el archivo /etc/bash.bashrc

FIN