# Linux en el instituto

Antonio Gómez

**En un centro educativo, resulta muy atractiva la idea de un ordenador central que cumpla funciones web, de filtrado de contenidos no aptos para menores y que permita centralizar y coordinar muchas actividades de enseñanza-aprendizaje basadas en las TIC.**

linux@software.com.pl

Vamos a tratar de reflejar, a lo largo de dos artículos, la experiencia del Instituto de Educación Secundaria Eduardo Valencia, en Calzada de Calatrava (Ciudad Real) en lo referente a la reinstalación de la red informática del centro, centralizándola en la figura de un ordenador servidor con Ubuntu Server 9.04. Las funciones principales de dicho servidor dentro de la red serán:

- Racionalización de la conexión compartida a Internet de los equipos del centro, así como la posible denegación de acceso de ciertos equipos a determinadas direcciones, si así lo solicita el profesorado del centro(Squid).
- El trabajo como servidor web con Apache2.
- Facilitar la compartición de carpetas entre equipos Windows, Molinux y Ubuntu con Samba.
- Disponer de un sistema de correo interno (externalizable) que posibilite el intercambio de documentación entre el personal del centro.

Las razones para llevar a cabo un experimento tan arriesgado (en la medida en que durante el tiempo que duran los distintos experimentos de instalación, testeo, reinstalación, etc... mantenemos interrumpida parte de la normal actividad informática dentro del centro), son muchas y muy variadas; mencionemos sólo las más importantes:

- Incluso un instituto pequeño como el nuestro puede llegar a mantener más de 90 equipos conectados a Internet, compartiendo una conexión de apenas 3 Mb (muchos días, sólo en teoría). La presencia de un servidor proxy en el sistema que guarde en su caché las páginas más solicitadas aceleraría el proceso. Si tenemos en cuenta que además, en un centro educativo, se solicitan muchas veces las mismas direcciones web (varias páginas de corte educativo, o el portal de la Junta de Comunidades de Castilla la Mancha, por poner sólo dos ejemplos), el aumento en la velocidad de conexión resultará mucho más visible.
- Es muy interesante, a todos los niveles, disponer de un servidor web accesible, al menos en parte, a profesores

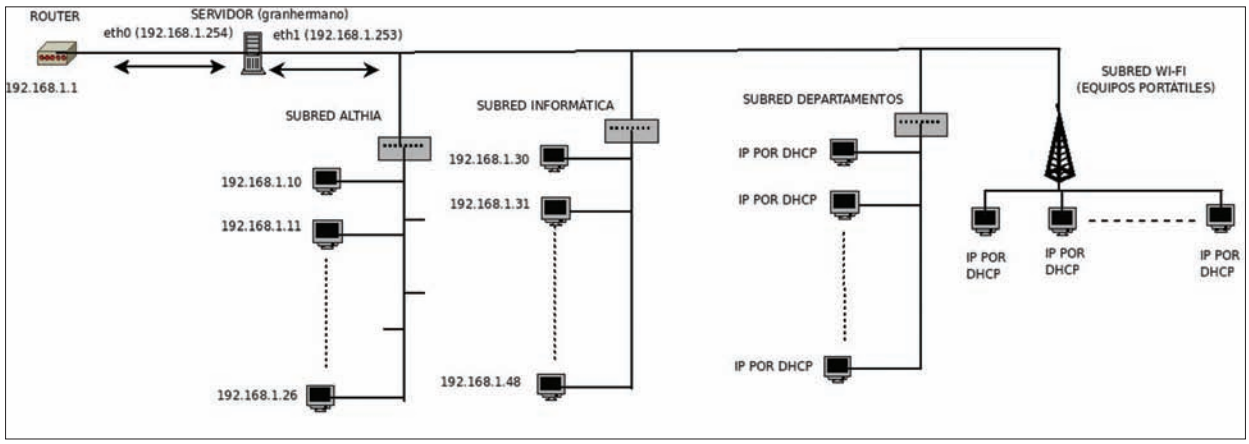


**Figura 1.** Configuración original de la red

y alumnos, con la autonomía propia que concede la presencia en el instituto de un servidor dedicado. Apache2 está demostrando ser muy potente y sencillo de configurar.

- Hay muchos equipos con Linux (Ubuntu y Molinux) instalados en el centro (de hecho, el último año, la Junta de Comunidades de Castilla la Mancha dotó a todos los profesores de la Comunidad Autónoma con un ordenador portátil con arranque dual en Windows XP y Molinux Hidalgo), pero el sistema de Microsoft sigue siendo aún mayoritario. Es muy necesario facilitar la cohabitación de ambos sistemas en el intercambio de archivos entre los distintos miembros de nuestra comunidad. Samba resolverá este conflicto.
- El correo electrónico es una herramienta muy presente hoy en día en las distintas actividades burocráticas y de gestión administrativa que conlleva el día a día en un centro educativo. Hemos dotado a cada Departamento y al Equipo Directivo del centro con sus propias direcciones web, que evitan al docente tener que utilizar su dirección privada de correo (que no tiene por qué verse obligado a utilizar en estos avatares), y dotan a nuestra institución de una pátina de autonomía y respetabilidad que supondrán una mejor imagen externa.

En efecto, para el redactor de este artículo son razones más que suficientes para iniciar una pequeña "guerra interna" a lo largo de dos o tres semanas con sus compañeros de profesión; una sucesión de pequeños conflictos que, me alegra decirlo, no pasaron del rango de pequeñas reclamaciones, casi siempre por la no disponibilidad de la red en varias ocasiones, lo que provocó unas cuantas veces la interrupción de determinadas actividades de aprendizaje que algún profesor llevaba a cabo con su grupo (realización de webquests, trabajo con blogs,...). Desde aquí mi agradecimiento a todos mis compañeros por la paciencia que demostraron en todo momento. Este tipo de situaciones refuerzan siempre nuestra sensación de auténtica pertenencia a una comunidad.

## Instalación física del servidor

El sistema elegido finalmente para nuestro ordenador central ha sido Ubuntu Server 9.04, dada su sencillez, fiabilidad y eficiencia. Además, ya dispone de la mayoría de los paquetes necesarios para iniciar nuestra actividad, y sus repositorios oficiales ya disponen de la mayoría de los que aún no estarán instalados pero necesitaremos. Recordemos que la herramienta más adecuada para instalar un paquete desde consola será aptitude. Ej: `# aptitude install openssh`.

Bien. No es objeto de este texto indagar sobre la instalación de un sistema operativo Linux en un equipo, además de que en la actualidad se ha convertido en una actividad de lo más sencilla. Recordemos que Ubuntu Server, como todos sus homónimos, no dispone en un principio de un entorno de escritorio, ni Gnome ni Kde, dado que no está pensado para equipos personales, y su presencia sólo redundaría en una menor eficiencia. Así que será imprescindible tener un cierto nivel de manejo con la shell.

Nuestro equipo en cuestión dispone de dos tarjetas de red, eth0 (conexión al router) y eth1 (a la red local). Su nombre de equipo, en un alarde de humor negro, será *granhermano*, y el nombre de usuario con posibilidades de root *jefazo*.

La red original en la que queremos implantar este equipo estaba configurada del siguiente modo: la conexión al exterior se hacía sobre un router, que alimentaba a cuatro subredes locales:

Aula Althia: sala con dieciséis ordenadores con arranque dual Windows y Molinux, parte de un proyecto de la JCCM de hace un par de años, para mejorar la informatización de los colegios e institutos. Aula de informática: sala con dieciocho ordenadores con arranque dual Windows y Ubuntu.

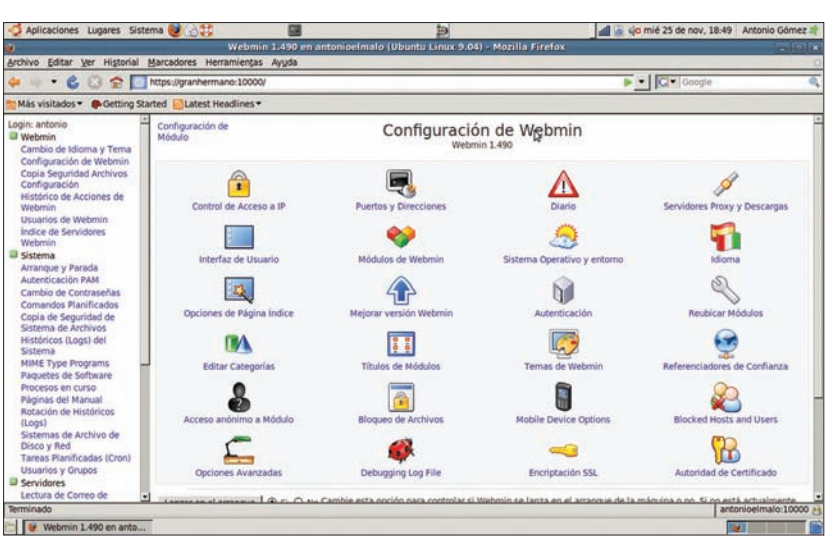


**Figura 2.** Explorando un poco, se puede encontrar rápidamente la función que necesitamos

Departamentos Didácticos: desde un switch, se cableó a lo largo de todo el centro el acceso a Internet del ordenador de cada Departamento. Unos veinte ordenadores más, contando los tres de la biblioteca del instituto.

Red Wi-Fi. Desde hace dos años, la Junta de Comunidades dotó también de los recursos necesarios para garantizar el acceso wi-fi a cualquier ordenador desde cualquier punto del instituto. A la sazón, tenemos instalada la red correspondiente de puntos de acceso por todo el edificio.

La inserción de nuestro servidor en este conglomerado nos dejaría en una situación como la de la Figura 1.

Los pasos que daremos, una vez instalado físicamente el servidor dentro de la red, serán los siguientes:

- Actualizar adecuadamente el servidor, y añadir los usuarios (departamentos didácticos y profesores) que deseamos tengan acceso al servidor.
- Configurar el sistema de acceso remoto al equipo *granhermano* mediante consola (*OPEN SSH*) y explorador web (*WEBMIN)*.
- Preparar el sistema de carpetas compartidas mediante *SAMBA*.
- Configurar el servidor proxy *SQUID*, que deseamos que funcione en modo transparente. Para ello, también tendremos que configurar un servidor *DHCP* e *IPTABLES*.

Hasta aquí, lo que trataremos a lo largo del presente artículo. En el próximo número, trataremos también la instalación y configuración de *APACHE2*, así como la creación de un servicio interno (accesible desde el exterior) basado en *POSTFIX*, *DOVECOT* y *SQUIRRELMAIL*, así como la posibilidad de dar acceso limitado al servidor a los profesores para realizar tareas personales sencillas, como cambiar su propia clave de acceso.

Bien, vayamos a ello. La historia no la escriben los cobardes (aunque bien es cierto que éstos, al menos, tienen la oportunidad de leerla después).

## Configuración del acceso remoto del servidor

De acuerdo. Hemos instalado Ubuntu Server en nuestro equipo. De manera provisional, mientras duren los testeos, el equipo dispone de los periféricos mínimos (teclado y monitor), pero en la instalación definitiva, no será así, dado que estará debidamente retirado de la vista al público. Por ello, el primer paso es asegurar que podemos acceder a *granhermano* tanto dentro como fuera de nuestra red LAN. Para ello, instalaremos *OPENSSH* para el acceso remoto por terminal, y la utilidad *WEBMIN* para el acceso gráfico vía web.

**Listado 1.** Archivo /etc/network/interfaces para configurar dos tarjetas de red, eth0 y eth1

```
auto lo
iface lo inet loopback
auto eth0
iface eth0 inet static
        address 192.168.1.254
        gateway 192.168.1.1
        netmask 255.255.0.0
        network 192.168.1.0
        broadcast 192.168.1.255
     auto eth1
iface eth1 inet static
address 192.168.1.253
netmask 255.255.0.0
```

**Listado 2.** Configurando el servidor DHCP /etc/dhcp3/dhcpd.conf

```
#Opciones generales
#Establecemos como servidor dns primario el propio router, que
normalmente estará configurado #para funcionar como tal.
option domain-name-servers 192.168.1.1;
#Broadcast-address fija la máscara que se utilizará para distribuir la
señal (toda la red)
option broadcast-address 192.168.255.255;
#Fijamos como puerta de enlace primaria la dirección IP de eth1
option routers 192.168.1.253;
#Fijamos como nombre de dominio el de nuestro servidor
option domain-name "granhermano";
#Desactivamos la actualización automática de dns dinámicas.
ddns-update-style none;
#La opción authoritative marca nuestro servidor como el principal
proveedor DHCP de direcciones; #si comentamos esta opción, deberíamos
desactivar el servidor DHCP de nuestro router;
authoritative;
# Creamos nuestra subred
subnet 192.168.1.0 netmask 255.255.255.0 {
         range 192.168.1.50 192.168.1.200;
         default-lease-time 21600;
         max-lease-time 43200;
}
```

### Configuración del router para el acceso remoto

Dependiendo del modelo de router que se utilice en cada red, los pasos de configuración a seguir pueden variar, pero siempre se vertebrarán en torno a un protocolo parecido a éste:

- El router puede configurarse mediante nuestro explorador web favorito (*Firefox*, por ejemplo), normalmente en la dirección 192.168.1.1 (naturalmente, necesitaremos el nombre y la clave de usuario autorizado).

**Tabla 1.** Redireccionamiento de puertos web en el router

| PUERTO | DIRECCIÓN IP |
|---|---|
| 8080 | 192.168.1.1 |
| 80 | 192.168.1.254 (eth0 en *granhermano*) |

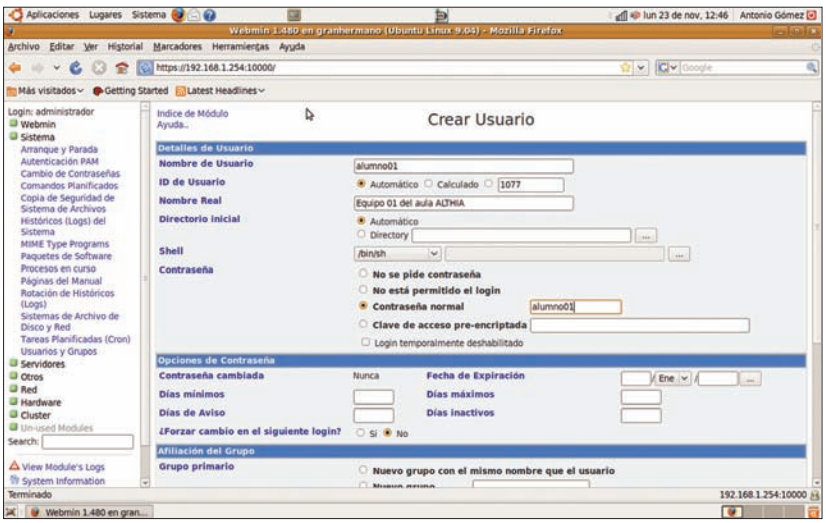


**Figura 3.** Creando usuarios y grupos desde WEBMIN

- Necesitamos indicar al router a qué ordenador debe desviar las peticiones externas realizadas a determinados puertos. En nuestro caso, debemos reconducir a la IP 192.168.1.254 (eth0) todas las peticiones hechas al puerto 22 (ssh), 10000 (webmin), 80 (apache) y 25 (postfix). Una opción para evitar problemas los primeros días sería aprovechar la característica DMZ (Demilitarized Zone) que suelen ofrecer los router estándar, y que abren directamente todos los puertos de un ordenador en particular dentro de la red, si bien es recomendable limitarse a los puertos señalados una vez pasado el período de pruebas.
- En nuestro caso particular, y para evitar las molestias derivadas de los cambios periódicos que pueden derivarse de la utilización de IP's dinámicas, recurrimos al servicio gratuito NO-IP, que nos permite recurrir a sus servicios de DNS y contar con nuestra propia URL (*http:/eduardovalencia.no-ip.org)*.

## Configuración del servicio de DNS dinámicas

Como ya sabemos, lo que conocemos como direcciones web o URL son transcripciones a lenguaje "normal" de las direcciones IP que corresponden a las ubicaciones físicas de los servidores web cuyos servicios deseamos utilizar. Los responsables de estas relaciones entre IP y URL son los servidores DNS. Tener nuestro servidor localizable en Internet nos ofrecía dos problemas.

- Necesitamos elegir dicho servidor DNS, en nuestro caso de carácter gratuito. Optamos por NO-IP (*http://www.no-ip.com*).
- Nuestro servidor se encuentra dentro de una red local LAN. El servidor DNS mantendrá en su base de datos la dirección IP general (la de nuestro router), que deberá redireccionar las peticiones web a *granhermano*, pero teniendo en cuenta, además, que la política de los actuales proveedores de ADSL considera una práctica aconsejable el cambio de dirección IP en los router de manera periódica. Deberemos, pues, asegurarnos de que cada vez que dichos cambios de dirección se produzcan se comunique a la base de datos de nuestro DNS, lo que se logra mediante la instalación de un programa cliente.

La ventaja de estos problemas estriba en que conocemos, por lo menos, su naturaleza, así que procedemos a resolverlos.

a) Daremos por supuesto que estamos registrados en uno de estos servicios (*dyndns* y *no-ip.com* son los más populares), de modo que tenemos una dirección asignada, y hemos descargado ya el programa cliente que avisa a *dyndns* o *no-ip* del cambio en la IP general cuando éste se produce. Dicho programa se puede descargar de la página web correspondiente, y habría que descomprimir la carpeta, compilar y configurar. El archivo suele incluir una sección de documentación, donde algún fichero del tipo LEEME.PRIMERO nos indicará cómo debemos actuar. Los S.O. de base Debian (nuestro queridísimo Ubuntu entre ellos), incorporan ya, en sus repositorios, dicho paquete: `# apt-get install no-ip` o `# aptitude install no-ip`.

Sólo restaría configurarlo (se nos pide la dirección de correo y la contraseña con la que constamos como usuarios registrados en dicho servicio).

b) Normalmente, accedemos a la utilidad web de configuración del router en la dirección 192.168.1.1, previa identificación como administrador de la red LAN (esta identificación no tiene nada que ver con el usuario *jefazo* de nuestro equipo). En un principio, la mayoría de los router incorporan directamente una opción de configuración de IP dinámica (por ejemplo, en los de marca COMTREND aparecen en el menú: *DNS->Dynamic DNS,* o en los de tipo *U.S. ROBOTICS* se ofrece en *DNS->DNS Dinámicas)*. En esa opción debería pedir nuestra identificación como usuarios del servicio (es decir, habría que repetir el paso anterior, pero en el router en vez de en el servidor).

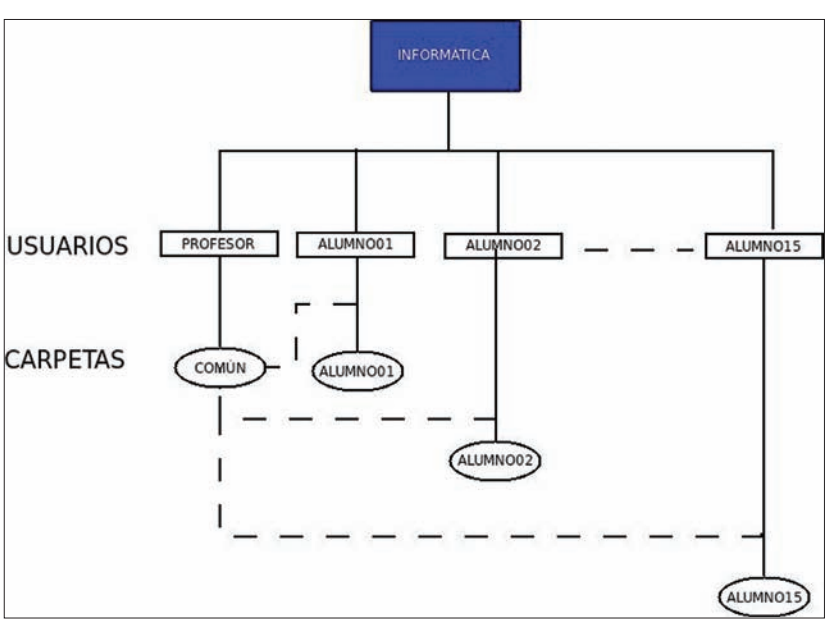


**Figura 4.** Estructura del grupo de trabajo INFORMÁTICA

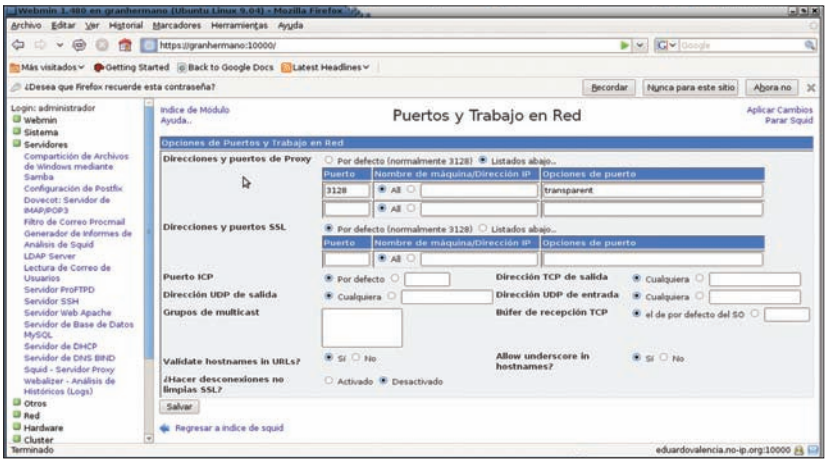


**Figura 5.** En WEBMIN, podemos configurar nuestro servidor Squid como transparente

c) El tercer paso es donde tropezamos varias veces, antes de caer en lo obvio: nuestro router está preparado para recibir peticiones del puerto 80 desde los servidores dns de *dyndns* o *no-ip*, *granhermano* está preparado para avisar a dichos servidores de los cambios que se puedan producir en la IP del router (aunque creo, en el fondo, que este segundo aspecto suele ser totalmente inútil en el conjunto); pero... cuando se hace una petición a nuestro router, ¿Cómo sabe el router a qué ordenador dirigir la petición web dentro de nuestra red? Por eso, tenemos que configurar el router para que envíe cualquier petición que se le haga, al puerto 80 por ejemplo, en el caso del servidor web, a la dirección IP de nuestro *APACHE*. Investigando en Internet, descubrimos que un problema añadido en varios casos es que el router puede tener reservado dicho puerto 80 a su propia página web de configuración. La solución que se propone consistiría, si eso nos da problemas, en cambiar el acceso a dicha web interna al puerto 8080, y dejar el 80 a tu APACHE. En nuestro caso, quedaría algo como en la Tabla 1.

## Instalación de *OPENSSH*

Esta utilidad nos permitirá gobernar de manera remota el ordenador desde cualquier otro equipo que disponga de un programa cliente que permita la conexión, sea Windows (*PUTTY es la mejor opción para mí), o Linux (desde consola, SSH-CLIENT),* con la ventaja añadida de que el intercambio de información entre equipos se realiza de manera encriptada, lo que aumenta la seguridad.

Los pasos a seguir serán los siguientes:

- Instalación del servidor: *aptitude install openssh-server.*
- Instalación de la utilidad cliente, tanto en el servidor como en nuestro equipo personal:*aptitude install openssh-client.*
- En el servidor, configuración de la utilidad: *ssh localhost.*

Este tercer paso implica la creación de un par de claves (keys), una pública y una privada, cuya combinación entre nuestro equipo personal y el servidor asegurará nuestra identificación adecuada en cada conexión como usuario autorizado.

Para comprobar si todo ha funcionado, bastará con abrir una consola en nuestro equipo personal (en el caso de Microsoft, iniciar *PUTTY*), y tratar de conectarnos: `# ssh jefazo@granhermano` (dentro de la red local) o `# ssh jefazo@IPPUBLICADENUESTRARED` (por ejemplo, desde nuestro domicilio, siempre que hayamos configurado correctamente el router).

Si todo ha ido bien, aparecerá un mensaje indicando la comprobación de la *public key* y pidiendo confirmar la conexión, hecho lo cual se nos pedirá la contraseña del usuario *jefazo*, que tiene los permisos necesarios para acceder al sistema.

## Instalación de WEBMIN

Webmin es una utilidad escrita en PHP que nos permite la gestión remota de equipos con Linux instalado a través de una interfaz gráfica que facilita, no mucho, sino muchísimo, la gestión de un equipo cuando somos usuarios poco avezados, y por qué no decirlo, un poco sobrepasados por nuestra falta de experiencia.

Partimos de que en este momento, nos encontramos en nuestra carpeta *HOME*. Vamos a descargarnos el paquete mediante la herramienta *wget*, le daremos los permisos necesarios al archivo descargado, un autoinstalable *.deb, y finalizaremos instalándolo:

```
# wget http://prdownloads.sourceforg
e.net/webadmin/webmin_1.490_all.deb
# chmod 755 webmin_1.490_all.deb
# ./webmin_1.490_all.deb
```

Para comprobar la funcionalidad de esta interfaz, nos conectaremos desde el explorador web de nuestro equipo personal: *https://192.168.1.254:10000* desde dentro de la red local, aunque también podría servir: *https://granhermano:10000.*

La dirección de acceso externo sería: *https://IPPUBLICADENUESTROROUTER:10000.*

El correcto desarrollo de esta fase de nuestro trabajo permitirá el acceso desde cualquier equipo, dentro o fuera de la red, al servidor central. Los prudentes y los cobardes (métaseme en el grupo que más agrade a nuestro dignísimo lector), omitirán la desconexión de teclado y monitor del equipo mientras no se demuestre la estabilidad del conjunto por activa y por pasiva, pero está claro que es algo que ya podría hacerse.

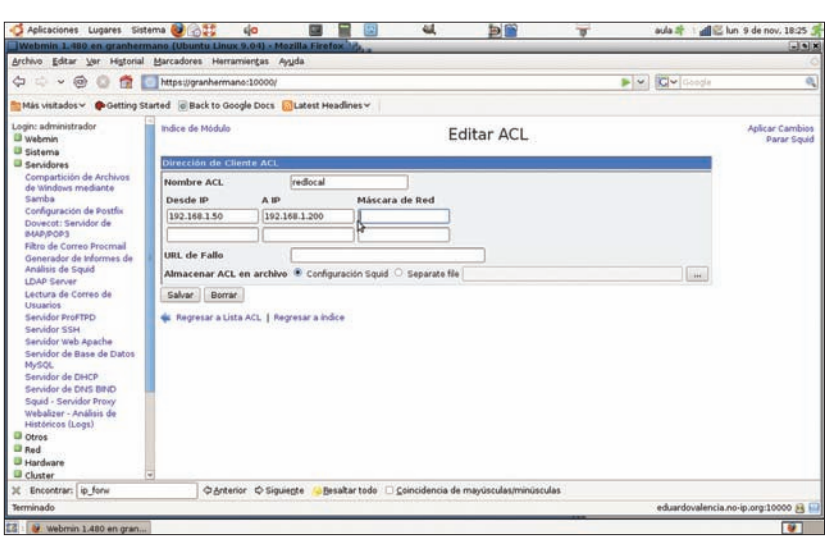


**Figura 6.** En el ejemplo, definimos la red con acceso a Internet desde 192.168.1.50 a 192.168.1.200

**Listado 3.** Contenido del archivo configurandoiptables.sh

```
#Habilitamos el ruteo
sudo echo 1 > /proc/sys/net/ipv4/ip_forward
#Redireccionamos todos los paquetes que vienen por el puerto 80 a eth1
hacia el paquete 3128
sudo iptables -t nat -A PREROUTING -i eth1 -s 192.168.1.0/24 -d !
192.168.1.0/24 -p tcp --dport 80 -j REDIRECT --to-port 3128
```

En ocasiones, después de instalar nuevas utilidades, es posible que echemos en falta el correspondiente módulo en el menú de *WEBMIN*. Es posible que esta interfaz web no sepa cómo manejar esta aplicación, pero eso sólo sucederá en contadas y muy específicas situaciones. Lo más normal es que baste con *Refresh Modules* (si es que todavía no hemos cambiado el lenguaje por defecto, *WEBMIN ->Webmin->Change Language and Theme*; por cierto, ¿sabía el lector que puede, incluso, operar en lenguaje Klingon?).

### Configuración de las tarjetas de red

Durante la instalación de nuestro sistema operativo, en un momento determinado se nos preguntará sobre la configuración de nuestra red. En ese momento, por razones de comodidad, dejaremos que el propio programa configure dichas tarjetas automáticamente, por DHCP, pero cuando terminemos, deberemos configurar dichas tarjetas en el archivo /etc/network/interfaces (el editor de textos *nano* o *vi* servirá de sobra). El contenido de dicho archivo debería ser (ver Listado 1).

### Creación de usuarios y grupos en el sistema

Muy importante, de cara a otros pasos que se detallan después, es tener muy claro qué usuarios deberían tener acceso a *granhermano* en cada caso, así como en qué grupos, primarios y secundarios, querremos encuadrarlos, y qué permisos de lectura y escritura les vamos a asignar en cada servicio. Este es uno de los momentos en que nos sentiremos agradecidos a *WEBMIN* por su simplicidad, aunque esto menoscabe, en cierto modo, la filosofía del usuario de la shell de sacrificar simplicidad por potencia.

En un principio, la política de cualquier sistema con base UNIX es orientarse al trabajo con múltiples usuarios. Cada vez que alguien se conecta al ordenador, se identifica con su nombre y su clave, y el equipo identifica automáticamente cuál es su carpeta de trabajo (*/home/nombredeusuario*), donde tiene permisos de lectura y escritura, así como el grupo o grupos a los que dichos usuarios pertenecen, y donde también pueden configurarse las atribuciones que deseemos asignarle. Recordemos que cada archivo y carpeta (como archivo "especial", que es como están considerados) incluyen un sistema de tres tipos de permisos (lectura, escritura y ejecución) para tres tipos distintos de usuarios (propietario, grupo de propietarios y otros), expresados por un sistema octal, en cuya base no entraremos ahora.

En cuanto a los grupos de usuarios de tipo "normal", lo cierto es que el administrador de la red no entró en excesivas consideraciones de tipo técnico ni organizacional: si los usuarios son alumnos y profesores, habrá un grupo *alumnos* y un grupo *profesores*.

Los usuarios individuales, a priori, se irían creando a petición de los distintos miembros de la comunidad que así lo deseen (siempre que el administrador lo considere conveniente, por supuesto). En un principio (como se verá en el apartado destinado a *SAMBA*), se crearán usuarios para cada ordenador individual de los espacios informáticos comunes (a la sazón, de *alumno01* hasta *alumno15* en el aula *ALTHIA*, y de *informatica01* hasta *informatica15* en el aula *INFORMÁTICA*, así como al ordenador de profesor de cada aula), y un usuario estándar por Departamento Didáctico (*tecnología, matemáticas, lengua, francés, inglés,...* etc).

Cada usuario tendrá permisos de lectura y escritura en su carpeta *home*, y podrá especificarse, en cada caso, si deseamos otorgar al resto de miembros del grupo permisos para escribir o ejecutar programas en dichas carpetas, o simplemente acceder a ellas. A la hora de intercambiar trabajos y documentos a través de la LAN, esta característica de los sistemas GNU/UNIX abrirá al profesor posibilidades muy interesantes. Así pues, entramos en nuestra interfaz *WEBMIN*, y una vez identificados como administradores con permisos de root, seleccionaremos la opción *Sistema->Usuarios y grupos->Grupos locales*. La pantalla de información, con los usuarios ya creados, que nos aparece, es de por sí suficientemente explicativa. Baste decir que crearemos los grupos de usuarios que consideremos necesarios (*Profesores* y *Alumnos*, como hemos dicho).

Del mismo modo, iremos creando cada uno de los usuarios individuales que consideremos necesario (*Sistema->Usuarios y grupos->Usuarios locales*), ateniéndonos siempre a las siguientes reflexiones:

- Los usuarios tienen un grupo primario (el que originalmente les está destinado), pero también se les puede asignar uno o más grupos secundarios.
- Todos los usuarios deberían tener una contraseña, por más simple que ésta sea.
- En casos especiales, pueden crearse usuarios sin carpeta *home*, o denegárseles el acceso a determinados servicios.

## Compartición de carpetas y archivos con SAMBA

A continuación, pasamos a configurar el sistema SAMBA de cara a compartir archivos en red en S.O. Windows y Linux.

### Organización de los grupos de trabajo

Muy bien, ya tenemos algo con lo que empezar. Si lo ya hecho hasta ahora no nos ha resultado lo suficientemente complicado, y a esta altura del artículo no nos hemos echado a llorar más de tres o cuatro veces, podemos enfrentarnos a *SAMBA*.

*SAMBA* (juego de palabras relacionado con SMB, *Server Message Block*, un protocolo de red diseñado para estas funciones) es una utilidad que permite compartir archivos entre varios ordenadores de una misma red local, independientemente del sistema operativo que éstos utilicen. Es altamente configurable, y trabaja no sólo con archivos, sino también con periféricos como las impresoras. Lo único que necesitamos tener claro a la hora de instalar samba es la organización de nuestros *GRUPOS DE TRABAJO*.

En el caso particular de nuestro instituto, nos interesa particularmente configurar dos grupos de trabajo: el del aula *ALTHIA* y el del aula *INFORMÁTICA*, dado que son conjuntos de ordenadores en los que interactuará el profesorado con varios grupos de alumnos, unas veces trabajando con Linux, otras veces no. Centrémonos en la estructura, por ejemplo, de *INFORMÁTICA* (Figura 4).

Disponemos de un usuario *profesorinformatica*, que debería tener permisos de lectura y escritura en todas las carpetas. Por otro lado, cada uno de los otros quince ordenadores deberá atender a un usuario denominado *alumnoXX* (siendo XX el número del equipo), que deberá tener permisos de lectura y escritura en su propia carpeta, no debería tener acceso directo a las carpetas de red de sus compañeros (para evitar la eterna tentación del *copiar y pegar*),

así como permisos de sólo lectura en una carpeta, *COMUN*, donde el profesor pueda crear los documentos de base a partir de los cuales el alumnado empiece a trabajar, pero que no sean borrables desde los otros equipos, por evitar errores más que por otra cosa (no olvidemos que trabajamos con grupos de jóvenes que suelen ser muy heterogéneos; siempre hay algún chico o chica que cometerá la peor equivocación posible en el peor momento). Utilicemos este ejemplo como base de configuración de nuestro sistema de compartición.

## Instalando y configurando *SAMBA*

Si no hemos instalado este paquete durante la instalación de *UBUNTU 9.04*, podemos hacerlo desde consola:

```
# aptitude install samba
```

Ante la pregunta que el sistema nos hace sobre si deseamos hacer funcionar SAMBA como daemon o desde inet-d, elegimos la primera opción .

La configuración de *SAMBA* puede hacerse de manera directa, desde el terminal, editando el archivo */etc/samba/smb.conf*. Sin embargo, como hemos remarcado varias veces a lo largo de este texto, nos estamos dirigiendo a lectores con un nivel de dominio de la shell bajo o incluso nulo, así que recurriremos a *WEBMIN*.

Una vez conectados a *WEBMIN*, y partiendo de que el usuario ya habrá configurado la herramienta a su gusto, idioma castellano incluido, desde el apartado *Servidores,* subapartado *SAMBA*, deberemos dar los siguientes pasos:

- Creación de los usuarios en Ubuntu Server. Se puede hacer mediante terminal (comando *adduser)*, o desde *WEBMIN* (*Sistema->Usuarios y grupos).*
- En *SAMBA*, *Crear una nueva compartición de archivo*. En el formulario subsiguiente, nos limitaremos a marcar la opción *Compartición de directorios de inicio*, y pincharemos en Aceptar, lo que nos generará la compartición *homes* (la carpeta home de cada usuario estará compartida por el protocolo SMB). Hay que recordar que en el home de *profeinformática* deberemos crear una carpeta denominada *COMUN.*
- *WEBMIN* incluye una opción para configurar automáticamente la sincronización de usuarios de UNIX y SAMBA, pero no parece funcionar muy bien. Para evitar errores, volveremos a añadir los usuarios de samba por la terminal, con el comando *smbpasswd nombreusuario -a*, tras el cual deberemos introducir la contraseña de dicho usuario.

## Conectando los equipos WINDOWS a *SAMBA*

En cada ordenador del aula, en Windows, tendremos acceso a dichas carpetas creando una UNIDAD DE RED en MIPC (menú *Herramientas->Conectar a unidad de red).* El trayecto de cada carpeta será \192.168.1.253\alumnoXX y \192.168.1.253\profesor-informatica\comun.

El nombre de usuario y password que se solicitarán a continuación serán los del alumno en cuestión. ¡No debemos olvidar dejar marcada la casilla que recordará esos parámetros cuando el equipo se reinicie!

## Servidor SQUID transparente, iptables, filtro web

Como siempre, empezaremos asegurando que disponemos de los paquetes necesarios. Si alguno nos faltara, no hay más que recurrir a nuestra "identidad secreta" como superhéroe *root*:

```
# sudo su aptitude install squid
squid-common
```

Squid es un servidor proxy caché, cuya principal función es regular los distintos accesos a Internet por parte de todos los equipos dentro de la red local. Su instalación y arranque son sencillísimos, especialmente desde *WEBMIN*.

Ahora bien, para que dicho proxy tenga efectividad, deberíamos reconfigurar nuestro navegador especificando la dirección de dicho servidor (192.168.1.254), cosa a todas luces inútil cuando trabajamos con tantos equipos y con tanta gente joven, máxime cuando uno de nuestros objetivos es regular el acceso a ciertos sitios como las redes sociales.

Así pues, lo que queremos, no es sólo un proxy. Queremos un proxy *transparente*. Esto es, conseguir que cuando un equipo se conecte a Internet desde nuestra red local, lo haga utilizando como *puerta de enlace* a *granhermano*, en la dirección *192.168.1.254*.

Para hacerlo, necesitamos dar los siguientes pasos:

- Granhermano debe funcionar como servidor DHCP, de modo que cuando los equipos se conecten y busquen de manera automática la IP con la que van a funcionar, sea dicho servidor el que se la de, y se autoasigne como puerta de enlace.
- Debemos configurar *SQUID* como servidor transparente (desde el mismo *WEBMIN* se puede hacer, basta con especificar la opción *transparent* al puerto que viene por defecto en *Opciones de puertos y trabajo en red*).
- Como *SQUID* trabaja con peticiones al puerto 3128, pero las peticiones web siempre se hacen desde el 80, reconfiguraremos *IPTABLES* creando una regla en el servidor (funcionando como puerta de enlace) que desvíe todas las peticiones web al servidor *SQUID.*
- Una vez esté el sistema (más o menos estable) funcionando, podemos crear *reglas de acceso (ACL's)* para determinar en qué

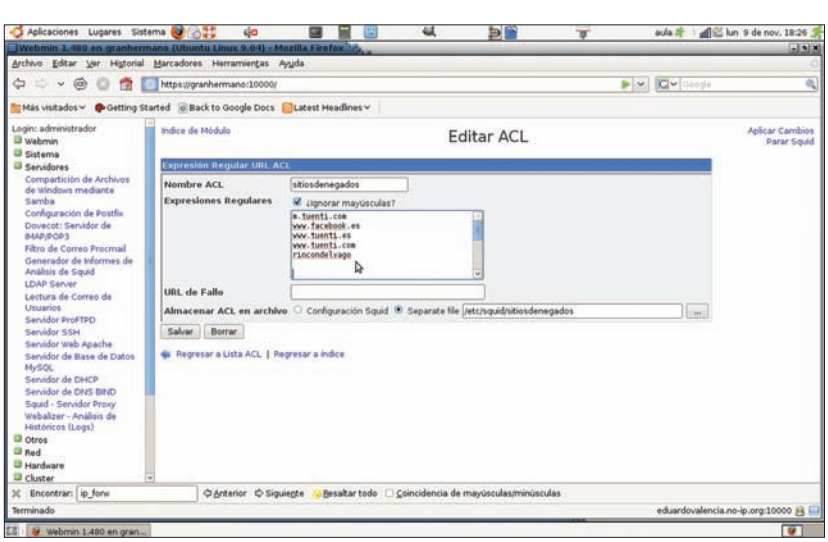


**Figura 7.** En el ejemplo, definimos las direcciones de Tuenti y Facebook, así como cualquier referencia al rincón del vago

condiciones pueden acceder a qué sitios de Internet cada uno de los equipos.

## Configuración del servidor DHCP

El paquete a configurar sería el *DHCPD:*

```
sudo aptitude install dhcp3-server.
```

La configuración del servidor DHCP puede configurarse por *WEBMIN* o de manera manual, en el archivo */etc/dhcp3/dhcpd.conf* (ver Listado 2). En el listado precedente, preparamos a *granhermano* para que conceda direcciones IP desde 192.168.1.50 a 192.168.1.250, reservándonos las otras direcciones para otras funciones (impresoras de red, servidores secundarios, prácticas de creación de redes locales con el alumnado de Bachillerato, etc...).

## IPTABLES

IPTABLES es una herramienta cortafuegos que permite a nuestro equipo interceptar y filtrar paquetes de red, de acuerdo a una serie de reglas de una muy sencilla gramática. En su estado original, SQUID trabaja con peticiones de otros equipos a través del puerto (por defecto) 3128, a diferencia de los programas exploradores como Firefox, que realizan sus peticiones al servidor a través del puerto 80. Por ello, es necesario introducir una regla que explique al ordenador que cualquier petición que venga desde la red local por el puerto 80 debe ser redireccionada al puerto 3128 (ver Listado 3).

Estas órdenes, que no deseamos tener que reescribir a cada reinicio del servidor, están en un script que llamamos configurandoiptables.sh y que se ha grabado (siempre como root) en la carpeta /etc/network/if-up.d

A partir de ahora, si todo ha ido bien (lo sabemos, lo sabemos, Murphy es especialmente estricto con los maestros y profesores; es posible que no haya salido a la primera, simplemente, repasemos lo ya hecho y volvamos a intentarlo); si todo ha ido bien, más tarde o más temprano, cada equipo de la red local que arranque a partir de ahora obtendrá su dirección IP de manera automática del servidor DHCP y realizará, sin saberlo, sus consultas TCP/IP a través de nuestro servidor.

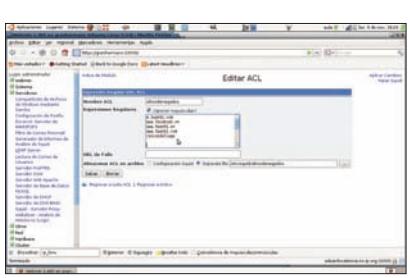


**Figura 8.** Dando acceso a la acl redlocal exceptuando sitiosdenegados

## Reglas de acceso en SQUID. Filtro web

Ya hemos mencionado un par de veces que uno de los mayores atractivos de utilizar *SQUID* en un centro educativo consiste en la posibilidad de sancionar de un modo muy sencillo el acceso a determinadas páginas por parte de nuestros alumnos.Una vez hemos conseguido que todos los navegadores del instituto pasen por nuestro querido *granhermano*, crearemos un par de reglas de control de acceso (*ACL*) muy sencilla a través de *WEBMIN*, la primera (*redlocal*), que definirá el conjunto de ordenadores que tendrá acceso a Internet dentro del instituto, y la segunda (*sitiosdenegados*), marcará el conjunto de páginas web o contenidos que queremos restringir. Una vez conectados al gestor web, en el apartado *Squid-Servidor Proxy*, subapartado *Control de acceso*, podremos hojear las distintas reglas que suelen venir por defecto con esta herramienta. Una de ellas suele ser *localnet*, que define la red 192.168.1.0, máscara de red 24, esto es, abarca todo el rango de direcciones IP 192.168.1.0/192.168.1.255. Podemos modificarla o definir nuestra propia ACL de este tipo, indicándole a *WEBMIN* que queremos definir una regla de tipo *Dirección de cliente.*

La segunda *ACL, sitiosdenegados*, es una regla de tipo *Expresión regular URL.* Podemos indicar direcciones web completas, o palabras relacionadas con los contenidos que queremos restringir. Este conjunto de palabras y direcciones puede guardarse en archivo aparte (es lo que hemos hecho nosotros), o directamente en el archivo de configuración de *SQUID, /etc/squid/squid.conf.*

Una vez definidas ambas reglas, sólo nos falta relacionarlas. En el mismo subapartado de *Control de acceso,* pestaña *Restricciones ICP*, añadiremos la regla *redlocal*, remarcando la excepción *sitiosdenegados*. De este modo, todos los ordenadores de nuestra red tendrán libre acceso, exceptuando las ocasiones en que hagan referencia a los sitios que hemos detallado en nuestra lista negra. Debo reconocer que esta parte de nuestro proyecto fue una de las más complejas, pero también las más satisfactorias. Uno de los mayores temores de muchos compañeros de profesión a la hora de abrirse al uso de las TIC ha estado siempre en el acceso, por parte de alumnos más rápidos que ellos, a sitios web digamos... bueno, más comprometidos. Desde ahora, podemos controlar y restringir el acceso a cualquier tipo de contenido que no sea específicamente educativo, léase diarios deportivos, pornografía, imágenes violentas...

## Conclusión

En este primer artículo, hemos emprendido la instalación básica de un ordenador lo suficientemente operativo como para incluirlo en la red local de nuestro centro educativo y permitir que el resto de la comunidad pueda seguir realizando sus actividades sin resultar afectados por ello. Nos hemos asegurado de que nuestra LAN tiene ahora un ordenador central que va a filtrar y controlar las entradas y salidas al exterior de los equipos con los que trabajan nuestros alumnos menores de edad, hemos logrado racionalizar el acceso a Internet, estamos preparados para compartir archivos y carpetas a través de la red de acuerdo a un sistema de permisos racional, y lo más importante, podemos administrar a *granhermano* desde el exterior, de modo que podamos realizar el resto de las tareas de instalación de forma externa; de hecho, en un centro educativo en el que varios grupos tienen que acceder a lo largo del día a las aulas con ordenadores, no podemos depender de tener un momento libre en nuestro horario como profesores, y esperar que justo en ese momento el aula no esté ocupada.

En una segunda parte, nos introduciremos en las apasionantes posibilidades que nos ofrece *APACHE2* para poner a disposición de cada miembro del personal, docente y discente, su propio espacio web (muy interesante para aumentar la cohesión de la comunidad desde el punto de vista de la comunicación), así como el proceso básico a llevar a cabo para instalar un sistema de correo para profesores y Departamentos, a través de *POSTFIX, DOVECOT* y *SQUIRRELMAIL*. También investigaremos en las posibilidades de ampliación de *WEBMIN* a través de los módulos *USERMIN*. ¡Hasta el próximo número!

## Sobre el autor

Ingeniero Técnico Industrial de formación, Antonio Gómez es profesor de Tecnologías en el IES Eduardo Valencia, en Calzada de Calatrava (Ciudad Real), desde el año 2004, donde desempeña el cargo de Responsable de Equipos Informáticos del centro. Anteriormente ha sido también asesor TIC en el Centro de Profesores de Puertollano (Ciudad Real), con el que sigue desarrollando diversos proyectos de innovación y formación relacionados con el uso del Software Libre en educación.