

CAPÍTULO 9

Configuración de un servidor web

Entre los usos más populares de Internet en segundo lugar nos encontramos con la navegación web, por detrás del correo electrónico. El servidor web más popular es con diferencia Apache; no solamente es gratis, un aspecto de máximo interés, sino que también es fuerte y rico en características. Además está disponible para cualquier plataforma.

El servidor web Apache tiene disponibles numerosas funcionalidades, que van aumentando con cada versión. Entre ellas están:

- Soporte para “cookies” (galletitas).
- Soporte para CGI (Common Gateway Interface, o Interfaz de Puerta Común).
- Autenticación de contraseña para las páginas o sitios web.
- Servidor proxy caché.
- Soporte de PHP y Perl (dos lenguajes de servidor para páginas dinámicas).
- Soporte de Java (a través de los proyectos “Jakarta” o “Java Apache Project”).
- Autenticación MySQL.
- Hosting virtual (varios sitios web en el mismo servidor).
- Soporte de XML (Extended Markup Language, o Lenguaje de Marcado Extendido).

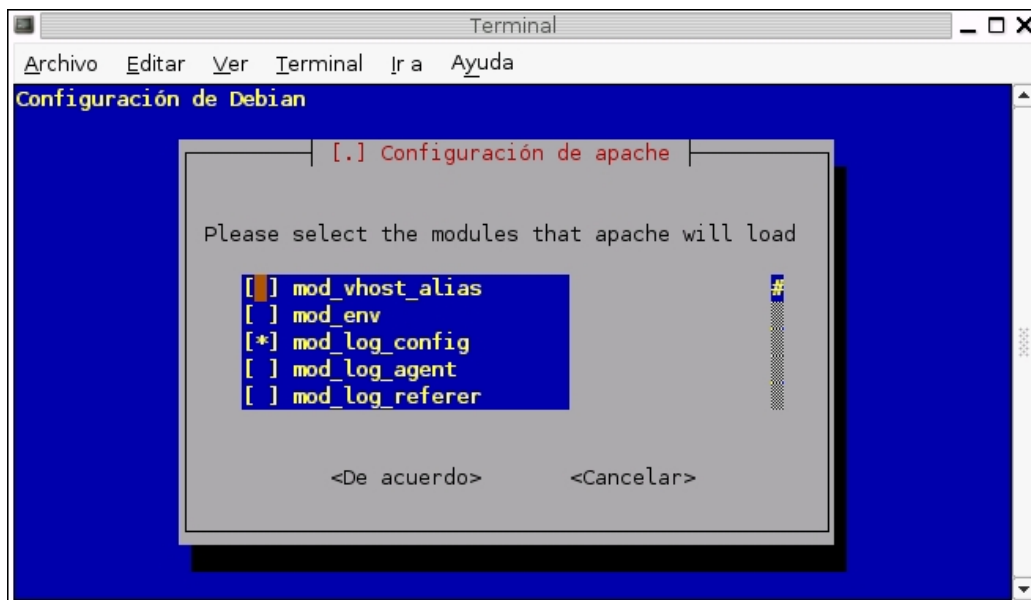
La lista es mucho más extensa, todo gracias a que Apache ha sido concebido bajo un diseño modular; esto quiere decir que existe un núcleo central que gobierna el servidor, y una serie de módulos cada uno con una funcionalidad específica o un grupo de funcionalidades; aunque esto resulta en una ejecución más lenta, siempre se puede no ejecutar aquellos módulos que no necesitamos, por lo que al desactivar estos módulos sí obtenemos un rendimiento superior del servidor web.

Para instalar Apache puedes utilizar apt (apt-get install apache) o synaptic; se instalarán varios paquetes para resolver las dependencias. Una vez concluida la descarga empieza la configuración con una serie de preguntas. Si alguna de estas preguntas no aparece, siempre puedes ejecutar “dpkg-reconfigure apache”, porque depende de la versión el que te haga todas las preguntas o sólo algunas durante la instalación.

La primera nos pregunta si queremos ejecutar el servidor Apache en el arranque del ordenador; contesta según quieras, pero recuerda que siempre podrás volver a cambiar esta decisión desde la aplicación “Servicios”, ya que Apache se ejecuta a través de un demonio propio.

La siguiente consiste en activar la opción de “suExec”; esta opción se utiliza para que los usuarios ejecuten los CGI (que son programas ejecutables) con los permisos del propietario del CGI en lugar de los suyos propios. Sigue el consejo y selecciona “no”.

A continuación debemos seleccionar los módulos que Apache cargará cuando se inicie; por ejemplo en la imagen puedes ver seleccionado el módulo “mod_log_config, encargado de almacenar información de funcionamiento de Apache; más adelante veremos algunos más de ellos. Como Apache ya viene con una serie de módulos necesarios en la mayoría de los casos, pulsa el tabulador y “De acuerdo” (si sabes ya que vas a necesitar alguno que no está marcado puedes activarlo ahora).

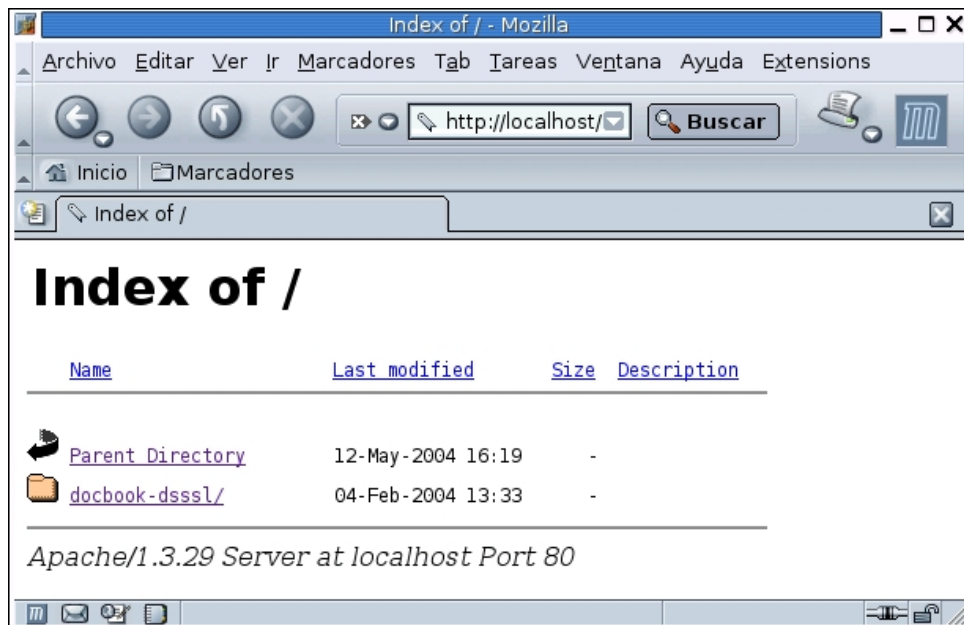


Debemos introducir ahora el nombre de nuestro dominio, si tenemos alguno; esto es lo que se conoce como FQND (Fully Qualified Domain Name, o nombre de Dominio Totalmente Cualificado); si no dispones de ninguno deja “localhost”.

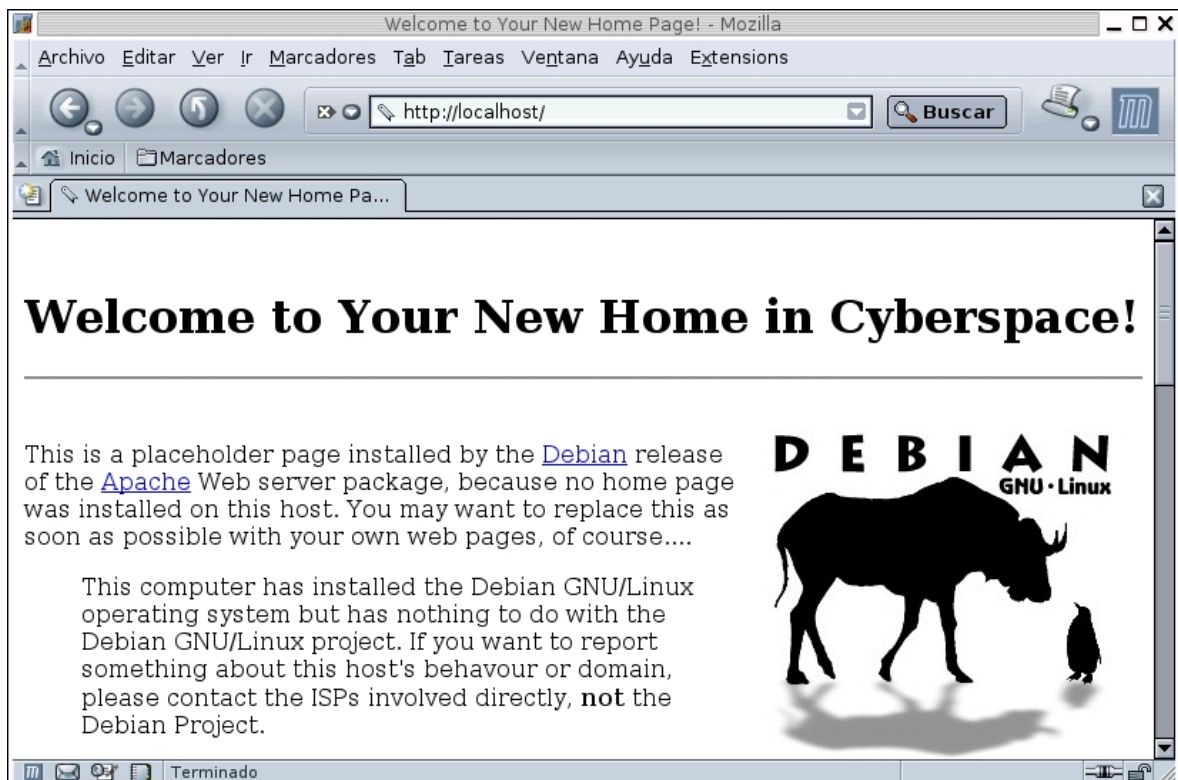
Lo siguiente que se necesita es la dirección del administrador de Apache, al que serán enviados los problemas en caso de haberlos; acto seguido hay que escribir la ruta local donde residirán las páginas web que Apache servirá; es buena idea dejar /var/www que es la ruta por defecto.

Por último nos pregunta el puerto en el que debe escuchar Apache;

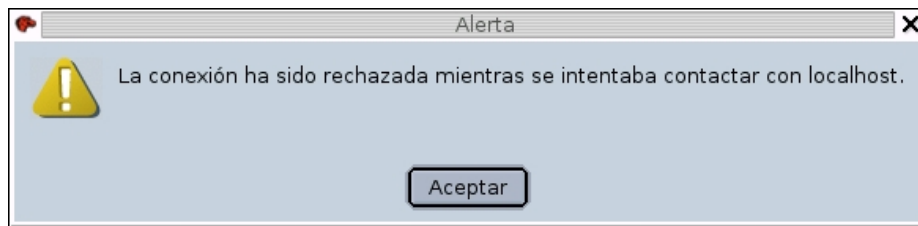
evidentemente debería ser el 80, a menos que tengas alguna buena razón para esconder tu servidor web (los navegadores siempre buscan en este puerto).



Ya puedes probar tu servidor apache; arranca tu navegador favorito y carga la dirección "localhost" (o tu nombre de dominio si lo tienes). Si utilizas Mozilla, deberías ver la imagen anterior.



Dependiendo de la versión de Apache podrías ver algo distinto, pero lo que nunca deberías ver es el mensaje de alerta; si es así, repite los pasos desde el principio, porque Apache se pone a funcionar en cuanto termina de instalarse.



Los archivos de configuración de Apache se encuentran en `/etc/apache`; hubo un tiempo en que se utilizaban tres archivos para configurar el servidor: `access.conf`, `httpd.conf` y `srm.conf`; el primero ofrecía el control de seguridad, el segundo controlaba el propio servidor, y el último de ellos seguía la pista de todos los elementos externos que necesitaba el servidor para cargar una página. Sin embargo las versiones más recientes centran toda la configuración en el archivo `httpd.conf` (los otros dos archivos siguen existiendo por compatibilidad con versiones anteriores, pero se recomienda dejarlos tal y como están).

Este archivo es muy extenso, pues dispone de muchas líneas de comentarios que sirven de descripción de las diferentes opciones. La primera de las directivas es `"ServerType standalone"`, que hace que funcione como demonio independiente; la otra opción es `"inetd"`, pero no es recomendable utilizar Apache así (además en la versión 1.3 está restringida esta opción a sistemas Unix).

A continuación se encuentran una serie de directivas para definir la localización de algunos archivos y directorios, que es mejor no modificar; a partir de `"Timeout 300"` es probable que sí quieras modificar alguna de ellas, por ejemplo `"StartServers 5"` que es el número de servidores que se crearán al iniciar Apache, o `"MaxClients 150"`, que especifica el número máximo de peticiones HTTP simultáneas que aceptara el servidor (cuanto mayor sea más memoria se necesitará).

La directiva `"Listen"` se utiliza para definir puertos o IPs de escucha adicionales a los utilizados por defecto; junto con `"BindAddress"` se utiliza para definir espacios virtuales; los veremos más adelante.

Acto seguido aparecen un par de opciones antes de llegar a la sección 2, que define la configuración principal del servidor. La primera de las directivas está bastante clara pues es `"Port 80"`, pero siempre debes recordar que esta directiva no se utiliza si se definen sitios virtuales (a menos que añadir una entrada `"Listen"` para el puerto 80). Las dos siguientes definen al usuario y el grupo bajo el cuál se ejecutará Apache (por lo general `"www-data"`). Tras ellas aparecen las directivas que contienen la dirección del administrador (el responsable del servidor, no del administrador del sistema) y el nombre del

servidor, así como el directorio donde se encuentran las páginas web (/var/www).

A continuación se configuran los parámetros para diferentes partes del sistema de archivos; la etiqueta "directory ruta" se utiliza para permitir o bloquear que una funcionalidad del servidor pueda acceder a dicha ruta del sistema de ficheros. Para toda la estructura del sistema (/) se establece no utilizar los enlaces simbólicos sólo si el destino es propiedad del que creó dicho enlace; es lo que hace "Options SymLinksIfOwnerMatch"; otras opciones son "FollowSymLinks", con la que se permite seguir los vínculos simbólicos, "ExecCGI", que permite que se ejecuten scripts CGI en esa ubicación, y "none" (ninguna opción activada).

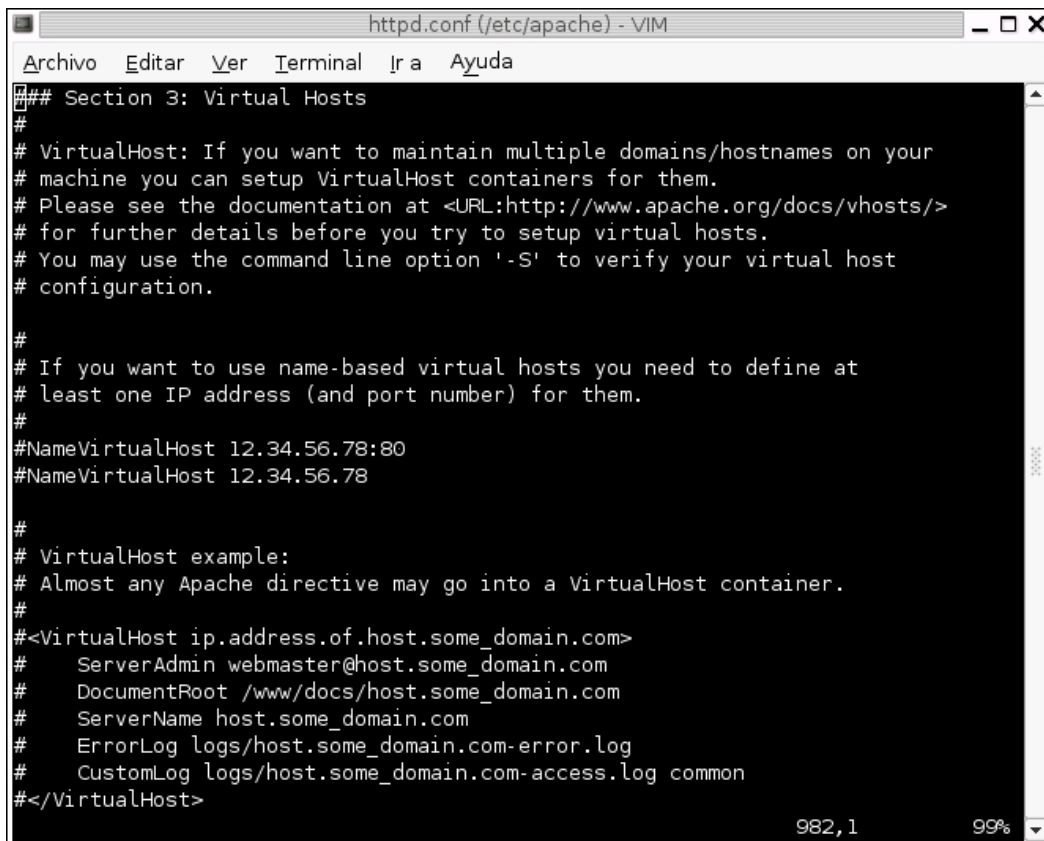
Por otro lado "AllowOverride" habilita las funcionalidades de autorización ("AuthConfig"), tipo de documento ("FileInfo"), indexación ("Indexes"), o la elegida en éste caso, ninguna ("none"), lo que es muy conveniente para el directorio raíz.

El siguiente directorio en aparecer es /var/www, que debe modificarse en el caso de haber definido otro en "DocumentRoot", que está configurado adecuadamente para la mayoría de los casos, ya que sigue la política de tener todos los sitios cerrados inicialmente, y decidir uno a uno los que tienen permiso; observa que las directivas de este tipo terminan en "</Directory>".

Así se definen varios directorios hasta llegar a la configuración del módulo "mod_dir.c", en el que se define cuál es la página inicial de los sitios web (DirectoryIndex), es decir, el nombre de la página principal que será cargado al acceder al sitio; el primero es "index.html", pero nada impide añadir delante "home.html" por ejemplo, o "principal.html". Si no existe la primera, se busca la siguiente en la lista hasta que se encuentra una; si no coincide ninguna se carga la página de error.

El resto son opciones variadas como "DefaultType text/plain", que le indica a Apache que si se encuentra con información de un tipo desconocido lo trate como texto, o los "LogFormat", que definen el nivel de detalle de los logs o información que se almacenará sobre el funcionamiento del servidor.

Avanza en el archivo y verás por ejemplo cómo se definen las extensiones de archivos y cómo asociarles un icono en Apache (directivas "AddIconByType", "AddIcon", y "AddDescription"); el resto son configuraciones que se realizan si se va a proceder a cargar el módulo ("IfModule").



```
httpd.conf (/etc/apache) - VIM
Archivo  Editar  Ver  Terminal  Ir a  Ayuda
### Section 3: Virtual Hosts
#
# VirtualHost: If you want to maintain multiple domains/hostnames on your
# machine you can setup VirtualHost containers for them.
# Please see the documentation at <URL:http://www.apache.org/docs/vhosts/>
# for further details before you try to setup virtual hosts.
# You may use the command line option '-S' to verify your virtual host
# configuration.
#
# If you want to use name-based virtual hosts you need to define at
# least one IP address (and port number) for them.
#
#NameVirtualHost 12.34.56.78:80
#NameVirtualHost 12.34.56.78
#
# VirtualHost example:
# Almost any Apache directive may go into a VirtualHost container.
#
#<VirtualHost ip.address.of.host.some_domain.com>
#   ServerAdmin webmaster@host.some_domain.com
#   DocumentRoot /www/docs/host.some_domain.com
#   ServerName host.some_domain.com
#   ErrorLog logs/host.some_domain.com-error.log
#   CustomLog logs/host.some_domain.com-access.log common
#</VirtualHost>
982,1 99%
```

La última parte del archivo es la que permite definir espacios virtuales (“Virtual Hosts”); hay que definir una IP (o IP:puerto) para cada sitio virtual, lo que se realiza con la directiva “NameVirtualHost”. Una vez configurado esto deberás introducir cierta información relativa a cada sitio virtual que definas mediante directivas “<VirtualHost>”.

Vamos a realizar una configuración típica: un segundo sitio web en el puerto 8080 (además del estándar del 80); para ellos introducimos lo siguiente:

```
NameVirtualHost 192.168.0.1:8080
```

```
<VirtualHost 192.168.0.1:8080>
    ServerAdmin webmaster@midominio.com
    DocumentRoot /var/www2
    ServerName midominio.com
</VirtualHost>
```

Dentro de “<VirtualHosts>” se pueden introducir cualquier directriz de las que hemos visto anteriormente; las que no se redefinen aquí toman el valor definido en la sección general (por ejemplo “DirectoryIndex” que podría haberse utilizado de nuevo).

No olvides definir la directiva “Listen” para el nuevo sitio virtual.

Recuerda que después de cualquier modificación en este archivos debes reiniciar el servidor.

Ejercicios

- 1) Crea un usuario llamado dominio; añade un sitio web virtual en su directorio bajo /home; este sitio deberá estar disponible en el puerto 8081 y la dirección de correo del responsable será dominio@undominio.com.
- 2) Copia un archivo html al directorio principal del sitio y haz que sea la página que se cargue al acceder al sitio web.

Soluciones

- 1) Ya sabes cómo crear un usuario; si no revisa el capítulo apropiado; ahora debes añadir casi al principio del archivo /etc/apache/httpd-conf:

Listen localhost:8081

Y al final del archivo:

NameVirtualHost localhost:8081

```
<VirtualHost localhost:8081>  
    ServerAdmin dominio@undominio.com  
    DocumentRoot /home/dominio  
</VirtualHost>
```

Si obtienes una ventana emergente con el mensaje “la conexión ha sido rechazada” es que no está bien configurado; si por el contrario obtienes una página “Forbidden” (prohibido) pero con la versión de Apache abajo junto con el puerto, entonces es correcto.

- 2) Puedes coger cualquier html; debes definir dentro de la directiva “VirtualHost” anterior “DirectoryIndex nombre_de_archivo.html”, o puedes renombrar el archivo a “index.html”.