





HyperText Transfer Protocol

Versión: 1.1.0

Alfredo Barrainkua Zallo

Marzo de 2008



Creative Commons – BY-SA-NC Lizentzia laburpena: <u>Euskaraz English Castellano</u>

Índice

1. Introducción	4
1.1. Hipertexto e Hipermedia	4
1.2. WWW (World Wide Web)	4
1.3. URL (Uniform Resource Locator)	4
1.4. HTML (HyperText Markup Language)	5
2. Funcionamiento del servicio HTTP.	6
2.1. Diálogo HTTP	6
2.2. Cabeceras	7
2.3. Códigos de estado	8
3. Trabajos previos a la instalación	9
3.1. Nombre del servidor	9
3.2. Dirección IP	9
3.3. Servidores DNS	10
3.4. La hora, la hora, la hora	10
3.5. Deshabilitar el reinicio por teclado	10
3.6. Acceso por la red : SSH	11
4. Instalación de Apache	12
4.1. Nuestra primera página web	14
4.2. Retoques iniciales	15
4.3. Permitir la sobreescritura de opciones	15
4.4. Icono de favoritos (Favicon)	16
4.5. Juego de caracteres	16
5. Módulos de Apache	17
5.1. mod_mime : Manejando documentos	17
5.2. mod_negotiation	17
5.3. mod_autoindex : Listando contenidos de directorio	18
5.4. mod_dir : Buscando el índice	21
5.5. mod_alias : Reubicando la información	21
5.6. mod_access : Limitando los accesos a directorios	22
5.7. mod_userdir : Páginas personales de usuarios del sistema	23
6. Autentificación	25
6.1. Basic	25
6.2. Active Directory (LDAP)	26
7. Servidores virtuales	28
8. Uso de SSL / TLS	
9. Apache y PHP	34
10. Herramientas de Estress	36
10.1. Webserver Stress Tool	36
10.2. OpenSTA	36
10.3. ab (Apache Benchmark)	36

Servicios de red con Linux

11. Referencias	37
12. Autor	

1. Introducción

Hay un antes y un después en Internet. La llegada de **HTTP** supuso el ocaso de servicios como **archie**, **verónica** y **gopher**. La **World Wide Web** trajo los contenidos multimedia y la búsqueda de información sin ningún tipo de conocimiento previo de uso de índices ni bases de datos. Simplemente, clicar en los enlaces.

1.1. Hipertexto e Hipermedia

La idea básica subyacente en el hipertexto es enlazar partes de un documento con otro documento. En la práctica, supone que una palabra o frase en un documento de hipertexto, nos llevan a otro documento de hipertexto cuando pulsamos sobre él. A esta palabra o frase se le denomina enlace (link). Cuando este concepto se extiende a otros medios, como fotos, sonido y vídeo, se convierte en hipermedia.

1.2. WWW (World Wide Web)

EL WWW extiende el uso de la hipermedia a Internet. Un click en una página de hipertexto puede enlazarnos con otra página de hipertexto alojada en un servidor al otro extremo del planeta. Es decir, nos enlaza a recursos remotos. Esta idea tan simple, ha generado el uso masivo de Internet.

La idea de la telaraña mundial que es la Worl Wide Web, proviene de la comunidad de investigación física, y en concreto de **Tim Berners-Lee**, del **CERN**.

Más tarde, cuando se crea **Mosaic**, en el **NCSA**, en 1992, es cuando empieza el **BOOM de la Web**. Mosaic es un navegador de Internet con capacidad para visualizar texto e imágenes, y mostrarlas de una forma elegante. Es capaz de acceder a recursos de múltiples fuentes, como hipertexto, servidores gopher, bases de datos de búsqueda, transferencia de ficheros y noticias. Realmente es el que ha marcado las directrices a los navegadores actuales.

1.3. URL (Uniform Resource Locator)

Con tantos protocolos, servicios, y clientes que acceden a ellos, pronto se crea la necesidad de definir el acceso de una forma homogénea. A la definición del recurso se le llama **URL** ó localizador de recursos homogéneo. URL es un caso especial del **URI** (**Universal Resource Identifier**). Aquí vemos las partes de que consta el URL:

http://www2.iurreta-institutua.net[:80]/sarea/ikastaroa/httpindex.html

Protocolo	Dirección del host	Puerto (opcional)	Lugar donde se encuentra el documento

Hay URLs para diferentes protocolos:

Diferentes tipos de URL		
http://	Acceso a documentos de hipertexto (Páginas web).	
gopher://	Servidores gopher.	
ftp://	Descarga de ficheros.	
file://	Descarga de ficheros.	
smb://	Acceso a recursos compartidos SMB.	
telnet://	Acceso e servicios de terminal.	
wais://	Acceso a servidor wais.	
news:	Acceso a los news. Se omite la parte host:puerto, pues el uso del servidor se define en la configuración.	
mailto:	Envío de correo. Se omite la parte host:puerto, pues el uso del servidor de correo se define en la configuración.	

1.4. HTML (HyperText Markup Language)

Los documentos HTTP (Páginas web) contienen información de enlaces de hipertexto y formateado de contenido que se describe utilizando el lenguaje de marcado de hipertexto o **HTML** (**HyperText Markup Language**). Es un lenguaje derivado de **SGML** (Standard Generalized Markup Language) y es más simple que éste. Las etiquetas de marcado (Tags) se definen entre el símbolo "<" y ">". El documento en sí es un documento de texto ASCII. Veamos el aspecto de un documento HTML.

<html>

2. Funcionamiento del servicio HTTP

El funcionamiento es muy simple. Un cliente se conecta al servidor en el puerto TCP/80. Solicita un documento, lo recibe y se desconecta. El usuario lee la información y piensa el siguiente paso. Nuevamente se realiza una conexión al mismo u otro servidor, y así sucesivamente.

Hay tres métodos que podemos utilizar con un servidor HTTP. El la tabla los tenemos.

Métodos utilizados por HTTP		
GET	Descarga un elemento. Un elemento puede ser un documento HTML, un fichero gráfico, etc.	
HEAD	Solicita ver las cabeceras de un elemento.	
POST	Enviar un elemento al servidor. Por ejemplo, los datos que se introducen en un formulario.	

2.1. Diálogo HTTP

El diálogo se realiza en modo texto (como casi todos los protocolos). Los formatos de tipos de datos y cabeceras son los estándares de correo electrónico y MIME. Aquí vemos un ejemplo de conexión. Vamos a solicitar el documento por defecto de la raíz del servidor. Utilizamos la versión 1.1 del protocolo, y aceptamos tipos texto/html. Después enviamos una nueva línea (pulsando ENTER).

```
root@zeus:/var/www# telnet localhost 80
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
GET /index.html HTTP/1.0
ACCEPT: text/html
HTTP/1.1 200 OK
Date: Sat, 23 Feb 2008 23:46:53 GMT
Server: Apache/2.2.4 (Ubuntu) PHP/5.2.3-1ubuntu6.3
Last-Modified: Sat, 23 Feb 2008 23:39:26 GMT
ETag: "90b5-fe-d9668780"
```

```
Accept-Ranges: bytes
Content-Length: 254
Connection: close
Content-Type: text/html
<html>
   <head>
        <title>Mi primer HTML</title>
   </head>
    <body>
        <h1>Kaixo mundua!</h1><br />
       <b>Kaixo mundua!</b><br />
        <u>Kaixo mundua!</u><br />
        <s>Kaixo mundua!</s><br />
        <center>----</center><br />
   </body>
</html>
Connection closed by foreign host.
root@zeus:/var/www#
```

La respuesta es una cabecera, una línea vacía y el documento solicitado. Finalmente, el servidor cierra la conexión.

2.2. Cabeceras

Las cabeceras aparecen al inicio de los mensajes intercambiados en la conexión HTTP. Las cabeceras generales pueden encontrarse tanto en las peticiones como en las respuestas. Hay cabeceras específicas de peticiones, específicas de respuestas, y cabeceras de entidad. Estas últimas contienen información sobre el elemento a que hacen referencia y pueden aparecer tanto en peticiones (POST) como en respuestas.

Cabeceras generales		
Date:	La fecha se encuentra en tiempo universal.	
MIME-version:	La versión de MIME de las cabeceras.	
Pragma:	Una directiva específica de la implementación. Por ejemplo: Pragma: no-cache. Indica al proxy que solicite nuevamente el elemento, aunque se encuentre en la caché.	

Cabeceras de petición		
Authorization:	Incluye la información del cliente para acceder a recursos protegidos.	
From:	Igual al campo "From:" del correo electrónico.	
If-Modified-Since:	Se usa para que el GET sea confidencial. Si el elemento no se ha modificado, la respuesta devuelve un código 304 sin	

Servicios de red con Linux

	cuerpo.
Referrer	La identidad del elemento de donde se obtuvo ese enlace.
User-Agent: Identifica el software del cliente.	

Cabeceras de respuesta		
Location:	La localización preferida del servidor para ese elemento.	
Server:	Identifica el software del servidor.	
WWW-Authenticate:	Son los datos que identifican el sistema de autenticación. Acuerda con el cliente que se autentifica, el mismo.	

Cabeceras de entidad		
Allow:	Lista de métodos que admite un recurso.	
Content-Encoding:	Si el cuerpo está codificado, indica el algoritmo.	
Content-Length:	Indica el tamaño del cuerpo que ha de transferir.	
Content-Type:	Tipos que recoge la IANA. Por ejemplo: text/html.	
Expires:	El elemento no será válido tras esa fecha.	
Last-Modified:	Última vez que se modificó el elemento.	

2.3. Códigos de estado

En cualquier respuesta, se envía un código. Hay una cantidad considerable de códigos. La asignación general de códigos se puede ver en la siguiente tabla.

Asignación general de códigos		
1xx	Informativo. No se utiliza en la actualidad.	
2xx	Correcto. La acción se ha recibido, interpretada y aceptada correctamente.	
3xx	Redirección. Se ha de realizar alguna acción adicional para completar la petición.	
4xx	Error del cliente. La petición tiene errores de sintaxis o no se puede concretar.	
5xx	Error del servidor. El servidor no ha podido realizar esa petición aparentemente válida.	

Con los códigos concretos, se envía información más detallada.

3. Trabajos previos a la instalación

Antes de instalar y configurar Apache, realizaremos una serie de configuraciones en el servidor, que si bien no son específicas de Apache, sí son imprescindibles para su correcto funcionamiento. Son detalles que hay que cuidar en un servidor, para este servicio o para cualquier otro.

3.1. Nombre del servidor

Necesitamos poder resolver el nombre de nuestro servidor independientemente de que tengamos un servidor DNS. Al inicio del sistema, puede suceder que necesitemos resolver el nombre de nuestra máquina, y no tengamos aún cargada la red o el servicio de interrogación DNS, o simplemente puede suceder que no podamos acceder al servidor DNS. Para ello, vamos a verificar que el nombre del servidor esté correctamente establecido en los ficheros **/etc/hostname** y **/etc/hosts.** He aquí el contenido del fichero **/etc/hostname**:

nireserver

El contenido del fichero /etc/hosts:

127.0.0.1 localhost 127.0.1.1 nireserver.nire-eskola.net nireserver # The following lines are desirable for IPv6 capable hosts ::1 ip6-localhost ip6-loopback fe00::0 ip6-localnet ff00::0 ip6-mcastprefix ff02::1 ip6-allnodes ff02::2 ip6-allrouters ff02::3 ip6-allhosts

3.2. Dirección IP

La IP deberá de ser fija. Por medio de **Escritorio, Configuración del sistema, Configuración de red, IP fija**, podremos hacerlo. Otra manera es en el fichero /etc/network/interfaces. En este fichero aparecen todos los interfaces de red. Deberá contener algo similar a esto:

```
auto eth0
iface eth0 inet static
address 10.22.1.12
netmask 255.255.255.0
gateway 10.22.1.1
```

3.3. Servidores DNS

Debemos configurar los servidores DNS a utilizar para la resolución de nombres. Se puede realizar esta configuración a través de **Escritorio, Configuración del sistema, Configuración de red, IP fija**. Otra forma es editar directamente el fichero /etc/resolv.conf. Deberá contener algo similar a :

```
search euskaltel.es
nameserver 212.55.8.132
```

3.4. La hora, la hora, la hora

Debemos asegurarnos de que la hora esté correctamente configurada. Para ello, configuraremos el cliente **NTP** (**Network Time Protocol**) para que el reloj del sistema se sincronice con un servidor NTP. Lo primero es instalar los paquetes necesarios:

aptitude install ntp ntpdate

Posteriormente, configuraremos el fichero **/etc/ntp.conf** con un servidor de tiempo. En el fichero debe aparecer una línea similar a esta:

server 0.pool.ntp.org
server 1.pool.ntp.org

Reiniciaremos el demonio.

/etc/init.d/ntp restart

En Redes de Área Local, lo más probable es que haya un servidor NTP, que sea el que se sincroniza con un reloj que haya en Internet. En este caso, los demás equipos de la red (incluidos los equipos que den algún tipo de servicio) se sincronizarán con este servidor. Así conseguimos que sea un solo equipo el que tenga que comunicarse con un servidor de Internet, reduciendo el tráfico entre la LAN e Internet.

3.5. Deshabilitar el reinicio por teclado

La distribución Debian, por defecto tiene activado el reinicio del sistema cuando detecta la combinación de teclado **CTRL+ALT+DEL**. Debemos desactivarlo. De este modo, se deberá introducir el comando oportuno para reiniciar el sistema.

El el fichero **/etc/inittab**, debemos comentar (poner "#" delante) la siguiente línea, que quedará así:

```
#ca:12345:ctrlaltdel:/sbin/shutdown -t1 -a -a now
```

3.6. Acceso por la red : SSH

En un servidor es común acceder al mismo por la red. Esto nos permite acceder al mismo desde lugares remotos, o aunque estemos en la misma localización, no tener que instalar periféricos para el mismo. el acceso se puede realizar tanto en modo consola o gráfico.

Las comunicaciones conviene realizarlas cifradas. De este modo, no se transmitirán por la red nuestros datos (entre otros, contraseñas de acceso) en testo plano. Los datos se cifran con algún método criptográfico. No vamos a entrar aquí en teoría ni sistemas de certificados, ni cosas de esas. Vamos a ver simplemente un acceso de clave compartida.

El sistema de acceso que vamos a utilizar el **ssh**. Hay que mencionar que **ssh** también permite la copia de ficheros entre máquinas, de forma cifrada con **scp**. Vamos a instalarlo.

aptitude install ssh

Vamos a configurar el paquete. El fichero de configuración es **/etc/ssh/sshd_config**. Vamos a poner las siguientes líneas de esta forma:

```
Protocol 2
PermitRootLogin no
```

Si querremos securizar más el sistema, podemos evitar el acceso con password y que se usen certificados, limitarlo a unas máquinas, etc.

Para acceder al sistema como el usuario elusuario:

ssh elusuario@lamaquina.eldominio.net

Para copiar el fichero **mifile** de **/home/mihome** al directorio home del usuario **elusuario** en el sistema 10.22.1.59:

scp /home/mihome/mifile elusuario@10.22.1.59:/home/elusuario/

4. Instalación de Apache

Vamos a instalar el servidor HTTP más utilizado de Internet. **Apache**. Concretamente, la versión 2.2.3-4 de **Apache**. Apache es un servidor muy flexible, con una arquitectura basada en módulos. Hay módulos de apache, para prácticamente cualquier necesidad. Desde soporte de lenguajes de scripting, hasta labores de autentificación.

```
aptitude install apache2 apache2-doc apache2-utils
```

La instalación de **apache2**, conlleva la instalación de numerosos módulos. Entre ellos el módulo de **php5**, **libapache2-mod-php5**. Concretamente, la versión 5.2.0-8. Veamos ahora, los directorios y ficheros más importantes para el funcionamiento de Apache.

Directorios y ficheros más importantes		
/etc/apache2/	Directorio general de configuración.	
/etc/apache2/mods-available/	Directorio con la relación de módulos disponibles.	
/etc/apache2/mods-enabled/	Directorio con la relación de módulos habilitados.	
/etc/apache2/sites-available/	Directorio con los sites disponibles.	
/etc/apache2/sites-enabled/	Directorio con los sites habilitados. Configuración de los hosts virtuales.	
/etc/apache2/apache2.conf	Fichero de configuración general de apache2.	
/etc/apache2/httpd.conf	Antiguo fichero de configuración de apache. Ahora se utiliza para las configuraciones del usuario.	
/etc/apache2/ports.conf	Puertos utilizados por apache2.	
/etc/apache2/envvars	Variables de entorno.	
/var/log/apache2/	Directorio de los ficheros de registro.	
/var/run/apache2.pid	Fichero con el PID del proceso.	
/usr/lib/apache2/modules/	Directorio con los módulos de apache2.	
/usr/lib/cgi-bin/	Directorio de los cgi-s. Programas de apache.	
/usr/bin/htpasswd	Programa para gestionar usuarios y passwords para ficheros de claves para la autentificación básica.	
/usr/share/apache2/error/	Directorio con los mensajes de error.	
/usr/share/apache2/icons/	Directorio con los iconos utilizados por apache2.	

/etc/init.d/apache2	Script de inicio / parada / reinicio / recarga del servidor.	
/var/www/	Raíz de los documentos (páginas web) del servidor.	

Fichero de configuración principal **/etc/apache2/apache2.conf**. Este fichero tiene innumerables directivas, como la definición de la raíz del servidor, el usuario y grupo con el que se ejecuta el servidor, modo en que se gestionan los mensajes de error, etc. Además incluye otros ficheros y directorios, con lo que no necesitamos modificarlo (generalmente), sino que añadiremos en nuestros ficheros, los cambios deseados para nuestros directorios. Echémosle un vistazo.

Algunas directivas del fichero /etc/apache2/apache2.conf		
ServerRoot	La raíz del servidor. Su configuración, mensajes de error, etc.	
PidFile	Lugar donde se encuentra el fichero que indica el PID.	
Timeout	Número de segundos antes de enviar un timeout.	
KeppAlive	Permite conexiones persistentes.	
MaxKeepAliveRequests	Número máximo de conexiones permitidas en el tiempo de una conexión persistente.	
KeepAliveTimeout	Número de segundos de espera al cliente para la próxima petición en la misma conexión persistente.	
User	Usuario con el que se ejecuta el servidor.	
Group	Grupo con el que se ejecuta el servidor.	
AccessFileName	Fichero que contiene directivas adicionales de configuración en cada directorio.	
DefaultType	Tipo MIME utilizado por defecto.	
HostnameLookups	Almacena el nombre del cliente que accede al servidor, en lugar de la dirección IP.	
ErrorLog	Lugar donde almacena los registros de error.	
Include	Incluye los ficheros mencionados. Las directivas que contiene esos ficheros se añaden a la configuración del servidor.	
ServerSignature	Incluye la versión del servidor y el nombre del host virtual en las páginas de error generadas por el servidor.	
ServerAdmin	Dirección de correo del administrador del servidor.	
DocumentRoot	Raíz de la jerarquía de directorios de contenidos del servidor.	
UserDir	Directorio de usuario para las páginas personales.	
Listen	Dirección y puerto en el que el servidor espera	

Servicios de red con Linux

	peticiones.
LanguagePriority	Prioridad de los idiomas durante la negociación de contenido.

Vamos a entrar ya en calor. Reiniciemos apache2

/etc/init.d/apache2 restart

Si todo ha ido bien, debemos de tener el servidor en marcha. Probémoslo poniendo lo siguiente en el navegador web:

http://localhost

It Works!. Si vemos la página, Bien! Nuestro servidor funciona!

4.1. Nuestra primera página web

Vamos a crear una página web, para poder ver si el servidor funciona como pretendemos.

```
<html>
<html>
<head>
<title>Mi primer HTML</title>
</head>
<body>
<h1>Kaixo Mundua!</h1><br />
<b>Kaixo Mundua!</b><br />
<u>Kaixo Mundua!</b><br />
<s>Kaixo Mundua!</s><br />
<center>-----</center><br />
</body>
</html>
```

Le vamos a poner como nombre **index.html**. Lo situaremos en **/var/www/index.html**. Seguidamente apuntamos el navegador como antes.

http://localhost

Lo vemos? No? Bien! La distribución Debian viene con una configuración que hace que las peticiones a la raíz del servidor sean redireccionadas al directorio apache2-default que se encuentra en la raíz del servidor. Vamos a deshabilitarlo. En el fichero /etc/apache2/sites-available/default, alrededor de la línea 17, comentamos la línea que comienza con **RedirectMatch**. Ha de quedar de esta manera:

```
# RedirectMatch ^/$ /apache2-default/
```

Tras realizar el cambio, le decimos a apache que recargue los ficheros de configuración.

/etc/init.d/apache2 reload

Ahora volvemos a apuntar el navegador a **http://localhost**. Aparece nuestra página? Si? Bien!

NOTA: Cada vez que realicemos un cambio en los ficheros de configuración de Apache, debemos de decirle que los vuelva a cargar para tenerlos en cuenta. Esto lo haremos con:

/etc/init.d/apache2 reload

4.2. Retoques iniciales

Vamos a decirle a apache que el administrador es **sare-admin@nire-eskola.net**. Para ello, vamos a utilizar el fichero **/etc/apache2/sites-available/default**. En el mismo, al principio, la línea siguiente queda como sigue:

ServerAdmin sare-admin@nire.eskola.net

Téngase en cuenta que podemos poner otra cosa en otro site. También le diremos, que en las páginas de errores, ponga la dirección de correo del administrador del site, para que los que encuentren algún error se puedan poner en contacto con él. Esto lo haremos poniendo ésta línea que está casi al final, de este modo:

ServerSignature EMail

Además le diremos que por defecto, no realice índices de los directorios en caso de que no encuentre ficheros de índice (index). En el mismo fichero, en el directorio raíz, lo indicaremos. Quedarán las líneas pertinentes de esta manera:

```
<Directory /var/www/>
Options -Indexes FollowSymLinks .....
```

Recargamos la configuración y lo probamos.

4.3. Permitir la sobreescritura de opciones

Algunas veces nos interesará que en un subdirectorio cambien las opciones establecidas. Para ello, pondremos una configuración similar a esta:

```
<Directory /var/www/informes >
Options Indexes
AllowOverride All
</Directory>
```

Y más tarde, en el directorio pertinente crearemos un fichero **.htaccess**, con este contenido:

Servicios de red con Linux

Options -Indexes

Esto hace que, aunque se hayan permitido los listados anteriormente, en este directorio es concreto, no se realicen. Se puede hilar un poco más fino, y permitir la sobreescritura de unas directivas, no todas. Lo veremos más tarde.

NOTA: El uso del fichero **.htaccess** sólo es recomendable como último recurso. Ralentiza el funcionamiento del servidor, pues ha de buscarlo en todos los directorios.

4.4. Icono de favoritos (Favicon)

Podemos crear un icono que aparezca al lado de la URL de nuestro servidor. Este pequeño icono, también se inserta en la lista de bookmarks (favoritos). Es un fichero gráfico en formato **ICO**. Su nombre es "**favicon.ico**". Lo proporcionamos para todo el servidor situándolo en la raíz de documentos del mismo. En nuestro caso será **/var/www**. También podemos crear diferentes iconos, para diferentes directorios. Sus medidas son de 16 x 16 pixels y 16 colores (4 bits). Si deseamos que una página, muestre otro icono, podemos poner lo siguiente en la cabecera:

<link rel="SHORTCUT ICON" href="http://www.nire-eskola.net/nirelekua/favicon.ico">

4.5. Juego de caracteres

Podemos definir el fjuego de caracteres a utilizar para poder ver correctamente los documentos y los índices de **mod_autoindex**. Esto se realiza con la directiva **AddDefaultCharset**. Conviene ponerla en los directorios de nuestros documentos. Por defecto es **ISO-8859-1**. Lo pondremos a **UTF-8** de la siguiente manera:

AddDefaultCharset UTF-8

5. Módulos de Apache

Apache tiene una estructura muy modular. Hace casi todo utilizando módulos. Algunos módulos están compilados en el servidor y otros se cargan al iniciar. En todas las distribuciones, se cargan numerosos módulos, pues su funcionalidad es bien apreciada en casi todos los entornos de servidor web. Otros módulos, los deberemos de hacer que los cargue. Los módulos se instalan en **/usr/lib/apache2/modules/**.

5.1. mod_mime : Manejando documentos

Este módulo proporciona a los clientes, metainformación sobre los documentos. También permite definir un manejador para un tipo determinado de documento para determinar el modo en que apache procesa los documentos. Las directivas de este módulo se encuentran en el fichero **/etc/apache/apache2.conf**, en las dos secciones correspondientes a éste módulo y delimitadas por **<IfModule mod_mime_.c>** y **</IfModule>**. Echémosle un vistazo.

Generalmente no será necesario modificar nada de la configuración de éste módulo.

NOTA: En **Ubuntu**, la configuración se realiza en el fichero **/etc/apache2/mods-enabled/mime.conf**.

5.2. mod_negotiation

Negocia con el cliente, el contenido a servir. La configuración se encuentra en **/etc/apache/apache2.conf**. La directiva más importante es **LanguagePriority**. Este es el aspecto por defecto:

LanguagePriority en ca cs da de el eo es et fr he hr it ja ko ltz nl nn no pl pt pt-BR ru sv zh-CN zh-TW

Lo podemos modificar para que priorice el español. Esto hará que por ejemplo, los mensajes de error nos los envíe en español en lugar de en inglés. Quedaría de esta manera:

LanguagePriority es en ca cs da de el eo et fr he hr it ja ko ltz nl nn no plptpt-BR ru sv zh-CN zh-TW

NOTA: Algunas veces no se negocia el lenguaje, o está fijado en el navegador, o puede

suceder que no existan los mensajes de error en otros idiomas.

NOTA: En **Ubuntu**, la configuración se realiza en el fichero **/etc/apache2/mods-enabled/negotiation.conf**.

5.3. mod_autoindex : Listando contenidos de directorio

Normalmente no nos interesa que el servidor liste los ficheros de un directorio, aunque algunas veces pueda ser útil. La opción **Indexes** nos permite habilitar o deshabilitar esta opción. Aquí vemos cómo habilitarlo en el directorio **/var/www/docs**. Para ello creamos el fichero **/etc/apache2/conf.d/docs** con el siguiente contenido:

```
<Directory /var/www/docs >
Options Indexes
AllowOverride All
</Directory>
```

Después creamos el directorio **/var/www/docs/** y tres ficheros en su interior, de esta forma:

```
mkdir /var/www/docs
touch /var/www/docs/file.a
touch /var/www/docs/file.b
touch /var/www/docs/file.c
```

Recargamos la configuración de apache y apuntamos el navegador a **http://localhost/docs**. Vemos los ficheros? Bien.

Ahora creamos un fichero **.htaccess** en el directorio **/var/www/docs/** con el siguiente contenido:

Options -Indexes

Volvemos a apuntar el navegador al mismo sitio. Los vemos? No? Bien! Adelante.

La configuración de **mod_autoindex** se encuentra en **/etc/apache2/apache2.conf**. En la sección correspondiente el módulo tenemos las definiciones de los iconos a utilizar con cada extensión de fichero. Puede ser interesante ampliarlo con los iconos de nuestras aplicaciones. Además, los iconos que trae Apache son algo anticuados. Podemos dar un aire más moderno y bonito a nuestro servidor, utilizando otros iconos. Los iconos originales de Apache son de 20 pixels de ancho y 22 pixels de alto. Nosotros utilizaremos los iconos del tema **Tango** de **freedesktop.org**. Utilizaremos los iconos de tamaño 22 x 22. Obtenemos los iconos de esta forma:

wget http://tango.freedesktop.org/releases/tango-icon-theme-0.8.1.tar.gz

Los descomprimimos.

```
tar zxf tango-icon-theme-0.8.1.tar.gz
```

Copiamos todos los iconos de **22x22/mimetypes** y algunos de **actions**, **places** y **devices** a **/usr/shatre/apache2/icons**. Lo haremos de este modo:

```
cp tango-icon-theme-0.8.1/22x22/mimetypes/*.png /usr/shatre/apache2/icons/
cp tango-icon-theme-0.8.1/22x22/actions/go-up.png /usr/shatre/apache2/icons/
cp tango-icon-theme-0.8.1/22x22/places/folder.png /usr/shatre/apache2/icons/
cp tango-icon-theme-0.8.1/22x22/devides/media-
optical.png /usr/shatre/apache2/icons/
```

Ahora modificamos el fichero de configuración, para que las siguientes líneas queden de este modo:

```
AddIconByEncoding (CMP,/icons/package-x-generic.png) x-compress x-gzip
AddIconByType (TXT, /icons/text-x-generic.png) text/*
AddIconByType (IMG, /icons/image-x-generic.png) image/*
AddIconByType (SND, /icons/audio-x-generic.png) audio/*
AddIconByType (VID, /icons/video-x-generic.png) video/*
AddIcon /icons/x-office-document.png .odt .sxw .doc
AddIcon /icons/x-office-document-template.png .ott .stw .dot
AddIcon /icons/x-office-drawing.png .odg .sxd
AddIcon /icons/x-office-drawing-template.png .otg .std
AddIcon /icons/x-office-presentation.png .odp .sxi .ppt .pps
AddIcon /icons/x-office-presentation-template.png .otp .sti .pot
AddIcon /icons/x-office-spreadsheet.png .ods .sxc .xls
AddIcon /icons/x-office-spreadsheet-template.png .ots .stc .xlt
AddIcon /icons/x-office-calendar.png .ics
AddIcon /icons/x-office-address-book.png .tab .vcf .ldif
AddIcon /icons/ooobase.png .odb
AddIcon /icons/ooomath.png .odm
AddIcon /icons/msaccess.png .mdb
AddIcon /icons/application-certificate.png .gpg .pgp
AddIcon /icons/font-x-generic.png .ttf
AddIcon /icons/media-optical.png .iso
AddIcon /icons/application-x-executable.png .bin .exe
AddIcon /icons/package-x-generic.png .Z .z .tgz .gz .zip
AddIcon /icons/text-html.png .html .shtml .htm .pdf
AddIcon /icons/text-x-generic.png .txt
```

```
AddIcon /icons/text-x-script.png .conf .sh .shar .csh .ksh .tcl
```

```
AddIcon /icons/go-up.png ..
AddIcon /icons/folder.png ^^DIRECTORY^^
```

Algunos iconos los tendremos aue obtener de otros sitios, como por ejemplo el de Access de MS.

También podemos obtener el icono que representa un documento PDF, desde los iconos de Scribus. Es este caso, pondremos las siguientes líneas de este modo:

```
AddIcon /icons/text-html.png .html .shtml .htm
AddIcon /icons/acrobat.png .pdf
```

Listo!. Nuestro servidor creará unos listados la mar de txulis.

Vamos a probarlo. Creamos ficheros con estas extensiones en nuestro directorio de documentos:

touch /var/www/docs/file.doc touch /var/www/docs/file.xls touch /var/www/docs/file.odt touch /var/www/docs/file.ods touch /var/www/docs/file.ods touch /var/www/docs/file.odg touch /var/www/docs/file.odg touch /var/www/docs/file.iso touch /var/www/docs/file.iso touch /var/www/docs/file.iso

Recargamos apache y apuntamos el navegador. Los vemos? No? Bien. Modificamos el fichero **.htaccess** y ponemos:

Options Indexes

Volvemos a apuntar con el navegador. Los vemos? Bien!

Si nuestros ficheros tienen caracteres acentuados o eñes, conviene utilizar un juengo de caracteres que no sea ISO-8859-1. Lo haremos poniendo la siguiente directiva en el fichero .htaccess o en el fichero de configuración /etc/apache2/conf.d/docs.

AddDefaultCharset UTF-8

NOTA: En **Ubuntu**, la configuración del módulo **mod_autoindex** se realiza en el fichero **/etc/apache2/mods-enabled/autoindex.conf**.

5.4. mod_dir : Buscando el índice

Cuando en el navegador ponemos el nombre de un directorio al final de la URL, este módulo realiza dos cosas. Primeramente pone la barra al final del nombre del directorio, y después busca la página que utilizará como índice, antes de realizar un índice automático de los contenidos del directorio, en caso de que fuera procedente. El fichero de configuración se encuentra en **/etc/apache/mods-enabled/dir.conf**. Esta es la directiva por defecto:

DirectoryIndex index.html index.cgi index.pl index.php index.xhtml

Por ejemplo, para decirle que busque primero el "**index.html**", después "**index.htm**", luego "**index.php**" y finalmente "**default.htm**" lo haríamos con la siguiente directiva:

DirectoryIndex index.html index.htm index.php default.htm

Luego recargamos la configuración.

/etc/init.d/apache2 reload

Vamos a probarlo. Primero modificamos el fichero **.htaccess** para que deje listar los contenidos. Quedará así:

Options Indexes

Ahora copiamos la página web que creamos al inicio, en el directorio **docs**, con el nombre **default.htm**. Este nombre para páginas de índice es común en los servidores web windows. Costumbres.

cp /var/www/index.html /var/www/docs/default.htm

Recargamos apache y apuntamos el navegador al directorio. Se ve el listado? Si? Bien.

Ahora modificamos la directiva **DirectoryIndex**, y al final ponemos **default.htm**. Recargamos apache y volvemos a apuntar con el navegador. Qué vemos? La pagina web? Bien!

NOTA: En **Ubuntu**, la configuración se realiza en el fichero **/etc/apache2/mods-enabled/dir.conf**.

5.5. mod_alias : Reubicando la información

Permite asociar partes del sistema de ficheros a otras. Veamos un ejemplo:

Alias /fotos/ "/var/tmp/images/"

Si recibimos una solicitud tal como http://localhost/fotos/photo1.jpg, nos devuelve el

fichero /var/tmp/images/photo1.jpg, en lugar del fichero /var/www/fotos/photo1.jpg.

Vamos a probarlo. Creamos el directorio **/var/tmp/images**, y dentro de él, una serie de ficheros. Lo haremos de ésta forma:

```
mkdir /var/tmp/images
touch /var/tmp/images/photo1.jpg
touch /var/tmp/images/photo2.jpg
touch /var/tmp/images/photo3.png
touch /var/tmp/images/photo4.jpg
touch /var/tmp/images/photo5.jpg
touch /var/tmp/images/photo6.png
touch /var/tmp/images/photo7.jpg
touch /var/tmp/images/photo8.jpg
```

Creamos un fichero de nombre **fotos** en **/etc/apache2/conf.d/**. Su contenido será el siguiente:

Alias /fotos/ "/var/tmp/images/"

Recargamos la configuración de Apache, y apuntamos el navegador a **http://localhost/fotos/**. Lo vemos? Si? Bien!

NOTA: Hay que tener cuidado con esta directiva, pues se puede salir de la jerarquía de ficheros del sitio web.

NOTA: En **Ubuntu**, la configuración se realiza en el fichero **/etc/apache2/mods-enabled/alias.conf**.

5.6. mod_access : Limitando los accesos a directorios

Muchas veces nos va a interesar limitar el acceso a un directorio, dependiendo de la dirección de origen de la petición. Es decir, la dirección IP del cliente. Vamos a ver cómo podemos hacerlo. Suponiendo que nuestro directorio es **/var/www/docs**, el fichero **/etc/apache2/conf.d/docs** tendrá el siguiente contenido:

```
<Directory /var/www/docs>
Options -Indexes
AllowOverride None
Order deny,allow
Deny from all
Allow from 192.168.201. 127.
</Directory>
```

De este modo, solamente se permite el acceso desde la red 192.168.201.0/24. Además hemos deshabilitado el listado del directorio y no se pueden sobreescribir estas opciones con un fichero **.htaccess**. Si deseamos que se acceda desde más sitios, los iremos listando uno detrás de otro, separados por espacios. Debemos de tener en cuenta que lo

que aquí se pone, se toma como strings que se hacen coincidir con las direcciones desde las que se accede. De este modo, si ponemos 192.168.16, se accederá desde las redes 192.168.16.0/24, 192.168.160.0/24, 192.168.161.0/24, ...

5.7. mod_userdir : Páginas personales de usuarios del sistema

Este módulo permite que los usuarios creen sus páginas web en su directorio home. Dentro de este directorio suele haber un subdirectorio **public_html**. Si no se especifica otra cosa, será utilizado para servir las páginas web, como raíz de ese usuario.

La forma como accedemos a las páginas es: **http://localhost/~usuario**, siendo usuario, el nombre del usuario del home.

Este módulo no está habilitado en la configuración inicial. Para habilitarlo ejecutamos lo siguiente:

a2enmod userdir

La configuración está en el fichero **/etc/apache2/mods-available/userdir.conf**. Si deseamos modificar algo que solamente atañe a un usuario (por ejemplo que no realice índices de sus directorio), lo podemos poner en el fichero

/home/usuario/public_html/.htaccess. En este caso, su contenido sería:

Options -Indexes

Sólo nos queda recargar la configuración.

/etc/init.d/apache2 reload

Vamos a probarlo. Vamos a crear tres usuarios en el sistema. Como el esqueleto de los directorios **/home** en Debian, no contiene el directorio **public_html**, vamos a crearlo primeramente.

mkdir /etc/skel/public_html

Ahora creamoxs los usuarios:

useradd -m -s /bin/bash -c 'User 1' user1 useradd -m -s /bin/bash -c 'User 2' user2 useradd -m -s /bin/bash -c 'User 3' user3

Les vamos a poner la misma contraseña que su nombre.

```
passwd user1
passwd user2
passwd user3
```

Ahora vamos a copiar la página web que hicimos al principio, en el directorio **public_html** de cada usuario.

cp /var/www/index.html /home/user1/public_html/ cp /var/www/index.html /home/user2/public_html/ cp /var/www/index.html /home/user3/public html/

Las vamos a modificar ligeramente, para poder distinguirlas. En el título, poner el nombre de usuario en cada una de ellas. Por ejemplo:

<title>Hau nire (User 1) orria da</title>

Ahora vamos a intentar acceder a ellas. Apuntar el navegador sucesivamente a las tres páginas:

http://localhost/~user1

http://localhost/~user2

http://localhost/~user3

Las vemos? SI? Bien!

6. Autentificación

En apache podemos utilizar diversos métodos de autentificación. Vamos a comenzar por el más sencillo y después veremos otros.

6.1. Basic

La autentificación básica viene habilitada por defecto en nuestra instalación de apache. Este método toma los usuarios y contraseñas de un fichero. Vamos a crearlo. Para ello utilizaremos el comando **htpasswd**. Crearemos tres usuarios, cuyo nombre será **user1**, **user2** y **user3**. Las claves serán las mismas. El fichero de contraseñas será /var/lib/apache2/user.pass. Vamos a ello. Primero hemos de crear el directorio /var/lib/apache2, que no existe en Debian, y vamos a darle la propiedad al usuario www-data.

```
mkdir /var/lib/apache2
chown www-data:www-data /var/lib/apache2
```

Ahora creamos el fichero de contraseñas.

```
htpasswd -cb /var/lib/apache2/user.pass user1 user1
htpasswd -b /var/lib/apache2/user.pass user2 user2
htpasswd -b /var/lib/apache2/user.pass user3 user3
```

La primera vez le pasamos el parámetro **-c** para que cree el fichero. El parámetro **-b** indica que lo haga en modo batch. En este modo, la clave se introduce por línea de comandos, en lugar de tener que teclearlo cuando nos lo pida.

Ahora vamos a añadir autentificación básica al ejemplo del directorio **docs**. El fichero **/etc/apache2/conf.d/docs** nos quedará de la siguiente manera:

```
<Directory /var/www/docs>
AllowOverride None
Order deny,allow
Deny from all
Allow from 192.168.201. 127.
AuthName "Docs"
AuthType Basic
```

```
AuthUserFile "/var/lib/apache2/user.pass"
Require valid-user
</Directory>
```

Le indicamos con ello que use la autentificación básica y la localización del fichero de contraseñas a utilizar. Además indicamos que conceda el acceso a todos los usuarios.

Podemos realizar otro ejemplo, con grupos de Usuarios. Podemos indicar que conceda el acceso a todos los usuarios del grupo jefes. Primero creamos el fichero de grupos /var/lib/apache2/groups, que tendrá éste aspecto.

```
managers: user1 user2
workers: user3 user4
```

Ahora ponemos de esta manera la configuración del directorio:

```
<Directory /var/www/docs>
AllowOverride None
Order deny,allow
Deny from all
Allow from 192.168.201. 127.
AuthName "Informes"
AuthType Basic
AuthUserFile "/var/lib/apache2/user.pass"
AuthGroupFile "/var/lib/apache2/groups"
Require group managers
</Directory>
```

Listo! Esto ya tiene mejor aspecto!

6.2. Active Directory (LDAP)

Para utilizar la autentificación contra **Active Directory** de Microsoft, debemos habilitar dos módulos. Lo haremos de esta forma:

```
a2enmod ldap
a2enmod authnz ldap
```

Vamos a ver ahora las directivas más importantes que soporta este módulo.

Directivas del módulo ldap				
AuthBasicProvider	Quién provee la autentificación.			
AuthzLDAPAuthoritative	Si el módulo LDAP tiene la última palabra respecto de la autorización.			
AuthLDAPUrl	Lugar desde el que se obtiene la información de los usuarios.			
	ldap	Protocolo a utilizar.		
	SERVER	Nombre DNS del servidor.		

Servicios de red con Linux

	nire-eskola.net	Dominio DNS del servidor.
	389	Puerto TCP del protocolo LDAP.
	usuarios	Unidad organizativa "usuarios".
	nire-eskola.net	Componentes de dominio del árbol LDAP.
	sAMAccountName	Atributo LDAP que contiene el nombre del usuario en el esquema de AD.
	sub	Profundidad de la búsqueda: Todos los niveles inferiores.
	objectClass=*	Filtro para el tipo de objeto: Todos.
AuthLDAPBindDN	Usuario que realiza la búsqueda en el directorio LDAP (Necesario en AD).	
	cn	Nombre común (Common Name).
	DAVBind	Nombre del usuario. Puede ser cualquiera que exista en el directorio.
	Users	Por defecto, todos los usuarios de AD se encuentran en este contenedor.
AuthLDAPBindPassword	Password del usuario que realiza la búsqueda. (DAVBind)	

Veamos ahora el ejemplo anterior, autentificando contra un Directorio Activo.

```
<Directory /var/www/informes>
```

AllowOverride None Order deny,allow Deny from all Allow from 192.168.201. 127.

AuthName "Informes" AuthType Basic AuthBasicProvider ldap AuthzLDAPAuthoritative off

AuthLDAPUrl "ldap://192.168.201.200:389/ou=usuarios,dc=nire-eskola,dc=net? sAMAccountName?sub?(objectClass=*)"

```
AuthLDAPBindDN "cn=ADBind,cn=Users,dc=nire-eskola,dc=net"
AuthLDAPBindPassword clave_usuario_ADBind
Require valid-user
</Directory>
```

Algunas veces hay cierto problemilla entre el LDAP de Linux y AD. Para solventarlo pondremos lo siguiente en **/etc/ldap/ldap.conf**:

REFERRALS off

Listo.

7. Servidores virtuales

Un servidor virtual por nombre es aquél en el que se referencia un nombre de host determinado, pero se redirecciona a un directorio de otro host. En este caso, el nombre de host referenciado es un alias del host real. Esto nos permite tener varios sites en un mismo host. Veamos cómo hacerlo utilizando dos sitios (sites) llamados **docs.nire-eskola.net** y **documentos.zure-eskola.net**.

Si tenemos servidor DNS, habremos de dar de alta el servidor virtual en el servidor a través de un registro **CNAME** en la zona directa. Aquí vemos cómo, en el ejemplo de **informes**:

docs IN CNAME nireserver.nire-eskola.net.

Si no tenemos un servidor DNS, podemos ponerlos en el fichero **/etc/hosts** de nuestra máquina, para realizar las pruebas. Pongamos lo siguiente:

127.0.2.1 docs.nire-eskola.net 127.0.3.1 documentos.zure-eskola.net

Al ser un sitio, conviene tener una estructura de ficheros diferenciada del resto. Tendremos los logs en un directorio, y los documentos en otro. También deberemos hacer que el propietario sea el usuario con el que se ejecuta apache. Lo haremos así:

```
mkdir -p /var/www/vhosts/docs/{logs,htdocs}
mkdir -p /var/www/vhosts/documentos/{logs,htdocs}
chown -R www-data /var/www/vhosts
```

Después crearemos los ficheros de configuración de los sitios. En este caso, /etc/apache2/sites-available/docs y /etc/apache2/sites-available/documentos. El aspecto del fichero docs que situamos en el directorio /etc/apache2/sites-available/ de apache quedará de este modo:

```
NameVirtualHost *:80
<VirtualHost *:80>
ServerName docs.nire-eskola.net
ServerAdmin sare-admin@nire-eskola.net
ServerSignature Email
DocumentRoot "/var/www/vhosts/docs/htdocs"
ErrorLog "/var/www/vhosts/docs/logs/error.log"
```

```
TransferLog "/var/www/vhosts/docs/logs/access.log"
      <Directory /var/www/vhosts/docs/htdocs>
            AllowOverride None
            Order deny, allow
            Deny from all
            Allow from 192.168.201. 127.
            AuthName "Docs"
            AuthType Basic
            AuthBasicProvider ldap
            AuthzLDAPAuthoritative off
            AuthLDAPUrl "ldap://192.168.201.200:389/ou=usuarios,dc=nire-
eskola,dc=net?sAMAccountName?sub?(objectClass=*)"
            AuthLDAPBindDN "cn=ADBind, cn=Users, dc=nire-eskola, dc=net"
            AuthLDAPBindPassword clave_usuario_ADBind
            Require valid-user
      </Directory>
</VirtualHost>
```

El del fichero **/etc/apache2/sites-available/documentos** lo haremos un poco más simple:

```
<VirtualHost *:80>
ServerName documentos.zure-eskola.net
ServerAdmin sare-admin@zure-eskola.net
ServerSignature Email
DocumentRoot "/var/www/vhosts/documentos/htdocs"
ErrorLog "/var/www/vhosts/documentos/logs/error.log"
TransferLog "/var/www/vhosts/documentos/logs/access.log"
<Directory /var/www/vhosts/documentos/htdocs>
AllowOverride None
Order deny,allow
Deny from all
Allow from 192.168.201. 127.
</Directory>
</VirtualHost>
```

Habilitamos los sites:

a2ensite informes a2ensite documentos

No debemos de olvidarnos de quitar el fichero **/etc/apache2/conf.d/docs**. Lo pondremos en **/etc/apache2/borradores/**, por si lo necesitamos:

mkdir /etc/apache2/borradores

mv /etc/apache2/conf.d/docs /etc/apache2/borradores/

Recargamos la configuración de Apache:

/etc/init.d/apache2 reload

Vamos a copiar el documento **index.html** creado anteriormente en /var/www/vhosts/docs/htdocs/ y /var/www/vhosts/documentos/htdocs. Además, vamos a modificarlo un poco para que sepamos que está accediendo al documento correcto. Pondremos algo así como "Kaixo mundu birtuala informes" y "Kaixo mundu birtuala documentos". Ahora podemos acceder a nuestros directorios con las URLs:

http://docs.nire-eskola.net

http://documentos.zure-eskola.net

Listo!

IMPORTANTE: Hemos modificado el lugar donde se crean los ficheros de registro. Ahora debemos de realizar un retoque en la rotación de logs. Vamos a modificar el fichero **/etc/logrotate.d/apache2**, y ponemos la primera línea así:

/var/log/apache2/*.log /var/www/vhosts/*/logs/*.log {

8. Uso de SSL / TLS

Cuando la información transferida es sensible o simplemente queremos evitar interferencias en la comunicación y asegurarnos de la identidad del servidor Web, es imprescindible encriptar la comunicación entre éste y el navegador web. Para ello utilizaremos **SSL** (**Secure Socket Layer**). **SSL** es compatible con la especificación **TLS** (**Transport Layer Security**) desarrollada por Netscape para el comercio electrónico.

Primero habremos de crear un certificado digital para el servidor. Podemos comprar estos certificados, firmados por una autoridad certificadora de prestigio, en la que tengamos confianza, o podemos crear nosotros la autoridad certificadora, y ahorrarnos la pasta. Si va a ser para uso interno de nuestra organización, podemos poner el certificado de nuestra **CA** (**Certification Authority**) en el navegador, y no tendremos más molestias. Si nos dedicamos al comercio electrónico, conviene tener un certificado de una autoridad de confianza para nuestros clientes potenciales.

Primero tenemos que activar el módulo SSL para apache. Como hemos hecho con otros módulos, lo haremos con éste:

a2enmod ssl

La conexión cifrada al servidor se realiza por el puerto TCP/443, con lo que deberemos de indicar a **Apache** que escuche en ese puerto. En el fichero **/etc/apache2/ports.conf** añadiremos lo siguiente:

```
<IfModule mod_ssl.c>
Listen 443
</IfModule>
```

NOTA: Ubuntu ya lo tiene incluido por lo que no es necesario añadirlo.

Ahora, veamos cómo crear nuestro certificado digital. Para ello, necesitamos el software criptográfico que nos permitirá crear los certificados. El paquete que vamos a utilizar es **openssl**. Vamos a instalarlo.

aptitude install openssl

El fichero de configuración se ha instalado en **/etc/ssl/openssl.cnf**. Vamos a crear un directorio para los certificados, y seguidamente generamos nuestro certificado.

mkdir /etc/apache2/certs

openssl req -new -x509 -days 1095 -nodes -keyout /etc/apache2/certs/docs.key -out /etc/apache2/certs/docs.crt

Seguidamente generará una clave, y nos pedirá que introduzcamos los datos que va a contener el certificado. Lo haremos de este modo:

```
Country Name (2 letter code) [AU]: EH
State or Province Name (full name) []: Bizkaia
Locality Name (eg, city) []: Iurreta
Organization Name (eg, company) []: Nire GLHB Eskola
Organizational Unit Name (eg, section) []: Sarea
Common Name (eg, YOUR name) [Sare Administraria]: docs.nire-eskola.net
Email Address []: sare-admin@nire-eskola.net
```

Debemos de proteger nuestra clave de la curiosidad ajena.

chmod 400 /etc/apache2/certs/docs.key
chown www-data /etc/apache2/certs/docs.key

Vamos a configurar el sitio **docs.nire-eskola.net** para que utilice solamente **SSL**. El fichero **/etc/apache2/sites-available/docs** quedará de esta forma:

```
<VirtualHost docs.nire-eskola.net:443>
      SSLEngine On
      ServerName docs.nire-eskola.net
      ServerAdmin sare-admin@nire-eskola.net
      ServerSignature Email
      DocumentRoot "/var/www/vhosts/docs/htdocs"
      ErrorLog /var/www/vhosts/docs/logs/error.log
      TransferLog /var/www/vhosts/docs/logs/access.log
      CustomLog /var/www/vhosts/docs/logs/ssl request.log ssl combined
      SSLCipherSuite \
            ALL: ! ADH: ! EXPORT56: RC4+RSA: +HIGH: +MEDIUM: +LOW: +SSLv2: +EXP: +eNULL
      SSLCertificateFile /etc/apache2/certs/docs.crt
      SSLCertificateKeyFile /etc/apache2/certs/docs.key
      SetEnvIf User-Agent ".*MSIE*." \
            nokeepalive ssl-unclean-shutdown \
            downgrade-1.0 forcersponse-1.0
      <Directory /var/www/vhosts/docs/htdocs>
            AllowOverride None
            Order deny, allow
            Deny from all
            Allow from 192.168.201. 127.
```

```
AuthName "Docs"
AuthType Basic
AuthBasicProvider ldap
AuthZLDAPAuthoritative off
AuthLDAPUrl "ldap://192.168.201.200:389/ou=usuarios,dc=nire-
eskola,dc=net?sAMAccountName?sub?(objectClass=*)"
AuthLDAPBindDN "cn=ADBind,cn=Users,dc=nire-eskola,dc=net"
AuthLDAPBindPassword clave_usuario_ADBind
Require valid-user
</Directory>
</VirtualHost>
```

Reiniciamos Apache:

/etc/init.d/apache2 reload

Ahora solamente nos queda probarlo. Apuntamos nuestro navegador a las direcciónes:

https://docs.nire-eskola.net

http://documentos.zure-eskola.net

Los vemos? Si? Bien!

9. Apache y PHP

El lenguaje de scripting más utilizado en Internet para realizar páginas dinámicas es sin duda **PHP** (**Personal HyperText Preprocessor**). La versión actual es la 5.0, y a ella nos atenemos. La distribución Debian no instala el módulo **mod_php** junto con el servidor **Apache**. Por ello, debemos de instalarlo explícitamente. Vamos a hacerlo.

aptitude install libapache2-mod-php5

Nos instalará el módulo y situará el fichero de configuración en **/etc/php5/apache2/php.ini**.

Hemos de realizar un par de retoques inicialmente. Muchas veces se utiliza **PHP** junto con la base de datos **MySQL**. MySQL puede escuchar conexiones en el puerto TCP/3306 o en un socket del dominio **UNIX**. En las configuraciones de DEBIAN y Ubuntu, MySQL sitúa este socket en **/var/run/mysqld/mysqld.sock**.

En la configuración de tiempo de compilación de PHP, se ha indicado que busque éste socket en **/tmp/mysql.sock**. El resultado es el esperado: Si nos conectamos a **localhost**, no se encuentra el susodicho socket. Para solventar este inconveniente, en el fichero de configuración **/etc/php5/apache2/php.ini**, donde pone:

mysql.default_socket =

pondremos:

mysql.default_socket = /var/run/mysqld/mysqld.sock

Asunto resuelto.

También podemos realizar otros ajustes. El límite de memoria que puede consumir una aplicación PHP, por ejemplo 32MB.

memory_limit = 32M

El tamaño máximo de un fichero que se suba al servidor, por ejemplo 8MB

upload_max_filesize = 8M

También nos interesará habilitar las dos extensiones más utilizadas por aplicaciones PHP. **MySQL** y **gd**. Las siguientes líneas deben quedar de esta manera (quitar el

carácter ";" que les precede):

extension=mysql.so
extension=gd.so

Para incrementar la seguridad, nos aseguraremos de que las siguientes variables aparecen de esta forma:

```
disable_functions = show_source, system, shell_exec, passthru, exec, popen,
proc_open, symlink
expose_php = Off
register_globals = Off
allow_url_fopen = Off
allow_url_include = Off
```

Recargar apache:

```
/etc/init.d/apache2 reload
```

Y para empezar, suficiente!!

10. Herramientas de Estress

Hay numerosas herramientas para medir la carga, someter a estress y hasta para someter al servidor al límite de su capacidad. Para páginas estáticas, páginas dinámicas, para aplicaciones Java, etc.

10.1. Webserver Stress Tool

Comercial. Le ponen muy bien. La edición Trial es igual a la Enterprise, pero limitada a 10 usuarios. Se puede descargar y comprar en el sitio web del fabricante:

https://www.paessler.com/download

10.2. OpenSTA

GPL. Le ponen muy bien. Infinidad de características, incluso test de accesos distribuidos. Muchas características siempre significan complejidad. Sitio:

http://www.opensta.org/

10.3. ab (Apache Benchmark)

Incluida en el paquete **apache2-utils**. Es la herramienta de la fundación apache. La hemos instalado en nuestras pruebas. Vamos a utilizarla. Vamos a decirle que abra 20 conexiones, y abrase a peticiones al servidor 192.168.201.48 durante 60 segundos.

ab -kc 20 -t 60 http://informes.nire-eskola.net/ ab -kc 20 -t 60 https://documentos.zure-eskola.net/

Le podemos decir que exporte los resultados como tablas HTML con la opción **-w**. Para la ayuda usar la opción **-h**. Como siempre, **man** es nuestro aliado.

11. Referencias

Para más información, en DEBIAN, mirar en /usr/share/doc/apache2.2-common.

Libros:

La Biblia : Servidor Apache 2, Mohammed J. Kabir

Apache Práctico, Ken Coar y Rich Bowen

En la web:

El RFC2616 se encuentra en:

ftp://ftp.rfc-editor.org/in-notes/rfc2616.txt

Documentación en el sitio de la fundación Apache:

http://httpd.apache.org/docs/2.0/

12. Autor

Alfredo Barrainkua Zallo

Iurreta Institutuko Sare Administraria

alfred obz @iurreta-institutua.net