IES Velázquez

Servicio HTTP

Despliegue de Aplicaciones WEB

Contenido

1 ¿Qué es Apache?

Apache es el servidor web más utilizado en sistemas GNU/Linux y Unix. Los servidores web se usan para el alojamiento de páginas web solicitadas por el cliente a través de internet desde los navegadores de web, actualmente el servidor apache se encuentra en la versión 2.

Viene con las siguientes ventajas y beneficios proporcionados por el software de servidor apache:

- Estabilidad
- Trabaja sobre una amplia gama de plataformas.
- Extremadamente flexible.
- Varios sitios alojados en un solo servidor apache.
- El servidor web más utilizado en www.

2 Los protocolos HTTP y HTTPS

El protocolo utilizado para ver páginas web es el HTTP, o `hyper text tranfer protocol` (protocolo de transferencia de hipertexto). Utiliza el protocolo de transporte TCP, concretamente el puerto 80 (de forma predeterminada), aunque también puede configurarse un servidor web para que utilice un puerto diferente.

El protocolo http no es seguro (las transmisiones se efectúan siempre en texto plano), por lo que se usa también una modificación, el protocolo HTTPS, que consiste en utilizar por debajo el protocolo SSL, que aporta cifrado de datos. De esta manera, el protocolo HTTPS ofrece estas ventajas:

1. Privacidad en las transmisiones, al ser cifradas.
2. Garantía de identidad, ya que usa firmas digitales en los servidores web para garantizar que son quien dicen ser.

Esto hace posible aplicaciones tales como el comercio electrónico o la administración electrónica.

El protocolo HTTPS utiliza el puerto TCP 443.

2.1 Formato de los mensajes

2.1.1 Peticiones

En las transacciones HTTP, el cliente envía al servidor un mensaje (petición), que contiene una cabecera y, opcionalmente, algún dato. El encabezado contiene la línea de petición (`request`), cuyos elementos son: método (GET en la petición), URL relativa `(/index.html` en el ejemplo) y la versión del protocolo HTTP (1.1 en el ejemplo):

Ejemplo:

```
GET /index.html HTTP/1.1
Host: www.google.com
Accept-Language: en
User-Agent: IE 7
{Línea en blanco}
```

Los métodos más usados en las peticiones son GET y POST:

- GET: pide la representación del recurso especificado.
- POST: presenta los datos que se procesarán al recurso especificado (por ejemplo un formulario, `form`, de HTML). Los datos se incluyen en el cuerpo de la petición. esto puede dar lugar a la creación de un nuevo recurso, a la actualización de recursos existentes o a ambos (páginas web dinámicas, consultas a bases de datos...). Este es el método idóneo para programación (PHP, ASP..).

Después de la línea de petición vienen varias cabeceras (headers), que contienen parámetros de petición, después can dos puntos (:) y, por último, el valor de esa petición. Ejemplos de parámetros son:

- Cache-control: no-cache // desactiva el poder quedarse en cache.
- Connection: close // indica al servidor el fin de la petición
- Date: Mod, 29 Nov 2010 08:15:31 GMT+1 // Fecha y hora (en inglés)
- Expires: Mod, 29 Nov 2010 08:15:31 GMT // Fecha de caducidad de la conexión.
- Pragma: no-cache // Pragma indica directivas a servidores proxy y gateway, desactiva la opción de cache en este caso.
- Transer-Encoding: chunked // Indica la codificación de transferencia en modos seguros.
- Ungrade: HTTP/2.0, SHTTP/1.3, IRC/6.9 // Indica para qué versiones de protocolos está actualizado.
- Accept-Language: EN, ES // Indica las lenguas aceptadas, según normas ISO, en el caso EN es inglés y ES español.

En encabezado termina con una línea en blanco (literalmente, debe llevar solo un salto de línea, nada de espacios o tabulaciones); después de esta, se pueden añadir datos adicionales, a los que se les llama cuerpo.

En la cabecera se puede especificar su longitud (en bytes) con el parámetro Content-Lenght. Este cuerpo suele ser una cadena de caracteres que dependerá de las gramáticas y reglas de los lenguajes de programación (XML, ASP, PHP, AJAX, DHTML...).

2.1.2 Respuestas

La respuesta del servidor HTTP suele contener un encabezado y un cuerpo.

Ejemplo:

```
HTTP/1.1 200 OK
Date: Fri, 31 Dec 2008 22:22:22 GMT
Content-Lenght: 1221
Server: Apache/1.3.3.7 (Unix) (Red-Hat/Linux)
Last-Modified: Wed, 08 Jan 2003 22:22:22 GMT
Etag: "3f80f-1b4-3e4vc02b"
Content-Type: text/html; charset=UTF-8
<html>
```

```
      <body>(la página web)</body>
</html>
```

La líneaprincipal o línea de respuesta, contiene la versión de HTTP soportada por el servidor y un código y mensaje de "error" o estado.

Después (en la segunda y sucesivas líneas) aparecen los parámetros de respuesta (algunos coinciden con los de petición). En el ejemplo podemos ver:

- Content-Lenght: 1221 // longitud del cuerpo en bytes
- Server: Apache/1.3.3.7 (Unix) (Red-Hat/Linux) // tipo del servidor.
- `Last-Modified: Wed, 08 Jan 2003 22:22:22 GMT` // Fecha y hora de última modificación del archivo o recurso requerido.
- `Content-Type: text/html; charset=UTF-8` // Tipo de contenido aceptado y el código aceptado.

A continuación le sigue una línea en blanco como separador y, después el cuerpo, que contendrá el recurso solicitado (una página web HTML en el ejemplo, pero puede ser cualquier de texto plano, lenguaje de etiquetas o de programación).

Existe la posibilidad de que se las conexiones son muy lentas se visualice la página web parcialmente o que falten imágenes u otros objetos.

2.2 Códigos de estado

Existen cinco tipos de códigos de error:

- Informativos del 100 al 109
- Petición correcta del 200 al299
- Redirección a otra URL del 300 al 399
- Petición incorrecta o error en el cliente del 400 al 499
- Error en el servidor del 500 al 599

2.2.1 Informativos

Son los que el cliente web recibe como mera información de manera que no tiene que interpretar nada ni responder a nada, simplemente darse por enterado.

Código	Mensaje	Descripción
100	Continue	Indica que puede recibir la siguiente petición
101	SwitchingProtocols	Indica que existe otro código mejor que el que utiliza actualmente. Esta función se usa para hacer upgrades de protocolo.

2.2.2 Petición correcta.

Códigos que envía el servidor indicando al cliente la recepción y aceptación de la petición.

Código	Mensaje	Descripción
200	OK	Acepta la petición y devuelve el documento solicitado.
201	Created	El servidor crea el URI solicitado en la cabecera de la petición del cliente.
202	Accepted	La petición del cliente se ha aceptado, pero no ha sido procesada aun o no ha finalizado su proceso.
203	Non-AuthoritativeInformation	La información de la cabecera de la petición no la ha generado el servidor, ha sido copiada de otro servidor.
204	No Content	Petición completada. No será necesario enviar más información para que el cliente tenga el documento completo.
205	Reset Content	Hay que reiniciar el documento actual, es útil cuando hay que borrar un formulario.
206	Partial Content	El documento ha sido enviado parcialmente al cliente. Junto con el código se indica el segmento de datos enviado.

2.2.3 Redirección a otra URL.

Códigos que se envían al cliente para indicarle que la acción solicitada requiere más acciones para completarse.

Código	Mensaje	Descripción
300	MultipleChoises	Se hace una petición que en realidad son varios documentos. El servidor puede enviar información sobre cada uno de los documentos de manera que el cliente pueda seleccionar cuál de ellos es el que está buscando.
301	Moved Permanently	El cliente solicita un documento que ha cambiado temporalmente de dirección. A partir de la recepción de este código el cliente hará las redirecciones automáticamente en futuras peticiones.
303	SeeOther	El cliente pide un documento que ya no está en la URL solicitada. El cliente deberá utilizar el método GET para recuperarla.
304	NotModified	Se indica que el documento no ha sido modificado desde la última visita del cliente. Se trata de utilizar la copia que reside en la cache del cliente en lugar de devolver el documento.
305	Use Proxy	El cliente debe utilizar el servidor Proxy que aparece en la cabecera de la petición, el documento se devuelve a través del servidor Proxy.

2.2.4 Petición incompleta o errores en el cliente

Son códigos que indicaran al cliente que debe enviar más información para completar la petición.

Código	Mensaje	Descripción
400	BadRequest	Es un error de sintaxis en la cabecera de la petición por parte del cliente
401	Unauthorized	Es necesaria la autentificación del cliente. El servidor devuelve una cabecera para indicar la autentificación y el alcance de esta.

402	PaymentRequired	Uso. Futuro. Se usará para indicar que se debe hacer un pago anterior a la obtención del documento.
403	Forbidden	El cliente no tiene acceso a la fuente solicitada.
404	NotFound	Se solicita un documento que no es posible encontrar en el servidor.
405	MethodNotAllowed	El método usado para la petición no es valido
406	Not Aceptable	El documento solicitado no está en un formato que el cliente pueda reconocer.
407	Proxy AuthentificationRequired	El cliente no se ha autentificado en el servidor proxy.
408	Request Time-Out	La petición no ha podido ser atendida en su tiempo establecido, por lo que deberá repetir la operación.
409	Conflict	La petición de documento solicitada por el cliente entra en conflicto con otra de otro cliente.
410	Gone	El documento solicitado ha sido eliminado del servidor.
411	LengthRequired	El cliente debe suministrar una cabecera Content-type para completar la petición
412	PreconditionFailed	Es posible hacer peticiones de documentos que dependan de una o más condiciones. El servidor usara esta misma cabecera para devolver la condición generada indicando cuál de ellas es falsa.
413	RequestEntitytooLarge	La petición tiene un cuerpo con una longitud excesiva. El servidor cierra la conexión para impedir la entrada de la petición.
414	Request-URI toolong	El servidor no procesará la petición porque la URI es demasiado larga.
415	Unsupoorted Media Type	El servidor no procesará la petición porque la URI es demasiado larga.

2.2.5 Errores en el servidor

Son códigos que el servidor estando activo (de otro modo no sería capaz de devolver ni siquiera los códigos de error) devuelve para indicar la causa por la que no puede devolver un documento o no puede finalizar la petición.

Código	Mensaje	Descripción
500	Internal Server Error	Hay un error en la configuración del servidor o de un programa ligado a este.
501	NotImplemented	El servidor no dispone de recursos suficientes como para atender a la petición o para completarla.
502	Bad Gateway	Se ha producido un error en el servidor Proxy o se ha obtenido una respuesta no valida del servidor.
503	ServiceUnavailable	El servicio no está disponible.
504	Gateway time-out	No se ha podido establecer una conexión con la puerta de enlace.
505	http versión notsupported	La versión del protocolo http que utiliza el cliente no se admite.

2.3 Cookies

Las cookies o "galletas" son archivos que el navegador del cliente graba en el disco duro a petición del servidor. Estos archivos almacenan datos que normalmente utiliza el servidor en otras conexiones.

Se suelen utilizar para:

- Guardar los nombres de usuario y contraseñas (son útiles para "cestas de la compra", blogs y otras páginas que necesitan mantener datos entre sesiones).
- Recopilar información de virus, hábitos de navegación de los usuarios, spyware y usos publicitarios.

Las cookies solo se podían generar con CGI, pero ahora también se generan con JavaScript, etc. Según la legislación europea, es obligatorio pedir la instalación de plugins o tipos MIME para el navegador (como PDF, Flash, ActiveX y Silverlight).

Cuando el servidor envía una respuesta incluye el parámetro *Set-Cookie:name=value* y, a partir de ese momento, el navegador del cliente añade el parámetro *Cookie:name=value* a todas las peticiones de ese servidor (y de los servidores de elementos externos que contenga ese recurso).

En los navegadores las cookies se pueden activar, desactivar o preguntar cada vez que se vaya a enviar alguna.

2.4 Protocolo HTTPS (HyperText Transfer ProtocolSecure)

El Protocolo Seguro de transferencia de HiperText, `HTTPS` (HyperText Transfer ProtocolSecure), es el mismo protocolo `HTTP` pero que ofrece más seguridad, bien sea con Protocolo de capa de conexión segura, `SSL` (Secure Sockets Layer) o con Seguridad de la capa de Transporte, `TLS` (TransportLayer Security, el servicio TLS 1.0 equivale al SSL 3.0).

Utiliza el puerto 443 y se emplea para la transferencia de contraseñas, pagos con tarjeta, bancos, etc. Las URL de las páginas empiezan pro HTTPS:// y su especificación sobre TLS está en el RFC 2818. Este protocolo, además de los objetivos del SSL, evita el Eavesdropping (ataques de escucha).

Este servicio requiere la confianza de la Autoridad de Certificación que necesita instalar plugins en el navegador (VeriSign, MS Live, etc.) Una vez instalado, el navegador nos confirma que estamos en zona segura o insegura (Internet Explorer, Firefox y Google Chrome lo hacen con un candado abierto en la parte derecha de la dirección o en la parte inferior derecha de la barra de estado, y Netscape con una llave completa) .

3 Instalación y arranque de Apache en Linux

Para poder instalar apache solamente tenemos que ejecutar el siguiente comando:

```
#apt-get  install apache2
```

El servicio de apache 2 en Linux se llama igual que el paquete, es decir, apache2. Las opciones de ejecución de que dispone son estas:

```
# /etc/init.d/apache2 {start | stop | restart | reload | force-reload}
```

o

```
# service apache2 {start | stop | restart | reload | force-reload}
```

- `start`: Inicia el servicio de apache.
- `stop`: Detiene el servicio apache.
- `restart`: Reinicia el servicio de apache. Equivale a un `stop`, seguido de un `start`.

- `reload`: Recarga los últimos cambios registrados dentro del servicio de apache.
- `force-reload`: Fuerza a realizar una recarga de los últimos cambios realizados dentro del servicio de apache.

4 Ficheros de configuración

La ruta de configuración de los archivos del servicio de apache se localiza en:

```
/etc/apache2/
```

Dentro de este directorio encontraremos varios archivos y directorios:

- `apache2.conf`: dentro de este archivo se encuentra la configuración del servidor web apache. Incluye «includes» a los demás ficheros.
- `conf.d`: Directorios en donde se encuentran otros archivos de configuración de apache.
- `envvars`: este archivo de configuración contiene la información del usuario, grupo Y PD del servicio de apache.
- `httpd.conf`: Este era el archivo de configuración, ya no se usa, al menos en distribuciones basadas en Debian. Pero en cualquier caso, el fichero de configuración principal apache2.conf, tiene un include a este fichero, por lo que podría usarse.
- `port.conf`: Archivo de configuración donde se especifican los puertos de escucha. Por defecto usará el `80` para `HTTP` y el `443` para `HTTPS`, si está instalado.
- `mods-available`: Directorio donde se encuentran los módulos que tenemos disponibles para instalar en el servidor apache.
- `mods-enabled`: Directorio que contiene un enlace débil a cada módulo del directorio anterior que esté activado en apache. de esta manera se sabe si un módulo está activado en apache o no.
- `sites-available`: directorio en donde se encuentran los archivos de configuración de sitios o páginas web con http y otro para sitios web con `HTTPS`. Aquí se crearán los archivos de configuración de sitio necesarios.
- `sites-enabled`: este directorio contiene un enlace débil a cada sitio del directorio anterior que esté activado en apache. de esta manera se sabe si un sitio está activado en apache o no.

5 Gestión de módulos

Un módulo en Apache, es un «paquete de código» que cumple una determinada función. Se puede ver una lista de los módulos disponibles de Apache en la dirección `http://modules.apache.org`.

La razón principal de esta estructura modular en Apache, es que no toda instalación requiere de las mismas funcionalidades. Si fueran incluidas todas las funcionalidades posibles en una versión única de Apache, esto lo haría sumamente pesado en cuanto a requerimientos de memoria, procesador y espacio en disco duro.

Los módulos pueden ser así activados, o desactivados, según muestras necesidades. La gestión de módulos se realiza así:

- Ver los módulos que tenemos activados.
  ```
  # apache2ctl -l
  ```

- Para activar un módulo:

  ```
  # a2enmod modulo (apache2 enable module)
  ```

- Para desactivar un módulo

  ```
  # a2dismod modulo (apache2 disable module)
  ```

6 Fichero httpd.conf

Apache, se encuentra dentro del directorio `conf`, en el directorio de instalación del Apache. En primer lugar hay que destacar que el fichero está dividido en tres secciones, que son:

1º Parámetros globales

2º Directivas de Funcionamiento

3º Host Virtuales

En el fichero se encuentran todos los parámetros de funcionamiento del Apache. Algunos parámetros son generales para la instalación y funcionamiento del Apache. Muchos otros de los parámetros se pueden configurar independientes para un conjunto de directorios y/o ficheros. En estos casos los parámetros se encuentran ubicados dentro de secciones donde se indica el ámbito de aplicación del parámetro.

6.1 Secciones importantes

Las secciones más importantes son:

- **`<Directory>`:** Los parámetros que se encuentran dentro de esta sección, sólo se aplicarán en el directorio especificado y a sus subdirectorios.
- **`<DirectoryMatch>`:** Igual que Directory, pero acepta en el nombre del directorio expresiones regulares.
- **`<Files>`:** Los parámetros de configuración proporcionan control de acceso de los ficheros por su nombre.
- **`<FilesMatch>`:** Igual que Files, pero acepta expresiones regulares en el nombre del fichero.
- **`<Location>`:** Proporciona un control de acceso de los ficheros por medio de la URL
- **`<LocationMatch>`:** Igual que Location, pero acepta expresiones regulares en el nombre del fichero.

Algunas veces las directivas de funcionamiento de las secciones anteriores se pueden cruzar en cuyo caso tienen el siguiente orden de preferencia:

1. `<Directory>` y `.htaccess` (`.htaccess` prevalece frente a `<Directory>`)
2. `<DirectoryMatch>` y `<Directory>`
3. `<Files>` y `<FilesMatch>`
4. `<Location>` y `<LocationMatch>`

También hay que destacar, que el fichero contiene un montón de comentarios para su correcta utilización, las líneas comentadas aparecen con el símbolo #.

Todos los parámetros que se establecen dentro de esta sección son globales para el funcionamiento del servidor, por lo que no admiten estar dentro de ninguna directiva.

6.2 Parámetros

6.2.1 ServerRoot

Especifica la ubicación del directorio raíz donde se encuentra instalado el Apache, a partir del cual se crea el árbol de directorios comentado anteriormente. Esta directiva no debería cambiar a no ser que se mueva la carpeta de instalación de apache a otro directorio. Se encuentra disponible a través del módulo `Core`.

6.2.2 Listen

Esta directiva permite especificar qué puerto se utilizará para atender las peticiones. Por defecto se utiliza el puerto 80 (www), también permite especificar que direcciones IP atenderá, por defecto todas. Para atender dos direcciones IP distintas, con distintos puerto, se utilizaría:

```
Listen 192.168.255.5:80
Listen 192.168.255.8:8080
```

Se encuentra disponible a través de varios módulos `beos, leader, mpm_winnt, mpmt_os2, perchild, prefork, thread pool ó worker`.

6.2.3 LoadModule

Directiva que sirve para cargar módulos que incluyen distintas funcionalidades. La sintaxis es:

```
LoadModule nombreModulo ubicacionFichero
```

Se encuentra disponible a través del módulo `mod_so`.

6.2.4 ServerAdmin

Especifica la dirección de correo electrónico del administrador, esta dirección aparece en los mensajes de error, para permitir al usuario notificar un error al administrador. No puede estar dentro de ninguna sección. Se encuentra disponible a través del módulo `Core`.

6.2.5 ServerName

Especifica el nombre y el puerto que el servidor utiliza para identificarse, normalmente se determina automáticamente, pero es recomendable especificarlo explícitamente para que no haya problemas al iniciar el servidor. Si el servidor no tiene un nombre registrado en las DNS, se recomienda poner su número IP. No puede estar dentro de ninguna sección.

La sintaxis es:

```
ServerName direccionIP:Puerto
```

Por ejemplo: `ServerName localhost:80`

Se encuentra disponible a través del módulo `Core`.

6.2.6 DocumentRoot

La carpeta raíz que se ubica en el servidor, desde la que se servirán los documentos. Por defecto, todas las peticiones, tendrán como raíz esta carpeta, a no ser que se utilicen alias (directorios virtuales en IIS)

Por defecto, la carpeta raíz es la carpeta `htdocs`, que se encuentra en la carpeta de instalación del Apache. No puede estar dentro de ninguna sección.

Si se cambia este directorio por otro, es muy importante que se ponga el nuevo valor, no solo en esta línea, sino también en la sección `<Directory>` en la que se establecen los parámetros de configuración de este directorio.

Esta línea empieza por `<Directory>` seguido de la carpeta raíz que originalmente hay en `DocumentRoot`.

Se encuentra disponible a través del módulo `Core`.

6.2.7 DirectoryIndex

Especifica el fichero por defecto que buscará en cada directorio, en caso de que no se especifique ninguno. Por defecto es `index.html`. Es decir, que si por ejemplo se pone en el navegador: www.iesalixar.org el servidor por defecto servirá www.iesalixar.org/index.html

En esta directiva se pueden especificar más de un fichero, la sintaxis es la siguiente:

```
DirectoryIndex fichero1 fichero2 fichero3
```

El orden con el que se especifica el nombre de fichero determinará la prioridad a la hora de decidir que fichero es el que se muestra.

La directiva se puede encontrar fuera de cualquier sección, dentro de una sección o dentro de un fichero `.htaccess`.

Se encuentra disponible a través del módulo `mod_dir`.

6.2.8 AccessFileName

Es el nombre del fichero de configuración que se buscará en cada una de los directorios del servidor para conocer la configuración del mismo. Este fichero permite configurar el comportamiento de cada uno de los directorios individualmente. Para que esta configuración funcione, la directiva `AllowOverride` tiene que tener un valor que lo permita. No puede estar dentro de ninguna sección.

El nombre de fichero que se especifica por defecto es el del fichero `.htaccess`.

Como medida de seguridad, la configuración del Apache establece que no se muestre la existencia de este fichero a ningún usuario, aunque este establecida la opción de listado de directorios. Si se decide cambiar al nombre, habrá que redefinir la seguridad para que no se muestre el contenido del nuevo fichero. Esto se hace en el fichero `httpd.conf` en una sección File como la que se presenta a continuación en la que se establece que todos los ficheros que comiencen por `.ht` no se mostrarán.

```
<Files ~ "^\.ht">
  Order allow,deny
```

```
    Deny from all
</Files>
```

Se encuentra disponible a través del módulo `Core`.

6.2.9 TypesConfig

Especifica el nombre del fichero que contiene la lista de tipos mime que conoce el servidor, y que determinará dependiendo de las extensiones para generar las cabeceras http. No puede estar dentro de ninguna sección.

Se encuentra disponible a través del módulo `mod_mime`.

6.2.10 DefaultType

Tipo mime que se servirá por defecto en caso de no conocer la extensión del fichero que se está sirviendo. Por defecto, se indicará que se sirve texto plano, con el valor text/plain. La directiva se puede encontrar fuera de cualquier sección, dentro de una sección o dentro de un fichero `.htaccess`.

Sintaxis:

```
DefaultTypetipoMime
```

Se encuentra disponible a través del módulo `Core`.

6.2.11 HostnameLookups

Se utiliza en los ficheros de registro. Por defecto cuando se produce un acceso, se guarda simplemente su número IP, si esta directiva se encuentra en On, el servidor buscará la correspondencia de ese número IP con su nombre, y almacenará el nombre. Establecer esta configuración en ON provocará que por lo menos se tenga que hacer una petición al servidor de nombres por cada una de las peticiones de usuario, por lo que el rendimiento de la máquina se puede ver decrementado. Esta directiva se puede encontrar dentro de una sección o fuera de cualquier otra.

Se encuentra disponible a través del módulo `Core`.

6.2.12 ErrorLog

Especifica la ubicación del fichero que contiene el registro de errores, por defecto en la carpeta logs. Esta directiva sólo se puede encontrar fuera de cualquier sección.

Se encuentra disponible a través del módulo `Core`.

6.2.13 LogLevel

Especifica el tipo de mensajes que se guardaran en el fichero de registro de errores, dependiendo de los valores especificados, se guardarán más o menos. Esta directiva sólo se puede encontrar fuera de cualquier sección.

Valor de más a menos son: `debug`, `info`, `notice`, `warn`, `error`, `crit`, `alert`, `emerg`

Se encuentra disponible a través del módulo Core.

6.2.14 LogFormat

La directiva permite definir el formato que se utilizará para almacenar los registros. A cada formato se le puede asignar un nombre, utilizándolo luego para crear distintos tipos de ficheros de registro. Pueden existir varios `LogFormat` distintos.

Sintaxis:

```
LogFormat "configuraciónError" nombre
```

Esta directiva se encuentra fuera de cualquier sección.

Se encuentra disponible a través del módulo `mod_log_config`.

6.2.15 CustomLog

La directiva se utiliza para especificar la ubicación y el tipo de formato que se utilizará en un fichero de registro. Pueden existir varios ficheros de registro distintos con configuraciones distintas. Para hacer esto, simplemente hay que poner varias líneas `CustomLog`.

Sintaxis:

```
CustomLog fichero formato
```

Esta directiva se encuentra fuera de cualquier sección.

Se encuentra disponible a través del módulo `mod_log_config`.

6.2.16 ServerTokens

Esta directiva establece la información que se devuelve dentro de la cabecera http que envía el servidor. Posibles valores de menor a mayor información son:

- *Pord*
- *Min*
- *Os*
- *Full*

Esta directiva se encuentra fuera de cualquier sección.

Se encuentra disponible a través del módulo `Core`.

6.2.17 IndexOptions

Esta directiva controla la apariencia de la página que se mostrará a un usuario cuando se pide la lista de ficheros de un directorio.

Sintaxis:

```
IndexOptions [+|-]opcion [[+|-]opcion] ... (Apache 1.3.3 en adelante)
```

Entre las opciones que se pueden poner, destaca:

`FancyIndexing`: que muestra los nombres de los ficheros, con iconos etc..

Se encuentra disponible a través del módulo `mod_autoindex`.

6.2.18 FoldersFirst

Hace que primero se muestren los directorios. Esta opción sólo se puede establecer en el caso de que `FancyIndexing` este activa.

Esta directiva se puede encontrar dentro del fichero `.htaccess`, dentro de una sección `<Directory>` y fuera de cualquier otra.

Se encuentra disponible a través del módulo `mod_autoindex`.

7 Servidores virtuales

Podemos tener asociadas ilimitadas DNS para una sola IP. Los servidores que alquilan servidores virtuales se llaman host o servidor web, la acción de alquilar es a la que llamamos hosting. Algunos ISP ofrecen alquiler de DNS y hosting desde 20 euros. Si deseamos un servidor propio, se debe alquilar un servidor dedicado, aunque muchos ya lo alquilan sobre máquinas virtuales.

7.1 Nombre de encabezado de host

Para crear servidores virtuales necesitamos activar esa opción. En apache solo debemos añadir una línea al archivo de configuración (`http.conf`), con la IP del servidor real:

```
NameVirtualHost 192.168.1.100:80  // sitio virtual
```

7.2 Identificación de un servidor virtual

En Apache añadimos la etiqueta `<VirtualHost *>` para cada servidor virtual. Como mínimo debemos usar los parámetros `DocumentRoot` y `ServerName` para poder diferenciar a los servidores y redireccionarlos a distintos sitios, pero podemos particularizar cada uno de los parámetros del servidor predeterminado (si no los especificamos, heredan los del servidor real).

```
<VirtualHost 192.168.2.125>
      ServerAdmin  webmaster@aulaDAW2.es
      DocumentRoot  /var/www/html/aulaDAW2
      ServerName www.aulaDAW2.es
      ErrorLog  /var/log/httpd/aulaDAW2_error_log
</VirtualHost>
<VirtualHost 192.168.2.125>
      ServerAdmin  webmaster@aulaDAW1.es
      DocumentRoot  /var/www/html/aulaDAW1
      ServerName www.aulaDAW1.es
      ErrorLog  /var/log/httpd/aulaDAW1_error_log
</VirtualHost>
```

Al crear un sitio o servidor virtual hay que guardarlo en el directorio `/etc/apache2/sites-available`. La extensión de dicho archivo debe ser `.conf`.

7.3 Los Acceso anónimo y autentificado

El acceso HTTP suele ser por defecto anónimo, pero existe la posibilidad de forzar la introducción de un nombre de usuario y contraseña. Hay mucho código escrito en PHP, ASP, CGI, C#, Java, etc., para realizarlo mediante bases de datos, pero tenemos la posibilidad de configurarlo en el servidor.

7.3.1 Métodos de autenticación

Hay 2 métodos para la autentificación de usuarios locales a la hora de acceder a un directorio: dentro del archivo principal de configuración con la sección `<Directory>`o creando un archivo `.htaccess` en cada directorio que requiera la autenticación. En cualquier caso, se necesita la directiva o el parámetro de configuración `AllowOverride` y los módulos `mod_access` y `mod_auth`:

```
LoadModule  mod_access libexec/mod_access.so
LoadModule  mod_auth libexec/mod_auth.so
AllowOverride AuthConfig
```

Primero se debe crear un archivo de claves (no accesible desde Internet). Por ejemplo, si el sitio web es `/usr/local/apache/htdocs/inetpub`, se guardará las contraseñas en `/usr/local/apache/passwd`. Para crear una contraseña nueva se tendrá que ejecutar:

```
htpasswd -c /usr/local/apache/passwd/passwords  minerva
```

Esta opción reclamará introducir una contraseña dos veces para cada usuario (en el ejemplo `minerva`).

Imaginar que se desea proteger la carpeta `/usr/local/apache/htdocs/secreto`, entonces se introduce la siguiente sección en el archivo de configuración:

```
<Directory /usr/local/apache/htdocs/secreto>
      AuthType Basic
      AuthName "Accesodenegado"
      AuthUserFile /usr/local/apache/passwd/passwords
      Requireuser minerva
</Directory>
```

O bien crear un archivo `.htaccess` en la carpeta e se introduce las anteriores directivas pero sin meterlas en una sección `<Directory>`.

También se puede dejar acceder a un grupo de usuarios locales con las directivas.

```
AutGroupFile /usr/local/apache/passwd/groups
```

7.3.2 Restricciones de acceso a recursos

Para filtros genéricos de Apache, en donde en vez del nombre de usuario prime la DNS o IP de la que provienen los usuarios, se usa `Allow` (para permitir) y `Deny` (para denegar). Se puede utilizar el comodín `all` (para todos), partes de URL o TLD (`.net.es`) e inicio de redes (`192.168.0`) o rangos de IP (`192`). Con `Order` se da preferencia a denegar o a permitir.

```
Order deny, allow
```

```
Deny from all
Deny from hackers.org
Deny from 192.168.2.111
Allow from aulaDAW2.es
```

7.3.3 Establecimiento de conexiones seguras (HTTPS)

El protocolo HTTP no es seguro ya que permite que las claves viajen a través de la red a merced de los programas sniffer. Para evitarlo podemos configurar accesos `HTTPS` con el servicio `SSL` o `TLS`. Para ello debemos crear nuestra propia Autoridad de Certificación (AC) o comprar una de las que comercializan VeriSign o utilizar las AC estatales. Para crear AC se puede hacer en Linux con el paquete `openssl`:

```
/usr/lib/ssl/misc/CA.sh -newca
```

Preguntará el nombre de la CA (si existe, si no pulsar `Intro` para crearlo), la `Pass Phrase` que sería la frase para acceder a la clave privada, así como otro tipo de datos (`país, provincia..`). Creará el archivo llamado `cacer.per` para firmar el certificado en `/private/carey.pem` donde estará la clave de la AC. Ahora crear una clave `CSR`, clave para nuestro servidor `triple-DES`.

```
opensslfenrsa -des3 -out server.key 1024
opensslreq -new -key sercer.ker -out sercer.csr
```

Cuando pregunte por el `CommonName`, se tendrá que poner el DNS (en el ejemplo www.aulaDAW2.es). Ahora crear el CRT, firmando CSR como AC.

```
ln .s sercer.csrnewreq.pem
CA.sh -signreq
mvnewcert.pem server.crt
```

A lo largo de este proceso preguntará por la **passphase** del CA. Cuando pregunte si se quiere firmar contestar que **SI**, así como también a la pregunta de hacer **comit**. Mover el archivo `server.key` a la carpeta `ssl.key` de Apache y el archivo `Server.crt` a `ssl.crt`. En el archivo de configuración añadir.

```
<Directory /usr/local/apache2/htdocs/secreto>
        SSLRequireSSL  //fuerza el acceso https
        AuthType Basic
        AuthName "privatefiles"
        AuthUserFile /usr/local/apache2/conf/passwd_basic
        Requirevalid-user
</Directory>
```

Ahora arrancar el servidor en modo seguro (HTTPS):

```
/usr/local/apache2/bin/apachectl start ssl
```

El archivo de configuración debe llamar al módulo `ssl_module`.

8 Gestión de sitios

El fichero de configuración general tiene un nclude a todos los enlaces que haya en el directorio `sites-enabled`. esto significa que de entre todos los ficheros que haya en el directorio `sites-available`, aquellos que tengan un enlace simbólico en el directorio `sites-enabled`, serán incluidos en la configuración.

Por lo tanto, la gestión de sitios se resuelve de la siguiente manera:

1. Generando los ficheros de configuración de sitio que deseemos en el directorio `sites-available`. Para ello se recomienda copiar la plantilla en un fichero nuevo y modificarlo.
2. Activando y desactivando sitios.

   Un fichero de configuración de sitio se activa con la orden

   ```
   # a2ensite fichero
   ```

   Un fichero de configuración de sitio se activa con la orden

   ```
   # a2dissite fichero
   ```

3. Es conveniente utilizar un solo fichero de configuración de sitio a la vez.

9 La herramienta Apache2CTL

Aunque el `daemon`(demonio) de Apache es `apache2`, también se dispone del comando `apache2ctl`, que permite ejecutar herramientas de administración del demonio. De hecho internamente, cuando ejecutamos las opciones básicas de apache2, éstas se traducen internamente en comandos `apache2ctl`.

Las opciones básicas de que dispone esta herramienta son:

- `start`: Inicia el servicio de apache.
- `stop`: Detiene el servicio de apache.
- `restart`: Reinicia el servicio de apache.
- `fullstatus`: Muestra un reporte del estado completo de apache.
- `status`: Muestra un reporte del estado breve de apache.
- `graceful`: Reinicia "delicadamente" el servicio apache enviando una señal SIGUSR1. La diferencia con `restart` consiste en que a diferencia de éste, `graceful` no cancela las conexiones que tenga previamente abiertas. Se puede considerar como un "`restart` en caliente".
- `configtest`: Ejecuta una prueba de los archivos de configuración de apache. en caso de que la configuración de apache esté bien, devuelve Syntax OK. En caso contrario, indica cuál es el error.

Se puede encontrar más información usando el comando `man apache2ctl`.

10 Tipos MIME

MIME (extensiones de correo de internet multipropósito), es un estándar que clasifica recursos y provee información a programas acerca de cómo manejarlos. Esto permite a los navegadores abrir correctamente un archivo .txt como un recurso de texto plano y no como un archivo de video o algún otro tipo. Cuando un tipo MIME no es especificado para un recurso, el programa que lo procese puede "suponer" su tipo por la extensión del recurso (por ejemplo, un archivo .bmp supone contener una imagen de mapa de bits).

Se recomienda proveer información sobre tipos MIME en todos los lugares en que sea posible hacerlo (atributo type en tag HTML script y tag HTML link, atributo enctype para el tag HTML form y en el tag HTML meta cuando se define el content-type del documento). Esto puede ayudar a lograr una mayor compatibilidad con los navegadores y al correcto funcionamiento del documento en sí.

10.1 Algunos tipos MIME

Tipo MIME	Extensión
application/acad	.dwg
application/arj	.arj
application/base64	.mm
application/base64	.mme
application/excel	.xl
application/excel	.xla
application/excel	.xls
application/excel	.xlt

11 Habilitar directorio de usuario (userdir) en Apache

El servidor web Apache por defecto guarda los archivos de las páginas web en la ubicación `/var/www`. Como cada usuario puede llegar a tener sus páginas webs, lo que se puede hacer es habilitar un módulo que permite crear un directorio en el home correspondiente de cada usuario y alojar allí sus webs.

Para ello, primero creamos en el directorio de cada *usuario* un directorio de nombre `public_html`

```
$ mkdir public_html
```

Luego vamos a habilitar el módulo de Apache que lo habilita a alojar páginas web en ese directorio.

```
$ sudo a2enmod userdir
```

Reiniciamos el servicio

```
$ sudo service apache2 restart
```

Ya se puede tener las páginas webs en el directorio `public_html`. Para acceder, poner la URL `http://localhost/~usuario/` en el navegador.

12 Directorios protegidos con WEBMIN

A través de Webmin, elegir la opción Directorios Web Protegidos.

Seleccionar el directorio que se quiere proteger, en el ejemplo `dir_protegido`.

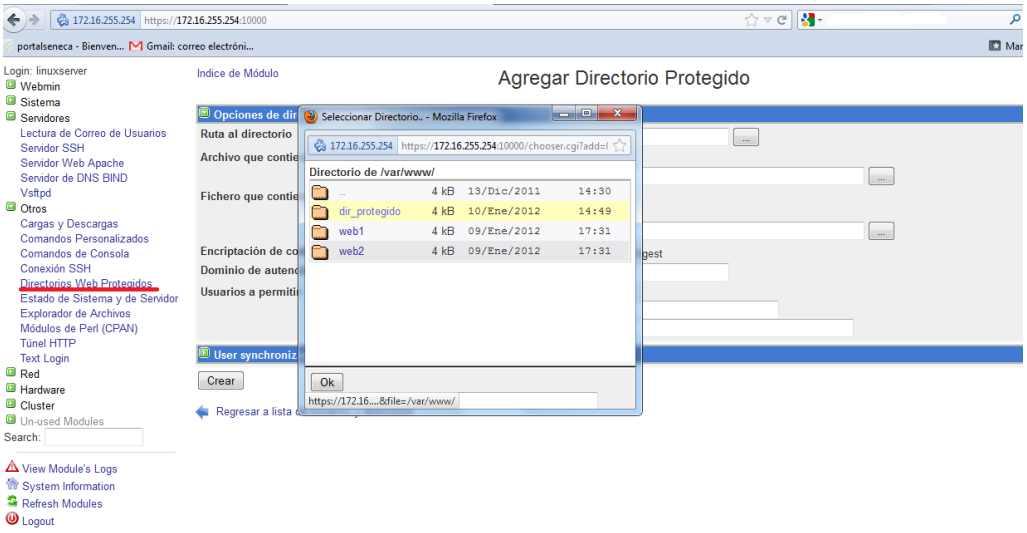

Ilustración 1 Agregar directorio protegido

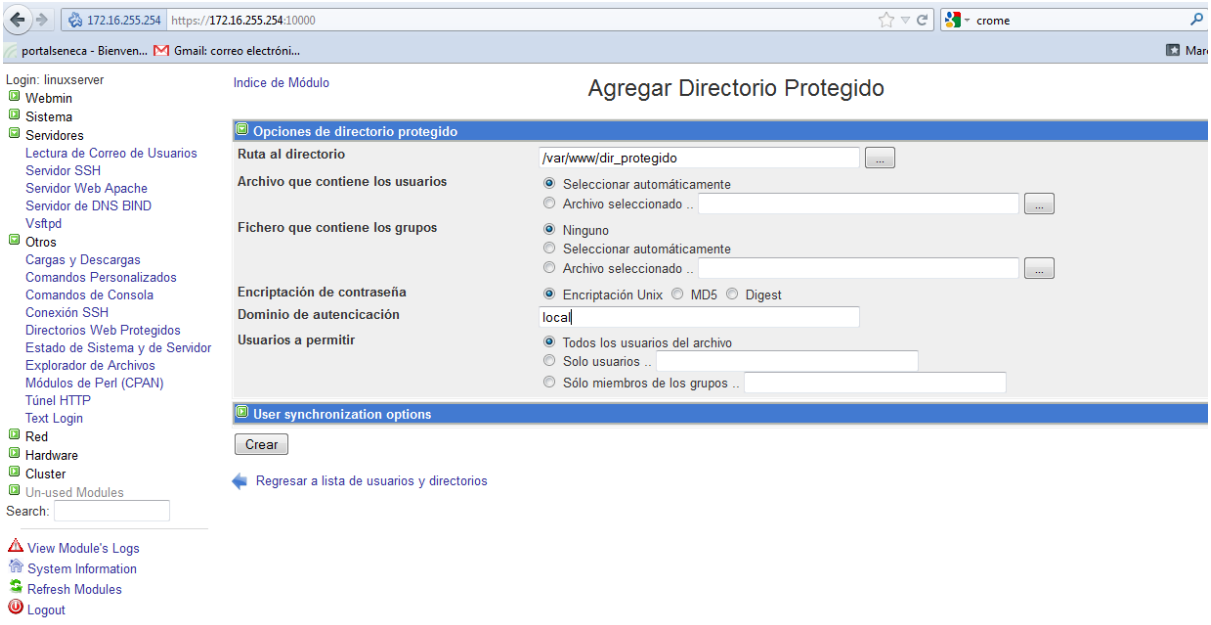

Ilustración 2 Directorio protegido agregado

Pulsar botón Crear.

Agregar los usuarios

Ilustración 3 Agregar usuarios al Dir. Protegido

Crear el usuario `usuario2` con contraseña `1234`.

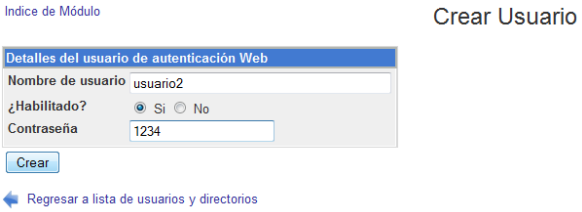

Ilustración 4 Crear usuario

Pulsar botón Crear.

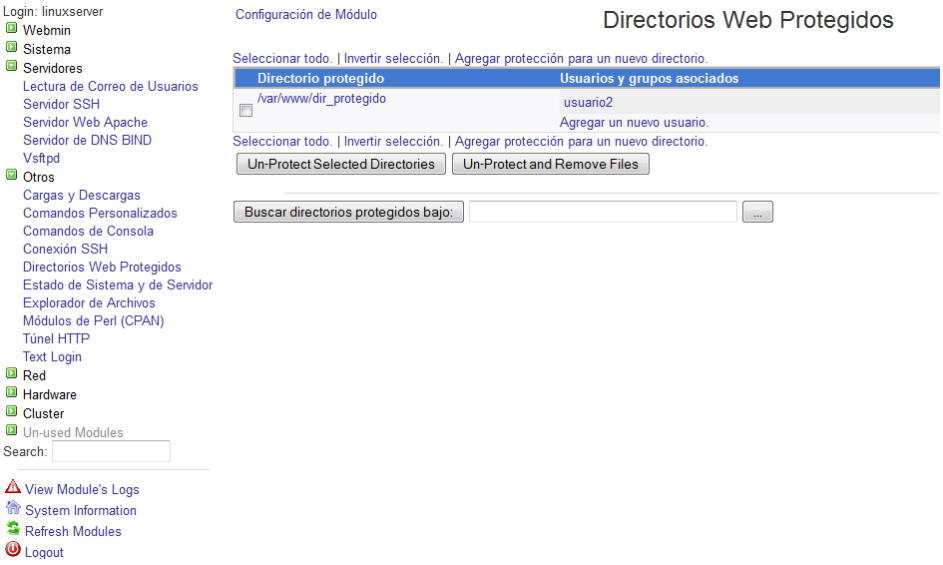

Ilustración 5 Usuario agregado

Al crear el usuario automáticamente se crean los archivos `.htaccess` y `.htpasswd`, en ellos están configurados los usuarios y contraseñas que tienen acceso al directorio. También se pueden crear con el comando.

Crear el fichero `.htaccess` con el editor de texto y poner

```
AuthUserFile "/var/www/dir_protegido/.htpasswd"
AuthType Basic
AuthName "localhost"
require valid-user
```

A través del comando `htpasswd` crear los usuarios y contraseñas que van a tener acceso al directorio protegido. Ejemplo:

```
root@ServidorSMR2:/var/www/paco# htpasswd -cm /var/www/paco/.htpasswd lino
New password:
Re-type new password:
Adding password for user lino
```

Ilustración 6 Directorio protegido "paco" se crea el usuario "lino"

Reiniciar el servicio y acceder con un navegador a `http://_IP_/dir_protegido`

Acceder al directorio nos he requerido el usuario y la contraseña.

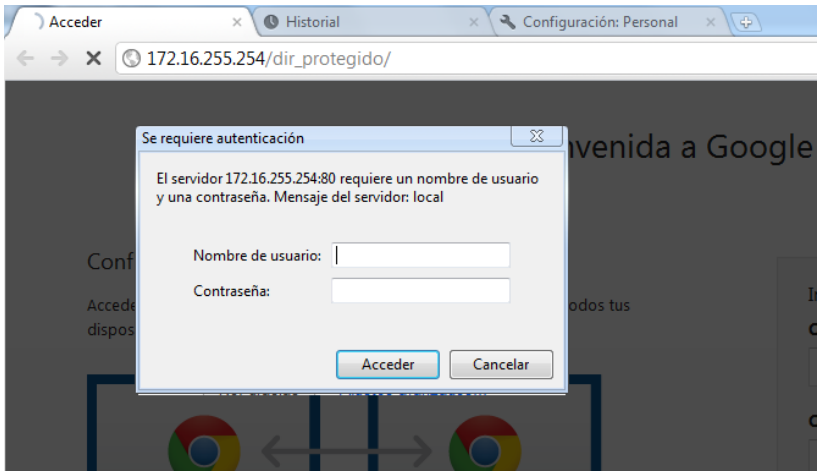

Ilustración 7 Acceso al directorio protegido

Y se accede

Ilustración 8 DIrectorio protegido

13 Segurida básica: .htaccess y .htpasswd (Directorios protegidos)

Los ficheros `.htaccess` permiten configurar determinados parámetros de `Apache` de manera que dicha configuración sólo afecte a un directorio y a sus subdirectorios .Observe que el punto inicial de los ficheros `.htaccess`y `.htpasswd` los convierte en ocultos. Para visualizar sus propiedades, sitúese en la carpeta de los ficheros y ejecute:

```
# ls -la
```

Los posibles parámetros que puede contener un fichero `.htaccess`vienen determinados por el parámetro `AllowOverride` dentro de la directiva `Directory`.Aunque los ficheros

`.htaccess` proporcionan una mayor flexibilidad a la hora de configurar Apache, tienen lossiguientes inconvenientes:

Pérdida de velocidad:Apache debe buscar en cada petición si existe el fichero `.htaccess` en el directorio del fichero o en sus directorios superiores. Seguridad: si los usuarios pueden crear sus propios ficheros `.htaccess`, se debe auditar qué modificaciones se permiten y su uso.

Seguridad: si los usuarios pueden crear sus propios ficheros `.htaccess`, se debe auditar qué modificaciones se permiten y su uso.

13.1 Restricción de acceso a carpetas

Para restringir el acceso al fichero `/var/www/dir_protegido/index.html` de manera que sólo puedan acceder los usuarios *usuario1* y *usuario2* con contraseñas respectivas `1234` y `5678`, edite el fichero `/etc/apache2/sites-available/default` (según la versión de `apache` puede que esté en `/etc/apache2/apache2.conf`) y sustituya las líneas:

```
<Directory /var/www/>
        Options Indexes FollowSymLinksMultiViews
        AllowOverride None
        Order allow, deny
        Allow from all
</Directory>
```

Por:

```
<Directory /var/www/>
        Options Indexes FollowSymLinksMultiViews
        AllowOverride AuthConfig
        Order allow, deny
        Allow from all
</Directory>
```

Para crear el fichero con los usuarios y las contraseñas, ejecute:

```
# htpasswd -c /etc/apache2/htpasswd secreto \usuario1
```

Para añadir nuevos usuarios, repita la orden anterior sin la opción–c.

Cree el fichero `/var/www/secreto/.htaccess` con el siguiente contenido:

```
AuthType Basic
AuthName "Necesita validarse para continuar"
AuthUserFile /etc/apache2/htpasswd secreto
```