Servidor Web Apache

El ***Servidor Web Apache*** es el servidor web más utilizado debido a su flexibilidad (*modular*), estabilidad y rendimiento.

Uno de los aspectos característicos del Servidor Apache es su ***modularidad***. Apache tiene un sinfín de características adicionales que si estuvieran siempre incluidas, harían de él un programa *demasiado grande y pesado*. En lugar de esto, ***Apache se compila de forma modular y se cargan en memoria sólo los módulos necesarios en cada caso.***

1. Instalación

1.1. Comprobaciones previas a la instalación: iptables

Antes de instalar el Servidor Apache es necesario comprobar si existe algún ***cortafuegos instalado*** en el Servidor que ***pueda estar bloqueando el tráfico*** a los puertos estándar del Servidor Web: el ***puerto 80*** y ***el puerto 443*** (*para conexiones seguras).*

Para comprobar las reglas configuradas del cortafuego iptables:

sudo iptables -L

1.2. Instalación de Apache:

sudo apt-get install **apache2**

```
gradosuperior@ubuntuServerLVM:~$ sudo apt-get install apache2
[sudo] password for gradosuperior:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes extras:
  apache2-mpm-worker apache2-utils apache2.2-bin apache2.2-common libapr1
  libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap
Paquetes sugeridos:
  apache2-doc apache2-suexec apache2-suexec-custom
Se instalarán los siguientes paquetes NUEVOS:
  apache2 apache2-mpm-worker apache2-utils apache2.2-bin apache2.2-common
  libapr1 libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap
0 actualizados, 9 se instalarán, 0 para eliminar y 72 no actualizados.
Necesito descargar 1.843 kB de archivos.
Se utilizarán 5.590 kB de espacio de disco adicional después de esta operación.
¿Desea continuar [S/n]? _
```

El comando anterior debería instalar apache junto con otras dependencias y paquetes de utilidades de configuración de apache, como por ejemplo el ***mpm-worker y el soporte SSL.***

Una vez instalado el paquete **apache**, tendremos el servidor web apache totalmente operativo.

1.3. Advertencia FQDN del Servidor

Al final de la instalación, y cuando *reiniciemos, paremos o iniciemos* el proceso Apache, es posible que aparezca la siguiente advertencia:

apache2: could not reliably determine the server's fully qualified domain name, using 127.0.01 for ServerName

```
Enabling module auth_basic.
Enabling module deflate.
Enabling module authz_default.
Enabling module authz_user.
Enabling module authz_groupfile.
Enabling module authn_file.
Enabling module authz_host.
Enabling module reqtimeout.
Configurando apache2-mpm-worker (2.2.22-1ubuntu1.4) ...
 * Starting web server apache2
apache2: Could not reliably determine the server's fully qualified domain name,
using 127.0.0.1 for ServerName
                                                                    [ OK ]
Configurando apache2 (2.2.22-1ubuntu1.4) ...
Procesando disparadores para libc-bin ...
ldconfig deferred processing now taking place
gradosuperior@ubuntuServerLVM:~$
```

Esta advertencia la indica el ***proceso "apache2"*** que es el *nombre del proceso de apache en ejecución*, señalando que desconoce el nombre completamente cualificado (FQDN) del Servidor, posteriormente habrá que ***utilizar la directiva ServerName en el fichero de configuración de apache (apache2.conf).***

1.4. Iniciar, Reiniciar o Detener el servidor Apache (*mediante la herramienta apache2ctl)*

Se puede utilizar la herramienta **apache2ctl** para iniciar, reiniciar o detener el servidor Apache de la misma forma que funciona el script ***/etc/init.d/apache2***.

Si se ejecuta el comando **apache2ctl** se obtiene la lista completa de argumentos que se pueden utilizar:

```
gradosuperior@ubuntuServerLVM:~$ sudo apache2ctl
Usage: /usr/sbin/apache2ctl start|stop|restart|graceful|graceful-stop|configtest
|status|fullstatus|help
       /usr/sbin/apache2ctl <apache2 args>
       /usr/sbin/apache2ctl -h           (for help on <apache2 args>)
```

Normalmente se utilizan los argumentos ***start, stop y restart*** para iniciar, parar y reiniciar el servidor apache. Pero existen **otros argumentos** disponibles:

- ***Argumento graceful:*** si se utiliza **apache2ctl** con el argumento **graceful**, se pide a apache que reinicie de tal forma que no interfiera con las conexiones existentes.

- ***Argumento graceful-stop***: es parecido a detener (***stop***) el servidor pero permitiendo que las conexiones existentes terminen su "trabajo".

En estos casos, el núcleo del servidor apache se reinicia o se detiene, dejando los procesos para que continúen manejando las conexiones antiguas con la configuración antigua. Es decir, "sobre el papel" habría que utilizar siempre graceful ya que puede reiniciar el servidor web sin tener que interrumpir las sesiones de los usuarios. Sin embargo, en la práctica graceful no siempre se ejecuta perfectamente: algunos módulos no funcionan correctamente con un reinicio graceful y pueden dejar "colgadas" algunas conexiones que no llegar a desaparecer.

- ***Argumento configtest***: simplemente comprueba si los archivos de configuración tienen errores de sintaxis. No interfiere con el servidor web en tiempo de ejecución. El **configtest** no garantiza si un cambio de configuración funcionará o no, simplemente detecta si hay algo mal escrito en el fichero de configuración.

- ***Argumento status o fullstatus:*** indica la situación general de lo que está haciendo el servidor web, aunque por lo general tarda un poco de tiempo, ya que el resultado obtenido es más complejo que indicar simplemente si el servidor web se está ejecutando o no. Para que se puedan utilizar estos argumentos hay que:
  - ***Tener habilitado y configurado el módulo mod_status***.
  - ***Tener instalado en el servidor un navegador web basado en texto como "lynx".***

Leer el siguiente artículo para ***habilitar el módulo mod_status en apache*** y saber interpretar correctamente la salida del comando:

http://solovidabien.blogspot.com/2018/02/como-habilitar-el-modstatus-de-apache.html

1.5. Iniciar, Reiniciar o Detener el servidor Apache (mediante init.d)

Además de los típicos ***start, restart, stop y status***, el script /etc/init.d/apache2 también dispone de las opciones ***graceful (reload) y graceful-stop***.

```
/* Iniciar el servidor Apache */

        sudo /etc/init.d/apache2 start

/* Reiniciar el servidor Apache */

        sudo   /etc/init.d/apache2   restart

/* Parar el servidor Apache */

        sudo /etc/init.d/apache2   stop
```

/* **Ver el estado** del servidor Apache: saber si ***está en ejecución o no*** */

sudo /etc/init.d/apache2 **status**

/* **Recargar** la configuración del servidor Apache: ***opción "reload" o "graceful"***

sudo /etc/init.d/apache2 **reload** sudo

/etc/init.d/apache2 **graceful**

/* **Parar** el servidor Apache permitiendo que las conexiones terminen su "trabajo" */

sudo /etc/init.d/apache2 **graceful-stop**

1.6. Apache LOGS

De forma predeterminada, los registros de apache se encuentran en el directorio:

/var/log/apache2

Para echar un vistazo hay que utilizar sudo. Los dos archivos principales de log son ***"access.log"*** y ***"error.log"***.

- ***"access.log"***: almacenará todos los *intentos de acceso que recibe apache*. Puede resultar útil *para análisis de tráfico* y para detectar *intentos de conexiones al servidor* que hayan sido *bloqueadas por iptables*.

- ***"error.log"***: almacena los *informes de error de apache*. Tanto errores de módulos del propio servidor como errores que se han enviado al usuario a partir de una petición del tipo *"File not found" (fichero no encontrado)*.

1.7. Probar el servidor web

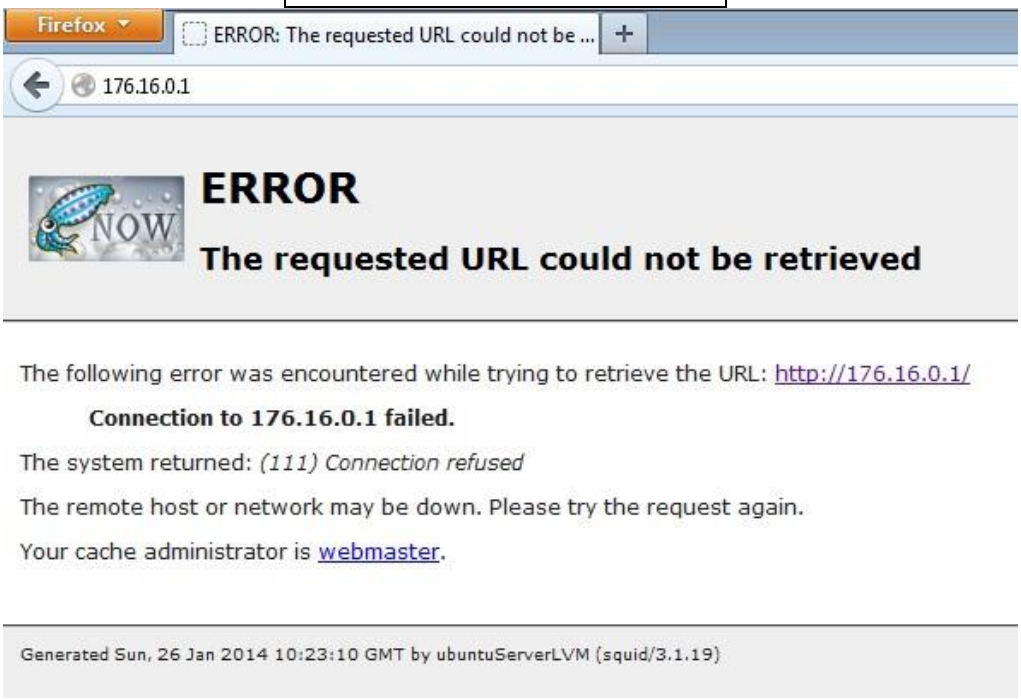
Para probar si el servidor web está funcionando o no, bastará con indicar la ***dirección IP del servidor en un navegador web*** de un cliente.

Si se ve una página predeterminada con el ***texto "It Works!"*** significa que el servidor está correctamente instalado. En caso contrario:

- Si el *navegador tardó un poco de tiempo* en responder el ***error de "Servidor no disponible",*** el acceso seguramente esté protegido mediante alguna ***regla del cortafuegos iptables*** (seguramente esté bloqueando el acceso al puerto 80).

- Si el *navegador de forma inmediata* ha respondido con un ***error "Conexión rechazada"*** (*Connection refused)* significa que apache no se está ejecutando. Arrancar el servidor mediante una de las siguientes instrucciones:

/etc/init.d/apache2 start apache2ctl start



- Si lo anterior no funciona, ***comprobar el registro de errores*** en el directorio log de apache. Los errores más comunes vienen por una ***mala configuración del archivo /etc/apache/apache2.conf***.

Recuerda que los clientes ***NO NAVEGAN CON DIRECCIONES IP***. Deberías configurar el ***servicio DNS*** para que también se resuelva el nombre **www.**

2. Ficheros/Carpetas de Configuración del Servidor Web Apache

Toda la configuración del Servidor Web Apache se encuentra en la ***carpeta /etc/apache2:***

2.1. Archivo apache2.conf

Dentro de esta carpeta, el **archivo principal de configuración de apache** es:

*/ etc/apache2/**apache2.conf***

Es un archivo bastante documentado de tal forma que ***indica los valores predeterminados y cómo se pueden cambiar***.

Las principales opciones a configurar de este archivo se verán a continuación, sin embargo existe la ***directiva include*** que merece tener en cuenta según se echa un vistazo al fichero de configuración.

2.1.1. La directiva "include"

Sería posible ***incluir todos los parámetros de configuración*** del servidor apache ***en un único fichero de configuración*** (de hecho antes se realizaba así), sin embargo ***resulta incómodo*** tener que manejar un fichero de configuración tan grande cada vez que haya que realizar algún cambio.

La ***directiva include*** indica a Apache que:

- Hay que *incluir otros ficheros* que también tienen *opciones de configuración*.
- También puede apuntar a un directorio, indicando que hay que incluir todos o algunos de los archivos de ese directorio.

De esta forma dentro del **fichero de configuración apache2.conf** se suelen encontrar las siguientes ***directivas include***:

Mediante rutas absolutas	Mediante rutas relativas
*Para incluir **"sentencias genéricas":***	
*Include /etc/apache2/**conf.d/***	*Include **conf.d/***
*Para incluir el **listado de puertos:***	
*Include /etc/apache2/**ports.conf***	*Include **ports.conf***

*Para incluir los **sitios webs habilitados (virtual hosts):***	
*Include /etc/apache2/**sites-enabled/***	*Include **sites-enabled/***
*Para incluir **módulos y sus configuraciones**:*	
*Include /etc/apache2/**mods-enabled/*.load*** *Include /etc/apache2/**mods-enabled/*.conf***	*Include **mods-enabled/*.load*** *Include **mods-enabled/*.conf***
*Para incluir **todas las configuraciones del usuario:***	
*Include /etc/apache2/**httpd.conf***	*Include **httpd.conf***

2.2. Directorio conf.d

- El directorio ***conf.d*** se debería utilizar para mantener "***declaraciones genéricas***".

- *Ejemplo*: la ***directiva ServerName*** se podría almacenar de forma directa en el fichero de configuración principal o se podría crear un nuevo archivo en el ***directorio conf.d*** para almacenar el valor de esta directiva.

- De los ficheros contenidos (por defecto) en el directorio **conf.d**, son interesantes:

    - ***Localized-error-pages:*** para establecer errores personalizados mediante la ***directiva ErrorDocument.***
    - ***Security:*** para establecer directivas relacionadas con la seguridad como ***ServerTokens y ServerSignature.***

```
gradosuperior@ubuntuServerLVM:/etc/apache2/conf.d$ ls -l
total 16
-rw-r--r-- 1 root root  269 feb  7  2012 charset
-rw-r--r-- 1 root root 3296 feb  7  2012 localized-error-pages
-rw-r--r-- 1 root root  143 feb  7  2012 other-vhosts-access-log
-rw-r--r-- 1 root root 1424 feb  7  2012 security
```

2.3. Fichero ports.conf

- El fichero ports.conf le indica a Apache el listado de puertos por los que debe "escuchar" peticiones HTTP. Por defecto está ***escuchando por el puerto 80*** (*puerto por defecto para el tráfico HTTP),* así como ***por el puerto 443 si el módulo SSL está instalado***.

- Habría que modificar los valores de este fichero si queremos que apache escuche peticiones por otros puertos o si queremos cambiar los valores por defecto (*cambiar el puerto 80 por el 8080, por ejemplo*).

- Contenido (por defecto) del fichero ***ports.conf:***

```
NameVirtualHost *:80
Listen 80

<IfModule mod_ssl.c>
    # If you add NameVirtualHost *:443 here, you will also have to change
    # the VirtualHost statement in /etc/apache2/sites-available/default-ssl
    # to <VirtualHost *:443>
    # Server Name Indication for SSL named virtual hosts is currently not
    # supported by MSIE on Windows XP.
    Listen 443
</IfModule>

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

2.4. Directorio sites-available

- El ***directorio sites-available*** contiene la configuración de los ***sitios web disponibles del servidor***. Por defecto suele contener dos archivos: "default" y "default-ssl":

    - **default**: contiene las *directivas que establecen el sitio web predeterminado*, indicando *la raíz de los documentos web* (***directiva DocumentRoot***) y algunos *permisos* para dicho directorio.

    - **default-ssl**: contiene las directivas que establecen el sitio web seguro predeterminado. Aparece *en caso de que esté instalado el módulo SSL*.

- Fragmento del contenido del ***fichero default (dentro del directorio sites-available):***

```
<VirtualHost *:80>
        ServerAdmin webmaster@localhost

        DocumentRoot /var/www
        <Directory />
                Options FollowSymLinks
                AllowOverride None
        </Directory>
        <Directory /var/www/>
                Options Indexes FollowSymLinks MultiViews
                AllowOverride None
                Order allow,deny
                allow from all
        </Directory>
```

Aunque posteriormente se verá cómo se pueden añadir dominios o sitios virtuales en el servidor, simplemente indicar que ***el archivo "default" es una buena plantilla para añadir sitios webs.***

Nota: con la directiva ***DocumentRoot /var/www***, se está indicando la carpeta raíz del sitio web por defecto. En esta carpeta se encuentra a partir de la instalación el fichero ***index.html***, con el siguiente contenido:

```
<html><body><h1>It works!</h1>
<p>This is the default web page for this server.</p>
<p>The web server software is running but no content has been added, yet.</p>
</body></html>
```

Importante:

- En el fichero de configuración ***apache2.conf,*** NO existe la directiva:

  *Include /etc/apache2/**sites-available***

- La directiva que ***realmente existe*** es:

  *Include /etc/apache2/**sites-enabled***

2.5. Directorio sites-enabled

- El **directorio sites-enabled** contiene los ***sitios webs habilitados para los clientes*** (navegadores web).

- Este directorio debe contener ***enlaces simbólicos (accesos directos) de los sitios disponibles***:

  - Por defecto aparece ***un único enlace simbólico llamado 000-default*** que enlaza con el fichero ***default*** incluido en ***sites-available***:

```
gradosuperior@ubuntuServerLVM:/etc/apache2/sites-enabled$ ls -l
total 0
lrwxrwxrwx 1 root root 26 ene 26 09:57 000-default -> ../sites-available/default
```

  - Apache lee los archivos de configuración por ***orden alfabético***: por eso aparece ***000 delante del fichero de configuración por defecto:*** simplemente para asegurarse que *se va a leer el primero*, es decir que se trata del fichero de configuración por defecto.

¿Por qué un sitio predeterminado?

Si alguien accede al servidor web mediante una conexión www.ejemplo.com en su navegador.

Si el servidor apache no tiene una configuración específica para el dominio www.ejemplo.com (no existe un archivo de configuración que coincida con el nombre de dominio o no está habilitada), utilizará la configuración por defecto.

2.5.1. Habilitar y deshabilitar sitios web

Para habilitar y deshabilitar sitios webs se pueden utilizar los ***comandos a2ensite y a2dissite*** que se instalaron con apache:

- Comando **a2ensite**: permite ***habilitar un sitio web***.
- Comando **a2dissite**: permite ***deshabilitar un sitio web***.

Ejemplo "deshabilitar el sitio web por defecto":

- Si se ejecuta el siguiente comando:

*sudo **a2dissite** default*

- Y si reiniciamos el servidor apache para activar la nueva configuración:

sudo apache2ctl graceful

- Se habrá deshabilitado el sitio web por defecto (default). En este caso apache al reiniciar el servicio (o al recargar la configuración de los ficheros) ***da un aviso "has no VirtualHosts"***, indicando que no existe ningún sitio virtual activo.

- En este caso si un *cliente intenta acceder a la página web del servidor*, obtendría un ***"Error 404: Not found"***. Hemos desactivado el sitio web predeterminado.

- Si comprobamos el contenido del ***directorio sites-enabled*** veremos que está vacío: se ha *eliminado el enlace simbólico hacia sites-available*.

Ejemplo "habilitar el sitio web por defecto":

- Para activar de nuevo el sitio web predeterminado, hay que ejecutar el comando a2ensite y volver a recargar la configuración de apache:

*sudo **a2ensite** default*

sudo apache2ctl graceful

2.6. Directorio mods-available

- El ***directorio mods-available*** almacena los ***módulos disponibles para la configuración del servidor apache***. Los módulos tienen que ser *cargados por Apache antes de que puedan ser configurados*.

- Los ***archivos*** contenidos en el *directorio mods-available* que terminan con ***".load"*** contiene la directiva necesaria para que apache pueda cargar el módulo. El archivo que termina con ***".conf"*** contiene alguna *configuración adicional* necesaria para ese módulo (*en caso de existir*).

```
gradosuperior@ubuntuServerLVM:/etc/apache2/mods-available$ ls
actions.conf          cern_meta.load     ident.load           proxy_http.load
actions.load          cgid.conf          imagemap.load        proxy.load
alias.conf            cgid.load          include.load         proxy_scgi.load
alias.load            cgi.load           info.conf            reqtimeout.conf
asis.load             charset_lite.load  info.load            reqtimeout.load
auth_basic.load       dav_fs.conf        ldap.conf            rewrite.load
auth_digest.load      dav_fs.load        ldap.load            setenvif.conf
authn_alias.load      dav.load           log_forensic.load    setenvif.load
authn_anon.load       dav_lock.load      mem_cache.conf       speling.load
authn_dbd.load        dbd.load           mem_cache.load       ssl.conf
authn_dbm.load        deflate.conf       mime.conf            ssl.load
authn_default.load    deflate.load       mime.load            status.conf
authn_file.load       dir.conf           mime_magic.conf      status.load
```

2.7. Directorio mods-enabled

- El ***directorio mods-enabled*** contiene ***enlaces simbólicos a los archivos de configuración de los módulos*** de apache que se **deseen cargar cuando se inicia**. Es decir este directorio contiene los módulos que estarán habilitados.

- El orden de las directivas include dentro del fichero de configuración apache2.conf es importante ya que primero debe aparecer:

*Include /etc/apache2/**mods-enabled/*.load***

  Para cargar los módulos y posteriormente para configurar dichos módulos (*los cargados previamente*):

*Include /etc/apache2/**mods-enabled/*.conf***

2.7.1. Habilitar y deshabilitar módulos

Para habilitar y deshabilitar módulos se pueden utilizar los ***comandos a2enmod y a2dismod*** que se instalaron con apache:

- Comando **a2enmod**: permite ***habilitar un módulo.***
- Comando **a2dismod**: permite ***deshabilitar un módulo.***

Ejemplo "habilitar el módulo userdir":

- Si se quiere ***habilitar el módulo userdir***, que permite que Apache sirva de forma automática *páginas webs de los directorios personales de los usuarios*, se deberá ejecutar:

*sudo **a2enmod** userdir*

Ejemplo "deshabilitar el módulo status":

- Si se quiere ***deshabilitar el módulo status***, que indica el estado actual del Servidor apache, se deberá ejecutar:

*sudo **a2dismod** status*

***Nota**: recuerda que para que cualquier cambio tome efecto hay que volver a cargar o reiniciar Apache !!!*

2.8. Archivo envvars

- El ***archivo envvars*** contiene algunas variables de entorno que son utilizados por algunos scripts relacionados con Apache como por el ejemplo apache2ctl. *Sólo es necesario modificar este archivo si cambian el **usuario o el grupo** que **utiliza Apache** por defecto*.
  - El usuario por defecto de apache es: ***www-data***
  - El grupo por defecto de apache es: ***www-data***

2.9. Archivo httpd.conf

- En versiones anteriores de apache (*o si se compila apache desde código fuente*), el archivo de configuración principal se llama ***httpd.conf***. Si se instala el paquete, el fichero de

configuración se llama "***apache2.conf***". En estas versiones se conserva el fichero "httpd.conf" por si el usuario quiere especificar alguna directiva aunque aparece ***vacío por defecto***.

3. Configuración del MPM (*"Multi-Processing Method")* Apache

El MPM determina el mecanismo que va a utilizar Apache para manejar las ***conexiones (múltiples) de los usuarios*** afectando a la ***cantidad de memoria que utilizará apache en el servidor*** *(por tanto afecta al tiempo de respuesta del servidor si se incrementa el tráfico del sitio web*).

Aunque existen varias MPMs, las principales son: ***mpm_worker y mpm_prefork***.

- ***mpm_worker***: maneja las conexiones creando nuevos hilos de ejecución (*threads*) dentro de cada proceso hijo.
- ***mpm_prefork***: genera un proceso nuevo para manejar cada conexión.

*Se considera que **mpm_worker es más eficaz, pero algunos módulos no son estables** cuando apache está trabajando en modo mpm_worker. Si se instala alguno de estos módulos el mpm_worker será reemplazado por el modo mpm_prefork que es más "viejo" pero más compatible.*

3.1. ¿En qué modo MPM se está ejecutando Apache?

La forma más fácil de ver ***qué modo mpm está utilizando Apache*** es ver los paquetes mpm que se han instalado. Para ver la ***lista de todos los mpm disponibles para instalar*** se puede utilizar el siguiente comando:

aptitude search apache2-mpm-

```
gradosuperior@ubuntuServerLVM:/etc/apache2$ aptitude search apache2-mpm-
p   apache2-mpm-event              - Servidor Apache HTTP - modelos de controla
p   apache2-mpm-event:i386         - Servidor Apache HTTP - modelos de controla
p   apache2-mpm-itk                - MPM multiusuario para Apache 2.2
p   apache2-mpm-itk:i386           - MPM multiusuario para Apache 2.2
p   apache2-mpm-prefork            - Apache HTTP Server - traditional non-threa
p   apache2-mpm-prefork:i386       - Apache HTTP Server - traditional non-threa
i A apache2-mpm-worker             - Servidor HTTP Apache, modelo de hilos de a
p   apache2-mpm-worker:i386        - Servidor HTTP Apache, modelo de hilos de a
gradosuperior@ubuntuServerLVM:/etc/apache2$
```

Del resultado que aparezca por pantalla interesa ***fijarse en la primera columna***:

- Una **"p"** indica un *paquete disponible para instalar*. **-**
  Una **"i"** indica que *el paquete ya está instalado*.

En este caso el paquete que ya está instalado es **"apache2-mpm-worker"**, por lo que la instalación de apache está utilizando ***mpm_worker***.

También se debería ejecutar el ***comando apache2ctl –l***, que muestra la lista de los nombres de los módulos compilados que utiliza Apache:

```
gradosuperior@ubuntuServerLVM:/etc/apache2$ sudo apache2ctl -l
[sudo] password for gradosuperior:
Compiled in modules:
  core.c
  mod_log_config.c
  mod_logio.c
  worker.c
  http_core.c
  mod_so.c
gradosuperior@ubuntuServerLVM:/etc/apache2$
```

No es la lista completa de módulos que se cargan *cuando Apache se inicia, sólo una lista de los* ***módulos que se han compilado en la instalación base de Apache****. En este caso se observa que el mpm es el worker.*

3.2. Directiva IfModule

- La ***directiva IfModule*** permite incluir una serie de instrucciones en un bloque, que sólo serán tenidos en cuenta por Apache si el módulo indicado está cargado.

- *El nombre del módulo que hay que utilizar con IfModule* es el nombre tal cual lo conoce el código del servidor, no su "descripción".
    - Ejemplo: Un bloque ***IfModule para el módulo SSL (mod_ssl)*** se debería incluir como: *<IfModule mod_ssl.c>*

- En el ***fichero de configuración apache2.conf*** se pueden ver distintos ***bloques IfModule*** para realizar los ajustes de configuración necesarios para **cada modo *mpm***:

```
<IfModule mpm_prefork_module>
    StartServers          5
    MinSpareServers       5
    MaxSpareServers      10
    MaxClients          150
    MaxRequestsPerChild   0
</IfModule>

<IfModule mpm_worker_module>
    StartServers          2
    MinSpareThreads      25
    MaxSpareThreads      75
    ThreadLimit          64
    ThreadsPerChild      25
    MaxClients          150
    MaxRequestsPerChild   0
</IfModule>
```

3.3. Configuración de los MPMs

Los ***ajustes mpm predeterminados*** seguramente son los más adecuados para un entorno de servidor con ***1 GB de memoria*** disponible para apache.

3.3.1. Directivas para configurar mpm_prefork

El ***mpm_prefork*** crea un nuevo proceso por cada conexión que maneja el servidor web. Dispone de las siguientes **directivas:**

- ***StartServers***: número de procesos hijos del servidor que se crean al iniciar Apache.
- ***MinSpareServers***: número mínimo de procesos hijos en el servidor que se mantendrán en espera para atender "picos" de demanda en las peticiones.
- ***MaxSpareServers***: número máximo de procesos hijos en el servidor que se mantendrán en espera para atender "picos" de demanda en las peticiones.
- ***MaxClients***: número máximo de procesos hijo que serán creados para atender peticiones.
- ***MaxRequestPerChild***: límite en el número de peticiones que un proceso hijo puede atender durante su vida. Con el valor por defecto 0 no se establece ningún límite.

3.3.2. Directivas para configurar mpm_worker

El ***mpm_worker*** crea varios hilos (*thread*) de ejecución dentro de cada proceso hijo arrancado por apache en el servidor. Dispone de las siguientes **directivas:**

- ***StartServers***: número de procesos hijos del servidor que se crean al iniciar Apache.
- ***MinSpareThreads***: número mínimo de hilos (*thread*) en espera para atender "picos" de demanda en las peticiones.
- ***MaxSpareThreads:*** número máximo de hilos (*threads*) en espera.
- ***ThreadLimit:*** establece el límite superior (valor máximo) del número de hilos (*threads*) por proceso hijo que pueden especificarse.
- ***ThreadsPerChild:*** número de hilos (*threads*) creados por cada proceso hijo.
- ***MaxClients***: número máximo de procesos hijo que serán creados para atender peticiones.
- ***MaxRequestPerChild***: límite en el número de peticiones que un proceso hijo puede atender durante su vida. Con el valor por defecto 0 no se establece ningún límite.

4. Directivas de Configuración de Apache

4.1. Directivas de identificación del servidor

ServerName

- La directiva ServerName especifica el nombre de host (*nombre FQDN*) y el puerto que usa el Servidor para identificarse. Si el nombre del servidor es "servidor.ejemplo.com", y la máquina tiene como alias DNS "www.ejemplo.com", se debería utilizar:
    - ***ServerName www.ejemplo.com:80***

- Hay que incluir esta directiva para que no aparezca el siguiente aviso cada vez que se reinicia apache: "*apache2: could not reliably determine the server's fully qualified domain name, using 127.0.01 for ServerName"*

- En caso de utilizar ***hosts virtuales basados en nombre***, la directiva ServerName dentro de una sección <VirtualHost> debe especificar el nombre de host que debe aparecer en la cabecera de petición para coincidir con ese host virtual.

ServerAdmin

- Valor por defecto: ***webmaster@localhost*** (*serverAdmin del sitio Default*)
- Especifica la dirección de e-mail que el Servidor incluye en cualquier mensaje de error que envía al cliente.

4.2. Directivas de localización de ficheros

DocumentRoot

- Valor por defecto: ***/var/www*** *(para el sitio por defecto "Default"*)
- Especifica el directorio principal que contiene la estructura de directorios visibles desde la web. A menos que se especifique alguna otra equivalencia mediante una ***directiva Alias***, el Servidor añade la ruta de la URL solicitada a este directorio para construir la ruta del documento a servir.

Alias

- La directiva alias permite que se almacenen documentos web en el sistema de ficheros en otra ubicación distinta a la marcada por la directiva ***DocumentRoot***.
- A modo de ejemplo en el sitio por defecto "Default" está establecido el siguiente alias:
    - ***Alias /doc/ "/usr/share/doc/"***

ErrorLog

- Valor por defecto: ***${APACHE_LOG_DIR}/error.log*** *(en el fichero apache2.conf)* ○ Donde APACHE_LOG_DIR es una *variable de entorno* fijada en el fichero ***envvars***: *export APACHE_LOG_DIR = /var/log/apache2*
- Determina la ubicación del fichero en el que se almacenan los mensajes de error.

CustomLog

- Valor por defecto: ***${APACHE_LOG_DIR}/access.log*** *(en el sitio por defecto "Default")* - Determina la ubicación del fichero de registro ***access.log.***

PidFile

- Valor por defecto: ***${APACHE_PID_FILE}*** *(en el fichero apache2.conf)* ○ APACHE_PID_FILE es una variable de entorno fijada en el fichero **ennvars:**

  export APACHE_PID_FILE = /var/run/apache2.pid
- Almacena el fichero en el que el servidor guarda su ID de proceso (*pid)*.

ServerRoot

- Valor por defecto: ***/etc/apache2*** *(en el fichero apache2.conf)*
- Especifica la ubicación del directorio raíz donde se encuentra instalado el servidor web apache. NO es el árbol de directorios públicos correspondientes a los portales web. Esta directiva no debería cambiar a no ser que se mueva la carpeta de instalación del servidor web apache a otro directorio.

AccessFileName

- Valor por defecto: ***.htaccess*** *(en el fichero apache2.conf)*
- AccessFileName indica el nombre de archivo que el servidor buscará para ver la información de control de acceso en cada directorio.

DirectoryIndex

- Especifica el fichero por defecto que entrega el servidor para cada directorio en caso de que no se especifique ninguno en la URL.
- Su valor por defecto suele ser: ***index.html***. Se puede especificar más de un fichero y el orden con el que se especifican determinará la prioridad para decidir cuál se sirve.
- Ejemplo: ***DirectoryIndex index.php index.html***

ErrorDocument

- Esta directiva establece la configuración del servidor en caso de error. Se pueden establecer cuatro configuraciones distintas:
  - Mostrar un texto de error. ○ Redirigir a un fichero de nuestro servidor. ○ Redirigir a un fichero fuera de nuestro servidor.

    - Ejecutar un programa CGI que realice algún tipo de operación y que genere su propia página web.
- Sintaxis: ***ErrorDocument código_error acción***.
- Ejemplo: ***ErrorDocument 404 /pagina_no_encontrada.html***
- El valor global para todo el servidor se suele establecer en el fichero ***conf.d/localized_error_pages***.

UserDir

- UserDir es el nombre del subdirectorio dentro del directorio principal de cada usuario dónde estarán los archivos HTML personales que serán servidor por el servidor Web.
- Ejemplo: ***userdir mis_paginas***
- Para acceder a la página "ejemplo.html" del usuario "gradosuperior" ubicada físicamente en: (***/home/gradosuperior/mis_paginas/ejemplo.html)***, habría que especificar la siguiente ruta (*siempre que se hubiera establecido la directiva UserDir):* ○ ***http://www.corteingles.lan/~gradosuperior/ejemplo.html***

DefaultType

- Valor por defecto: ***none*** *(en el fichero apache2.conf)*
- Especifica el tipo MIME que se servirá por defecto en caso de no conocer la extensión del archivo que se está sirviendo. Si se cambia el valor ***none***, se debería establecer el valor ***text/plain*** para indicar que es "texto plano".

4.3. Directivas de seguridad

ServerTokens

- Valor por defecto: ***OS*** (*se suele establecer en* ***conf.d/security*****)**
- Esta directiva gestiona si el campo ***Server*** de las ***cabeceras de las respuestas*** que se envían a los clientes incluye una descripción del Sistema Operativo genérico del Servidor así como información sobre los módulos compilados.
- Posibles valores:

Valores ServerTokens	Información enviada
ServerTokens **Prod**	*Server: Apache*

ServerTokens **Major**	*Server: Apache/2*
ServerTokens **Minor**	*Server: Apache/2.2*
ServerTokens **Min**	*Server: Apache/2.2.21*
ServerTokens **OS**	*Server: Apache/2.2.21 (Ubuntu)*
ServerTokens **Full**	*Server: Apache/2.2.21 (Ubuntu) PHP 4.2.2*

ServerSignature

- Valor por defecto: ***On*** (*se suele establecer en* ***conf.d/security*****)**
- Añade una línea que contiene el valor de la ***directiva ServerTokens*** y la ***directiva ServerName*** para cualquier documento generado por el servidor, tales como mensajes de error devueltos a los clientes.
- El valor de ***ServerSignature*** puede ser: ***On, Off o Email***.
    - ***Off:*** desactiva esta información.
    - ***Email***, añade además una etiqueta HTML ***mailto:*** con el valor de la ***directiva ServerAdmin***.

4.4. Directivas de control de la conexión

Timeout

- Valor por defecto: ***300 (segundos)*** *(en el fichero apache2.conf)*
- Timeout define, en segundos, el tiempo que el Servidor esperará para que ocurran determinados eventos (*como la cantidad de tiempo que tarda en recibir una petición GET*) antes de cerrar una conexión.

KeepAlive

- Valor por defecto: ***On*** *(en el fichero apache2.conf)*
- KeepAlive permite que se establezcan conexiones HTTP persistentes, es decir el servidor permitirá más de una petición por conexión.

MaxKeepAliveRequests

- Valor por defecto: ***100 (peticiones)*** *(en el fichero apache2.conf)*
- MaxKeepAliveRequests limita el número de peticiones permitidas en una conexión persistente (*debe estar activado KeepAlive*). Si se especifica el valor 0, el número de peticiones permitidas es ilimitado. Se recomienda que se especifique un valor alto (*500*) para obtener el máximo rendimiento del servidor.

KeepAliveTimeout

- Valor por defecto: ***5 (segundos)*** *(en el fichero apache2.conf)*
- Es el tiempo en segundos que Apache esperará peticiones subsiguientes antes de cerrar una conexión persistente.

4.5. Directivas de Sección (contenedores)

Un contenedor es una agrupación de ficheros a los que se aplica un conjunto de reglas. Un contenedor puede ser un directorio o directorios determinados, un tipo de fichero concreto, una URL (localización) o un servidor virtual alojado en nuestro servidor apache.

Secciones	Descripción
<Directory>	Los parámetros que se encuentran dentro de esta sección sólo se aplicarán al *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
<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
<VirtualHost>	Los parámetros sólo se aplicarán a las ***peticiones*** que vayan dirigidas a este ***"host" (nombre de servidor, o dirección IP, o puerto TCP)***

Ejemplo de Secciones (contenedores) del sitio por defecto (*"default"*):

```
<VirtualHost *:80>
        ServerAdmin webmaster@localhost
        DocumentRoot /var/www
        <Directory />
                Options FollowSymLinks
                AllowOverride None
        </Directory>
        <Directory /var/www/>
                Options Indexes FollowSymLinks MultiViews
                AllowOverride None
                Order allow,deny
                allow from all
        </Directory>
```

Dentro de una **sección de Directorio (*directory*)** se definen directivas para establecer distintos permisos:

Directiva	Opciones	Función
Options	***None***	No establece ninguna opción

	All	Establece todas las opciones
	Indexes	Permite visualizar las páginas (índice) existentes en el directorio
	FollowSymlinks	Permite seguir los enlaces simbólicos del directorio.
	Multiviews	Permite servir distintas versiones de fichero (*conjunto de caracteres, idiomas, ..)* a los clientes en función de las características establecidas en el navegador.
	ExecCGI	Admite la ejecución de scripts CGI.
AllowOverride	***None***	No establece ninguna opción. El servidor no leerá los archivos .htaccess
	All	Establece todas las opciones. Permite usar las directivas especificadas en .htaccess
	FileInfo	Muestra la información de los archivos del directorio.
Order	***allow,deny***	Primero aplicará los permisos de allow y luego los de deny.
	deny,allow	Primero aplicará los permisos de deny y luego los de allow.
Allow	***from all***	Admite cualquier acceso al directorio.
	from IP	Admite cualquier acceso al directorio proveniente de la dirección IP indicada.
	from dominio	Admite cualquier acceso al directorio desde el dominio especificado.
Deny	***from all***	Deniega cualquier acceso al directorio.
	from IP	Deniega cualquier acceso al directorio proveniente de la dirección IP indicada.
	from dominio	Deniega cualquier acceso al directorio desde el dominio especificado.

5. Hosts Virtuales (*Servidores Virtuales)* en Apache

Un servidor Apache puede servir múltiples "*portales o sitios web*" a la vez utilizando una única dirección IP. Para aplicar esta técnica, Apache necesita diferenciar el acceso a cada "portal web":

- ***Configurando puertos distintos para cada portal web***: el acceso se hará a través de cada puerto.

- ***A través de diferentes nombres de dominio*** alojados en el servidor y que se conectan a un mismo puerto (*normalmente al puerto 80)*

5.1. Servidores virtuales: Acceso a través de puertos distintos

Hay que configurar los ***VirtualHosts*** del Servidor Apache, indicando los puertos dónde queremos que escuche el servidor web.

a. Editar el fichero /etc/apache2/ports.conf:

- Mediante la ***directiva NameVirtualHost*** indicar que se va a trabajar con varios hosts virtuales indicando los puertos de escucha.
    i. Ejemplo: *NameVirtualHost *:80*
    *NameVirtualHost *:9999*

- Indicar los puertos a "escuchar" mediante la ***directiva Listen***. Si se quiere que Apache escuche por direcciones IP específicas habría que indicar en la ***directiva Listen*** la dirección IP y el puerto (*Listen 172.16.0.1:80).*
    i. Ejemplo: *Listen 80*
    Listen 9999

El asterisco (*) indica que vamos a recibir peticiones de VirtualHost en todas las IP por las que esté escuchando el servidor. Si interesa escucha por un IP específica cambiar el asterisco por la dirección IP.

b. Dentro de la ***carpeta sites-available***, crear el fichero de configuración de otro "Sitio Virtual" a partir del contenido el fichero ***default:***

sudo cp default default_9999

c. Cambiar las directivas del ***sitio que va a escuchar por el puerto 9999*** *(fichero default_9999):*

- Cambiar el valor de ***sección VirtualHost*** para indicar el ***puerto a escuchar***:

*<VirtualHost *:9999>*

- Cambiar el valor de la directiva: ***DocumentRoot*** *(para indicar la ruta del nuevo sitio web)*. También se puede cambiar el valor del resto de directivas.

```
<VirtualHost *:9999>
        ServerAdmin webmaster@localhost
        DirectoryIndex index.php index.html
        DocumentRoot /var/www_puerto_9999
```

d. Crear la carpeta indicada por la directiva ***DocumentRoot*** alojando el ***portal web*** en su interior.

cd /var sudo mkdir www_puerto_9999 cd /www_puerto_9999 sudo vim index.html

Nota: Habitualmente los "sitios web" son almacenados dentro de las carpetas **/var/www** o **/srv/www**. Y una buena recomendación es hacer que el propietario de estos directorios sea el ***usuario definido por Apache www-data:***

cd /var *sudo chown –R www-data:www-data www_puerto_9999*

e. Habilitar el sitio web creado para atender las peticiones por el puerto 9999, mediante el comando ***a2ensite:***

sudo a2ensite default_9999

f. Verificar que existe el ***enlace simbólico*** del fichero ***default_9999*** en la carpeta ***sitesenables***. Y reiniciar el servidor apache.

sudo /etc/init.d/apache2 reload

g. Comprobar desde un ***cliente el acceso al VirtualHost*** por el puerto ***9999***:

5.2. Servidores virtuales: Acceso a través de nombres de dominios diferente

a. Fichero /etc/apache2/ports.conf:

- En este caso el ***puerto será común a todos los servidores virtuales***. En principio, el fichero ***ports.conf*** debe indicar que el servidor escuche por el puerto 80:

```
NameVirtualHost *:80
Listen 80
```

b. Dentro de la ***carpeta sites-available***, crear el fichero de configuración de otro "Sitio Virtual" a partir del contenido el fichero ***default:***

```
sudo cp default sitio_carrefour
```

c. Cambiar las directivas del ***Nuevo Sitio Virtual***:

- Establecer el valor de ***la directiva ServerName***, para indicar el nombre de dominio del Servidor Virtual.

```
ServerName www.carrefour.lan
```

- Cambiar el valor de la directiva: ***DocumentRoot*** *(para indicar la ruta del nuevo sitio web)*. También se puede cambiar el valor del resto de directivas como ***ServerAdmin, DirectoryIndex, ..***

d. Crear la carpeta indicada por la directiva ***DocumentRoot*** alojando el ***portal web*** en su interior.

```
cd /var sudo mkdir www_carrefour cd /www_carrefour
sudo vim index.html
```

Hacer que el propietario de esta carpeta (y de las subcarpetas que contenga) sea el ***usuario definido por Apache www-data:***

```
cd /var
sudo chown –R www-data:www-data www_carrefour
```

e. Habilitar el sitio web creado para atender las peticiones del sitio ***www.carrefour.lan*** mediante el comando ***a2ensite:***

```
sudo a2ensite sitio_carrefour
```

f. Verificar que existe el ***enlace simbólico*** del fichero ***sitio_carrefour*** en la carpeta ***sitesenables***. Y reiniciar el servidor apache.

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

g. Comprobar desde un ***cliente el acceso al VirtualHost*** por el ***nombre de dominio "www.carrefour.lan":***

Obtendremos el error de "Servidor no encontrado": problemas de resolución de nombres DNS!!!!

Modificar el servicio DNS para que resuelva los nombres de la zona "carrefour.lan". Una vez resueltos los problemas:

6. Seguridad en Apache: Proteger directorios/Autentificación

6.1. Autentificación Básica (módulo auth_basic)

Hay que verificar que el ***módulo auth_basic*** está habilitado en ***/etc/apache2/mods-enabled***

```
gradosuperior@ubuntuServerLVM:/etc/apache2/mods-enabled$ ls
alias.conf            authz_user.load  dir.conf          reqtimeout.conf
alias.load            autoindex.conf   dir.load          reqtimeout.load
auth_basic.load       autoindex.load   env.load          setenvif.conf
authn_file.load       cgid.conf        mime.conf         setenvif.load
authz_default.load    cgid.load        mime.load         userdir.conf
authz_groupfile.load  deflate.conf     negotiation.conf  userdir.load
authz_host.load       deflate.load     negotiation.load
```

Se puede proteger un directorio ***mediante un fichero .htaccess en el directorio a proteger***.

a. ***Crear el fichero .htaccess*** en el directorio a proteger (*por ejemplo /var/www/.htaccess*), con el siguiente contenido:

AuthName "Acceso restringido a este recurso"

AuthType Basic ***AuthUserFile*** /var/.htpasswd ***Require valid-user***

- Si solo se quiere permitir el **acceso a un usuario específico**:
  - ***Require user nombre_usuario***
  - Ejemplo: *requiere user gradosuperior*

- Si se quiere permitir el **acceso a varios usuarios**:
  - ***Require user nombre_usuario1 nombre_usuario2 …***
  - Ejemplo: *requiere user gradosuperior gradomedio profesor*

- El fichero de contraseñas que indica ***AuthUserFile*** normalmente se ubica en un directorio fuera de la raíz web para tener más seguridad.

b. Crear el ***fichero de contraseñas*** mediante el ***comando htpasswd –c*** (*con el parámetro –c se crea el fichero*)

htpasswd –c /var/.htpasswd gradosuperior

```
gradosuperior@ubuntuServerLVM:/var/www$ sudo htpasswd -c /var/.htpasswd gradosup
erior
New password:
Re-type new password:
Adding password for user gradosuperior
gradosuperior@ubuntuServerLVM:/var/www$
```

c. Para añadir más usuarios al fichero de contraseñas volver a ejecutar el comando SIN el parámetro –c

htpasswd /var/.htpasswd gradomedio

Contenido del fichero .htpasswd

```
gradosuperior:$apr1$slo592R0$oqYW9r6Lot2O/6UnKpfiY0
gradomedio:$apr1$qhfKUOCs$nSNLfEbr.MyT7rdHHN4Ea0
```

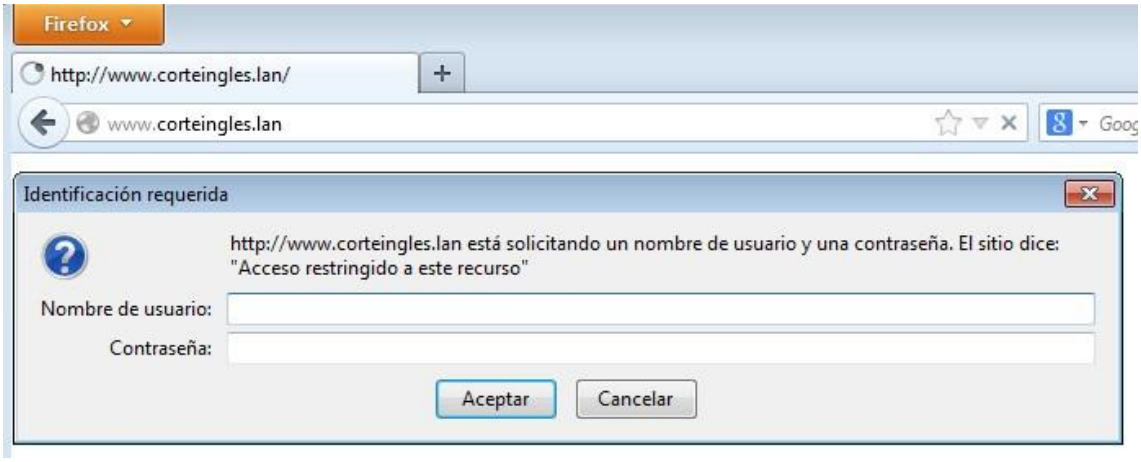
(La contraseña se almacena cifrada mediante la implementación estándar de crypt())

d. Editar el ***directorio virtual*** en el fichero de configuración de apache a proteger, para activar la ***opción AllowOverride All,*** que permite el uso de los ***archivos .htaccess***

```
<VirtualHost *:80>
        ServerAdmin webmaster@localhost
        DirectoryIndex index.php index.html
        DocumentRoot /var/www
        <Directory />
                Options FollowSymLinks
                AllowOverride None
        </Directory>
        <Directory /var/www/>
                Options FollowSymLinks MultiViews
                AllowOverride All
                Order allow,deny
                allow from all
        </Directory>
```

e. ***Reiniciar*** el servicio Apache.

f. Acceso con el cliente al ***directorio protegido***:



<u>Autentificación de Grupos</u>

Se puede ***autentificar grupos de usuarios*** en el ***fichero .htaccess*** incorporando la ***directiva AuthGroupFile*** y cambiando el valor de ***Require*** a ***group:***

AuthName "Directorio Secreto"
AuthType Basic
AuthUserFile /var/.htpasswd
AuthGroupFile /var/.htgroup
Require group administradores

- **AuthUserFile:** sigue indicando el fichero de passwords (no hace falta crear uno adicional).
- **AuthGroupFile:** indica la ubicación completa del fichero que relaciona los grupos y los usuarios que pertenecen a cada grupo.

○ El formato de este fichero es: ***nombre_del_grupo:usuario1 usuario2 …*** ○
Ejemplo: ***administradores: gradosuperior profesor***

alumnos: juan luis antonio raquel

```
administradores:gradosuperior profesor
alumnos:juan luis antonio raquel
```

(*donde gradosuperior, profesor, juan, luis, antonio y raquel son usuarios autentificados en el fichero indicado por* ***AuthUserFile****)*

```
gradosuperior:$apr1$M/JKCdDA$XY4788uXzX3mTLjSIOMAB1
profesor:$apr1$5grSlUKi$0bGkdKqREyX4c6GVZJFPA0
juan:$apr1$XJNaSMFO$ZSjMciLprBaWk0OKN1eDe/
luis:$apr1$IlS5kpSY$7WAn4Oq4SDrQqnfERIJL10
antonio:$apr1$eHVn0o81$a7F6xkvzcfSaBLUrWskNH0
raquel:$apr1$3AgYh2QO$Vr/0RXKOhHhdJkMRP3RfP0
```

- ***Require group administradores***: permite el acceso únicamente a los usuarios que pertenezcan al *grupo administradores*.

6.2. Autentificación Digest (por resúmenes MD5)

Para habilitar una ***autentificación Digest con resúmenes MD5***, hay que proceder de la misma forma que para una ***autentificación Basic***, teniendo en cuenta las siguientes consideraciones:

- ***Hay que activar el módulo auth_digest*** *(reiniciando el servicio posteriormente):*

sudo a2enmod ***auth_digest*** *sudo*
/etc/init.d/apache2 reload

- ***Crear el fichero .htaccess*** en el directorio a proteger (*por ejemplo /var/www/.htaccess*), estableciendo:

AuthType Digest
AuthName "Registro de visitantes"
AuthUserFile /var/.htpassw_digest
Require valid-user

- ***Crear el fichero de contraseñas*** *con resumen MD5* mediante el ***comando htdigest***, de la misma forma que se utilizó el *comando htpasswd* (*utilizar la* ***opción –c*** *para crear el fichero solo la primera vez)*

Lo único que hay que incorporar es *"a modo de identificación" para facilitar la administración cuanto tenemos varias secciones protegidas,* el valor especificado en la ***directiva AuthName*** después del nombre del fichero.

```
root@ubuntuServerLVM:/var# htdigest -c /var/.htpassw_digest "Registro de visitan
tes" gradosuperior
Adding password for gradosuperior in realm Registro de visitantes.
New password:
Re-type new password:
```

7. Instalación de certificados y Acceso Seguro con HTTPS

Si se quieren utilizan conexiones seguras mediante el ***protocolo HTTPS*** para acceder a determinadas páginas, hay que ***configurar Apache*** para que cargue el ***módulo SSL*** (*en algunas versiones de Apache2 suele venir ya integrado).*

Comprobar si está habilitado el módulo ssl

- Consultando el directorio ***/etc/apache2/mods-enabled***, y en caso de que no esté, habilitarlo mediante el comando ***a2enmod***.

sudo a2enmod ssl

```
root@ubuntuServerLVM:/etc/apache2/mods-enabled# sudo a2enmod ssl
Enabling module ssl.
See /usr/share/doc/apache2.2-common/README.Debian.gz on how to configure SSL and
 create self-signed certificates.
To activate the new configuration, you need to run:
  service apache2 restart
```

(Recuerda reiniciar el servidor Apache para activar la nueva configuración)

Instalar OpenSSL

Además, también se necesita tener instalado ***OpenSSL***, que se encarga de la ***implementación SSL*** para llevar a cabo ***transferencias seguras***.

sudo apt-get install openssl

Nota: el módulo ***OpenSSL*** *no es recomendado en configuraciones con más de un servidor virtual que utilicen el mismo puerto de escucha*, ya que la conexión SSL del cliente se establece con una IP y puerto determinado del Servidor. Apache no sabría con qué "Servidor Virtual" debería utilizar SSL, escogiendo el primero que encuentra, no utilizando conexiones seguras para el resto de servidores virtuales.

Sí se podría tener un "Servidor Virtual" principal para el sitio principal que atienda las peticiones del puerto 80, y un "Servidor Virtual" para un ***sitio seguro*** que atienda peticiones por el ***puerto 443 y que utilice SSL.***

Para corregir este problema se utiliza la ***extensión TLS denominada SIN (Server Name Indication)***. En este caso es necesaria la instalación del ***paquete libapache2-mod-gnutls***, para cargar el ***módulo gnutls*** en vez de SSL.

Certificado de Autenticidad

Para ofrecer páginas de forma segura en un sitio web***, el servidor necesita un certificado que apruebe su autenticidad***.

Para generar un ***certificado "autofirmado"*** hay que ejecutar la ***instrucción make-ssl-cert*** *(pedirá cierta información para completar el certificado)* indicando como parámetro el ***fichero con extensión ".pem"***, que se generará al final de la ejecución y que contendrá las claves para efectuar el cifrado asimétrico por el servidor:

```
make-ssl-cert /usr/share/ssl-cert/ssleay.cnf  /etc/apache2/certificados/micertificado.pem
```

```
root@ubuntuServerLVM:/etc/apache2# make-ssl-cert /usr/share/ssl-cert/ssleay.cnf
/etc/apache2/certificados/micertificado.pem
```

(Crear previamente la carpeta /etc/apache2/certificados)

- El comando pide que se indique el ***Nombre del equipo*** (*campo commonName)*: nombre FQDN del servidor que quiera utilizar el certificado.

Revisar si está habilitado el puerto de escucha 443 (directiva Listen)

En el fichero ***ports.conf***, revisar que exista la siguiente configuración: se estará escuchando por el puerto 443 (***Listen 443***), si está ***activo*** el ***módulo mod_ssl.c***

```
NameVirtualHost *:80
Listen 80

<IfModule mod_ssl.c>
    # If you add NameVirtualHost *:443 here, you will also have to change
    # the VirtualHost statement in /etc/apache2/sites-available/default-ssl
    # to <VirtualHost *:443>
    # Server Name Indication for SSL named virtual hosts is currently not
    # supported by MSIE on Windows XP.
    NameVirtualHost *:443
    Listen 443
</IfModule>

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

Revisar y configurar los contenedores de los "Sitios Disponibles"

En la ***carpeta /etc/apache2/sites-available*** se encuentra el fichero ***default-ssl***, que se puede utilizar directamente o como patrón para generar ***"Sitios Virtuales" seguros.*** En el "Sitio Virtual" seguro serán necesarias establecer (al menos) las siguientes directivas:

- ***SSLRequireSSL***: se usa en el ***contenedor <Directory>*** y fuerza la utilización de SSL, evitando su desactivación.
- ***SSLEngine***: habilita **SSL** (valor **on**) en un servidor virtual, si se deshabilitó en el servidor principal.
- ***SSLCertificateFile***: indica la ruta del ***fichero*** (***.pem***) que contiene el ***certificado***.

En el fichero ***default-ssl*** del siguiente ejemplo, se han cambiado:

- El valor del contenedor **VirtualHost: (*VirtualHost *:443)***
- La ruta de la carpeta que se quiere establecer como segura mediante ***DocumentRoot: (DocumentRoot /var/www/seguro)***
- El contenedor ***Directory,*** la ruta de la carpeta segura: ***<Directory /var/www/seguro>*** ○ Y dentro del Directorio, la ***directiva SSLRequireSSL.***

```
<IfModule mod_ssl.c>
<VirtualHost *:443>
        ServerAdmin webmaster@localhost

        DocumentRoot /var/www/seguro
        <Directory />
                Options FollowSymLinks
                AllowOverride None
        </Directory>
        <Directory /var/www/seguro>
                Options Indexes FollowSymLinks MultiViews
                AllowOverride None
                Order allow,deny
                allow from all
                SSLRequireSSL
        </Directory>
```

- Verificar el valor de ***SSLEngine on***

```
#   SSL Engine Switch:
#   Enable/Disable SSL for this virtual host.
SSLEngine on
```

- Establecer el valor de ***SSLCertificateFile*** con el ***certificado generado anteriormente.***

```
SSLCertificateFile /etc/apache2/certificados/micertificado.pem
#SSLCertificateKeyFile /etc/ssl/private/ssl-cert-snakeoil.key
```

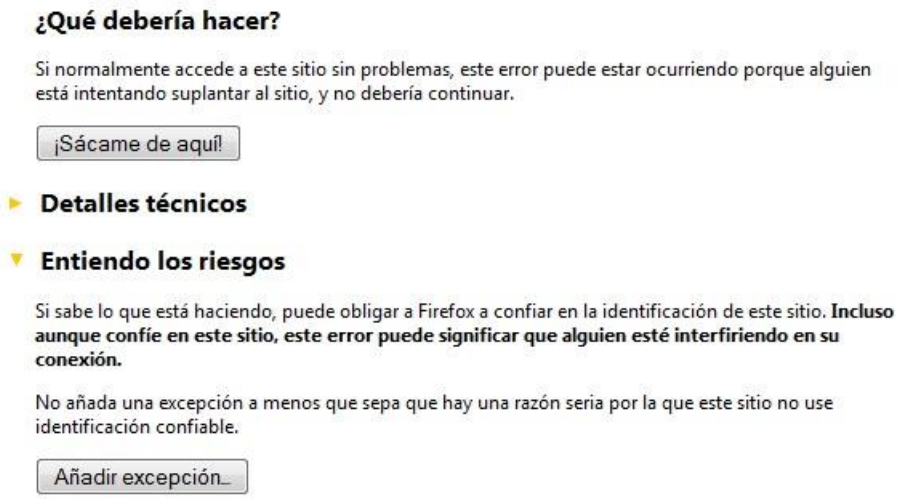
Habilitar el nuevo sitio web seguro y reiniciar el servidor apache

```
sudo a2ensite default-ssl sudo
/etc/init.d/apache2 reload
```

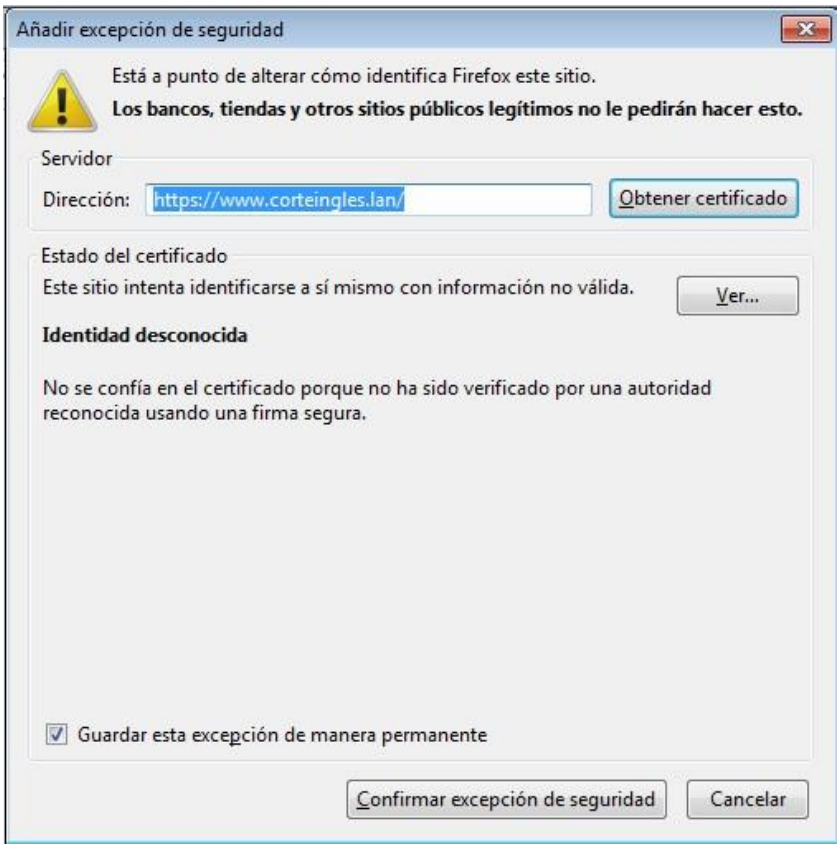
Probar la conexión segura desde un cliente

- Conectarse a la dirección: ***https://www.corteingles.lan***

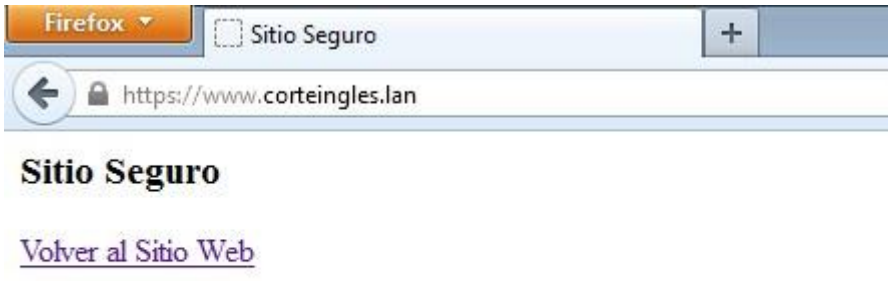
- Detecta la conexión con un ***certificado "autofirmado***", pulsar en ***Entiendo los Riesgos y en "Añadir Excepción".***



- Finalmente pulsar sobre el ***botón "Confirmar excepción de seguridad":***



- Acceso al sitio web por ***https:***



8. Analizadores de registros

8.1. Awstats

AWStats (Advanced Web Statistics) es un software libre que se distribuye bajo la ***licencia GNU GPL*** (General Public License), ***que generar estadísticas gráficas a partir de los logs del sistema.***

AWStats es capaz de ***visualizar logs generados por distintos servidores***, entre los que merece la pena destacar:

- ***Servidores web***: como *Apache e IIS*.
- ***Servidores FTP***: como *ProFTP*.

- ***Servidores de Correo SMTP***: como *Postfix y Sendmail*. - ***Servidores Proxy Web***: como *Squid*.

AWStats soporta cualquier sistema operativo, debido a que está escrito íntegramente en Perl, basta con que el servidor que lo va a interpretar tenga el módulo correspondiente instalado. Siendo así, las estadísticas de los LOGS se pueden generar tanto desde la línea de comandos, como un CGI y mostrar la información en una o varias páginas Web.

La página oficial es http://awstats.sourceforge.net/

8.1.1. Instalación de awstats

Para ***instalar awstats*** utilizamos el ***comando apt-get*** de la siguiente manera:

sudo apt-get install awstats

Entre los archivos instalados, vamos a destacar los siguientes:

Archivos instalados	Descripción
/usr/share/doc/awstats	Directorio donde se guardan archivos de ejemplo y ayuda.
/usr/share/awstats/lang/awstats-es.txt	Archivo de idioma castellano. Es un simple archivo de texto.
/usr/share/awstats/plugins	Directorio en el que se instalan los plugins que
	viene por defecto con el paquete.
/usr/share/awstats/icon	Directorio donde se instalan los iconos, que se van a utilizar para mostrar las estadísticas.
/usr/lib/cgi-bin/awstats.pl	Es el archivo que va a generar las estadísticas a partir de los LOGS del sistema. Escrito en Perl.
/etc/cron.d/awstats	Entrada de cron para actualizar las estadísticas periódicamente.
/etc/awstats/awstats.conf	Archivo de configuración de AWStats.

8.1.2. Configuración de awstats

Vamos a realizar una configuración para poder generar estadísticas para el servidor Apache.

Cambiar el idioma

- Lo primero que vamos a configurar es el idioma. AWStats viene preconfigurado para mostrar los LOGS y estadísticas en inglés.
- Los archivos de idioma han sido instalados en el ***directorio /usr/share/awstats/lang/.***
- Verificar que el archivo de configuración de AWStats (**/etc/awstats/awstats.conf**) conoce la ruta donde debe de buscar los archivos de idioma.

DirLang="/usr/share/awstats/lang"

- Para que el idioma por defecto sea el castellano, hay que sustituir en el fichero de configuración el valor de la directiva Lang "auto" por "es". El valor "auto" indica que mostrará el primer lenguaje disponible aceptado por el navegador.

Lang="es"

Verificar ruta de imágenes y Alias en Apache

- Para que las imágenes que contiene la página que se va a generar se vean correctamente, hay que indicarle a AWStats dónde están ubicadas.
- El fichero de configuración (***awstats.con)*** dispone de la ***directiva DirIcons***, que debe ***ser una ruta relativa a la ubicación de la página web***, es decir NO puede contener el valor *DirIcons="/usr/share/awstats/icon/",* ya que realmente intentaría acceder a: ***http://mi_dominio/usr/share/awstats/icon/***.
- Para resolver este problema, se pueden ***utilizar los alias del servidor apache***, añadiendo la siguiente línea en el *fichero de configuración de apache*:

Alias /awstats-icon/ /usr/share/awstats/icon/

- Y el valor que tiene que tener la ***directiva DirIcons*** es el nombre del ***alias que hemos creado en Apache:***

DirIcons="/awstats-icon"

Cambiar la directiva SiteDomain

- ***SiteDomain*** es la directiva que indica el nombre del dominio. En caso de manejar varios dominios en un mismo servidor/host (*Virtual Host*) es recomendable generar LOGS independientes, es decir, uno para cada dominio.
- Cambiar el valor de la directiva ***SiteDomain*** con el nombre del dominio:

SiteDomain="www.corteingles.lan"

Verificar la directiva LogFile (LOG de Apache2)

- Por último hay que indicar donde se encuentra el LOG de Apache2, verificando el valor de la directiva ***LogFile***:

LogFile="/var/log/apache2/access.log"

8.1.3. Puesta en marcha de awstats

- Llegado a este punto, deberíamos ser capaces de ***visualizar las estadísticas*** por medio del siguiente enlace:

http://dominio/cgi-bin/awstats.pl http://www.corteingles.lan/cgi-bin/awstats.pl

- Sin embargo, las ***estadísticas están vacías***, porque el archivo no lee los datos de los LOGS directamente sino que *genera un archivo de texto* y muestra la información a partir de dicho texto.
- Los *archivos de texto generados* se encuentran en: ***/var/lib/awstats***
- Para ***generar estos archivos por primera vez*** o actualizarlos manualmente, hay que ejecutar el archivo awstats.pl con los siguientes parámetros:

*/usr/lib/cgi-bin/awstats.pl -config=**dominioServidor** –update*

*/usr/lib/cgi-bin/awstats.pl -config=**www.corteingles.lan** –update*

- Para realizar esta tarea automáticamente, se puede modificar la ***tarea cron*** que instala awstats en ***/etc/cron.d/awstats***:

*0,10,20,30,40,50 * * * * root /usr/lib/cgi-bin/awstats.pl –config=www.corteingles.lan –update > /dev/null*

```
0,10,20,30,40,50 * * * * root /usr/lib/cgi-bin/awstats.pl -config=www.corteingle
s.lan -update > /dev/null
```

(Actualiza las estadísticas cada 10 minutos)

8.2. Webalizer

Webalizer es una herramienta ***de análisis de servidores*** rápida, fiable y fácil de usar. Genera informes con información detallada sobre todos los movimientos que se producen en un ***servidor web***.

Estos informes están en ***formato HTML,*** por lo que se pueden visualizar con cualquier navegador web, y son sencillos de configurar y totalmente personalizables.

8.2.1. Instalación de Webalizer

Para la instalación de webalizer utilizamos apt-get de la siguiente forma:

```
sudo apt-get install webalizer
```

8.2.2. Configuración de webalizer

Sobre el fichero de configuración de ***webalizer (*/etc/webalizer.conf*)***, hay que realizar los *siguientes cambios* para poner en marcha las estadísticas web:

- Establecer el *fichero donde apache esta guardando sus LOGS* mediante la ***directiva LogFile***:

```
LogFile /var/log/apache2/access.log
```

- Indicar el *directorio donde se van a generar los ficheros HTML* que muestran la estadística mediante la ***directiva OutputDir:***

```
OutputDir /var/www/webalizer
```

- Indicar el nombre de dominio mediante la ***directiva HostName:***

```
HostName www.corteingles.lan
```

- Se puede ***personalizar la apariencia de la página web*** con las directivas:
    - ***HTMLPre***: define la etiqueta <html> a incluir. ○ ***HTMLHead***: define código html que se incluirá dentro de la etiqueta <head>. ○ ***HTMLBody***: define el contenido de la etiqueta <body>.
    - ***HTMLPost***: define el código html que se insertará inmediatamente antes de la primera etiqueta <hr> del documento, que está después del título y el periodo resumido.

- ***HTMLTail:*** define código html que se puede añadir al final de cada documento HTML.
- ***HTMLEnd:*** define el código html de la última línea de cada página html. Debería incluir </body> y </html>

8.2.3. Puesta en marcha de webalizer

- Para ***generar y actualizar las estadísticas*** sólo es necesario ejecutar el comando:

webalizer

(*El comando webalizer también actualiza los cambios generados sobre el fichero de configuración)*

- Para ver las estadísticas sólo hay que indicar la siguiente dirección URL:

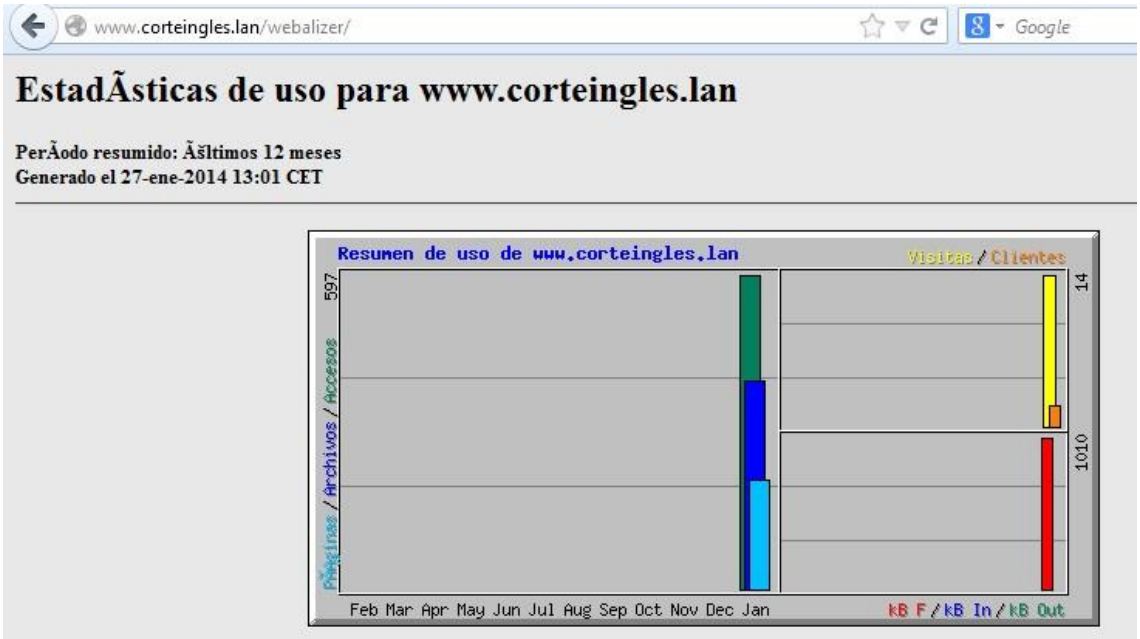
*http://www.corteingle.lan/**webalizer***

- Para automatizar la tarea de generación de las estadísticas, se puede crear un fichero (*webalizer por ejemplo*) en la carpeta **/etc/cron.d***,* que contenga el siguiente comando:

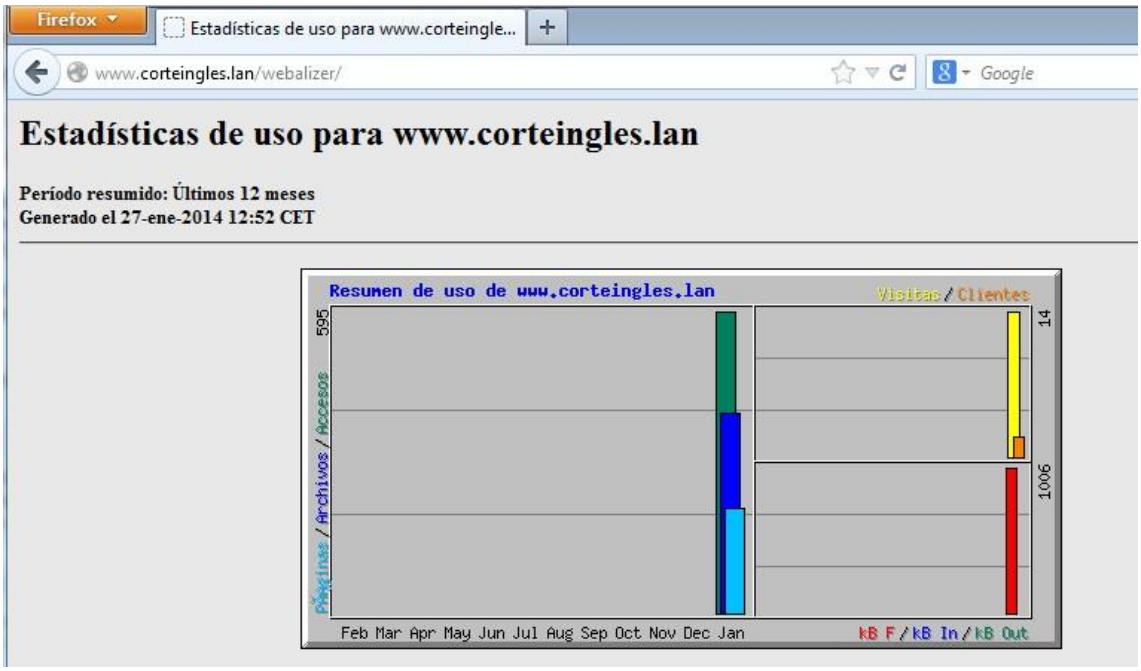
*0,10,20,30,40,50 * * * * root **webalizer** > /dev/null*

```
0,10,20,30,40,50 * * * * root webalizer > /dev/null
```

(Actualiza las estadísticas cada 10 minutos)



¿Y si queremos que la página aparezcan sin problemas de acentos? ¿Cómo se consigue?


Firefox
Estadísticas de uso para www.corteingle...
www.corteingles.lan/webalizer/
Google
Estadísticas de uso para www.corteingles.lan
Período resumido: Últimos 12 meses
Generado el 27-ene-2014 12:52 CET
Resumen de uso de www.corteingles.lan
Visitas / Clientes
595
Páginas / Archivos / Accesos
14
1006
Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec Jan
kB F / kB In / kB Out