

5.11. Ficheros `.htaccess`

En la máquina **ServidorLinuxXX**

- Habilita en *Apache* el uso de ficheros de configuración personalizada de directorios (`.htaccess`) en el directorio `/home/profesor/blog` para que sea el propio usuario **profesor** el que pueda controlar como sirve *Apache* los contenidos de ese directorio.
- Configura el alias `/blog` que permita acceder al directorio `/home/profesor/blog`.
- Como usuario **profesor** haz uso del fichero `.htaccess` y configura sobre el directorio `/home/profesor/blog`.
 - Que solo se pueda acceder desde **DesarrolloW7XX**.
 - Autenticación HTTP *Digest* para que solo pueda acceder el usuario **blog**.

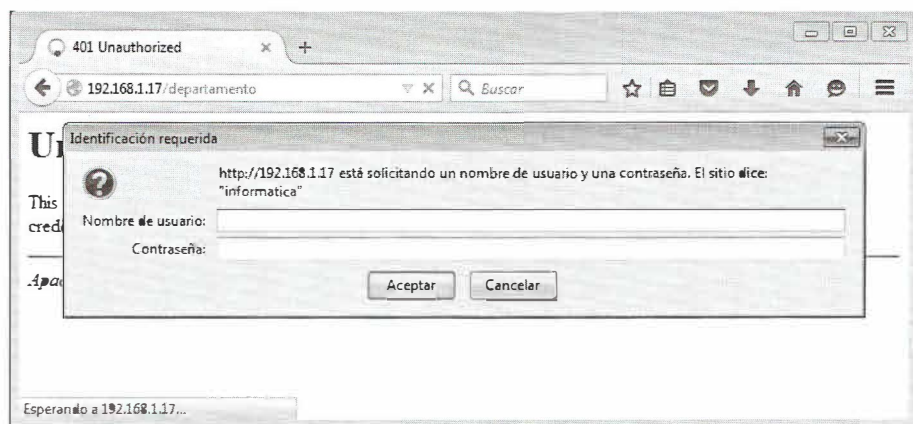


Figura 5.71: Conexión desde **DesarrolloW7XX**

1. Inicia una sesión en **ServidorLinuxXX** con un usuario con privilegios de administrador.
2. Crea el usuario **profesor**.

`sudo adduser profesor`
3. Edita el fichero de configuración `/etc/apache2/sites-available/000-default.conf`. Crea el alias y habilita el uso de ficheros `.htaccess` permitiendo sobrescribir todas las directivas en el directorio `/home/profesor/blog`. Elimina las directivas anteriores y añade la directiva `AllowOverride All`, Figura 5.72.

```
Alias /blog /home/profesor/blog
<Directory /home/profesor/blog>
    AllowOverride All
</Directory>
```

Figura 5.72: Permitir el uso de ficheros `.htaccess`

4. Reinicia el servidor para que los cambios tengan efecto.
5. Inicia una sesión en **ServidorLinuxXX** como usuario **profesor**.
6. Crea el directorio `/home/profesor/blog/`. Crea dentro el fichero `blog.html` con el contenido que quieras.
7. Crea el fichero `/home/profesor/blog/.htdigest` y añade al usuario `blog`.

`htdigest -c /home/profesor/blog/.htdigest informatica blog`
8. Crea el fichero `/home/profesor/blog/.htaccess` y añade las directivas para realizar la configuración pedida (no es necesario incluir la directiva `<Directory>` porque el fichero ya está en el directorio en el que se aplicará su configuración), Figura 5.73.
9. Desde **DesarrolloW7XX** accede a `http://192.168.1.X7/blog` para probar la configuración, Figuras 5.74 y 5.75.

```
Options Indexes
AuthType Digest
AuthName "informatica"
AuthDigestProvider file
AuthUserFile /home/profesor/blog/.htdigest
Require user blog
```

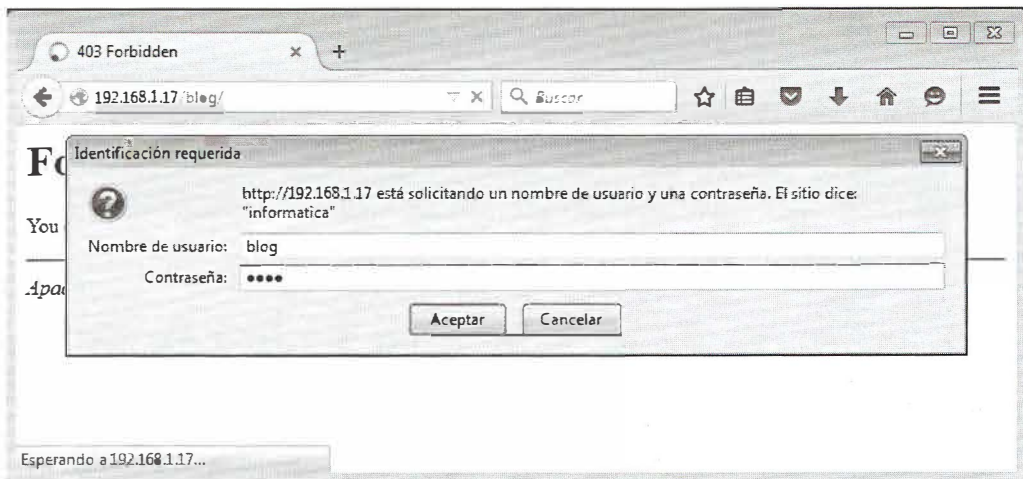
Figura 5.73: Fichero `.htaccess`

Figura 5.74: Conexión desde DesarrolloW7XX

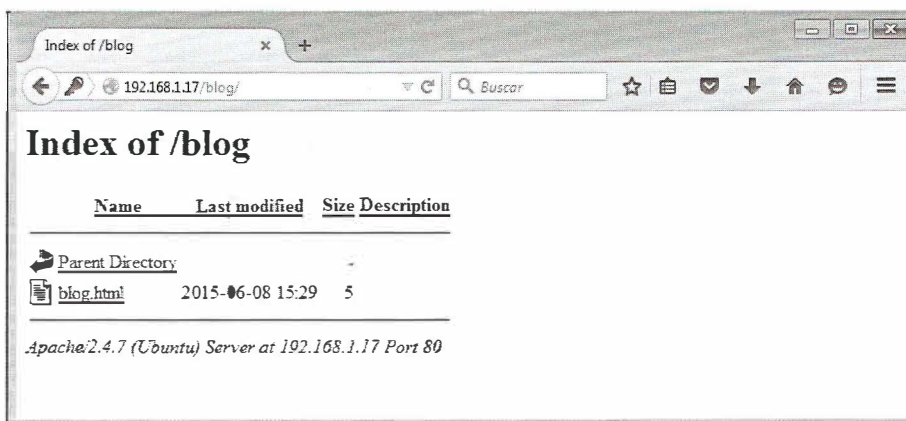


Figura 5.75: Conexión desde DesarrolloW7XX

5.12. Ficheros de registros (*logs*)

Sobre la máquina **ServidorLinuxXX** consulta los ficheros de configuración de *Apache* y observa las directivas utilizadas para definir la configuración de los ficheros de registros (*logs*). Posteriormente consulta estos ficheros.

1. Inicia una sesión en **ServidorLinuxXX** con un usuario con privilegios de administrador.
2. Consulta el archivo `/etc/apache2/sites-available/000-default.conf`.
3. ¿Cuál es el fichero de *logs* de errores (directiva `ErrorLog`) y cuál es su nivel de prioridad (directiva `LogLevel`)?, Figura 5.76.

```
ff Available loglevels: trace8, ..., trace!, debug, info, notice, warn,
ff error, crit, alert, emerg.
ff It is also possible to configure the loglevel for particular
ff modules, e.g.
LogLevel info ssl:warn

ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined
```

Figura 5.76: Configuración del fichero de *logs* de errores

4. ¿Cuál es el fichero de *logs* de accesos (directiva `CustomLog`) y cuál es su formato (como no especifica ningún formato con `LogFormat` se usa el definido para el servidor principal en el fichero `/etc/apache2/apache2.conf`)?, Figuras 5.77 y 5.78.

```
ff Available loglevels: trace8, ..., trace!, debug, info, notice, warn,
ff error, crit, alert, emerg.
ff It is also possible to configure the loglevel for particular
ff modules, e.g.
LogLevel info ssl:arn

ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined
```

Figura 5.77: Configuración del fichero de *logs* de accesos

5. Consulta el *log* de errores `/var/log/apache2/error.log`.
6. Consulta el *log* de accesos `/var/log/apache2/access.log`.

5.13. Módulos *mod_status* y *mod_info*

Sobre la máquina **ServidorLinuxXX** prueba la funcionalidad de los módulos *mod_status* y *mod_info*.

1. *mod_status*

Módulo que permite monitorizar el rendimiento del servidor *Apache*. Genera un documento en HTML con información sobre el estado actual del servidor.

```
#
# The following directives define some format nicknames for use with
# a CustomLog directive.
#
# These deviate from the Common Log Format definitions in that they use %D
# (the actual bytes sent including headers) instead of %b (the size of the
# requested file), because the latter makes it impossible to detect partial
# requests.
#
# Note that the use of %{X-Forwarded-For}i instead of %h is not recommended.
# Use mod_remoteip instead.
#
LogFormat "%v:%p %h %l %u %t \"%r\" %>s %D \"%{Referer}i\" \"%{User-Agent}i\"" %s
LogFormat "%h %l %u %t \"%r\" %>s %D \"%{Referer}i\" \"%{User-Agent}i\"" combin$
LogFormat "%h %l %u %t \"%r\" %>s %D" common
LogFormat "%{Referer}i -> %U" referer
LogFormat "%{User-agent}i" agent
```

Figura 5.78: Configuración del fichero de *logs* de accesos
(Fichero `/etc/apache2/apache2.conf`)

- 1.1. Inicia una sesión en **ServidorLinuxXX** con un usuario con privilegios de administración.
- 1.2. Habilita el módulo si no está habilitado.

```
sudo a2enmod status
```

- 1.3. Edita el fichero de configuración del módulo `/etc/apache2/mods-enabled/status.conf` y habilita el acceso a `/server-status` desde **DesarrolloW7XX**, Figura 5.79.

```
<Location /server-status>
    SetHandler server-status
    Require local
    Require ip 192.168.1.16
    #Require ip 192.0.2.0/24
</Location>
```

Figura 5.79: Fichero de configuración `status.conf`

- 1.4. Reinicia el servidor para aplicar los cambios.
- 1.5. Desde **DesarrolloW7XX** conéctate a `http://192.168.1.X7/server-status`, Figura 5.80.



Figura 5.80: Estado del servidor

- 1.6. También es posible consultar la información desde un terminal ejecutando el siguiente comando.

```
sudo apache2ctl status
```

2. mod_info

Módulo que proporciona una vista resumida de la configuración del servidor.

- 2.1. Abre un terminal y habilita el módulo.

```
sudo a2enmod info
```

- 2.2. Edita el fichero de configuración del módulo `/etc/apache2/mods-enabled/info.conf` y habilita el acceso a `/server-info` desde **DesarrolloW7XX**, Figura 5.81.

```
<Location /server-info>
    SetHandler server-info
    Require local
    Require ip 192.168.1.16
    #Require ip 192.0.2.0/24
</Location>
```

Figura 5.81: Fichero de configuración `info.conf`

- 2.3. Reinicia el servidor para aplicar los cambios.
- 2.4. Desde **DesarrolloW7XX** conéctate a `http://192.168.1.X7/server-info`, Figura 5.82.

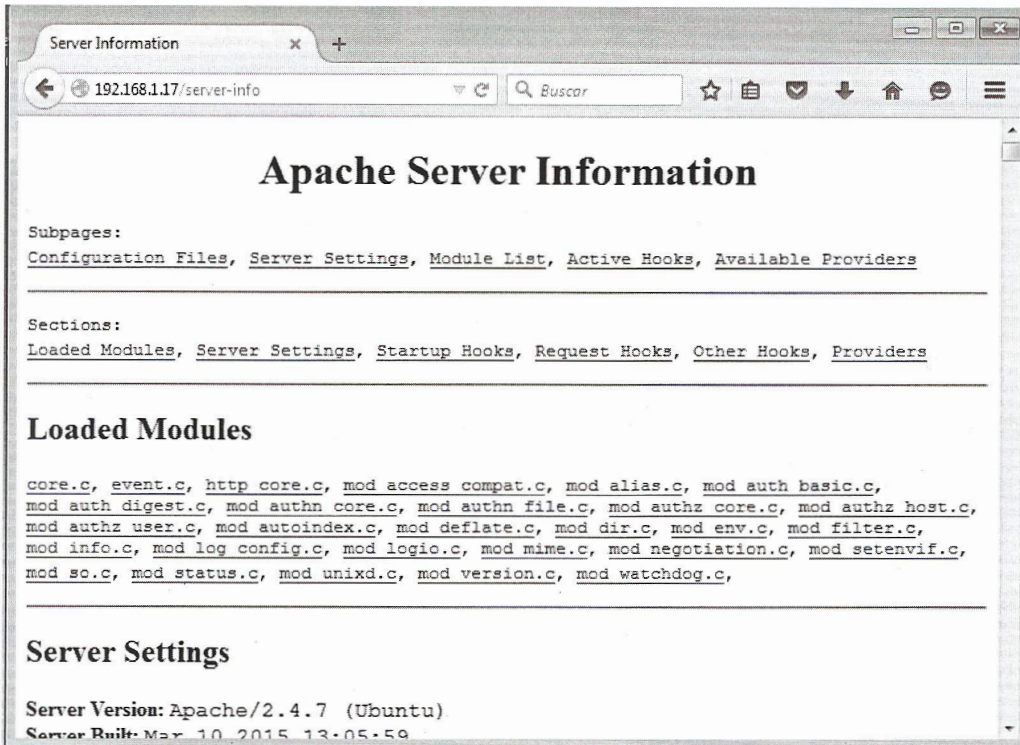


Figura 5.82: Información sobre la configuración del servidor

5.14. Webalizer

Sobre la máquina **ServidorLinuxXX** instala y prueba la funcionalidad de *Webalizer*.

1. Inicia una sesión en **ServidorLinuxXX** con un usuario con privilegios de administración.
2. Abre un terminal e instala el software.

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

3. Consulta el fichero de configuración `/etc/webalizer/webalizer.conf` y observa que se analizará el fichero de *logs* de accesos del servidor virtual por defecto. Quita la extensión `.1`, e indica `/var/www/html/webalizer` como directorio de salida, Figura 5.83.
4. Crea el directorio `/var/www/html/webalizer`.
5. Lanza el programa para que lea el fichero de *log* y genere el documento html con las estadísticas, Figura 5.84.

```
sudo webalizer
```

6. Desde **DesarrolloW7XX** conéctate a `http://192.168.1.X7/webalizer/index.html`, Figura 5.85.

Para que funcione (osea, que lea el fichero log), primero hay que acceder a alguna web para que cargue la información de acceso y esta pueda ser analizada por el Webalizer.

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

# LogType defines the log type being processed. Normally, the Webalizer
# expects a CLF or Combined web server log as input. Using this option,
# you can process ftp logs (xferlog as produced by wu-ftp and others),
# Squid native logs or W3C extended format web logs. Values can be 'clf',
# 'ftp', 'squid' or 'w3c'. The default is 'clf'.

#LogType          clf

# OutputDir is where you want to put the output files. This should
# should be a full path name, however relative ones might work as well.
# If no output directory is specified, the current directory will be used.

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

Figura 5.83: Fichero /etc/webalizer/webalizer.conf

```
alumno@ServidorLinux01:/var/www/webalizer$ sudo webalizer
Webalizer V2.23-08 (Linux 3.16.0-30-generic x86_64) locale: UTF-8
Utilizando histórico /var/log/apache2/access.log (clf)
Creando informe en /var/www/html/webalizer
El nombre de máquina en el informe es 'ServidorLinux01.daw01.net'
Leyendo archivo... webalizer.hist
Generando informe de Junio 2015
Guardando información de archivo...
Generando informe resumido
312 registros en 1 segundos, 312/sec
alumno@ServidorLinux01:/var/www/webalizer$
```

Figura 5.84: Ejecución de *webalizer*

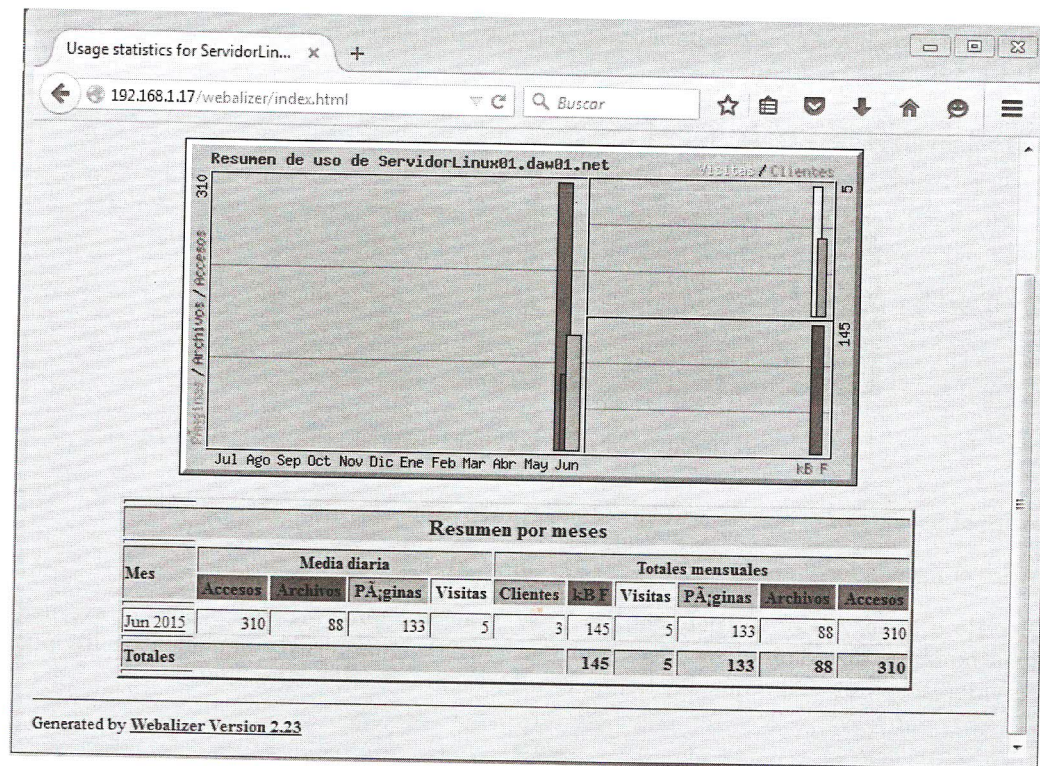


Figura 5.85: Estadísticas de *webalizer*