

4.1. Protocolo HTTP

En esta práctica se analiza la información de los mensajes de petición y respuesta del protocolo HTTP.

1. Inicia sesión con un usuario con privilegios de administrador en **DesarrolloW7XX**.
2. Abre el navegador *Google Chrome*.
3. Inicia una captura con *Wireshark* (Inicio, Todos los programas, *Wireshark, Capture, Interfaces, Start*).
4. Desde el navegador *Google Chrome* establece una conexión a `http://www.apache.org/`.
5. Accede a *Wireshark* y para la captura (*Capture, Stop*).
6. Buscar una trama HTTP en donde la petición sea GET / HTTP/1.1, véase Figura 4.1.



Figura 4.1: *Wireshark*

7. Haz clic con el botón derecho del ratón y selecciona **Follow TCP Stream**, véase Figura 4.2.
8. Responde a las siguientes preguntas:
 - 8.1. ¿Cuál es la IP de la máquina donde se ejecuta el servidor Web? 140.211.11.131.
 - 8.2. ¿Qué versión de HTTP se utiliza? HTTP 1.1.
 - 8.3. ¿Qué método de petición se utiliza? GET.
 - 8.4. ¿Qué recurso se solicita al servidor? El directorio raíz.
 - 8.5. ¿Qué valor tiene la cabecera Host? www.apache.org
 - 8.6. ¿Se envían cookies en la petición HTTP? No.
 - 8.7. ¿Qué lenguaje utiliza el navegador? es-ES.
 - 8.8. ¿Qué código de estado tiene la respuesta HTTP? 200.
 - 8.9. ¿Qué servidor Web y versión se utiliza? Apache/2.4.1 (Unix).
 - 8.10. ¿De qué tipo MIME es el recurso recibido? text/html.
 - 8.11. ¿Se han utilizado conexiones persistentes, es decir, en la misma conexión TCP hay varias peticiones y respuestas HTTP? Sí.

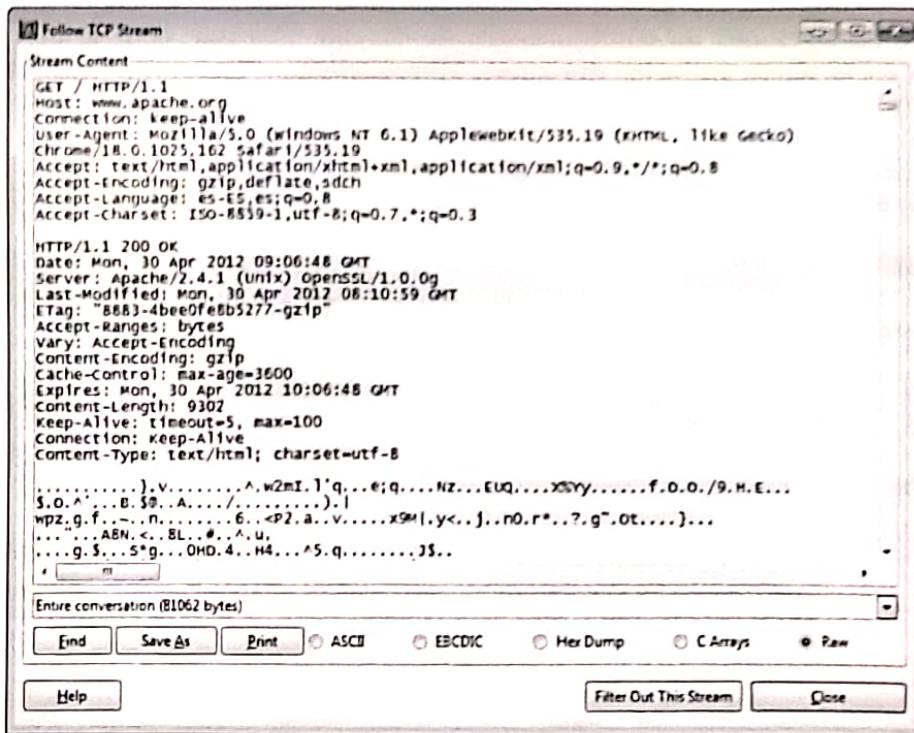


Figura 4.2: Tráfico HTTP

8.12. ¿Existen peticiones y respuestas de imágenes? Sí, véase Figura 4.3.

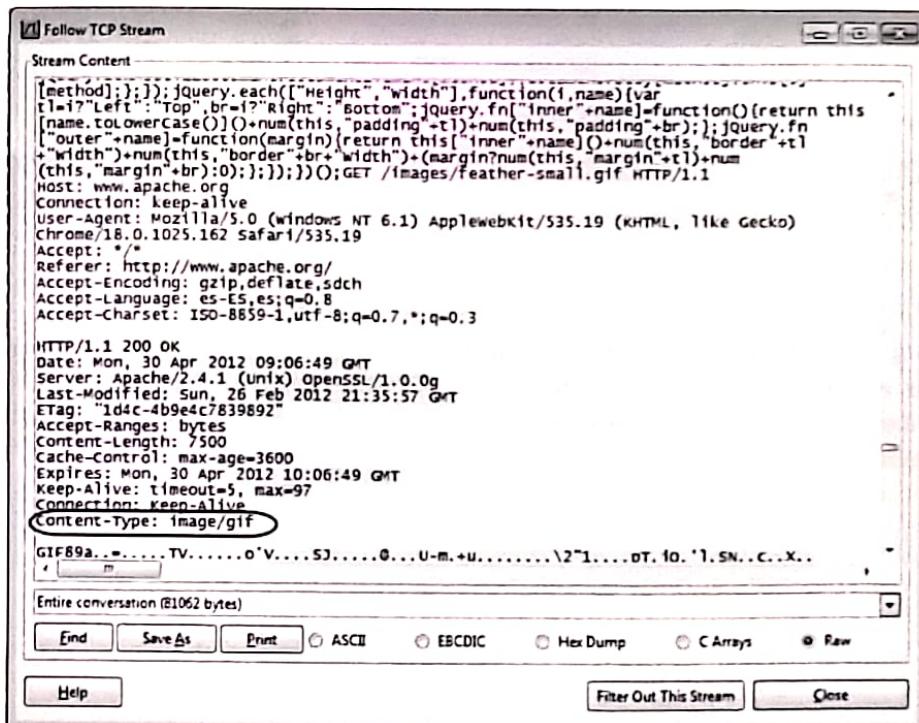
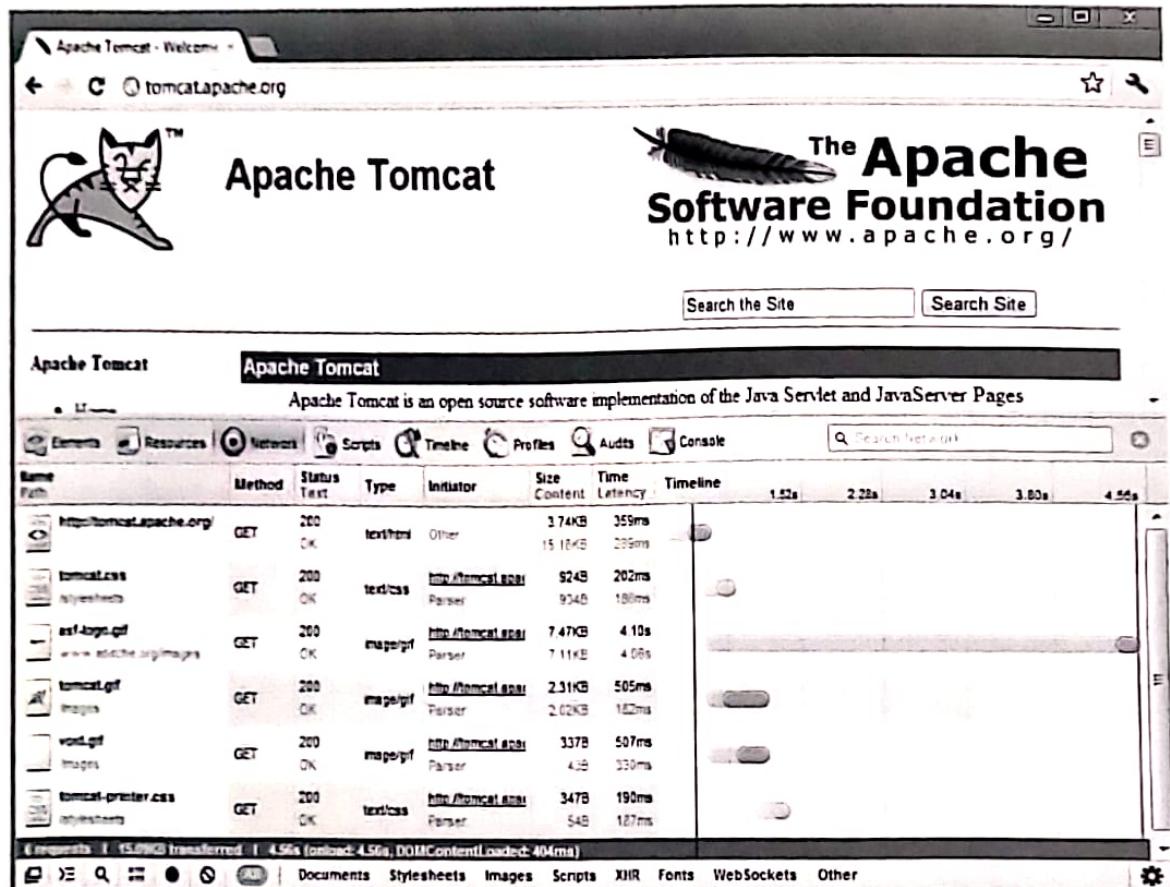


Figura 4.3: Envío de una imagen

9. Accede a las opciones de configuración de *Google Chrome* (botón con una herramienta en la parte superior derecha), Herramientas, Herramientas para desarrolladores.
10. Accede a <http://tomcat.apache.org/> y analiza la peticiones y respuestas HTTP, que métodos usan, los códigos de respuesta, los recursos que envía el servidor, ..., Figura 4.4.

Figura 4.4: Herramienta para desarrolladores de *Google Chrome*

11. Accede a las opciones de configuración de *Google Chrome* (botón con una herramienta en la parte superior derecha), Configuración, Avanzada, Configuración de contenido, Todas las cookies y los datos de sitios y observa las *cookies* que tiene almacenadas el navegador.

Administración de Apache

5.1. Instalación del servidor web *Apache 2.4* en *Linux*

Instala el servidor web *Apache 2.4* en la máquina **ServidorLinuxXX** desde los repositorios de software de *Ubuntu*. Prueba el funcionamiento del servidor estableciendo conexiones desde **DesarrolloW7XX** empleando tanto direcciones IP como nombres de dominio.

1. Instalación

- 1.1. Inicia sesión con un usuario con privilegios de administrador en **ServidorLinuxXX**.
- 1.2. Instala el servidor *Apache 2.4* desde los repositorios oficiales de *Ubuntu*.

```
sudo apt-get update
sudo apt-get install apache2
```

- Se crean los archivos de configuración.
- Se crean el usuario **www-data** que se añade al grupo **www-data**. Son el usuario y el grupo con el que se ejecutan los procesos de *Apache* que se encargan de atender peticiones.
- Se crea el directorio **/var/www/html**.
 - Su propietario es **root** y su grupo es **root**.
 - Es el directorio raíz del servidor virtual por defecto.

- 1.3. Comprueba que el servidor está iniciado y escuchando en el puerto 80/TCP, Figura 5.1.

```
ps -ef | grep apache
netstat -ltn
```

```
aumno@ServidorLinux01:~$ ps -ef | grep apache
root     1840     1  0 12:33 ?        00:00:00 /usr/sbin/apache2 -k start
www-data 1843  1840  0 12:33 ?        00:00:00 /usr/sbin/apache2 -k start
www-data 1846  1840  0 12:33 ?        00:00:00 /usr/sbin/apache2 -k start
alumno   1965    966  0 12:33 ttys1    00:00:00 grep --color=auto apache
alumno@ServidorLinux01:~$ netstat -ltn
Conexiones activas de Internet (solo servidores)
Proto Recib Enviad Dirección local           Dirección remota         Estado
tcp6      0      0 :::80                      ::::*                  ESCUCHAR
alumno@ServidorLinux01:~$
```

Figura 5.1: *Apache* iniciado y escuchando en el puerto 80/TCP

- 1.4. Comprueba que se ha creado el directorio **/var/www/html** y que su propietario es el usuario **root**.

```
ls -l /var
```

- 1.5. Ejecuta el siguiente comando para ver las opciones con las que se ha compilado la versión de *Apache* instalada.

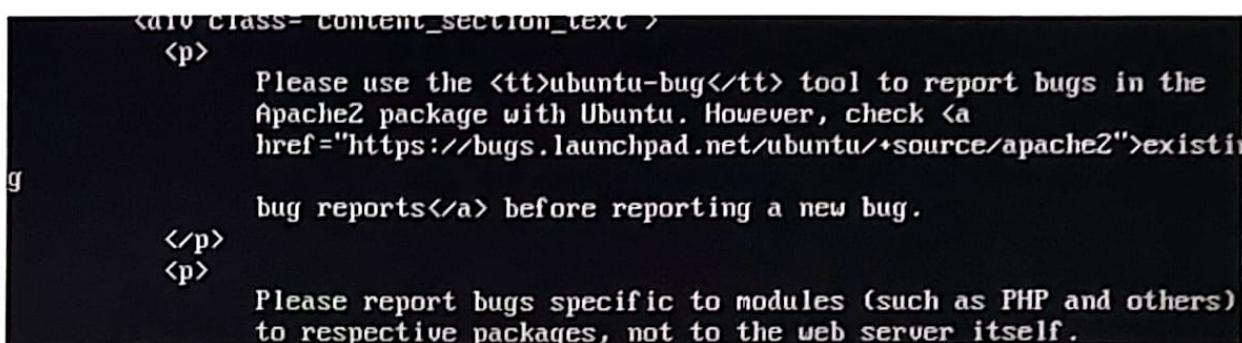
```
apache2 -v
```

- 1.6. Consulta el contenido del directorio **/var/www/html**, Figura 5.2.

- 1.7. Consulta el contenido del fichero **/var/www/html/index.html**, Figura 5.3.

```
alumno@ServidorLinux01:/var/www/html$ ls
index.html
alumno@ServidorLinux01:/var/www/html$
```

Figura 5.2: Directorio /var/www/html



```
<div class="content_section_text">
<p>
Please use the <tt>ubuntu-bug</tt> tool to report bugs in the
Apache2 package with Ubuntu. However, check <a
href="https://bugs.launchpad.net/ubuntu/+source/apache2">existing
bug reports</a> before reporting a new bug.
</p>
<p>
Please report bugs specific to modules (such as PHP and others)
to respective packages, not to the web server itself.
</p>
```

Figura 5.3: Fichero /var/www/html/index.html

2. Prueba de conexión al servidor

- 2.1. Inicia sesión DesarrolloW7XX.
- 2.2. Abre un navegador y accede a <http://192.168.1.X7> (como no se especifica el puerto se utiliza por defecto el puerto 80), Figura 5.4.

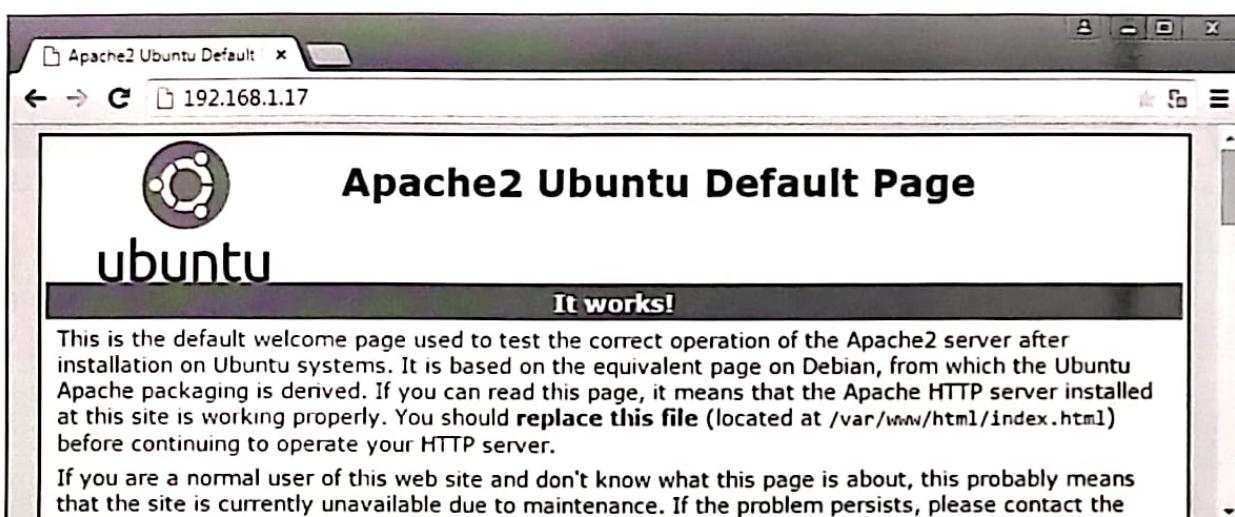


Figura 5.4: Conexión a Apache

- 2.3. Observa que Apache sirve por defecto el fichero **index.html** (que está en el directorio `/var/www/html`).
- 2.4. También puedes usar los nombres DNS `servidorlinuxXX.dawXX.net` y `obelix.dawXX.net`, véanse Figuras 5.5 y 5.6.

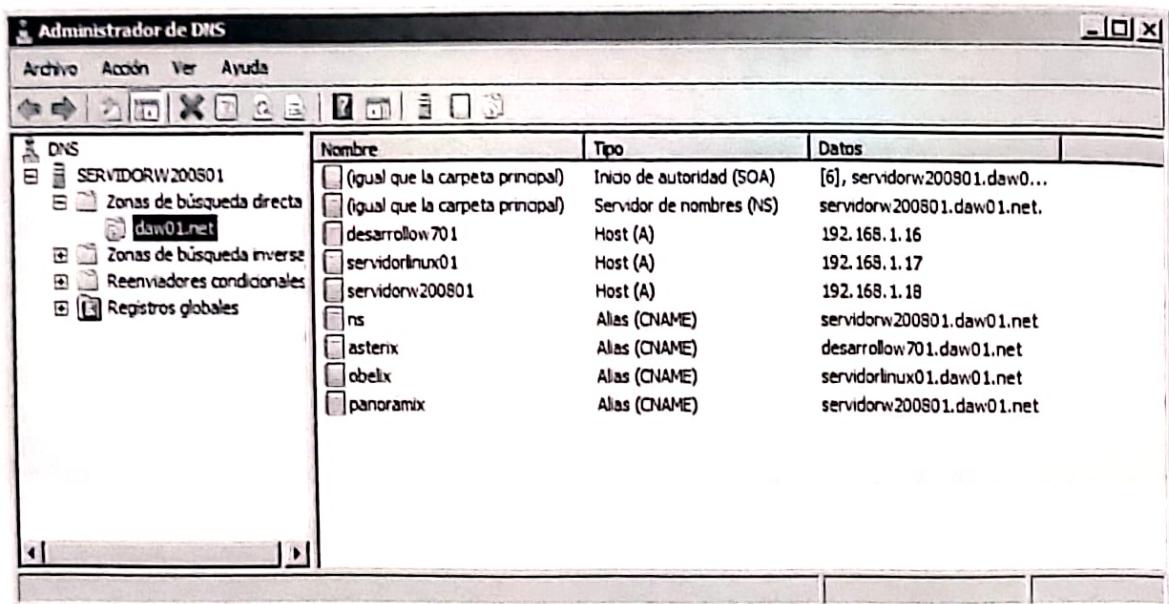


Figura 5.5: Zona dawXX.net

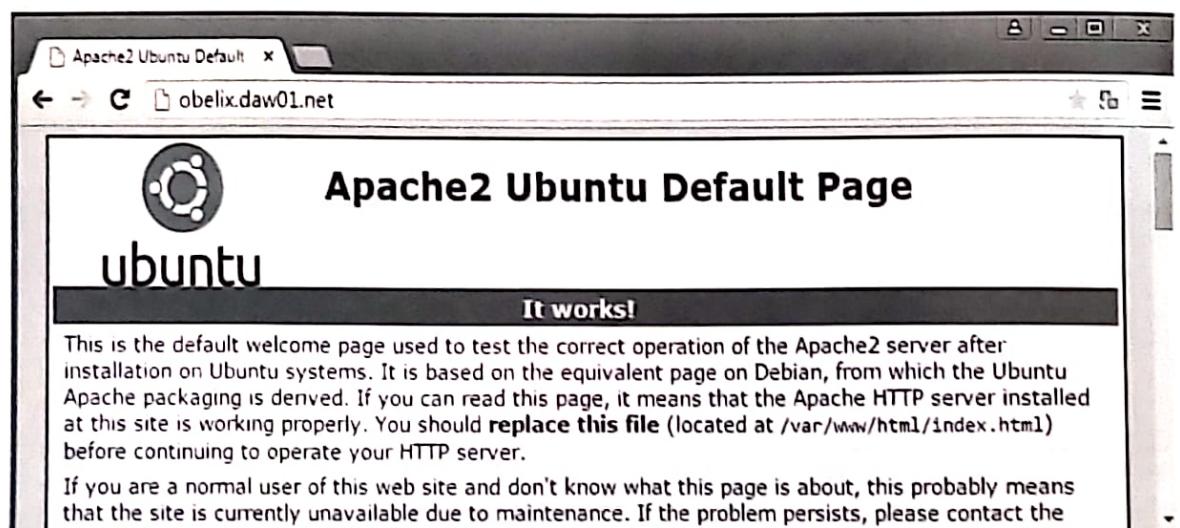


Figura 5.6: Conexión a Apache

5.2. Instalación del servidor web *Apache 2.2* en *Windows*

Instala el servidor web *Apache 2.2* en la máquina **ServidorW2008XX** o **ServidorW2012XX** usando los paquetes binarios. Prueba el funcionamiento del servidor estableciendo conexiones desde **DesarrolloW7XX** empleando tanto direcciones IP como nombres de dominio.

1. Instalación

- 1.1. Inicia sesión con un usuario con privilegios de administrador en **ServidorW2008XX**.
- 1.2. Inicia el navegador *Google Chrome* y accede a <http://httpd.apache.org/download.cgi>, Figura 5.7.

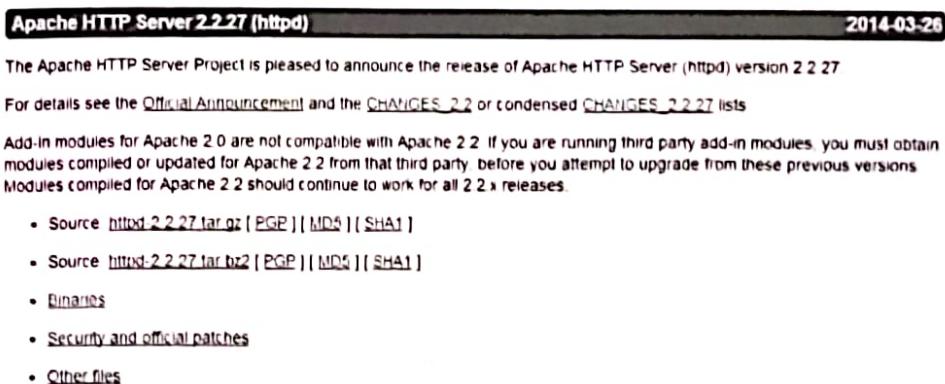
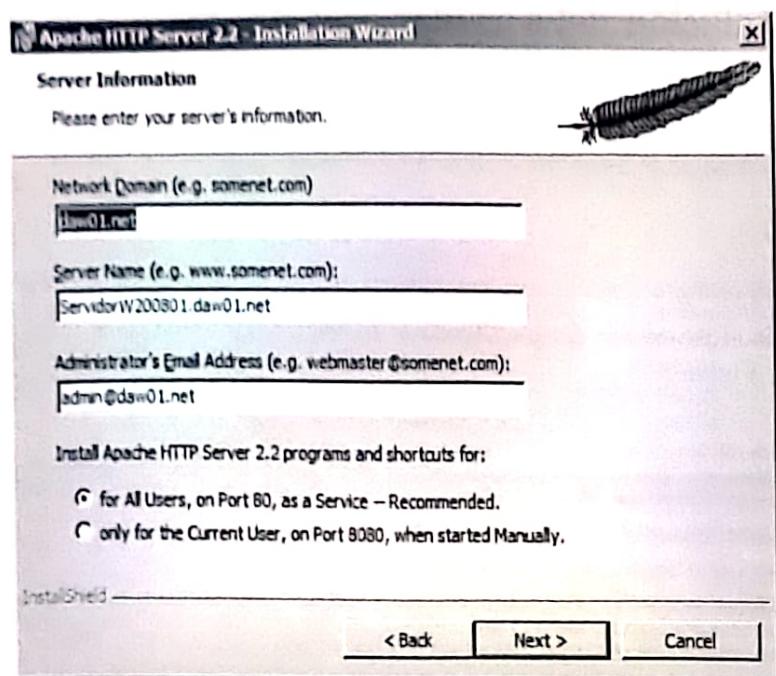
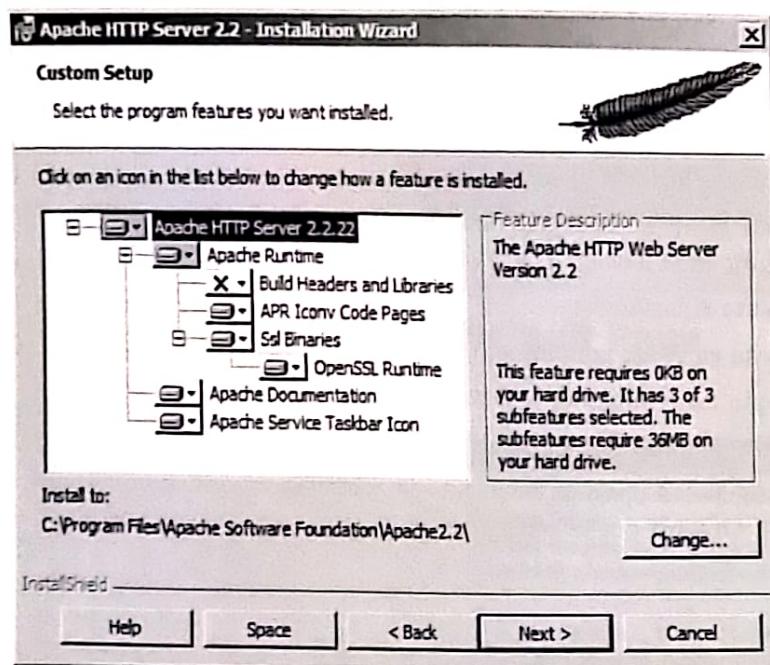


Figura 5.7: Descarga de *Apache* para *Windows*

- 1.3. Accede a ***Other Files***.
- 1.4. Accede a ***binaries*** y luego a ***win32***.
- 1.5. Descarga el archivo para *win32* con OpenSSL (<http://httpd-2.2.25-win32-x86-openssl-0.9.8y> en el momento de probar la práctica).
- 1.6. Ejecuta el instalador.
- 1.7. Pincha en ***Next*** para iniciar el asistente de instalación.
- 1.8. Acepta los términos de la licencia y pincha en ***Next***.
- 1.9. Lee la información y pincha en ***Next***.
- 1.10. Configura las opciones tal y como se muestran en la Figura 5.8 (adáptalo a tu grupo XX). Observa que se configurará el servidor para que escuche peticiones en el puerto 80.

Figura 5.8: Instalación de Apache (*Server Information*)

- 1.11. Selecciona el tipo de instalación *Custom* y pincha en *Next*.
- 1.12. Observa qué componentes se instalarán, Figura 5.9, y pincha en *Next*.

Figura 5.9: Instalación de Apache (*Custom Setup*)

- 1.13. Pincha en *Install* para continuar.
- 1.14. Una vez terminada la instalación pincha en *Finish*.

- 1.15. Comprueba que el servidor está iniciado, en la barra de tareas abajo a la izquierda hay un icono que muestra una herramienta para monitorizar el servidor. Figura 5.10.

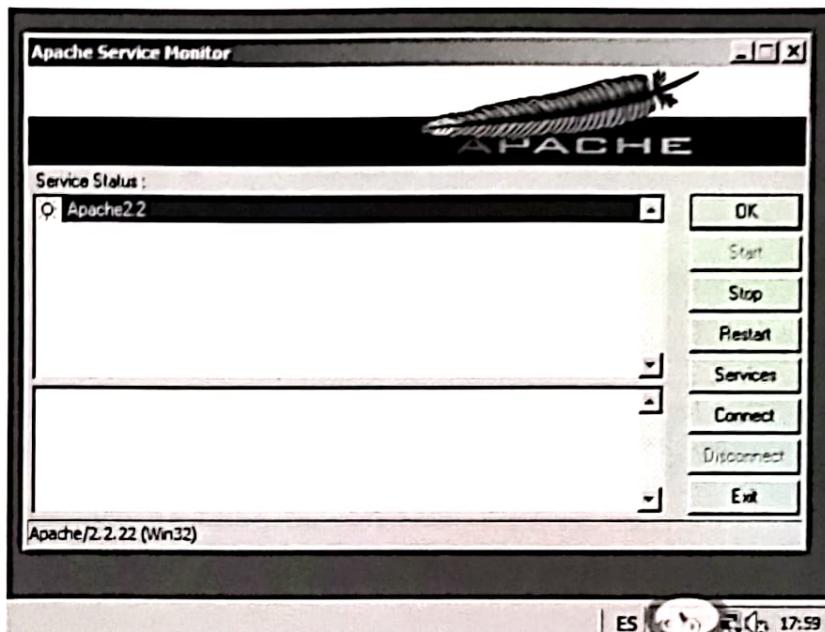


Figura 5.10: Monitorización del servidor

- 1.16. Comprueba que el servidor está escuchando en el puerto 80/TCP, Figura 5.11

```
netstat -a -p TCP -n
```

The screenshot shows a command prompt window titled "Administrador: Símbolo del sistema". It displays the output of the command "netstat -a -p TCP -n". The output lists active connections with columns for Proto, Dirección local, Dirección remota, and Estado. Most entries show "0.0.0.0" for both local and remote addresses, indicating they are listening ports. The "Estado" column shows all entries as "LISTENING".

Proto	Dirección local	Dirección remota	Estado
TCP	0.0.0.0:80	0.0.0.0:0	LISTENING
TCP	0.0.0.0:135	0.0.0.0:0	LISTENING
TCP	0.0.0.0:445	0.0.0.0:0	LISTENING
TCP	0.0.0.0:5357	0.0.0.0:0	LISTENING
TCP	0.0.0.0:49152	0.0.0.0:0	LISTENING
TCP	0.0.0.0:49153	0.0.0.0:0	LISTENING
TCP	0.0.0.0:49154	0.0.0.0:0	LISTENING
TCP	0.0.0.0:49155	0.0.0.0:0	LISTENING
TCP	0.0.0.0:49156	0.0.0.0:0	LISTENING
TCP	0.0.0.0:49157	0.0.0.0:0	LISTENING
TCP	127.0.0.1:53	0.0.0.0:0	LISTENING
TCP	192.168.1.18:53	0.0.0.0:0	LISTENING
TCP	192.168.1.18:139	0.0.0.0:0	LISTENING

Figura 5.11: Apache iniciado y escuchando en el puerto 80/TCP

- 1.17. Consulta el contenido del directorio C:\Program Files\Apache Software Foundation\Apache2.2\htdocs, Figura 5.12.

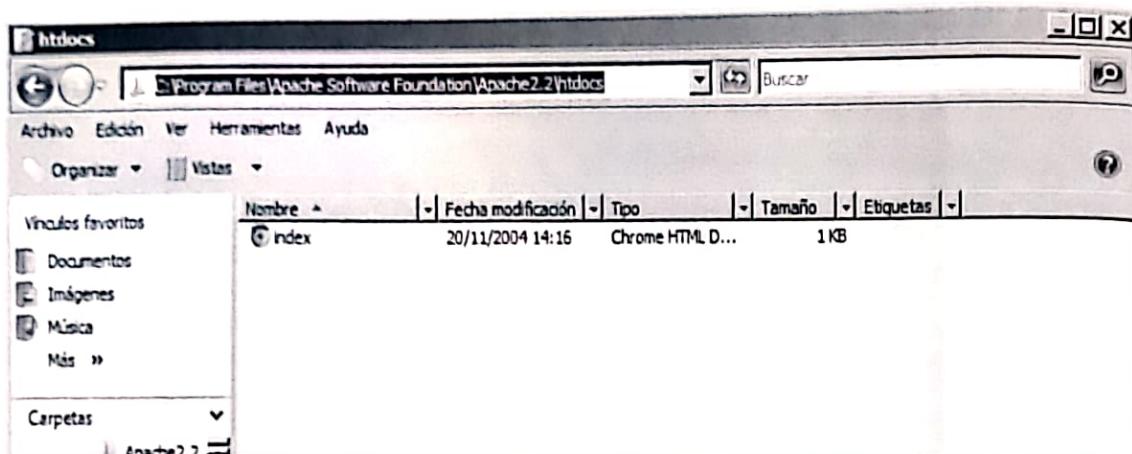


Figura 5.12: Directorio C:\Program Files\Apache Software Foundation\Apache2.2\htdocs

- 1.18. Consulta el contenido del fichero C:\Program Files\Apache Software Foundation\Apache2.2\index.html, Figura 5.13.

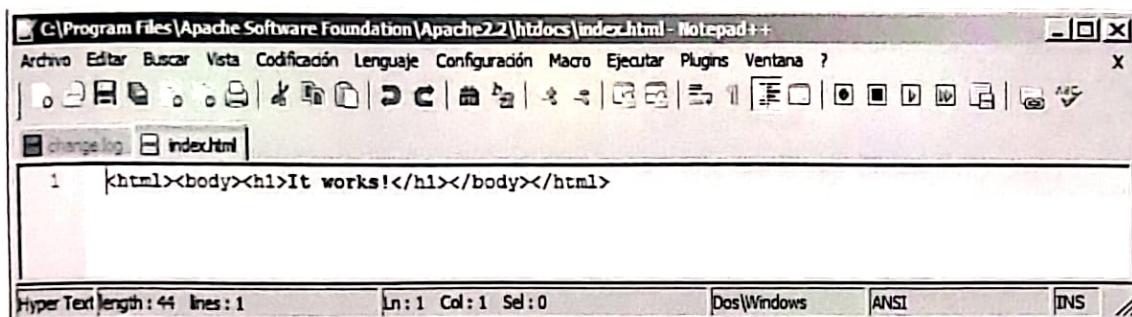


Figura 5.13: Fichero index.html

- 1.19. Habilita una regla en el *Firewall* para que se permitan conexiones a *Apache*.
- Accede a Inicio, Panel de control, Seguridad, Dejar pasar un programa a través de *firewall* de *Windows*.
 - Pincha en Agregar Programa.
 - Pincha en Examina y selecciona C:\Program Files\Apache Software Foundation\Apache2.2\bin\httpd.exe.
 - Acepta los cambios.
- 1.20. Si queremos utilizar las aplicaciones de *Apache* desde la cualquier localización en la línea de comandos podemos añadir la ruta de los binarios de *Apache* en la variable de entorno **PATH**.
- Accede a Inicio, Panel de control, Sistema y mantenimiento, Sistema, Configuración avanzada del sistema, Variables de Entorno.
 - En Variables de sistema selecciona la variable Path y pincha en Editar.
 - Añade la ruta C:\Program Files\Apache Software Foundation\Apache2.2\bin, véase Figura 5.14.
 - Acepta los cambios.

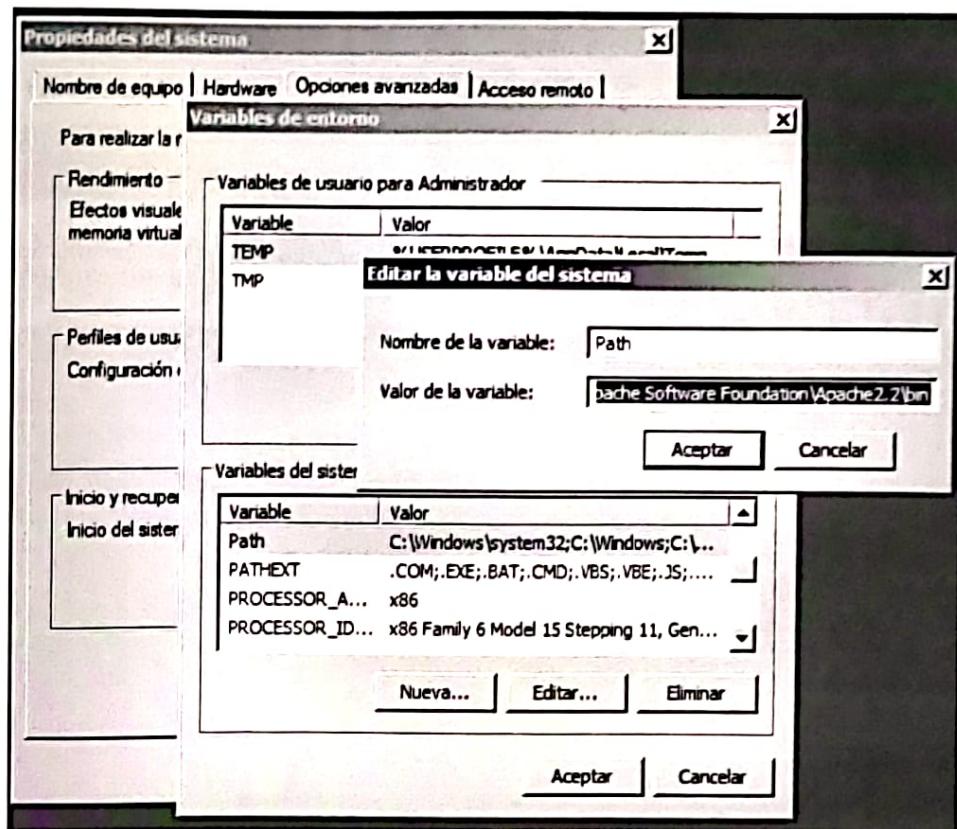


Figura 5.14: Variable de entorno Path

2. Prueba de conexión al servidor

- 2.1. Abre un navegador y accede a `http://127.0.0.1` o a `http://localhost` (como no se especifica el puerto se utiliza por defecto el puerto 80), Figura 5.15.

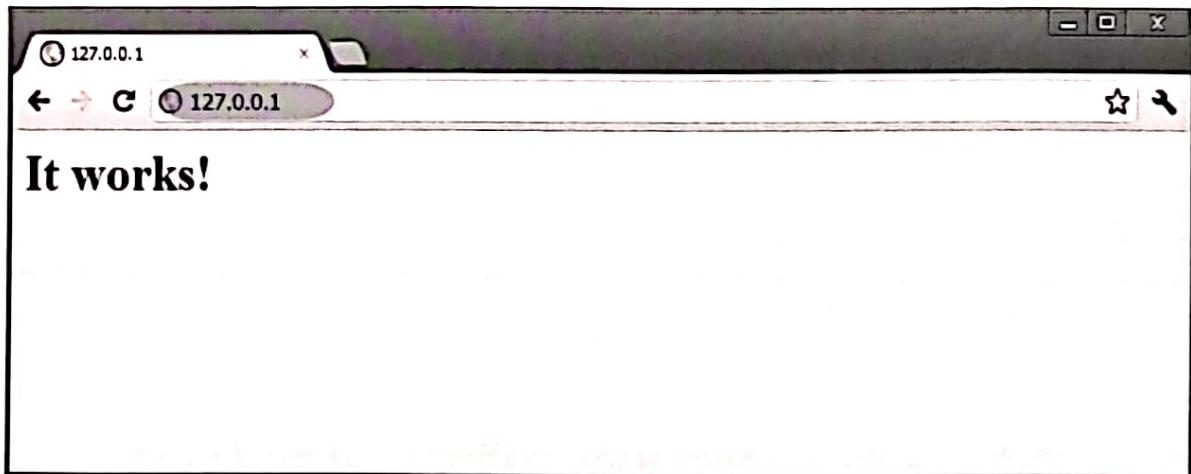


Figura 5.15: Conexión a Apache

- 2.2. Inicia sesión DesarrolloW7XX.

- 2.3. Abre un navegador y accede a `http://192.168.1.X8` (como no se especifica el puerto se utiliza por defecto el puerto 80), Figura 5.16.



Figura 5.16: Conexión a *Apache*

2.4. Observa que *Apache* sirve por defecto el fichero `index.html`.

2.5. También puedes usar los nombres DNS `servidorlinuxXX.dawXX.net` y `panoramix.dawXX.net`, Figura 5.17.



Figura 5.17: Conexión a *Apache*

5.3. Ficheros de configuración y directivas en *Linux*

En esta práctica analizaremos los principales ficheros de configuración y el valor de algunas directivas del servidor web *Apache2.4* instalado en la máquina **ServidorLinuxXX**.

1. Ficheros de configuración

- 1.1. Inicia sesión con un usuario con privilegios de administrador en **ServidorLinuxXX**.
- 1.2. Abre un terminal y accede al directorio **/etc/apache2**.
- 1.3. Haz un listado del directorio y observa los ficheros de configuración.
- 1.4. Abre el fichero **/etc/apache2/apache2.conf** y analiza su configuración. Observa que incluye con la directiva **include** a otros ficheros y directorios, Figura 5.18.

```
LogFormat "%h %l %u %t \"%r\" %>s %O \"%{Referer}i\" \"%{User-Agent}i\"\" combi
LogFormat "%h %l %u %t \"%r\" %>s %O" common
LogFormat "%{Referer}i -> %U" referer
LogFormat "%{User-agent}i" agent

# Include of directories ignores editors' and dpkg's backup files,
# see README.Debian for details.

# Include generic snippets of statements
IncludeOptional conf-enabled/*.conf

# Include the virtual host configurations:
IncludeOptional sites-enabled/*.conf

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

Figura 5.18: Fichero **/etc/apache2/apache2.conf**

2. Servidor virtual por defecto

- 2.1. Accede al directorio **/etc/apache2/sites-available** y comprueba que está creado el archivo **default** que contiene la configuración del servidor virtual por defecto.
- 2.2. Accede a **/etc/apache2/sites-enabled** y comprueba que existe el fichero **000-default.conf** que es un enlace simbólico a **default**, Figura 5.19 (recuerda que los ficheros incluidos en **/etc/apache2/sites-enabled** se incluyen en **/etc/apache2/apache2.conf** en orden alfabético).

```
alumno@ServidorLinux01:/etc/apache2/sites-enabled$ ls -l
total 0
lrwxrwxrwx 1 root root 35 jun  8 12:33 000-default.conf -> ../../sites-available/000-default.conf
alumno@ServidorLinux01:/etc/apache2/sites-enabled$
```

Figura 5.19: Fichero **/etc/apache2/sites-enabled/000-default.conf**

3. Directivas

- 3.1. Consulta el fichero **/etc/apache2/apache2.conf** y comprueba cuál es el valor de las siguientes directivas.

- **ServerRoot**
- **User y Group**
- **TimeOut**

- 3.2. Consulta la documentación de *Apache* y responde a las siguientes preguntas:
- ¿Se permiten conexiones persistentes (que todas las conexiones de un usuario se atienden en la misma conexión TCP)? Si ¿Qué directiva define este comportamiento? `KeepAlive`.
 - ¿Cuál es el fichero de errores? `/var/log/apache2/error.log` ¿Qué directiva lo define? `ErrorLog`.
- 3.3. Consulta el fichero `/etc/apache2/ports.conf`, Figura 5.20 y comprueba cuál es el puerto en el que escucha peticiones *Apache* (*puerto 80*) ¿En qué puerto escuchará también si se habilita el módulo modssl? `443`.

```
# If you just change the port or add more ports here, you will likely also
# have to change the VirtualHost statement in
# /etc/apache2/sites-enabled/000-default.conf

Listen 80

<IfModule ssl_module>
    Listen 443
</IfModule>

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

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

Figura 5.20: Contenido del fichero `/etc/apache2/ports.conf`

- 3.4. Consulta el fichero `/etc/apache2/sites-available/default` y observa, Figura 5.21.

- Dentro de la directiva `<VirtualHost> ... </VirtualHost>` se define el comportamiento del servidor virtual por defecto.
- El valor de la directiva `DocumentRoot` es `/var/www/html`.
- El valor de la directiva `ErrorLog`.

- 3.5. Consulta el fichero `/etc/apache2/apache.conf` y observa, Figura 5.22.

- La directiva contenedora `<Directory> ... </Directory>` que se utiliza para determinar cómo *Apache* sirve el contenido del directorio `/var/www`.

```
<VirtualHost *:80>
    # The ServerName directive sets the request scheme, hostname and port t
    # the server uses to identify itself. This is used when creating
    # redirection URLs. In the context of virtual hosts, the ServerName
    # specifies what hostname must appear in the request's Host: header to
    # match this virtual host. For the default virtual host (this file) thi
    # value is not decisive as it is used as a last resort host regardless.
    # However, you must set it for any further virtual host explicitly.
    #ServerName www.example.com

    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html

    # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
    # error, crit, alert, emerg.
    # It is also possible to configure the loglevel for particular
    # modules, e.g.
    #LogLevel info ssl:warn

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

    # For most configuration files from conf-available/, which are
    # enabled or disabled at a global level, it is possible to
    # include a line for only one particular virtual host. For example the
```

Figura 5.21: Contenido del fichero /etc/apache2/sites-available/default

```

# access here, or in any related virtual host.
<Directory />
    Options FollowSymLinks
    AllowOverride None
    Require all denied
</Directory>

<Directory /usr/share>
    AllowOverride None
    Require all granted
</Directory>

<Directory /var/www/>
    Options Indexes FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>

<Directory /srv/>
    Options Indexes FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>
```

Figura 5.22: Contenido del fichero /etc/apache2/apache2.conf

5.4. Ficheros de configuración y directivas en *Windows*

En esta práctica analizaremos los principales ficheros de configuración y el valor de algunas directivas del servidor web *Apache2.2* instalado en la máquina **ServidorW2008XX** o **ServidorW2012XX**.

1. Ficheros de configuración

- 1.1. Inicia sesión con un usuario con privilegios de administrador en **ServidorW2008XX/2012XX**.
- 1.2. Accede al directorio **C:\Program Files\Apache Software Foundation\Apache2.2\conf**.
- 1.3. Observa los ficheros de configuración.
- 1.4. Abre el fichero **C:\Program Files\Apache Software Foundation\Apache2.2\conf\httpd.conf** y analiza su configuración.

2. Directivas

- 2.1. Consulta el fichero **C:\Program Files\Apache Software Foundation\Apache2.2\conf\httpd.conf**
- 2.2. Comprueba cuál es el valor de la directiva **ServerRoot**.
- 2.3. Comprueba cuál es el puerto en el que escucha peticiones *Apache* (*puerto 80*), Figura 5.23.

```

httpd - Bloc de notas
Archivo Edición Formato Ver Ayuda
# ServerRoot at a non-local disk, be sure to point the LockFile directive
# at a local disk. If you wish to share the same ServerRoot for multiple
# httpd daemons, you will need to change at least LockFile and Pidfile.
#
ServerRoot "C:/Program Files/Apache Software Foundation/Apache2.2"

#
# Listen: Allows you to bind Apache to specific IP addresses and/or
# ports, instead of the default. See also the <VirtualHost>
# directive.
#
# Change this to Listen on specific IP addresses as shown below to
# prevent Apache from glomming onto all bound IP addresses.
#
#Listen 12.34.56.78:80
Listen 80

#
# Dynamic Shared Object (DSO) Support
#
# To be able to use the functionality of a module which was built as a DSO you
# have to place corresponding 'LoadModule' lines at this location so the
# directives contained in it are actually available _before_ they are used.
# statically compiled modules (those listed by 'httpd -l') do not need

```

Figura 5.23: *ServerRoot* y *Listen*

- 2.4. El valor de la directiva DocumentRoot es C:\Program Files\Apache Software Foundation\Apache2.2\htdocs, Figura 5.24.

```

httpd - Bloc de notas
Archivo Edición Formato Ver Ayuda
#ServerName Servidorw200801.daw01.net:80

#
# DocumentRoot: The directory out of which you will serve your
# documents. By default, all requests are taken from this directory, but
# symbolic links and aliases may be used to point to other locations.
#
DocumentRoot "C:/Program Files/Apache Software Foundation/Apache2.2/htdocs"

#
# Each directory to which Apache has access can be configured with respect
# to which services and features are allowed and/or disabled in that
# directory (and its subdirectories).
#
# First, we configure the "default" to be a very restrictive set of
# features.

```

Figura 5.24: *DocumentRoot*

- 2.5. La directiva contenedora <Directory> ...</Directory> que se utiliza para determinar cómo Apache sirve el contenido del directorio C:\Program Files\Apache Software Foundation\Apache2.2\htdocs, Figura 5.25.

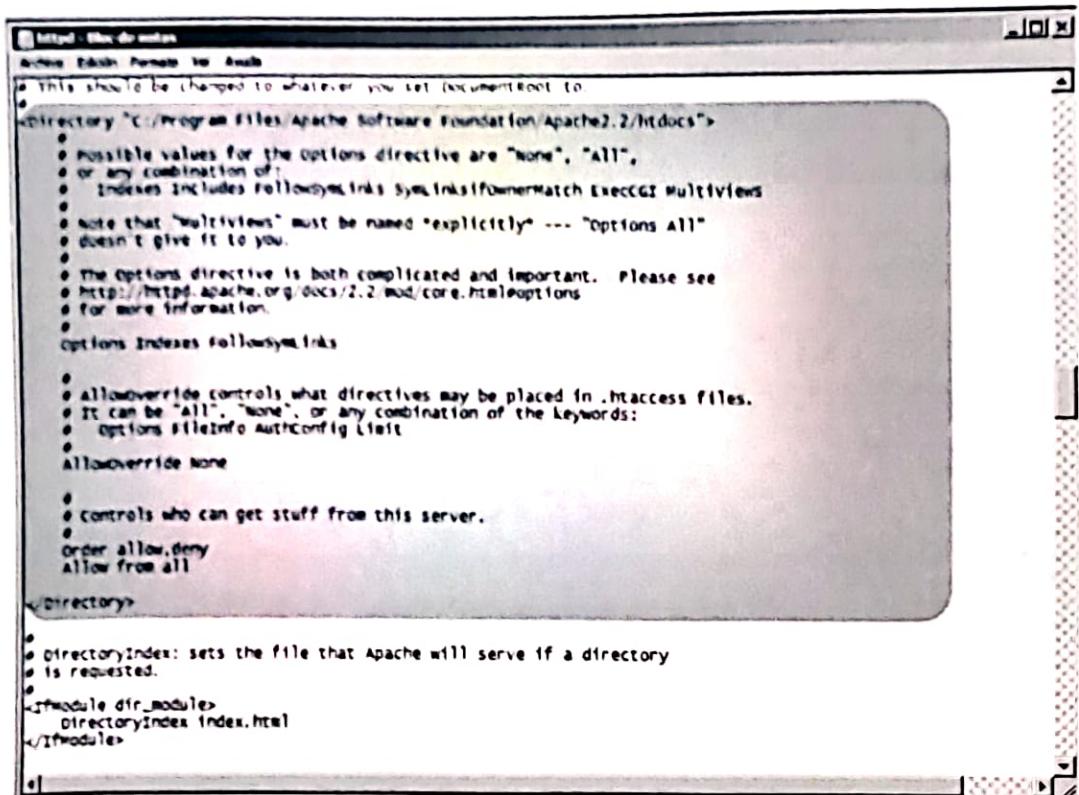


Figura 5.25: *Directory*

2.6. Observa el valor de la directiva ErrorLog, Figura 5.26.

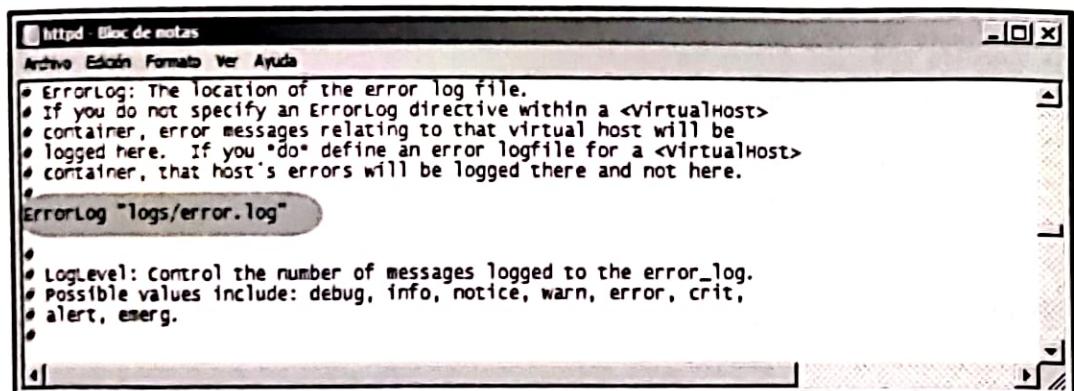


Figura 5.26: *ErrorLog*

2.7. Observa que existen varias directivas include comentadas para incluir nuevas funcionalidades, Figura 5.27.

2.8. Consulta los ficheros C:\Program Files\Apache Software Foundation\Apache2.2\conf\extra\httpd-default y C:\Program Files\Apache Software Foundation\Apache2.2\conf\extra\httpd-vhosts para ver cómo podrías modificar directivas por defecto y añadir servidores virtuales, Figuras 5.28 y 5.29.

```
# server-pool management (MPM specific)
#include conf/extra/httpd-mpm.conf

# Multi-language error messages
#include conf/extra/httpd-multilang-errordoc.conf

# Fancy directory listings
#include conf/extra/httpd-autoindex.conf

# Language settings
#include conf/extra/httpd-languages.conf

# user home directories
#include conf/extra/httpd-userdir.conf

# Real-time info on requests and configuration
#include conf/extra/httpd-info.conf

# virtual hosts
#include conf/extra/httpd-vhosts.conf

# Local access to the Apache HTTP Server Manual
#include conf/extra/httpd-manual.conf

# Distributed authoring and versioning (WebDAV)
#include conf/extra/httpd-dav.conf

# various default settings
#include conf/extra/httpd-default.conf

# Secure (SSL/TLS) connections
#include conf/extra/httpd-ssl.conf
```

Figura 5.27: *Include*

```
# This configuration file reflects default settings for Apache HTTP Server.
# You may change these, but chances are that you may not need to.

#
# Timeout: The number of seconds before receives and sends time out.
# Timeout 300

#
# KeepAlive: whether or not to allow persistent connections (more than
# one request per connection). Set to "Off" to deactivate.
# KeepAlive On

#
# MaxKeepAliveRequests: The maximum number of requests to allow
# during a persistent connection. Set to 0 to allow an unlimited amount.
# We recommend you leave this number high, for maximum performance.
# MaxKeepAliveRequests 100

#
# KeepAliveTimeout: Number of seconds to wait for the next request from the
# same client on the same connection.
# KeepAliveTimeout 5

#
# useCanonicalName: Determines how Apache constructs self-referencing
# URLs and the SERVER_NAME and SERVER_PORT variables.
```

Figura 5.28: Fichero httpd-default

```

httpd-vhosts - Bloc de notas
Archivo Edición Formato Ver Ayuda
NameVirtualHost *:80

# virtualHost example:
# Almost any Apache directive may go into a VirtualHost container.
# The first VirtualHost section is used for all requests that do not
# match a ServerName or ServerAlias in any <VirtualHost> block.

<VirtualHost *:80>
    ServerAdmin webmaster@dummy-host.daw01.net
    DocumentRoot "C:/Program Files/Apache Software Foundation/Apache2.2/docs/dummy
    ServerName dummy-host.daw01.net
    ServerAlias www(dummy-host.daw01.net
    ErrorLog "logs/dummy-host.daw01.net-error.log"
    CustomLog "logs/dummy-host.daw01.net-access.log" common
</VirtualHost>

<VirtualHost *:80>
    ServerAdmin webmaster@dummy-host2.daw01.net
    DocumentRoot "C:/Program Files/Apache Software Foundation/Apache2.2/docs/dummy
    ServerName dummy-host2.daw01.net
    ErrorLog "logs/dummy-host2.daw01.net-error.log"
    CustomLog "logs/dummy-host2.daw01.net-access.log" common
</VirtualHost>

```

Figura 5.29: Fichero httpd-vhosts

5.5. Configuración básica en *Linux*

En esta práctica realizaremos varias pruebas de configuración sobre el servidor web *Apache 2.4* instalado en la máquina ServidorLinuxXX. Usaremos las directivas: `DirectoryIndex`, `<Directory> ... </Directory>`, `Options Indexes`, `ErrorDocument`, `Alias` y `Redirect`.

1. Ficheros y directorios de prueba

- 1.1. Inicia sesión con un usuario con privilegios de administrador en ServidorLinuxXX.
- 1.2. Accede al directorio `/var/www/html` y crea los siguientes ficheros y directorios con el contenido que quieras, Figura 5.30.
 - `/var/www/html/despliegue.html`
 - `/var/www/html/fp.html`
 - `/var/www/html/ciclos/listado.html`
 - `/var/www/html/ciclos/asir.html`
 - `/var/www/html/ciclos/daw.html`
 - `/var/www/html/ciclos/dam.html`
- 1.3. Desde DesarrolloW7XX abre un navegador y establece las siguientes conexiones:
 - `http://192.168.1.X7`
 - `http://192.168.1.X7/despliegue.html`
 - `http://192.168.1.X7/ciclos`
 - `http://192.168.1.X7/ciclos/listado.html`

```
alumno@ServidorLinux01:~$ cd /var/www/html/
alumno@ServidorLinux01:/var/www/html$ ls -l
total 24
drwxr-xr-x 2 root root 4096 jun  8 13:23 ciclos
-rw-r--r-- 1 root root    26 jun  8 13:21 despliegue.html
-rw-r--r-- 1 root root    12 jun  8 13:22 fp.html
-rw-r--r-- 1 root root 11510 jun  8 12:33 index.html
alumno@ServidorLinux01:/var/www/html$ cd ciclos/
alumno@ServidorLinux01:/var/www/html/ciclos$ ls -l
total 16
-rw-r--r-- 1 root root   5 jun  8 13:22 asir.html
-rw-r--r-- 1 root root   4 jun  8 13:22 dam.html
-rw-r--r-- 1 root root   4 jun  8 13:22 daw.html
-rw-r--r-- 1 root root  13 jun  8 13:22 listado.html
alumno@ServidorLinux01:/var/www/html/ciclos$ _
```

Figura 5.30: Ficheros y directorios de prueba

2. Ficheros a servir por defecto (*Directory Index*)

- 2.1. Desde DesarrolloW7XX abre un navegador y establece una conexión a **http://192.168.1.X7**.
- 2.2. En la URL no se ha pedido ningún recurso en concreto. El servidor ha enviado por defecto **index.html** (valor de la directiva *DirectoryIndex* por defecto).
- 2.3. Renombra el fichero **/var/www/html/index.html** a **/var/www/html/indice.html**.
- 2.4. Desde DesarrolloW7XX abre un navegador y establece una conexión a **http://192.168.1.X7** (pulsa F5 para refrescar). Como no se ha encontrado **index.html** en el directorio raíz **/var/www/html** se muestra el contenido del directorio, Figura 5.31.

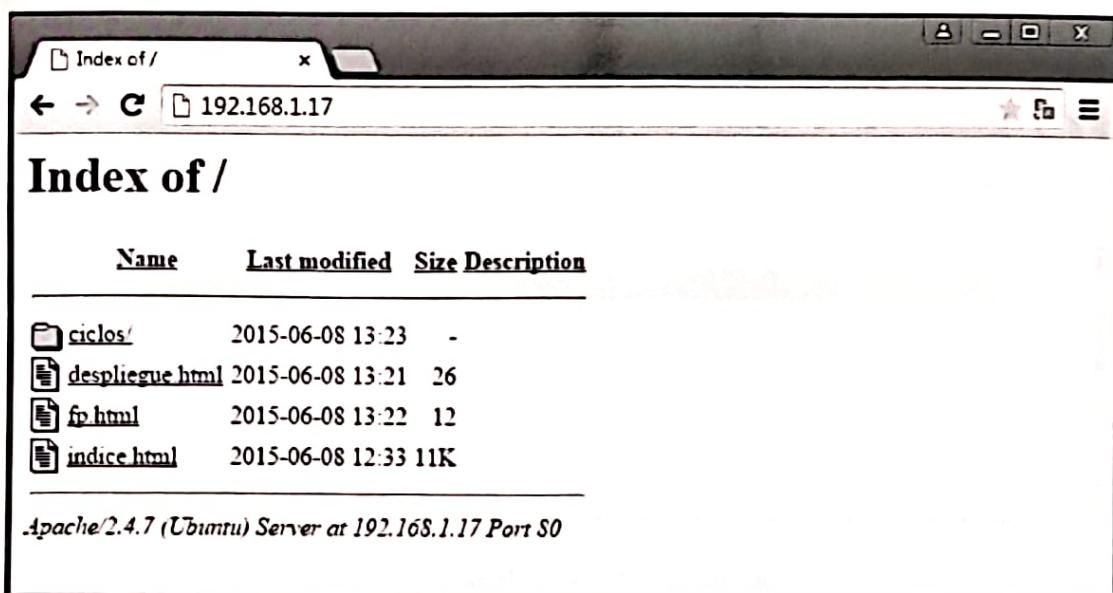


Figura 5.31: Listado del directorio raíz del servidor

- 2.5. Edita el archivo `/etc/apache2/sites-available/000-default.conf` y añade las siguientes directiva. Sección
`<Directory /var/www/html> ... </Directory>` que incluye la directiva `Directory` con el valor `despliegue.html` tal y como se muestra en la Figura 5.32.

```
# For most configuration files from conf-available/, which are
# enabled or disabled at a global level, it is possible to
# include a line for only one particular virtual host. For example the
# following line enables the CGI configuration for this host only
# after it has been globally disabled with "a2disconf".
#Include conf-available/serve-cgi-bin.conf

<Directory /var/www/html>
    DirectoryIndex despliegue.html
    Options Indexes FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>

</VirtualHost>
```

Figura 5.32: `DirectoryIndex`

- 2.6. Reinicia el servidor para que los cambios tengan efecto.

```
sudo service apache2 stop
sudo service apache2 start
```

- 2.7. Desde DesarrolloW7XX abre un navegador y establece una conexión a `http://192.168.1.X7`. Ahora se sirve el fichero `despliegue.html`, Figura 5.33.

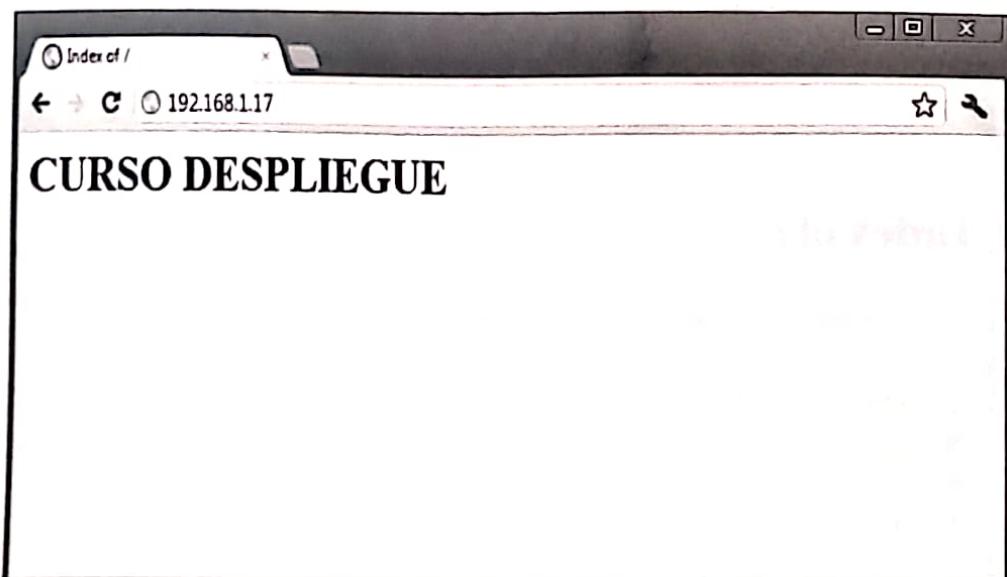


Figura 5.33: Fichero `despliegue.html`

3. Opciones sobre directorios (`Directory` y `Options Indexes`)

- 3.1. Edita el archivo `/etc/apache2/sites-available/000-default.conf` y observa que la directiva `<Directory> ... </Directory>` contiene las directivas que determinan cómo

Apache sirve el contenido de ese directorio. Todos los directorios que estén dentro de `/var/www/html` heredan su configuración, y `/var/www` hereda y sobrescribe la configuración del directorio raíz (/) (la configuración del raíz está definida en el fichero `apache2.conf`). Figura 5.34 .

```
<Directory /var/www/html>
    DirectoryIndex despliegue.html
    Options Indexes FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>

</VirtualHost>
```

Figura 5.34: *Directory*

- 3.2. Desde DesarrolloW7XX abre un navegador y establece una conexión a `http://192.168.1.X7/ciclos`. Dentro de `/var/www/html/ciclos` (ha heredado la configuración de `/var/www/html`) no existe el fichero `despliegue.html` (`DirectoryIndex`) y por eso se muestra su contenido.
- 3.3. Crea una nueva directiva `<Directory> ... </Directory>` para `/var/www/ciclos` tal y como se muestra en la Figura 5.35.

```
<Directory /var/www/html>
    DirectoryIndex despliegue.html
    Options Indexes FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>

<Directory /var/www/html/ciclos>
    Options FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>
```

Figura 5.35: *Directory* y opción *Indexes*

- 3.4. Observa que para `/var/www/html/ciclos` no se ha definido la opción `Indexes`. Cuando aparece esta opción, el servidor lista el contenido del directorio si no encuentra los ficheros definidos en `DirectoryIndex`, y si no aparece no se muestra el contenido del directorio indicando un mensaje de prohibición.
- 3.5. Reinicia el servidor para que los cambios tengan efecto.

```
sudo service apache2 stop
sudo service apache2 start
```

- 3.6. Desde DesarrolloW7XX abre un navegador y establece una conexión a `http://192.168.1.X7/`. Se muestra `despliegue.html`.

- 3.7. Desde DesarrolloW7XX abre un navegador y establece una conexión a `http://192.168.1.X7/ciclos`. Como en `/var/www/html/ciclos` no existe `despliegue.html` y no se permite el listado del directorio (`Options Indexes`) el servidor retorna el código **403 Forbidden**, Figura 5.36.

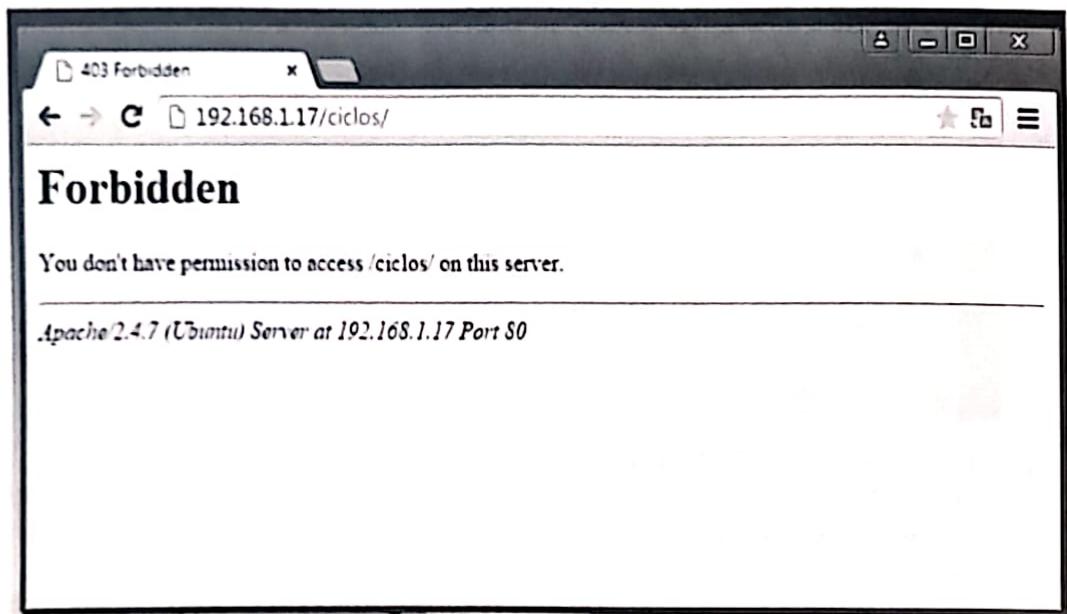


Figura 5.36: *DirectoryIndex* no encontrado en `/var/www/html/ciclos`

- 3.8. Desde DesarrolloW7XX abre un navegador y establece una conexión a `http://192.168.1.X7/ciclos/listado.html` y verifica que es posible acceder, Figura 5.37.

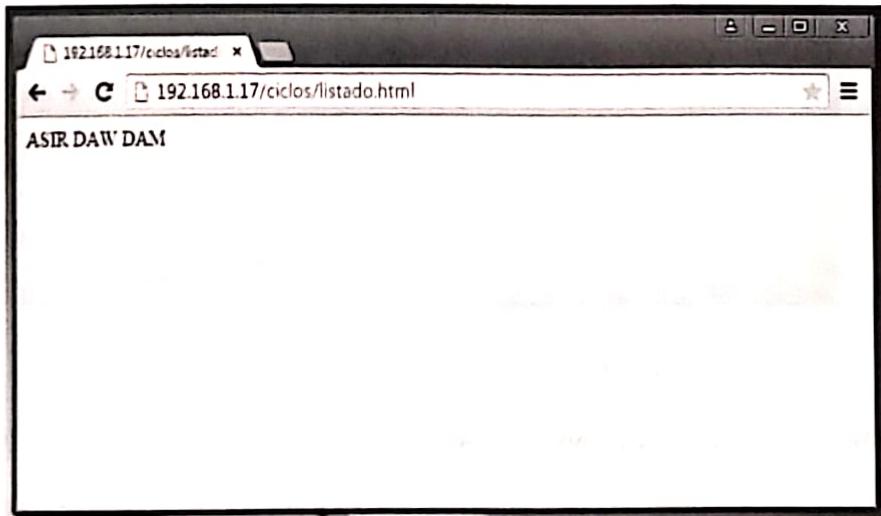


Figura 5.37: Acceso a un fichero dentro del directorio `/var/www/html/ciclos`

4. Códigos del error (*ErrorDocument*)

- 4.1. Configura el servidor virtual por defecto para que cuando retorne el código de error 404 (página no encontrada) envíe el texto “Página no encontrada en el servidor de la red

dawXX.net". Edita el archivo `/etc/apache2/sites-available/000-default.conf` tal y como se muestra en la Figura 5.38.

```
<Directory /var/www/html/ciclos>
    Options FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>

    ErrorDocument 404 "Página no encontrada en la red daw01.net"

</VirtualHost>
```

Figura 5.38: *ErrorDocument*

- 4.2. Reinicia el servidor para que los cambios tengan efecto.
- 4.3. Desde DesarrolloW7XX abre un navegador y establece una conexión a `http://192.168.1.X7/noexiste.html`, Figura 5.39.



Figura 5.39: Error 404

- 4.4. Configura el servidor virtual por defecto para que cuando retorne el código de error 404 (página no encontrada) envíe la página 404.html almacenada en el directorio raíz del servidor, Figura 5.40. Crea el fichero `/var/www/html/404.html` con el contenido que quieras.
 - 4.5. Reinicia el servidor para que los cambios tengan efecto.
 - 4.6. Desde DesarrolloW7XX abre un navegador y establece una conexión a `http://192.168.1.X7/noexiste.html`.
- ## 5. Directorios Virtuales (*Alias*)
- 5.1. Crea el directorio `/home/alumno/apuntes`. Dentro del directorio crea un fichero denominado `apuntes.html` con el contenido que quieras.

```
ErrorLog ${APACHE_LOG_DIR}/error.log
# Possible values include: debug, info, notice, warn, error, crit,
# alert, emerg.
LogLevel warn

CustomLog ${APACHE_LOG_DIR}/access.log combined

ErrorDocument 404 /404.html
```

Figura 5.40: Error 404

- 5.2. Edita el archivo `/etc/apache2/sites-available/000-default.conf`. Utiliza la directiva `Alias` para crear un directorio virtual denominado `/apuntes` que referencia a `/home/alumno/apuntes`. Usa la directiva `Directory` para definir las opciones de configuración del directorio `/home/alumno/apuntes`, Figura 5.41.

```
<Directory /var/www/html/ciclos>
    Options FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>

Alias /apuntes /home/alumno/apuntes
<Directory /home/alumno/apuntes>
    Options Indexes FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>

ErrorDocument 404 /404.html

</VirtualHost>
```

Figura 5.41: Directorios virtuales (*Alias*)

- 5.3. Reinicia el servidor para que los cambios tengan efecto.
5.4. Desde **DesarrolloW7XX** abre un navegador y establece una conexión a `http://192.168.1.X7/apuntes`, Figura 5.42.

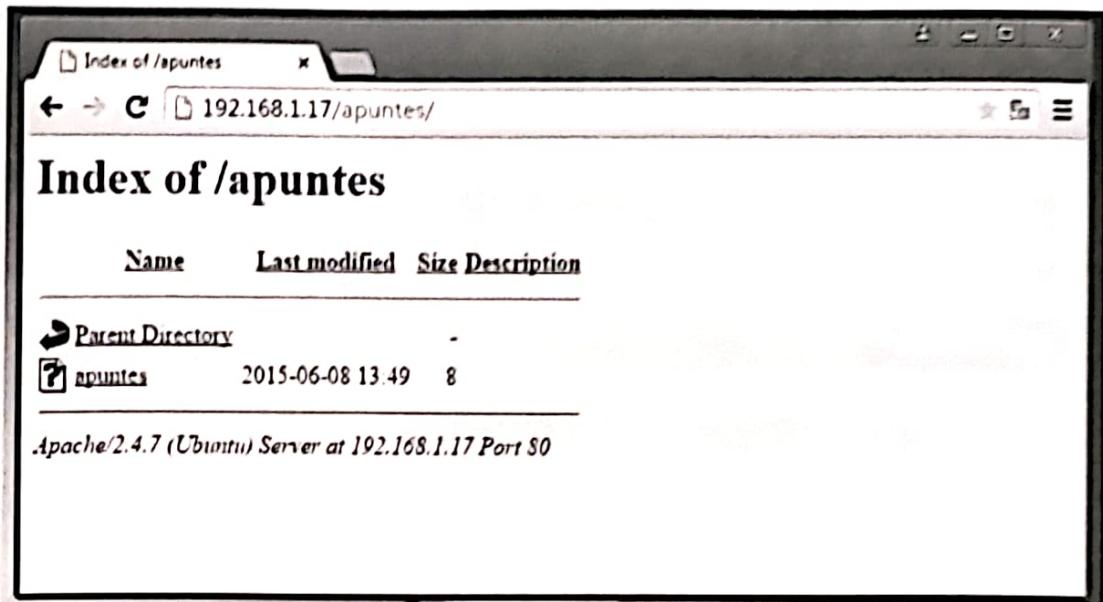


Figura 5.42: Acceso al directorio virtual apuntes

6. Redirecciones (*Redirect*)

- 6.1. Edita el archivo `/etc/apache2/sites-available/default`. Utiliza la directiva *Redirect* para crear una redirección de `/fp` hacia a `http://www.todosfp.es`, Figura 5.43.

```
Alias /apuntes /home/alumno/apuntes
<Directory /home/alumno/apuntes>
    Options Indexes FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>

Redirect /fp http://www.todosfp.es_

ErrorDocument 404 /404.html
```

Figura 5.43: Redirección (*Redirect*)

- 6.2. Reinicia el servidor para que los cambios tengan efecto.
 6.3. Desde DesarrolloW7XX abre un navegador y establece una conexión a `http://192.168.1.X7/fp`, Figura 5.44.

Figura 5.44: Redirección de <http://192.168.1.x7/fp> a <http://www.todofp.es>

5.6. Configuración básica en Windows

Replica la configuración de *Apache* que has realizado en la práctica anterior sobre el servidor Apache de la máquina ServidorW2008XX o ServidorW2012XX.

1. Ficheros y directorios de prueba

- 1.1. Inicia sesión con un usuario con privilegios de administrador en **ServidorW2008XX/2012XX**.
- 1.2. Accede al directorio **C:\Program Files\Apache Software Foundation\Apache2.2\htdocs** y crea los siguientes ficheros y directorios con el contenido que quieras, Figura 5.45.

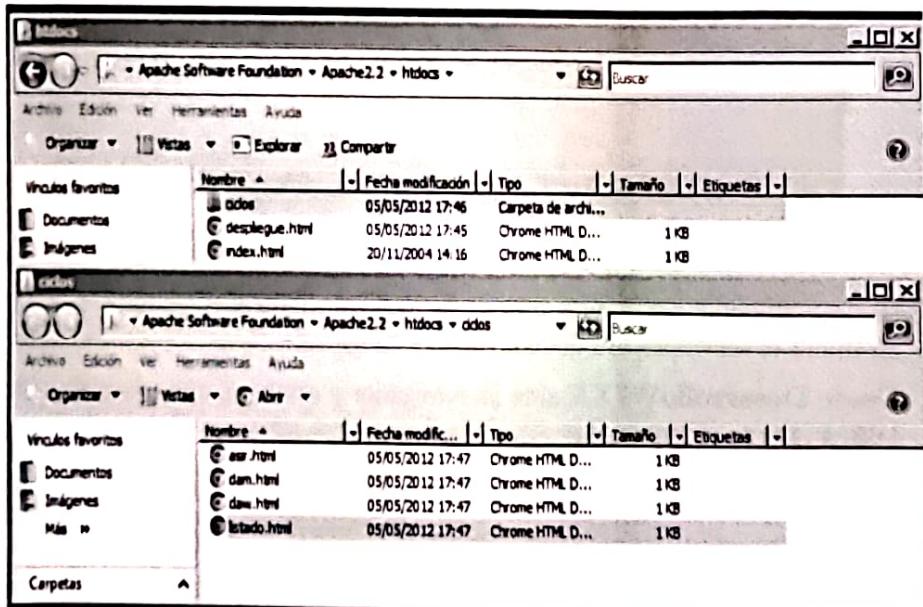


Figura 5.45: Ficheros y directorios de prueba

- ..\htdocs\despligue.html
- ..\htdocs\fp.html
- ..\htdocs\ciclos\listado.html
- ..\htdocs\ciclos\asir.html
- ..\htdocs\ciclos\daw.html
- ..\htdocs\ciclos\dam.html

1.3. Crear el fichero C:\Usuarios\Administrador\apuntes.html.

1.4. Desde DesarrolloW7XX abre un navegador y establece las siguientes conexiones:

- http://192.168.1.X8
- http://192.168.1.X8/despliegue.html
- http://192.168.1.X8/ciclos
- http://192.168.1.X8/ciclos/listado.html

2. Configuración

2.1. Edita el archivo C:\Program Files\Apache Software Foundation\Apache2.2\conf\httpd.conf y crea las directivas necesarias tal y como se muestra en las Figuras 5.48, 5.46 y 5.47.

```
<Directory "C:/Program Files/Apache Software Foundation/Apache2.2/htdocs/ciclos">
    Options FollowSymLinks
    AllowOverride None
    Order allow,deny
    Allow from all
</Directory>
```

Figura 5.46: *Directory*

```
Alias /apuntes C:/Users/Administrador/apuntes
<Directory "C:/Users/Administrador/apuntes">
    Options Indexes FollowSymLinks
    AllowOverride None
    Order allow,deny
    Allow from all
</Directory>

Redirect /fp http://www.todo fp.es
```

Figura 5.47: *Alias y Redirection*

2.2. Reinicia el servidor para que los cambios tengan efecto, Figura 5.49.

2.3. Desde DesarrolloW7XX abre un navegador y establece conexiones con http://192.168.1.X8 para probar la configuración, Figuras 5.50, 5.51, 5.52 y 5.53.

```
<Directory "C:/Program Files/Apache Software Foundation/Apache2.2/htdocs">
#
# Possible values for the Options directive are "None", "All",
# or any combination of:
#   Indexes Includes FollowSymLinks SymLinksIfOwnerMatch ExecCGI MultiViews
#
# Note that "MultiViews" must be named *explicitly* --- "Options All"
# doesn't give it to you.
#
# The Options directive is both complicated and important. Please see
# http://httpd.apache.org/docs/2.2/mod/core.html#options
# for more information.
#
# DirectoryIndex despliegue.html
Options Indexes FollowSymLinks

#
# AllowOverride controls what directives may be placed in .htaccess files.
# It can be "All", "None", or any combination of the keywords:
#   Options FileInfo AuthConfig Limit
#
# AllowOverride None

#
# Controls who can get stuff from this server.
#
# Order allow,deny
# Allow from all

</Directory>
```

Figura 5.48: *DirectoryIndex*

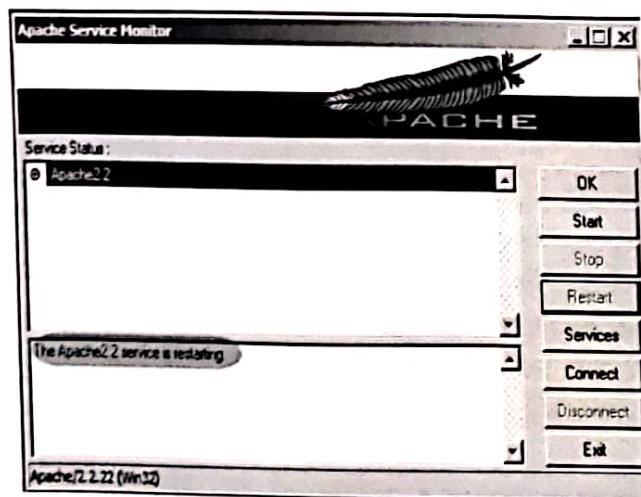


Figura 5.49: Reiniciar el servidor

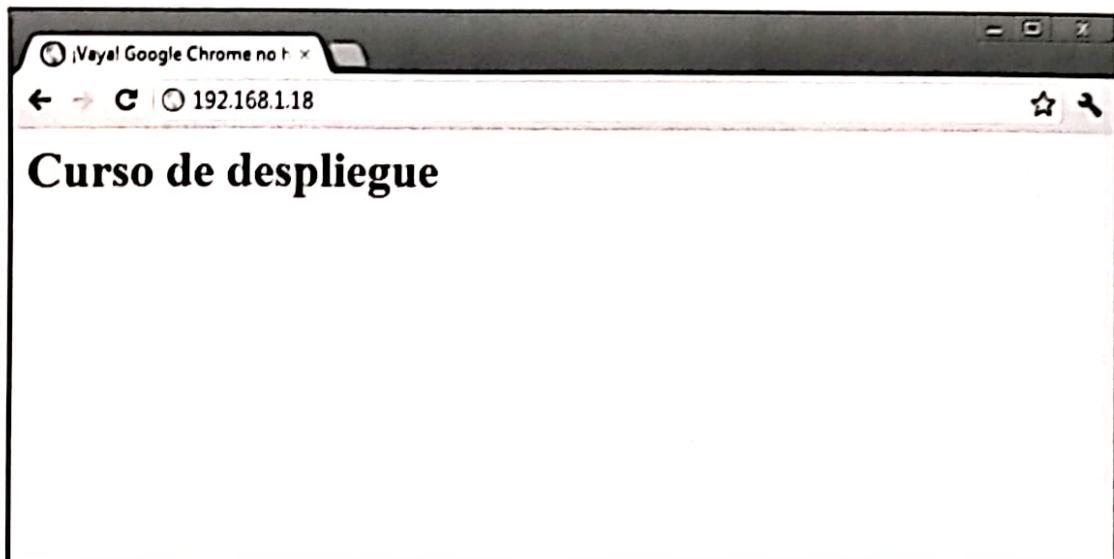


Figura 5.50: Conexión al servidor

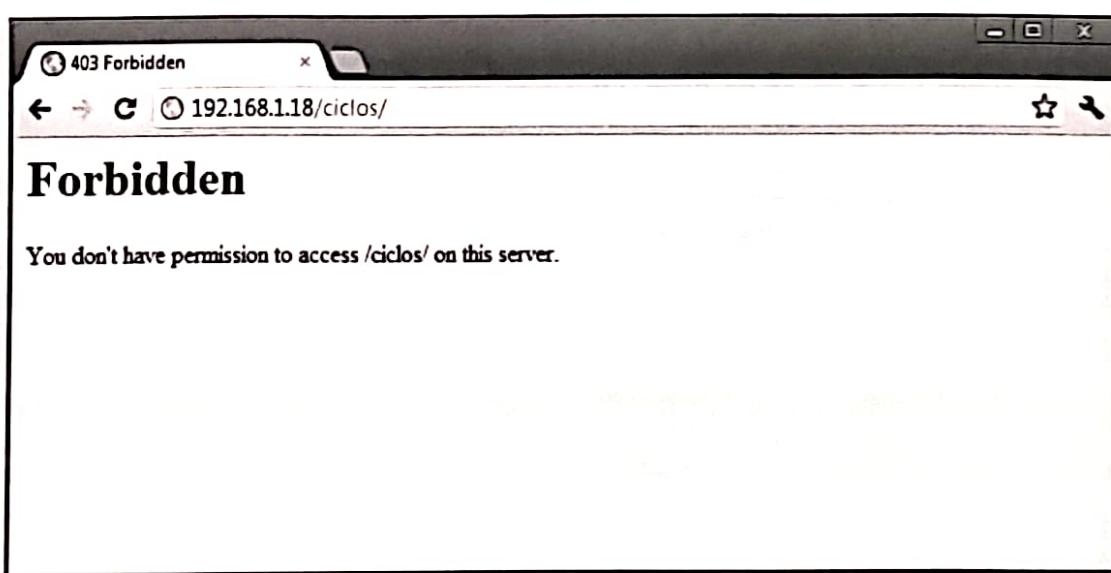


Figura 5.51: Conexión al servidor

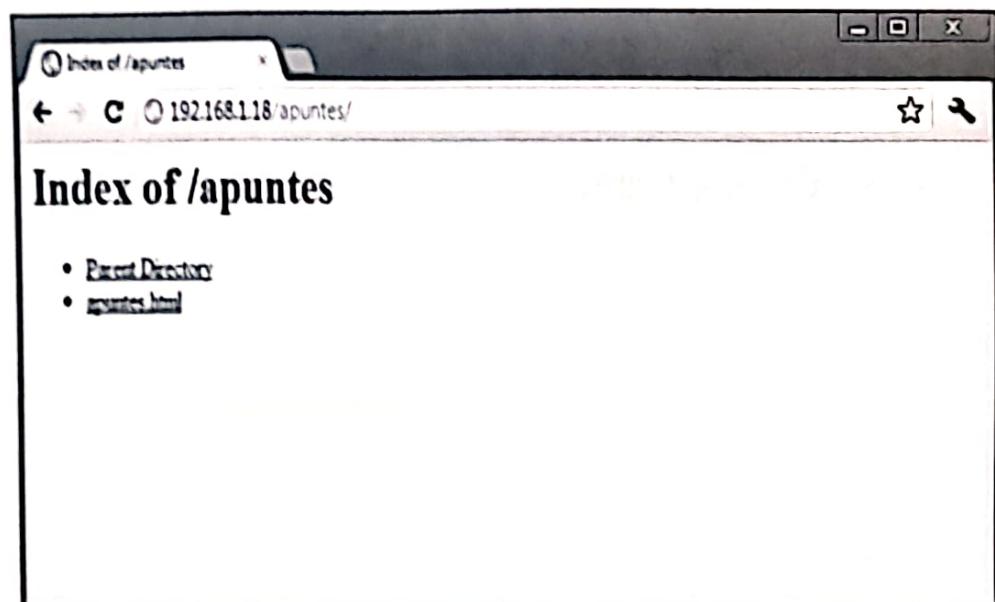


Figura 5.52: Conexión al servidor

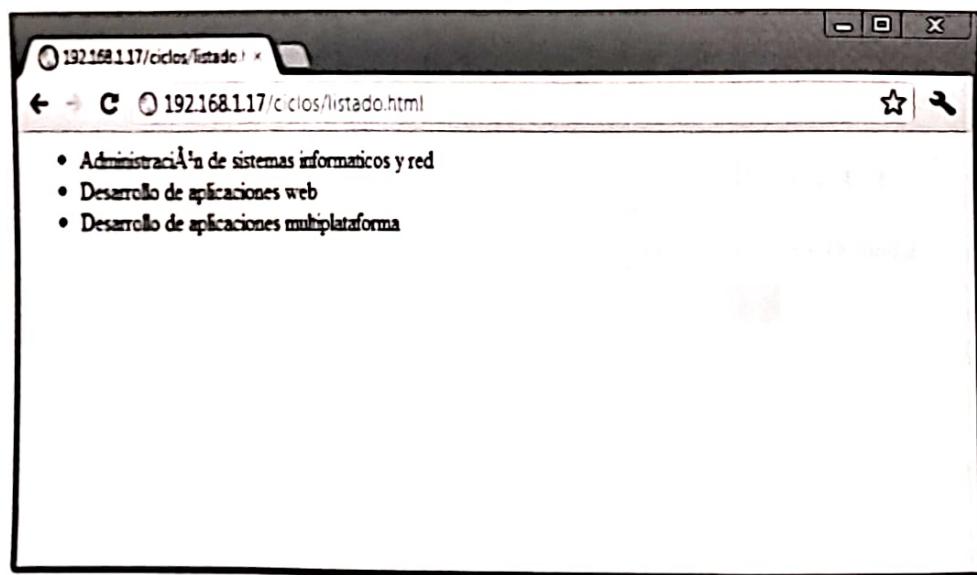


Figura 5.53: Conexión al servidor

5.7. Módulos en Linux

Consulta los módulos estáticos y los módulos dinámicos cargados por defecto en la versión de Apache instalada en la máquina ServidorLinuxXX. Posteriormente, investiga la funcionalidad del módulo `userdir`, habilítalo y prueba su funcionalidad.

1. Módulos

- 1.1. Inicia una sesión en ServidorLinuxXX con un usuario con privilegios de administrador.
- 1.2. Comprueba los módulos estáticos que se han cargado al compilar el servidor ejecutando el siguiente comando, Figura 5.54:

```
sudo apache2ctl -l
```

```
alumno@ServidorLinux01:~$ sudo apachectl -l
Compiled in modules:
  core.c
  mod_so.c
  mod_watchdog.c
  http_core.c
  mod_log_config.c
  mod_logio.c
  mod_version.c
  mod_unixd.c
alumno@ServidorLinux01:~$ _
```

Figura 5.54: Módulos estáticos

- 1.3. Comprueba los módulos que se han cargado dinámicamente al arrancar el servidor consultando el directorio `/etc/apache2/mods-enabled`. Observa que los ficheros que aparecen en este directorio son enlaces simbólicos a ficheros de `/etc/apache2/mods-available`.
- 1.4. Edita uno de los ficheros `.load` (por ejemplo `dir.load`) y observa cómo se utiliza la directiva `LoadModule`, Figura 5.55 para cargar el módulo. Comprueba cuál es la ruta donde está el código del modulo (archivo `.so`).

```
LoadModule dir_module /usr/lib/apache2/modules/mod_dir.so
```

Figura 5.55: Fichero `dir.load`

- 1.5. Edita uno de los ficheros `.conf` (por ejemplo `dir.conf`) y observa cómo se añaden directivas dentro de una declaración `<IfModule nombremodulo> ... </IfModule>`, Figura 5.56 que se ejecutarán si se carga el módulo.
- 1.6. Consulta el directorio `/usr/lib/apache2/modules/` y observa los módulos disponibles para cargar.

```
<IfModule mod_dir.c>
    DirectoryIndex index.html index.cgi index.pl index.php index.xhtml
</IfModule>
```

Figura 5.56: Fichero dir.conf

- 1.7. Ejecuta el siguiente comando para mostrar los paquetes disponibles en los repositorios de *Ubuntu* que permiten instalar módulos adicionales en *Apache*, Figura 5.57.

```
sudo apt-cache search libapache2-mod
```

```
libapache2-mod-apparmor - changehat AppArmor library as an Apache module
libapache2-mod-perl2-doc - Integration of perl with the Apache2 web server -
umentation
libapache2-mod-php5 - server-side, HTML-embedded scripting language (Apache 2
dule)
libapache2-mod-wsgi - Python WSGI adapter module for Apache
php5-cgi - server-side, HTML-embedded scripting language (CGI binary)
libapache2-mod-auth-kerb - modulo apache2 para la autentificación en Kerberos
libapache2-mod-auth-mysql - módulo apache 2 para autenticación MySQL
libapache2-mod-auth-pgsql - Módulo para Apache2 que proporciona autenticación
sql
libapache2-mod-auth-plain - Módulo para Apache2 que provee de autenticación e
exto plano
libapache2-mod-axis2c - Motor de servicios web Apache - módulo de apache
libapache2-mod-macro - Use macros en los archivos de configuración de apache2
libapache2-mod-perl2 - Integración de perl con el servidor web Apache 2
libapache2-mod-perl2-dev - Integración de perl con el servidor web Apache 2 - a
chivos de desarrollo
libapache2-mod-python - módulo integrado Python para Apache 2
libapache2-mod-python-doc - Modulo de empotrado Python para Apache 2 - docume
ción
libapache2-mod-fastcgi - módulo FastCGI de Apache 2 para scripts CGI de gran
cución
libapache2-mod-apreq2 - generic Apache request library - Apache module
libapache2-mod-auth-cas - CAS authentication module for Apache2
```

Figura 5.57: Módulos disponibles en los repositorios de *Ubuntu*

2. Módulo userdir

- 2.1. Comprueba, consultando el directorio /etc/apache2/mods-enabled que el módulo *userdir* no está habilitado.

- 2.2. Habilita el módulo ejecutando el siguiente comando:

```
sudo a2enmod userdir
```

- 2.3. Verifica que dentro del directorio /etc/apache2/mods-enabled se han creado enlaces simbólicos del módulo *userdir* (ficheros .conf y .load) hacia /etc/apache2/mod_available. Figura 5.58.

- 2.4. Reinicia el servidor para que los cambios tengan efecto.

```
lrwxrwxrwx 1 root root 30 may  4 23:09 userdir.conf -> ../mods-available/userdir
.conf
lrwxrwxrwx 1 root root 30 may  4 23:09 userdir.load -> ../mods-available/userdir
.load
alumno@ServidorLinux01:/etc/apache2/mods-enabled$
```

Figura 5.58: Módulo userdir habilitado

- 2.5. Consulta el fichero `/etc/apache2/mod_enabled/userdir.conf`. Observa que está habilitado el uso de directorios personales para todos los usuarios excepto para el usuario `root` y que `public_html` es el nombre del subdirectorio que pueden crear los usuarios en su directorio `home` para poner sus páginas personales.
- 2.6. Crea en directorio `/home/alumno/public_html`. Dentro del directorio crea un fichero denominado `personal.html` con el contenido que quieras.
- 2.7. Desde `DesarrolloW7XX` accede a `http://192.168.1.17/~alumno`, Figura 5.59.

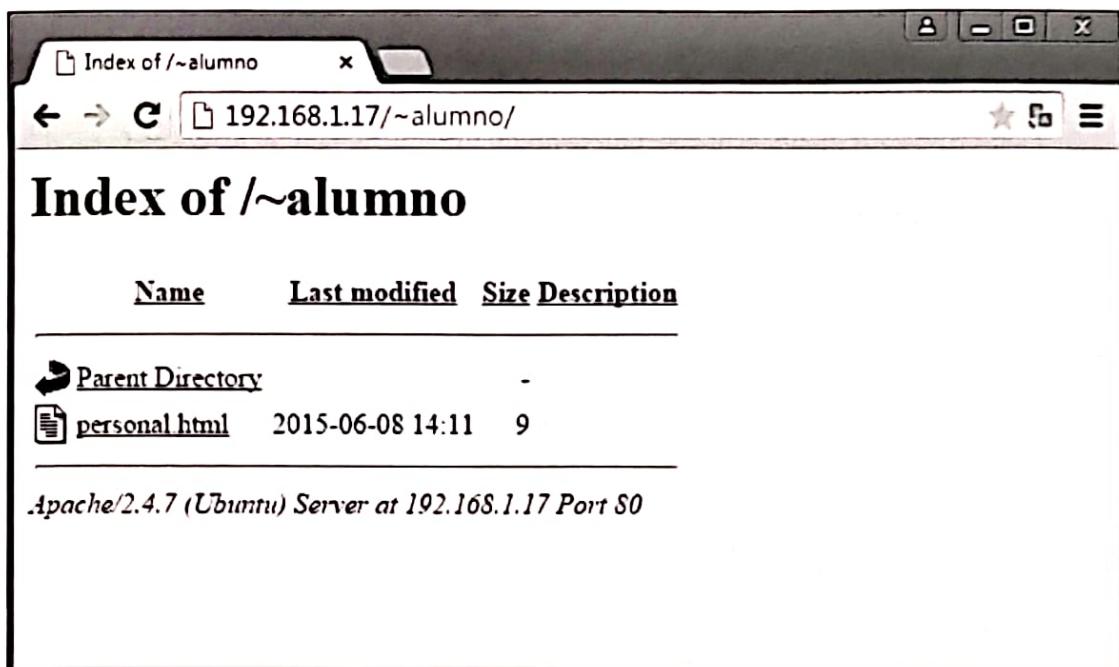


Figura 5.59: Acceso al directorio personal del usuario alumno

- 2.8. Deshabilita el módulo ejecutando el siguiente comando:

```
sudo a2dismod userdir
```

- 2.9. Reinicia el servidor para que los cambios tengan efecto.

5.8. Módulos en Windows

Consulta los módulos estáticos y los módulos dinámicos cargados por defecto en la versión de Apache instalada en la máquina `ServidorW2008XX` o `ServidorW2012XX`. Posteriormente, investiga la funcionalidad del módulo `userdir`, habíitalo y prueba su funcionalidad.

1. Módulos

- 1.1. Inicia una sesión en **ServidorW2008XX/2012XX** con un usuario con privilegios de administrador.
- 1.2. Comprueba los módulos estáticos que se han cargado al compilar el servidor ejecutando el siguiente comando, Figura 5.60:

```
httpd -l
```

```
C:\Users\Administrador>httpd -l
Compiled in modules:
  core.c
  mod_win32.c
  mpn_winnt.c
  http_core.c
  mod_so.c

C:\Users\Administrador>_
```

Figura 5.60: Módulos estáticos

- 1.3. Comprueba los módulos que se han cargado dinámicamente al arrancar el servidor consultando el fichero C:\Program Files\Apache Software Foundation\Apache2.2\conf\httpd.conf.
- 1.4. Consulta el directorio C:\Program Files\Apache Software Foundation\Apache2.2\conf\modules y observa los módulos disponibles para cargar.

2. Módulo (userdir)

- 2.1. Edita el fichero C:\Program Files\Apache Software Foundation\Apache2.2\conf\httpd.conf y habilita el módulo userdir eliminando el comentario de las directivas LoadModule e Include, Figuras 5.61 y 5.62.

```
#LoadModule spelling_module modules/mod_spelling.so
#LoadModule ssl_module modules/mod_ssl.so
#LoadModule status_module modules/mod_status.so
#LoadModule substitute_module modules/mod_substitute.so
#LoadModule unique_id_module modules/mod_unique_id.so
LoadModule userdir_module modules/mod_userdir.so
#LoadModule usertrack_module modules/mod_usertrack.so
#LoadModule version_module modules/mod_version.so
#LoadModule vhost_alias_module modules/mod_vhost_alias.so
```

Figura 5.61: Habilitar el módulo userdir

```
# Fancy directory listings
#Include conf/extra/httpd-autoindex.conf

# Language settings
#Include conf/extra/httpd-languages.conf

# User home directories
Include conf/extra/httpd-userdir.conf
```

Figura 5.62: Habilitar directivas del módulo userdir

- 2.2. Edita el fichero C:\Program Files\Apache Software Foundation\Apache2.2\conf\extras\httpd-userdir.conf. Observa que está habilitado el uso de directorios personales para todos los usuarios y que My Documents/My Website es el nombre del subdirectorio que pueden crear los usuarios para poner sus páginas personales. Edita el fichero y cambia My Documents/My Website por Documents/Website, Figura 5.63.

```
# Settings for user home directories
# Required module: mod_userdir

# UserDir: The name of the directory that is appended onto a user's home
# directory if a ~user request is received. Note that you must also set
# the default access control for these directories, as in the example below.
# UserDir "Documents/website"

# Control access to userdir directories. The following is an example
# for a site where these directories are restricted to read-only.
#
<Directory "C:/users/*/*/Documents/website">
    AllowOverride FileInfo AuthConfig Limit Indexes
    Options Multiviews Indexes SymLinksIfOwnerMatch IncludesNOEXEC
    <Limit GET POST OPTIONS>
        Order allow,deny
        Allow from all
    </Limit>
    <LimitExcept GET POST OPTIONS>
        Order deny,allow
        Deny from all
    </LimitExcept>
</Directory>
```

Figura 5.63: Fichero httpd-userdir.conf

- 2.3. Reinicia el servidor para que los cambios tengan efecto.
- 2.4. Crea en directorio C:\Usuarios\Administrador\Documents\Website. Dentro del directorio crea un fichero denominado personal.html con el contenido que quieras.
- 2.5. Desde DesarrolloW7XX accede a <http://192.168.1.18/~administrador>, Figura 5.64.

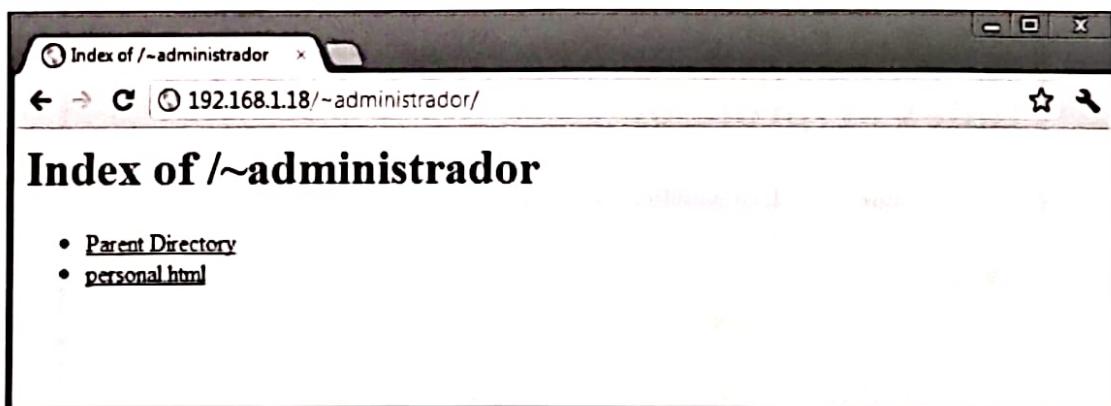


Figura 5.64: Acceso al directorio personal del usuario administrador

- 2.6. Edita el fichero C:\Program Files\Apache Software Foundation\Apache2.2\conf\httpd.conf y deshabilita el módulo userdir poniendo el comentario de las directivas LoadModule e Include.
- 2.7. Reinicia el servidor para que los cambios tengan efecto.

5.9. Control de acceso por IP y nombre de dominio

En la máquina **ServidorLinuxXX** crea el directorio **/var/www/html/profesor** y configura **Apache** para que solo se pueda acceder desde el equipo local y desde **DesarrolloW7XX** (192.168.1.X6).

1. Inicia una sesión en **ServidorLinuxXX** con un usuario con privilegios de administrador.
2. Crea en directorio **/var/www/html/profesor**. Dentro del directorio crea un fichero denominado **profesor.html** con el contenido que quieras.
3. Edita el fichero de configuración **/etc/apache2/sites-available/000-default.conf** y utiliza la directiva **<Directory>** junto con la directivas **Require** para denegar el acceso al directorio a todos los equipos excepto al local y a **DesarrolloW7XX**, Figura 5.65.

```
<Directory /var/www/html/profesor>
    Options Indexes FollowSymLinks
    AllowOverride None
    Require ip 127.0.0.1
    Require ip 192.168.1.16
</Directory>
```

Figura 5.65: Control de acceso por IP y nombre de dominio

4. Reinicia el servidor para que los cambios tengan efecto.
5. Comprueba que se puede acceder a **http://192.168.1.X7/profesor/** desde **DesarrolloW7XX** pero no desde la máquina real, Figuras 5.66 y 5.67.

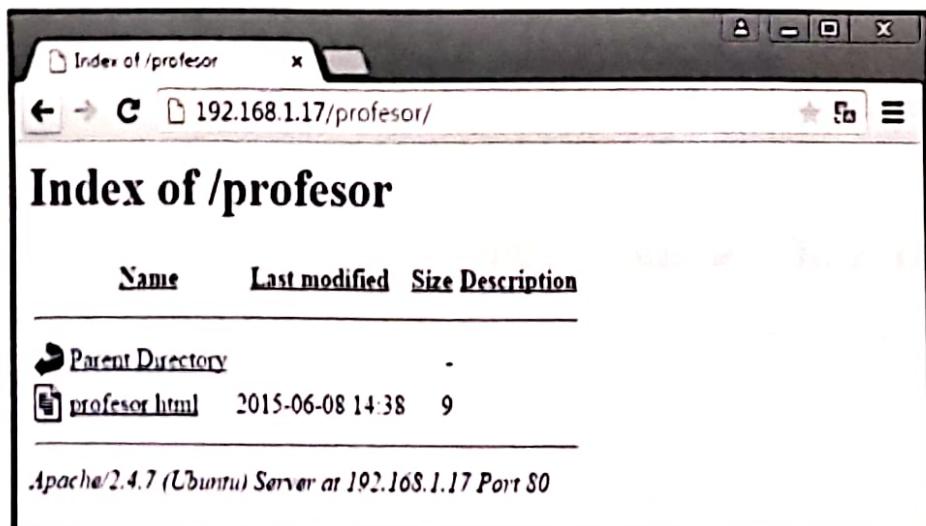


Figura 5.66: Conexión desde **DesarrolloW7XX**

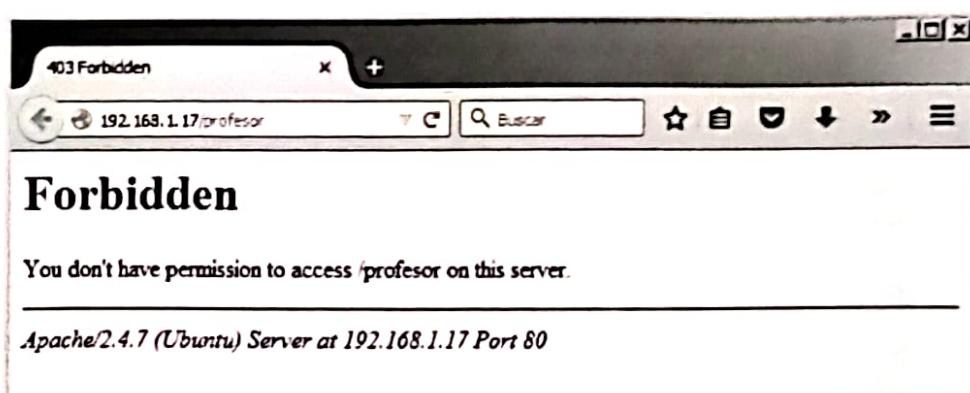


Figura 5.67: Conexión desde la máquina real

6. Prueba a permitir/denegar el acceso desde las máquinas de tus compañeros.

5.10. Autenticación y autorización *Basic* y *Digest*

En la máquina ServidorLinuxXX configura la autenticación HTTP *Basic* sobre el directorio /var/www/html/profesor para que solo puedan acceder los usuarios **profesor1** y **profesor2**. Configura la autenticación HTTP *Digest* sobre el directorio /var/www/html/departamento para que solo puedan acceder los usuarios **admin1** y **admin2**.

1. Autenticación HTTP Basic

- 1.1. Inicia una sesión en ServidorLinuxXX con un usuario con privilegios de administrador.
- 1.2. Comprueba, consultando el directorio /etc/apache2/mods-enabled, que el módulo **auth_basic** está habilitado.
- 1.3. Para usar la autenticación *basic* hay que crear un fichero accesible por *Apache* en el que se guardarán los usuarios y sus contraseñas. Para crear este fichero se utilizará el comando **htpasswd** (<http://httpd.apache.org/docs/2.4/programs/htpasswd.html>).
- a Instala el paquete **apache2-utils** que contiene **htpasswd**.

```
sudo apt-get install apache2-utils
```

- b Crea el fichero y añade el usuario **profesor1** (la opción -c es para crear el fichero).

```
sudo htpasswd -c /etc/apache2/passwd profesor1
```

- c Añade el usuario **profesor2** (no se usa la opción -c porque el fichero ya existe).

```
sudo htpasswd /etc/apache2/passwd profesor2
```

- 1.4. Edita el fichero de configuración /etc/apache2/sites-available/000-default.conf y permite el acceso a directorio /var/www/html/profesor a los usuarios **profesor1** y **profesor2**, es necesario utilizar las directivas **<RequireAll>** y **<RequireAny>** para controlar cuáles de las directivas **Require** queremos que se cumplan. Figura 5.68.

- 1.5. Reinicia el servidor para que los cambios tengan efecto.
- 1.6. Desde DesarrolloW7XX accede a <http://192.168.1.17/profesor/> con el usuario **profesor1**, Figura 5.69. Intenta el acceso con otro usuario ¿Es posible?

```
<Directory /var/www/html/profesor>
    Options Indexes FollowSymLinks
    AllowOverride None
    AuthType Basic
    AuthName "Acceso restringido"
    AuthUserFile /etc/apache2/passwd
    <RequireALL>
        Require user profesor1 profesor2
    <RequireAny>
        Require ip 127.0.0.1
        Require ip 192.168.1.16
    </RequireAny>
</RequireAll>
</Directory>
```

Figura 5.68: Autenticación *Basic*

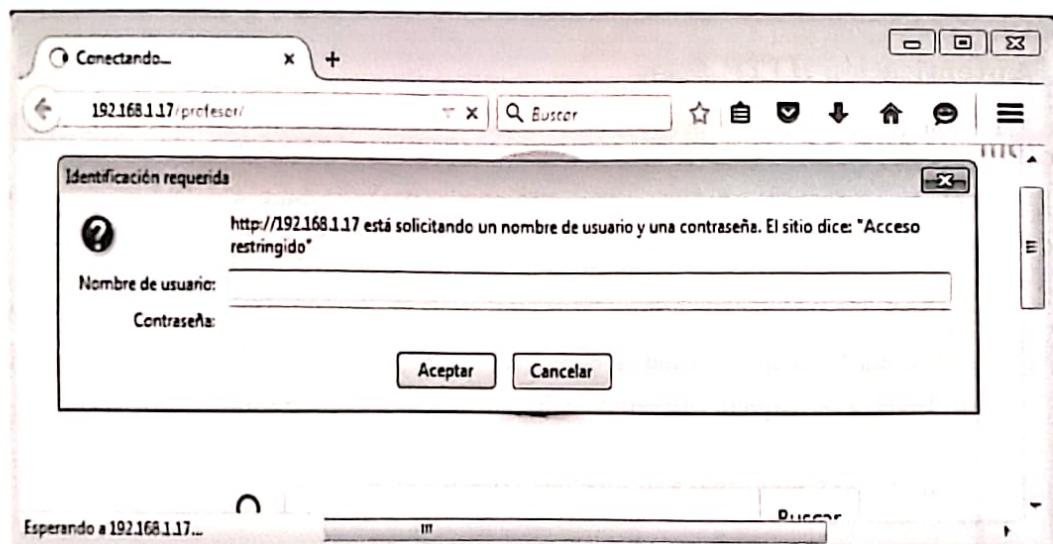


Figura 5.69: Conexión desde DesarrrolloW7XX

2. Autenticación HTTP *Digest*

- 2.1. Crea en directorio `/var/www/html/departamento`. Dentro del directorio crea un fichero denominado `departamento.html` con el contenido que quieras.
- 2.2. Habilita el módulo `auth_digest`.

```
sudo a2enmod auth_digest
```

- 2.3. Reinicia el servidor para que los cambios tengan efecto.

- 2.4. Para usar la autenticación *digest* hay que crear un fichero accesible por *Apache* en el que se guardarán los usuarios y sus contraseñas asociados a un dominio (*realm*). Para crear este fichero se utilizará el comando **htdigest** (<http://httpd.apache.org/docs/2.4/es/programs/htdigest.html>).

- a Crea el fichero y añade el usuario **admin1** al dominio **informatica** (la opción **-c** es para crear el fichero).

```
sudo htdigest -c /etc/apache2/digest informatica admin1
```

- b Añade el usuario **admin2** (no se usa la opción **-c** porque el fichero ya existe).

```
sudo htdigest /etc/apache2/digest informatica admin2
```

- 2.5. Edita el fichero de configuración **/etc/apache2/sites-available/000-default.conf** y permite el acceso a directorio **/var/www/html/departamento** a los usuarios **admin1** y **admin2**, Figura 5.70.

```
<Directory /var/www/html/departamento>
    Options Indexes FollowSymLinks
    AllowOverride None
    AuthType Digest
    AuthName "informatica"
    AuthDigestProvider file
    AuthUserFile /etc/apache2/digest
    Require user admin1 admin2
</Directory>
```

Figura 5.70: Autenticación *Digest*

- 2.6. Desde **DesarrolloW7XX** accede a **http://192.168.1.X7/departamento/** con el usuario **admin1**, Figura 5.71. Intenta el acceso con otro usuario ¿Es posible?

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**.

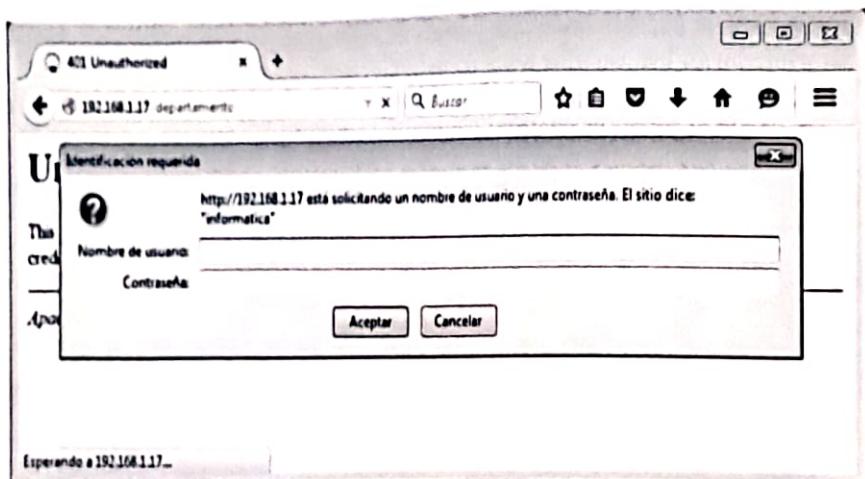


Figura 5.71: Conexión desde DesarrrolloW7XX

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 sobreescribir 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/.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 **DesarrrolloW7XX** 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

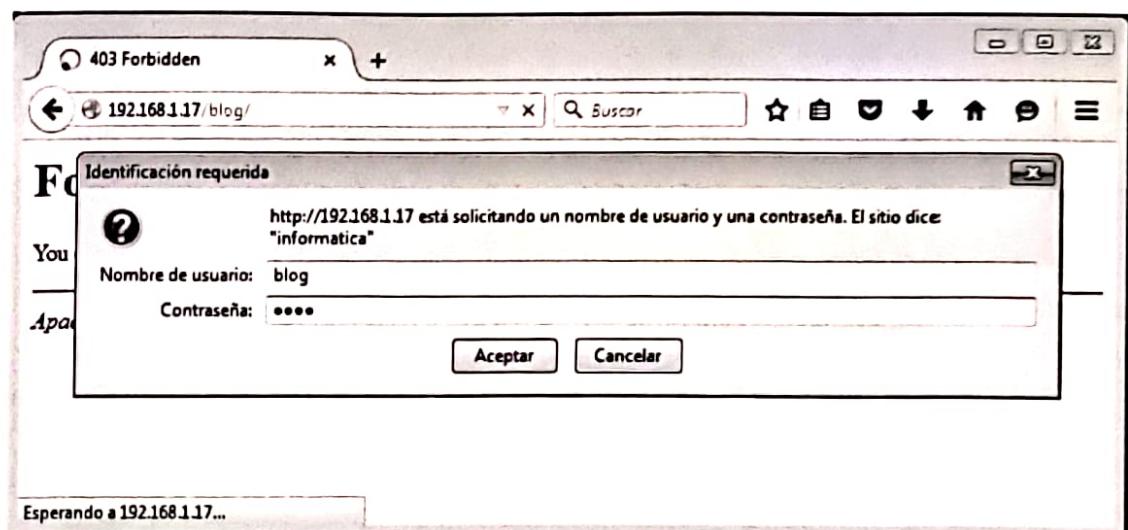


Figura 5.74: Conexión desde DesarrolloW7XX

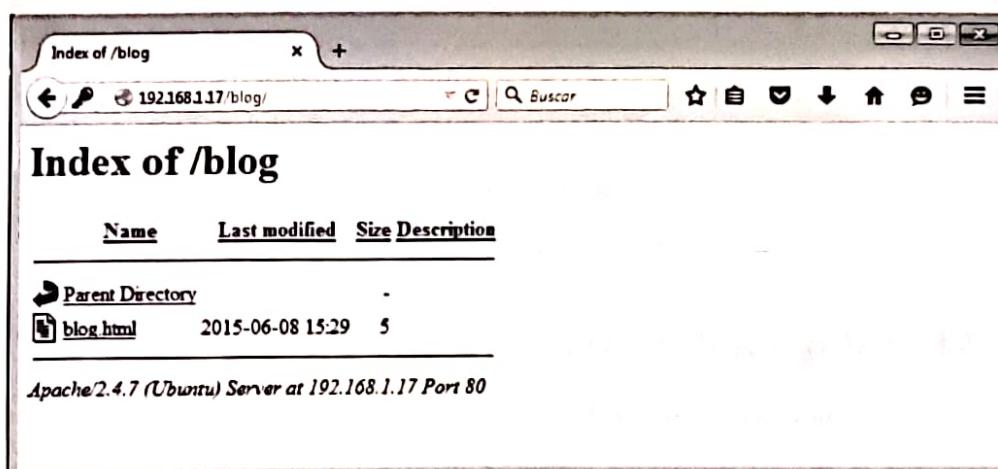


Figura 5.75: Conexión desde DesarrolloW7XX

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.

```
# Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
# error, crit, alert, emerg.
# It is also possible to configure the loglevel for particular
# 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.

```
# Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
# error, crit, alert, emerg.
# It is also possible to configure the loglevel for particular
# modules, e.g.
LogLevel info ssl:warn

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 %O
# (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 %O \"%{Referer}i\" \"%{User-Agent}i\""
LogFormat "%h %l %u %t \"%r\" %>s %O \"%{Referer}i\" \"%{User-Agent}i\""
LogFormat "%h %l %u %t \"%r\" %>s %O" 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.

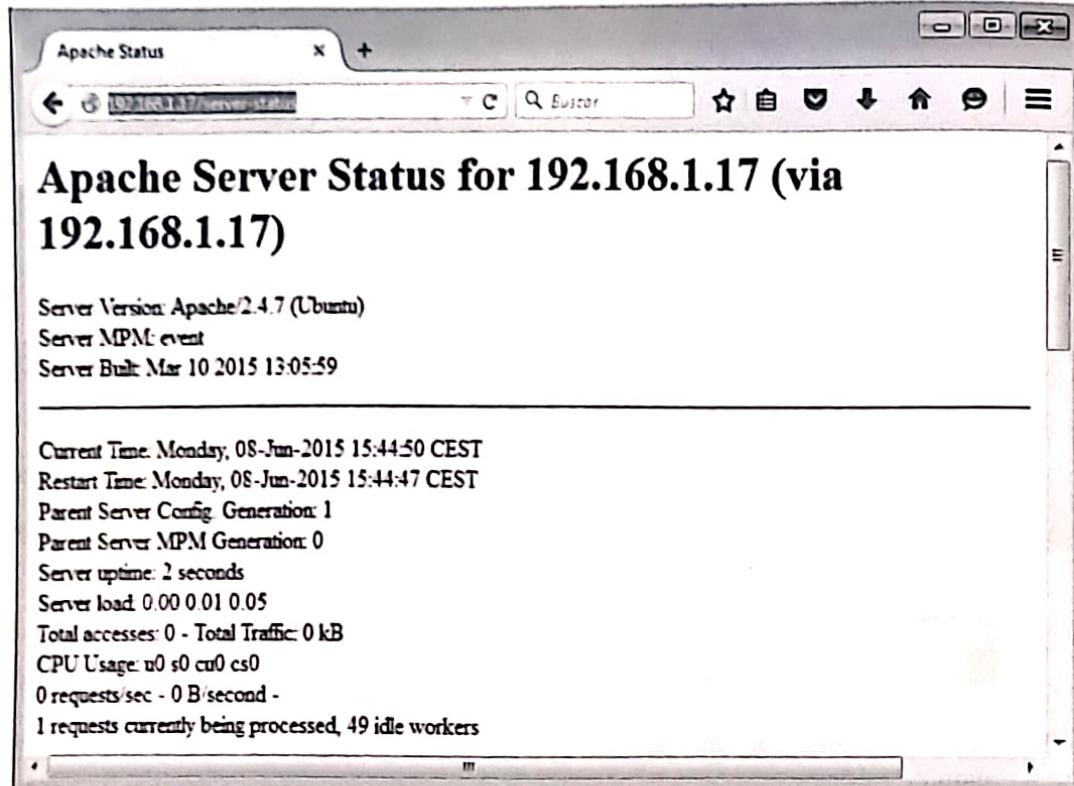


Figura 5.80: Estado del servidor

- 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.

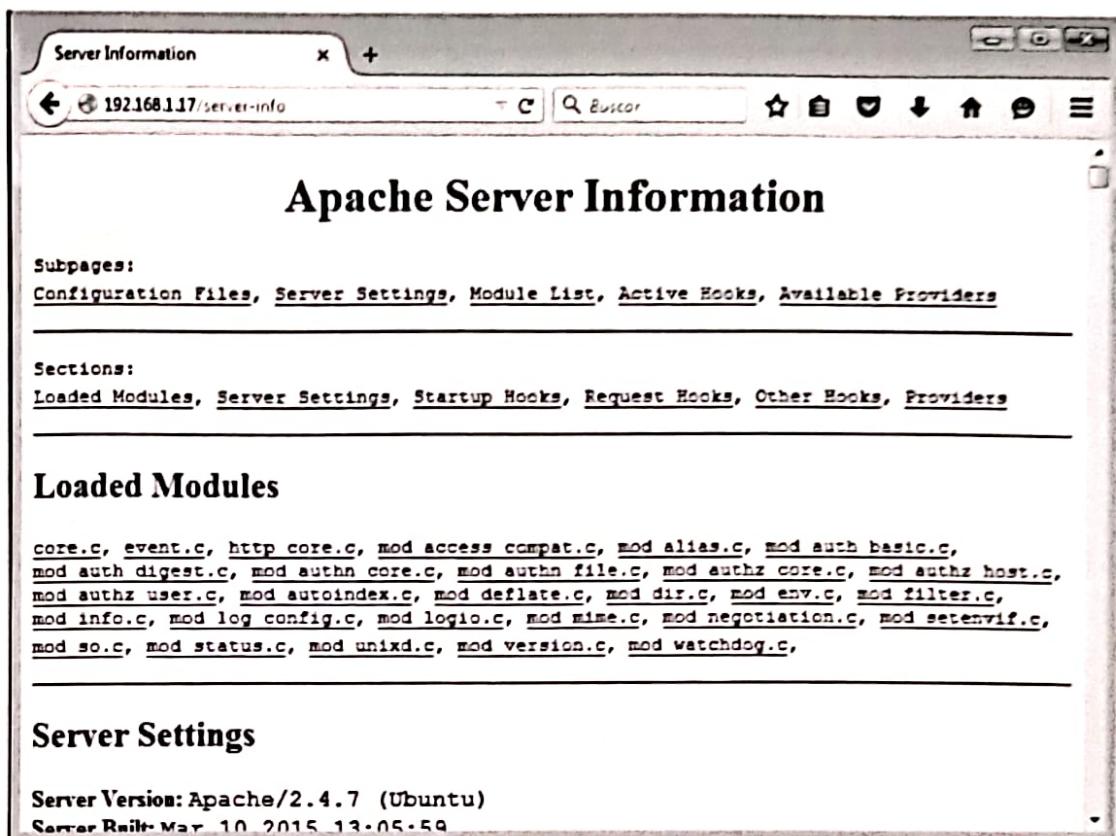


Figura 5.82: Información sobre la configuración del servidor

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.

```
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*

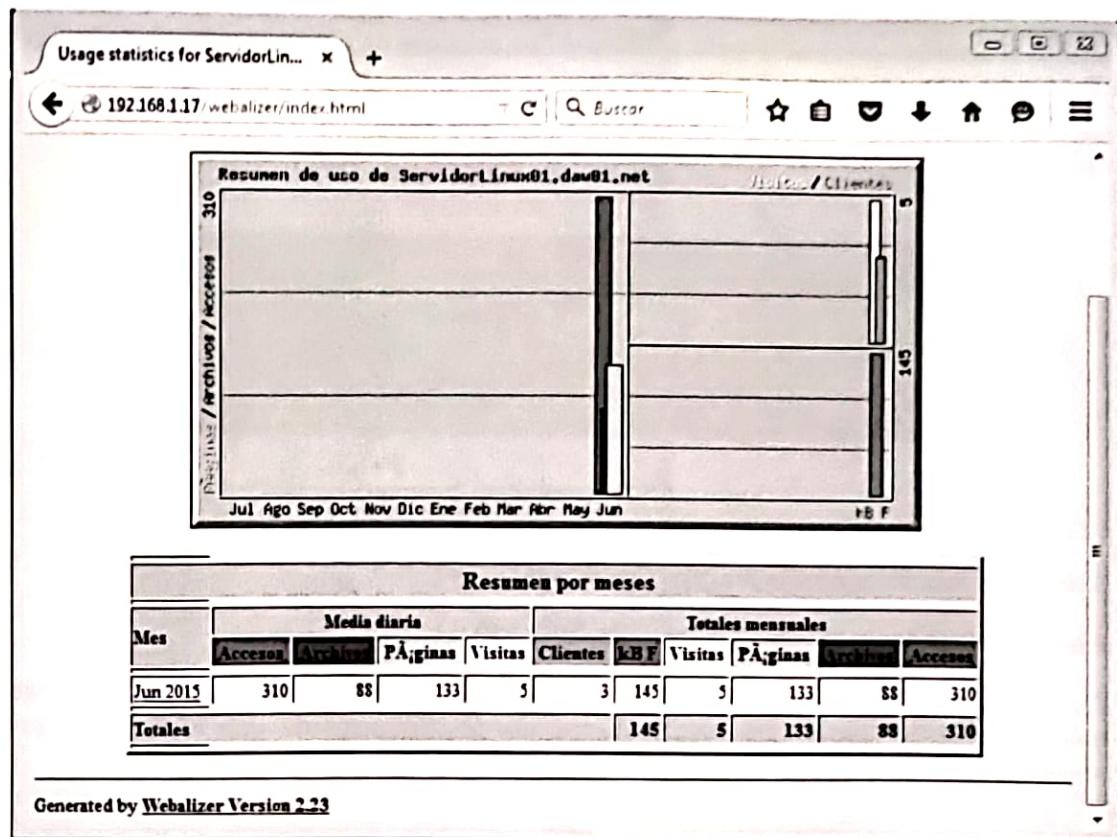


Figura 5.85: Estadísticas de webalizer

## 5.15. Alojamiento virtual de sitios web en Internet

Consulta las siguientes webs <http://www.myipneighbors.com/> y <http://www.robtex.com/> y averigua información sobre nombres de dominio alojados en la misma IP.

1. Inicia una sesión en DesarrolloW7XX con un usuario con privilegios de administración.
2. Utiliza el comando nslookup para obtener las direcciones IP asociadas al nombre de dominio `www.agssf.net`, Figura 5.86.

```
nslookup www.agssf.net
```

```
C:\>nslookup www.agssf.net
Servidor: servidoro200801.dau01.net
Address: 192.168.1.18

Respuesta no autoritativa:
Nombre: agssf.net
Address: 91.199.120.11
Aliases: www.agssf.net
```

Figura 5.86: Resolución directa

- Utiliza el comando nslookup para obtener el/los nombres de dominio asociados a la dirección IP 91.199.120.11, Figura 5.87.

```
nslookup 91.199.120.11
```

```
C:\Users\alumno>nslookup 91.199.120.11
Servidor: servidorw200801.daw01.net
Address: 192.168.1.18

Nombre: ela.h3m.com
Address: 91.199.120.11
```

Figura 5.87: Resolución inversa

- Abre el navegador y accede a <http://www.agssf.net>.
- Abre el navegador y accede a <http://91.199.120.11>.
- Accede a <http://www.myipneighbors.com/> y a <http://www.robtex.com/> y averuga qué más nombres de dominio se relacionan con esa IP. Esto nos da una idea de los posibles servidores web virtuales que existen en esa IP, Figuras 5.88, 5.89 y 5.90.



#### Results for www.agssf.net

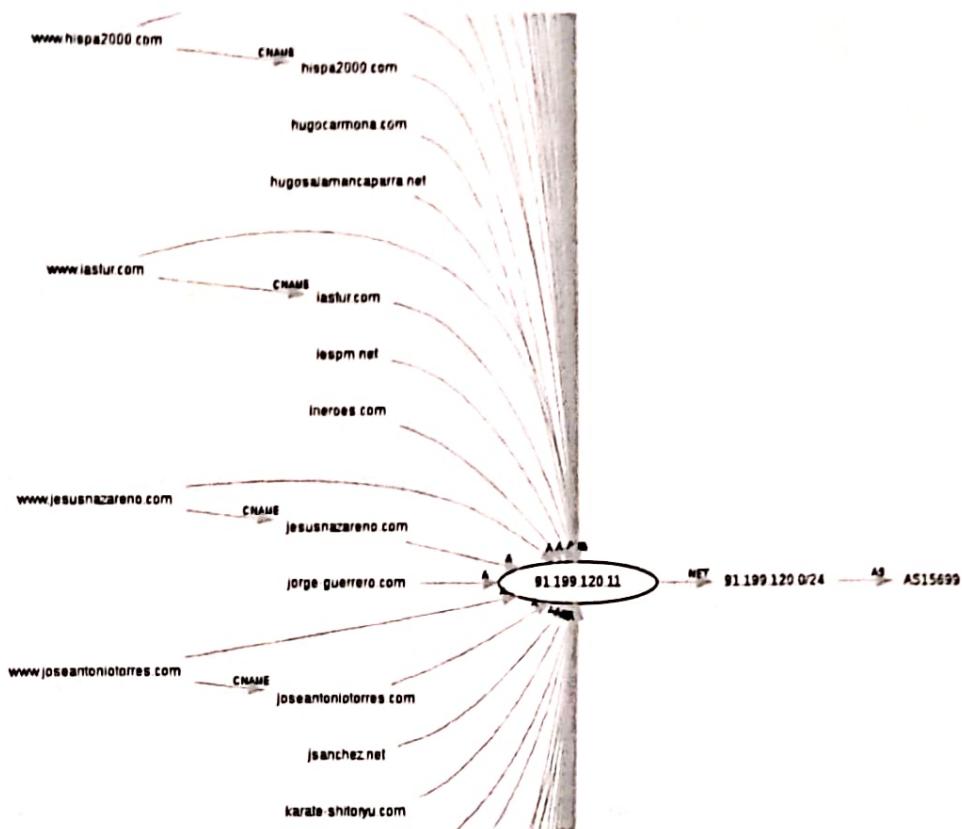
These 10 of 373 domains are at 91.199.120.11. [Subscribe to see them all.](#)

- 1937.es
- ab-informatica.org
- academia.starweb.com
- acebaché.com
- acorn-salvajescaleras.es
- aerosalvia.com
- afrikalakorps.ewrwan-network.net
- agenciagranada.com
- aikidolanzarote.com
- airsoftjeen.com

Figura 5.88: <http://www.myipneighbors.com>

[www.agssf.net](http://www.agssf.net) has one IP number (91.199.120.11), which is the same as for [agsfp.net](http://www.agssf.net), via cname , but the reverse is [ela.h3m.com](http://ela.h3m.com).  
[Miedoalaverdad.com](http://Miedoalaverdad.com), [webdeportes.es](http://webdeportes.es), [mugurza.es](http://mugurza.es), [igalpel.com](http://igalpel.com), [gatsrocafort.com](http://gatsrocafort.com) and at least 200 other hosts point to the same IP.  
[Disparatefarmaceutico.com](http://Disparatefarmaceutico.com), [ibaplicaciones.com](http://ibaplicaciones.com), [hermandaddesantiago.org](http://hermandaddesantiago.org), [treshormigas.es](http://treshormigas.es), [tutecnico.com](http://tutecnico.com) and at least ten other hosts use [www.agssf.net](http://www.agssf.net) as a mail server under another name.

Figura 5.89: <http://www.robtex.com/>

Figura 5.90: <http://www.robtex.com/>

7. Busca otros sitios web que ofrezcan el mismo servicio.

## 5.16. Alojamiento virtual de sitios web en Linux

Realiza la siguiente configuración en el servidor *Apache* instalado en ServidorLinuxXX.

- Configura el servidor DNS de ServidorW2008XX o ServidorW2012XX para que resuelva los nombres **software.dawXX.net** y **hardware.dawXX.net**.
- Deshabilita el servidor virtual por defecto.
- Crea y habilita un servidor virtual para el dominio **software.dawXX.net**.
  - Directorio raíz **/var/www/html/software**.
    - Se servirá el fichero **index.html** si no se indica ningún fichero en la URL.
    - No se mostrará un listado del directorio raíz si no se solicita ningún fichero.
    - Podrán acceder todos los usuarios.
  - Directorio **/var/www/software/privado**.
    - Solo podrá acceder el usuario **linux** desde el equipo **192.168.1.X6** (**DesarrolloW7XX**).
    - Se mostrará un listado del directorio raíz si no se solicita ningún fichero.
  - El *log* de errores será **/var/log/apache2/software.error.log**.
  - El *log* de accesos será **/var/log/apache2/software.access.log**, con formato *combined*.

- Crea y habilita un servidor virtual para el dominio hardware.dawXX.net.
  - Directorio raíz /var/www/html/hardware.
    - Se servirá el fichero index.html si no se indica ningún fichero en la URL.
    - Se mostrará un listado del directorio raíz si no se solicita ningún fichero.
    - Podrán acceder todos los usuarios.
  - Se creará el alias /alumno que permitirá acceder al directorio virtual /home/alumno.
    - Se permitirá sobreescibir todas las directivas haciendo uso de ficheros .htaccess en /home/alumno.
  - El log de errores será /var/log/apache2/hardware.error.log.
  - El log de accesos será /var/log/apache2/hardware.access.log, que tendrá formato combined.

## 1. Configuración del servidor DNS

- 1.1. Configura el servidor DNS de ServidorW2008XX o ServidorW2012XX en para que resuelva los nombres software.dawXX.net y hardware.dawXX.net. La dirección IP asociada a los nombres será la IP de ServidorLinuxXX es decir 192.168.1.X7, Figura 5.91.
- 1.2. Asegúrate que DesarrolloW7XX utiliza el servidor DNS que has configurado, Figura 5.92.

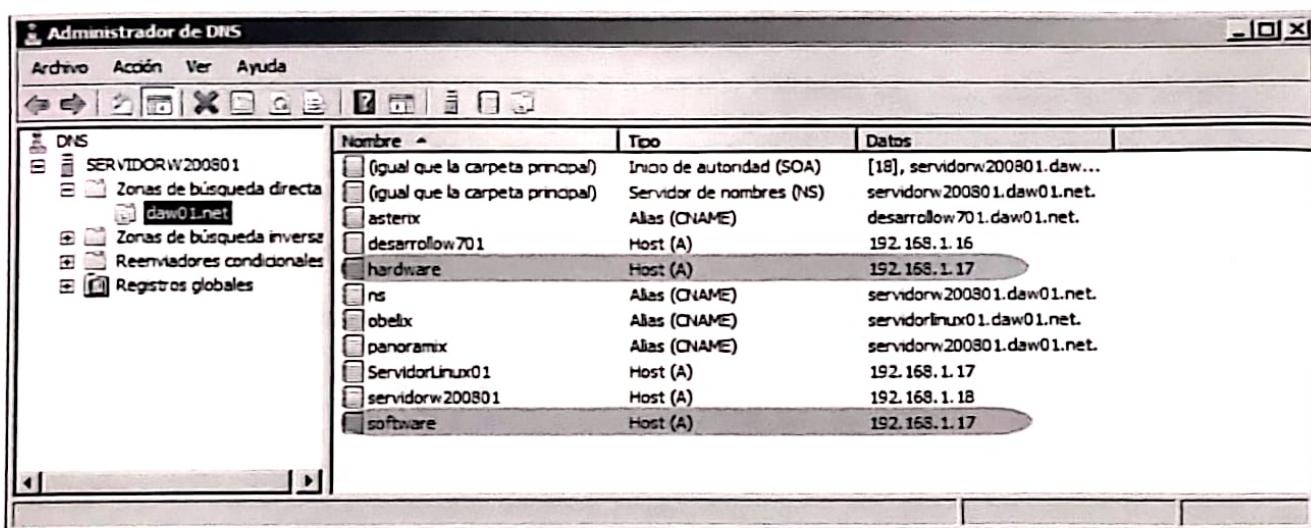


Figura 5.91: Configuración del servidor DNS en ServidorW2008XX

## 2. Deshabilitar el servidor virtual por defecto

- 2.1. Deshabilita el servidor virtual por defecto.

```
sudo a2dissite 000-default
```

- 2.2. Verifica que dentro del directorio /etc/apache2/sites-enabled se ha borrado el fichero 000-default.conf.

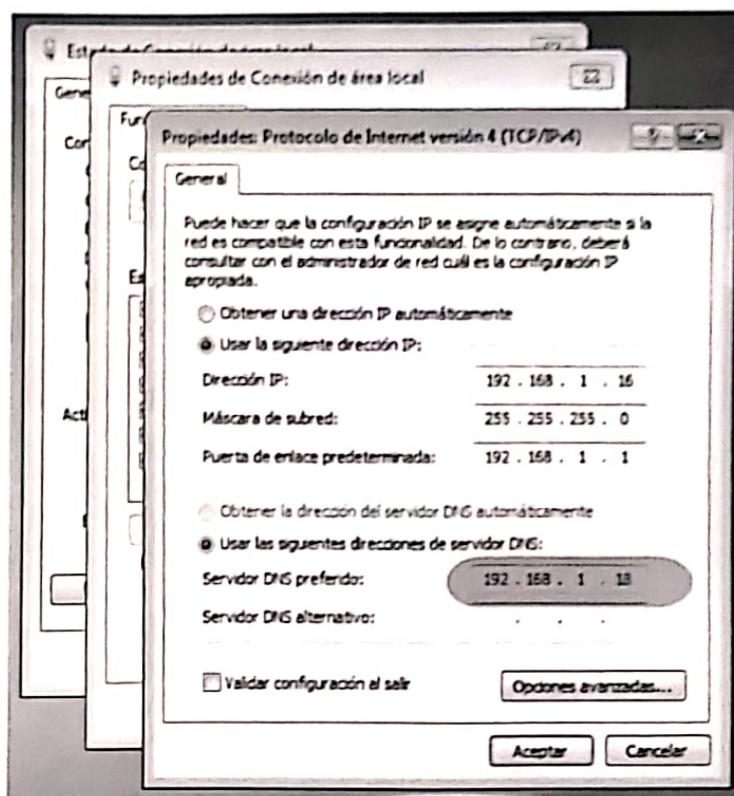


Figura 5.92: Servidor DNS en DesarrolloW7XX

- 2.3. Reinicia el servidor para que los cambios tengan efecto. Observa que se muestra un mensaje que indica que no hay servidores virtuales configurados.
- 3. Servidor virtual para el dominio software.dawXX.net**
- 3.1. Crea el directorio `/var/www/html/software`.
  - 3.2. Crea el fichero de texto `/var/www/html/software/index.html` con contenido que quieras.
  - 3.3. Crea el directorio `/var/www/html/software/privado`.
  - 3.4. Crea el fichero de texto `/var/www/html/software/privado/privado.html` con contenido que quieras.
  - 3.5. Crea el fichero `/etc/apache2/software.digest` y añade el usuario `linux` al dominio `software` (la opción `-c` es para crear el fichero).
- ```
sudo htdigest -c /etc/apache2/software.digest software linux
```
- 3.6. Crea el fichero `/etc/apache/site-available/software.conf` con las siguientes directivas, Figura 5.93.
 - 3.7. Habilita el servidor virtual de `software`.
- ```
sudo a2ensite software
```
- 3.8. Verifica que dentro del directorio `/etc/apache2/sites-enabled` se ha creado el enlace `software.conf`, Figura 5.94.

```
<VirtualHost *:80>
 ServerName software.daw01.net

 DocumentRoot /var/www/html/software

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

 <Directory /var/www/html/software>
 DirectoryIndex index.html
 Options Indexes FollowSymLinks
 AllowOverride None
 Require all granted
 </Directory>

 <Directory /var/www/html/software/privado>
 Options Indexes FollowSymLinks
 AllowOverride None
 AuthType Digest
 AuthName "software"
 AuthDigestProvider file
 AuthUserFile /etc/apache2/software.digest
 <RequireALL>
 Require user linux
 Require ip 192.168.1.16
 </RequireALL>
 </Directory>

</VirtualHost>
```

Figura 5.93: Fichero de configuración del servidor virtual software.daw01.net

```
alumno@ServidorLinux01:/etc/apache2/sites-enabled$ ls -l
total 0
lrwxrwxrwx 1 root root 32 jun 8 16:37 software.conf -> ../../sites-available/software.conf
```

Figura 5.94: Directorio /etc/apache2/sites-enabled

- 3.9. Reinicia el servidor para que los cambios tengan efecto.
- 3.10. Desde DesarrolloWindowsXX accede a <http://software.dawXX.net>, <http://software.dawXX.net/privado>, Figuras 5.95 y 5.96.

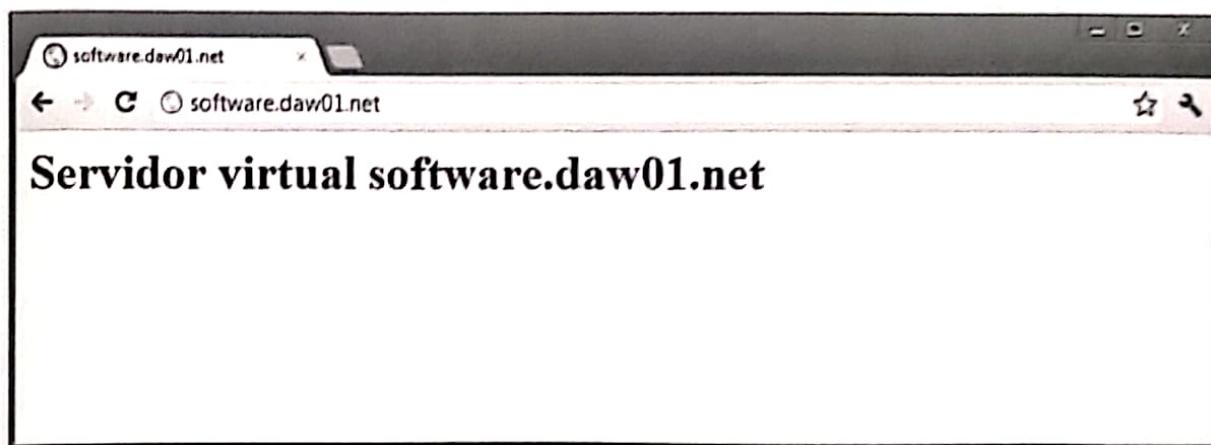


Figura 5.95: Acceso al servidor virtual software.daw01.net

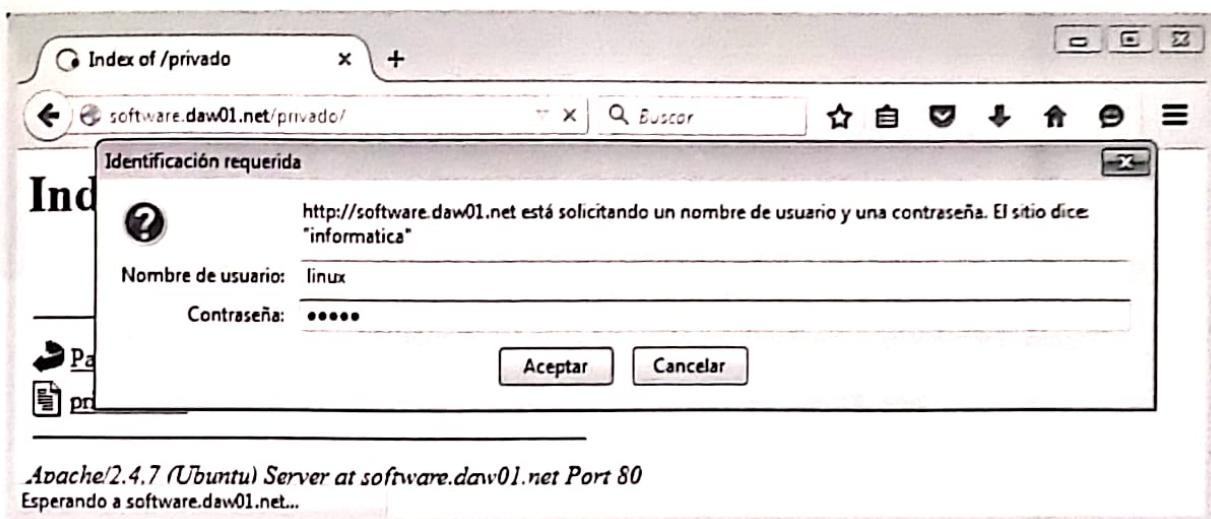


Figura 5.96: Acceso al servidor virtual software.daw01.net

#### 4. Servidor virtual para el dominio hardware.dawXX.net

- 4.1. Crea el directorio /var/www/html/hardware.
- 4.2. Crea el fichero de texto /var/www/html/hardware/index.html con contenido que quieras.
- 4.3. Crea el fichero /etc/apache/site-available/hardware.conf con las siguientes directivas, Figura 5.97.
- 4.4. Habilita el servidor virtual de hardware.

```
sudo a2ensite hardware
```

```
<VirtualHost *:80>
 ServerName hardware.daw01.net

 DocumentRoot /var/www/html/hardware

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

 <Directory /var/www/html/hardware>
 DirectoryIndex index.html
 Options Indexes FollowSymLinks
 AllowOverride None
 Require all granted
 </Directory>

 Alias /alumno /home/alumno
 <Directory /var/www/html/software/privado>
 AllowOverride All
 </Directory>

</VirtualHost>
```

Figura 5.97: Fichero de configuración del servidor virtual hardware.daw01.net

```
alumno@ServidorLinux01:/etc/apache2/sites-enabled$ ls -l
total 0
lrwxrwxrwx 1 root root 32 jun 8 16:49 hardware.conf -> ../sites-available/hardware
lrwxrwxrwx 1 root root 32 jun 8 16:37 software.conf -> ../sites-available/software
alumno@ServidorLinux01:/etc/apache2/sites-enabled$ █
```

Figura 5.98: Directorio /etc/apache2/sites-enabled

- 4.5. Verifica que dentro del directorio `/etc/apache2/sites-enabled` se ha creado el enlace `hardware.conf`, Figura 5.98.
- 4.6. Reinicia el servidor para que los cambios tengan efecto.
- 4.7. Desde DesarrolloW7XX accede a `http://hardware.daw01.net`, Figura 5.99.



Figura 5.99: Acceso al servidor virtual `hardware.daw01.net`

## 5.17. Alojamiento virtual de sitios web en *Windows*

Realiza la siguiente configuración en el servidor *Apache* instalado en **ServidorW2008XX**.

- Configura el servidor DNS de **ServidorW2008XX** para que resuelva los nombres `www.tierra.com` y `www.marte.com`.
- Crea y habilita un servidor virtual para el dominio `www.tierra.com`.
  - Directorio raíz `C:\Program Files\Apache Software Foundation\Apache2.2\htdocs\tierra`.
    - Se servirá el fichero `index.html` si no se indica ningún fichero en la URL.
    - No se mostrará un listado del directorio raíz si no se solicita ningún fichero.
    - Podrán acceder todos los usuarios.
  - El *log* de errores será `C:\Program Files\Apache Software Foundation\Apache2.2\logs\tierra.error.log`.
  - El *log* de accesos será `C:\Program Files\Apache Software Foundation\Apache2.2\logs\tierra.access.log`, con formato *combined*.
- Crea y habilita un servidor virtual para el dominio `www.marte.com`.
  - Directorio raíz `C:\Program Files\Apache Software Foundation\Apache2.2\htdocs\marte`.
    - Se servirá el fichero `index.html` si no se indica ningún fichero en la URL.
    - Se mostrará un listado del directorio raíz si no se solicita ningún fichero.

- Podrán acceder todos los usuarios.
- El *log* de errores será C:\Program Files\Apache Software Foundation\Apache2.2\logs\marte.error.log
- El *log* de accesos será C:\Program Files\Apache Software Foundation\Apache2.2\logs\marte.access.log, con formato *combined*.

### 1. Configuración del servidor DNS

- 1.1. Configura el servidor DNS de **ServidorW2008XX** para que resuelva los nombres **www**, **marte.com** y **www.tierra.com**. La dirección IP asociada a los nombres será la IP de **ServidorWindowsXX** es decir 192.168.1.X8, Figuras 5.100 y 5.101.

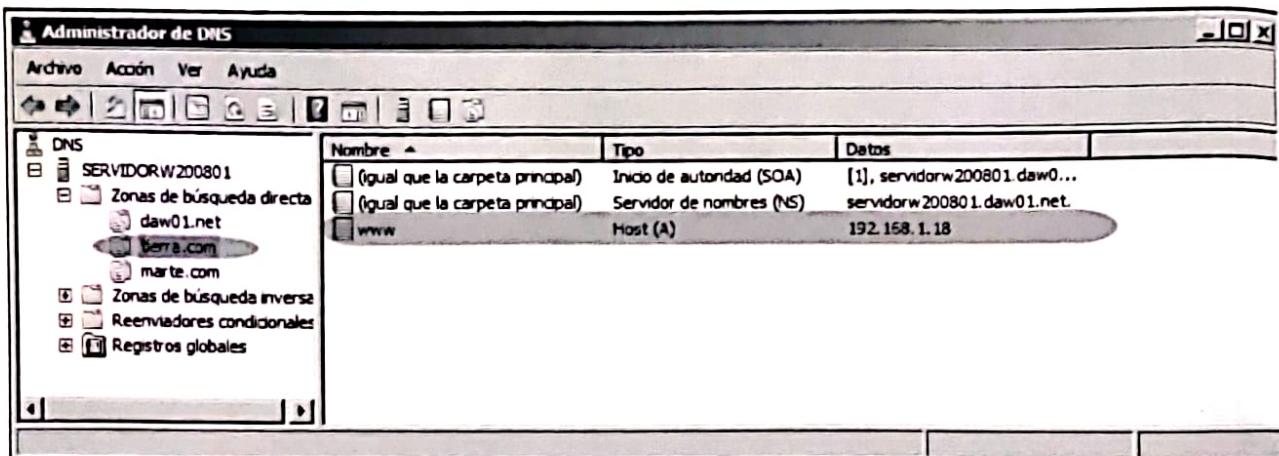


Figura 5.100: Configuración del servidor DNS en **ServidorW2008XX**

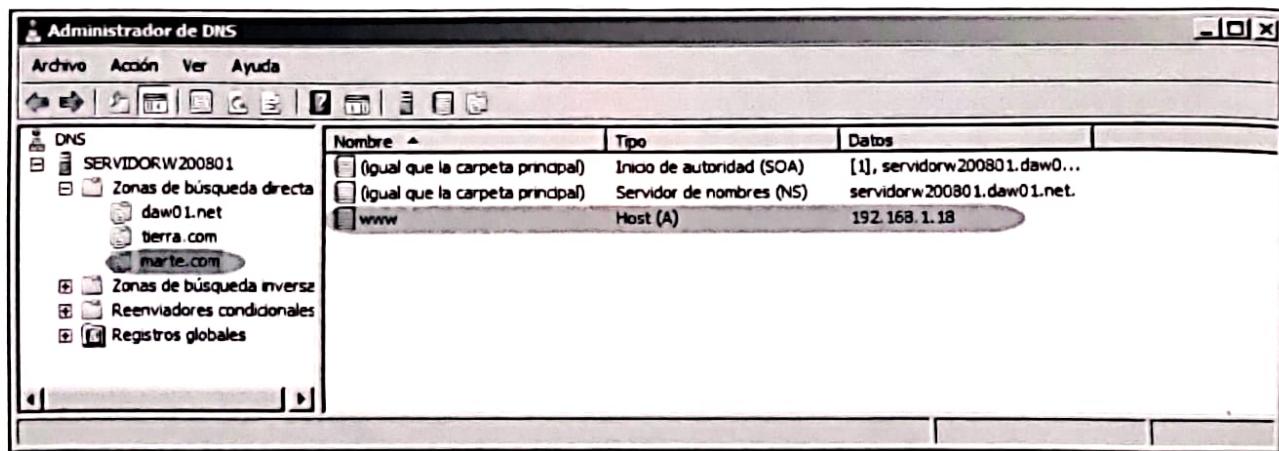


Figura 5.101: Configuración del servidor DNS en **ServidorW2008XX**

- 1.2. Asegúrate que **DesarrolloW7XX** utiliza el servidor DNS que has configurado.

**2. Configuración de servidores virtuales**

- 2.1. Edita el fichero C:\Program Files\Apache Software Foundation\Apache2.2\conf\httpd.conf y eliminana el comentario de la directiva **Include** del fichero conf/extrá/httpd-vhost.conf, Figura 5.102.

```
Real-time info on requests and configuration
#Include conf/extrá/httpd-info.conf

Virtual hosts
Include conf/extrá/httpd-vhosts.conf

Local access to the Apache HTTP Server Manual
#Include conf/extrá/httpd-manual.conf

Distributed authoring and versioning (webDAV)
#Include conf/extrá/httpd-dav.conf

Various default settings
#Include conf/extrá/httpd-default.conf
```

Figura 5.102: Fichero httpd.conf

- 2.2. Crea el directorio C:\Program Files\Apache Software Foundation\Apache2.2\htdocs\tierra.
- 2.3. Crea el fichero de texto C:\Program Files\Apache Software Foundation\Apache2.2\htdocs\tierra\index.html con contenido que quieras.
- 2.4. Crea el directorio C:\Program Files\Apache Software Foundation\Apache2.2\htdocs\marte.
- 2.5. Crea el fichero de texto C:\Program Files\Apache Software Foundation\Apache2.2\htdocs\marte\marte.html con contenido que quieras.
- 2.6. Edita el fichero C:\Program Files\Apache Software Foundation\Apache2.2\conf\extras\httpd-vhost.conf y configura las siguientes directivas, Figura 5.103.
- 2.7. Reinicia el servidor para que los cambios tengan efecto.
- 2.8. Desde DesarrolloW7XX accede a <http://www.tierra.com> y <http://www.marte.com>, Figuras 5.104 y 5.105.

```

Use name-based virtual hosting.

NameVirtualHost *:80

virtualHost example:
Almost any Apache directive may go into a VirtualHost container.
The first VirtualHost section is used for all requests that do not
match a ServerName or ServerAlias in any <VirtualHost> block.

<VirtualHost *:80>
 ServerName www.tierra.com
 DocumentRoot "C:/Program Files/Apache Software Foundation/Apache2.2/htdocs/tierra"
 <Directory "C:/Program Files/Apache Software Foundation/Apache2.2/htdocs/tierra">
 DirectoryIndex index.html
 Options Indexes FollowSymlinks
 AllowOverride None
 Order allow,deny
 Allow from all
 </Directory>

 ErrorLog "logs/tierra.error.log"
 CustomLog "logs/tierra.access.log" common
</VirtualHost>

<VirtualHost *:80>
 ServerName www.marte.com
 DocumentRoot "C:/Program Files/Apache Software Foundation/Apache2.2/htdocs/marte"
 <Directory "C:/Program Files/Apache Software Foundation/Apache2.2/htdocs/marte">
 DirectoryIndex index.html
 Options Indexes FollowSymlinks
 AllowOverride None
 Order allow,deny
 Allow from all
 </Directory>

 ErrorLog "logs/marte.error.log"
 CustomLog "logs/marte.access.log" common
</VirtualHost>
```

Figura 5.103: Fichero de configuración de los servidores virtuales



Figura 5.104: Acceso al servidor virtual **www.tierra.com**



Figura 5.105: Acceso al servidor virtual **www.marte.com**

## 5.18. HTTPS y certificados digitales

### 1. Certificado digital verificado

- 1.1. Inicia sesión en DesarrolloW7XX.
- 1.2. Inicia *Firefox*.
- 1.3. Conéctate a <https://www.bbva.es>.
- 1.4. Observa en la URL que el protocolo usado es https.
- 1.5. Pincha en la parte izquierda de la URL.
- 1.6. Pincha sobre **Más información** para consultar el certificado digital que ha enviado el servidor web y responde de a las siguientes preguntas, Figura 5.106.
  - a) ¿Qué algoritmo de clave simétrica se ha utilizado para cifrar la información que viaja por la red? *AES* ¿Cuál es la longitud de la clave utilizada? *128 bits*.
  - b) ¿Cuál es el periodo de validez del certificado? *Del 18/08/2014 al 14/08/2015*.

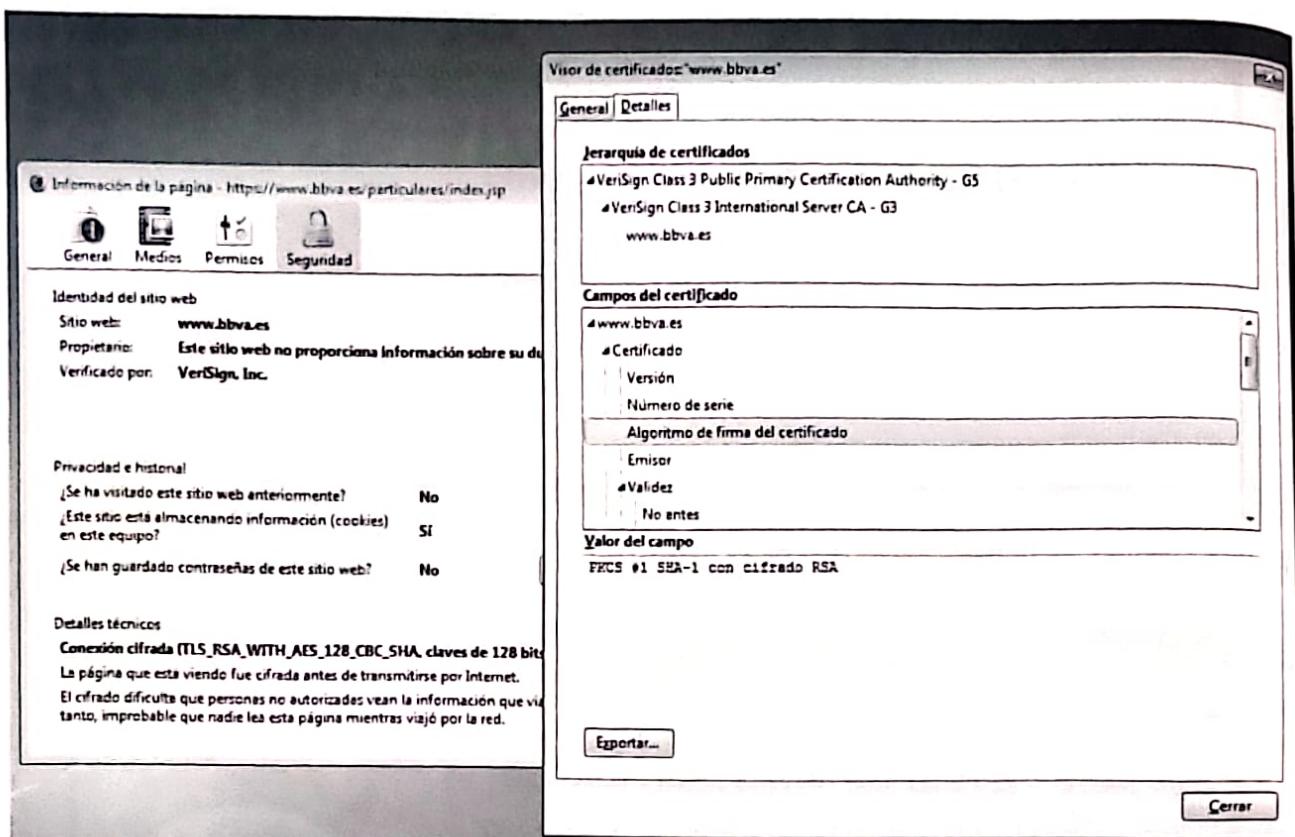


Figura 5.106: Certificado digital

- c) ¿Qué función resumen (hash) ha utilizado la autoridad de certificación para firmar el certificado? *SHA1*.
  - d) ¿Qué algoritmo de clave asimétrica ha utilizado la autoridad de certificación para firmar el certificado? *RSA*.
  - e) ¿De qué tamaño es la clave pública del certificado? *2048 bits*.
  - f) ¿Qué autoridad de certificación ha firmado el certificado? *VeriSign Class 3 International Server CA - G3* ¿De quién depende? *VeriSign Class 3 Public Primary Certification Authority - G5*.
- 1.7. En el menú de *Firefox* accede a **Opciones, Opciones, Pestaña Avanzado, Pestaña Cifrado, Ver certificados** y busca el certificado de la autoridad certificadora que ha firmado el certificado, Figura 5.107.

## 2. Certificado no verificado

- 2.1. Inicia Firefox.
- 2.2. Conéctate a la url que indique el profesor.
- 2.3. El navegador muestra un mensaje de error indicando que no ha podido verificar el certificado que le ha enviado el servidor web, Figura 5.108.
- 2.4. Pincha en **Entiendo los riesgos**.
- 2.5. Pincha en **Añadir Excepción**, Figura 5.109. Observa que está marcada la opción **Guardar excepción de forma permanente**.

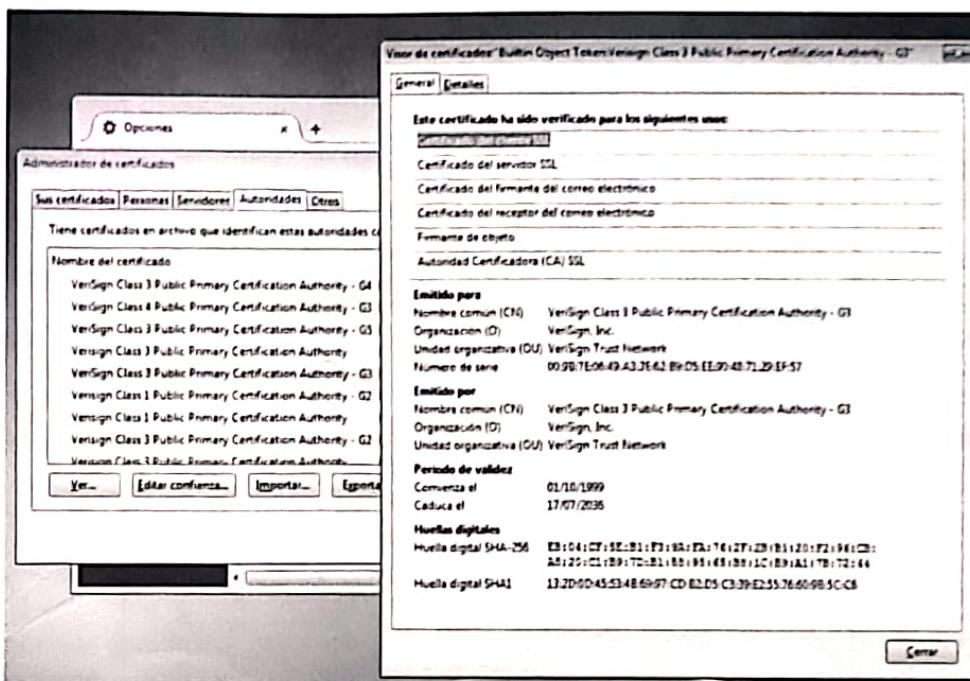


Figura 5.107: Certificado digital de la autoridad de certificación

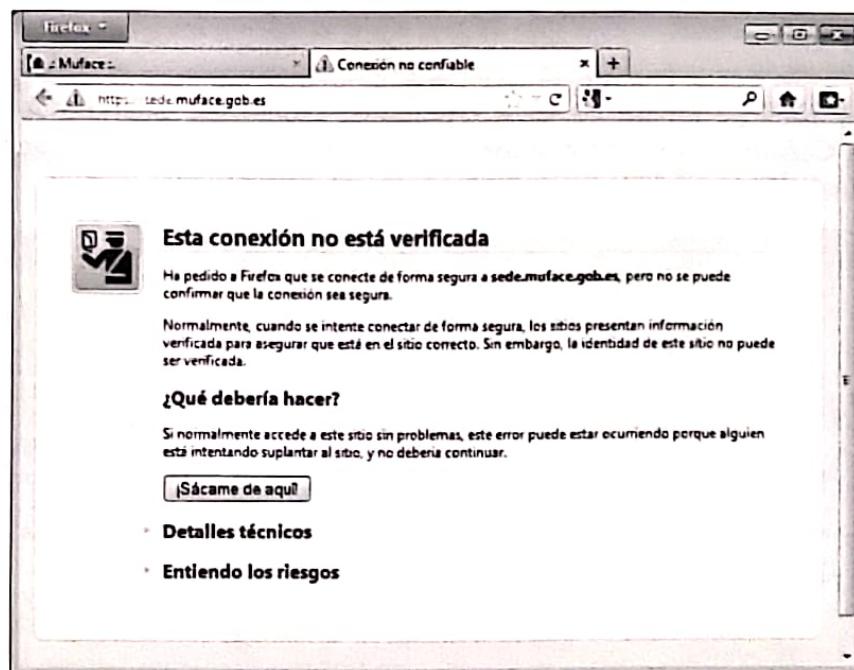


Figura 5.108: Aviso de certificado digital no verificado

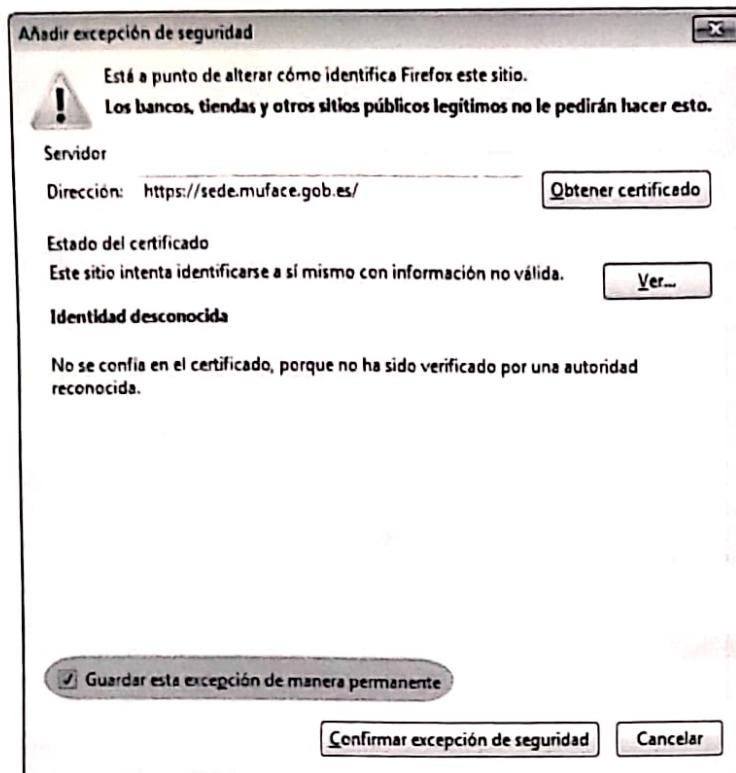


Figura 5.109: Ver el certificado digital no verificado

- 2.6. Pincha en **Obtener certificado** y en **Ver** para mostrar los datos del certificado digital que ha enviado el navegador.
- 2.7. Pincha en confirmar excepción de seguridad.
- 2.8. En el menú de Firefox accede a **Opciones, Opciones, pestaña Avanzado, Pestaña Cifrado, Ver certificados** busca el certificado del servidor que has aceptado y elimínalo, Figura 5.110.

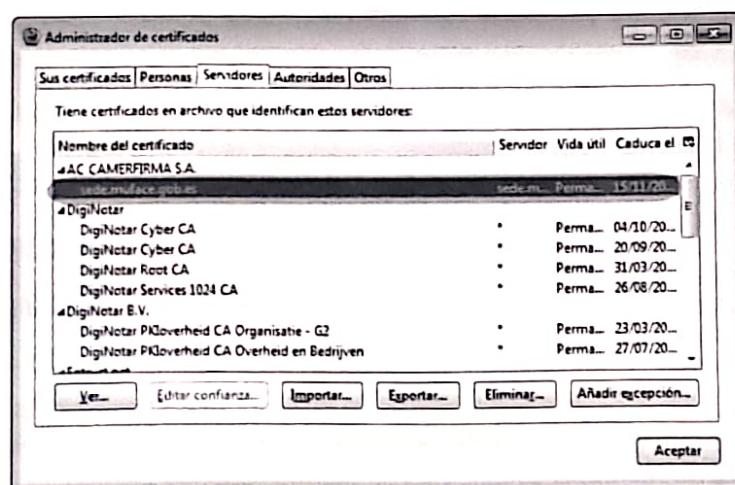


Figura 5.110: Eliminar certificado del servidor

## 5.19. Servidor virtual HTTPS por defecto en Linux

Realiza la siguiente configuración en el servidor *Apache* instalado en **ServidorLinuxXX**.

- Habilita el servidor virtual por defecto.
- Deshabilita los servidores virtuales creados en las prácticas anteriores.
- Habilita el modulo *mod\_ssl*.
- Habilita el servidor virtual *ssl* por defecto.

Prueba la configuración.

1. Inicia una sesión en **ServidorLinuxXX** con un usuario con privilegios de administración.
2. Habilita el servidor virtual por defecto de *Apache*.

```
sudo a2ensite 000-default
```

3. Verifica que dentro del directorio */etc/apache2/sites-enabled* se ha creado el enlace **000-default.conf**.
4. Deshabilita los servidores virtuales creados en prácticas anteriores.

```
sudo a2dissite software
sudo a2dissite hardware
```

5. Reinicia el servidor para que los cambios tengan efecto.
6. Habilita el módulo *modssl* que permite usar *https*, Figura 5.111.

```
sudo a2enmod ssl
```

```
alumno@ServidorLinux01:/etc/apache2$ sudo a2enmod ssl
[sudo] password for alumno:
Considering dependency setenvif for ssl:
Module setenvif already enabled
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Module socache_shmcb already enabled
Enabling module ssl.
See /usr/share/doc/apache2/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
alumno@ServidorLinux01:/etc/apache2$
```

Figura 5.111: Habilitar el modulo modssl

7. Reinicia el servidor para que los cambios tengan efecto.

```
If you just change the port or add more ports here, you will likely also
have to change the VirtualHost statement in
/etc/apache2/sites-enabled/000-default.conf

Listen 80

<IfModule ssl_module>
 Listen 443
</IfModule>

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

vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

Figura 5.112: Fichero /etc/apache2/port.conf

8. Consulta el fichero /etc/apache2/port.conf y observa que si habilita el modulo *ssl* el servidor escuchará en el puerto 443, Figura 5.112.
9. Verifica que el servidor escucha en los puertos 80/TCP y 443/TCP.

```
netstat -ltn
```

10. Accede al directorio /etc/apache2/sites-availables y observa que existe un fichero denominado **default-ssl.conf** que contiene la configuración por defecto de un servidor HTTPS
11. Habilita el servidor virtual ssl defecto (default-ssl.conf) de *Apache*.

```
sudo a2ensite default-ssl
```

12. Reinicia el servidor para que los cambios tengan efecto.
13. Consulta el fichero /etc/apache2/sites-availables/default-ssl.conf y observa su configuración. Fíjate en las directivas que habilitan SSL y que definen la ruta del certificado digital que usará el servidor, Figuras 5.113 y 5.114.

El servidor utiliza por defecto un certificado digital autofirmado que se ha creado al instalar *Apache*. Un certificado autofirmado no está firmado por una autoridad de certificación (tercera parte de confianza ) y por tanto, no existen mecanismos automáticos que garanticen su autenticidad. Por eso los navegadores nos pedirán confirmación cuando el servidor se lo envíe.

14. Desde DesarrolloW7XX abre el navegador y establece una conexión a <http://192.168.1.X7>, Figura 5.115.
15. Desde DesarrolloW7XX abre el navegador y establece una conexión a <https://192.168.1.X7>, Figuras 5.116 y 5.117.

```
<IfModule mod_ssl.c>
 <VirtualHost _default_:443>
 ServerAdmin webmaster@localhost
 DocumentRoot /var/www/html

 # Available loglevels: trace0, ..., trace1, debug, info, notices,
 # error, crit, alert, emerg.
 # It is also possible to configure the loglevel for particular
 # modules, e.g.
 LogLevel info ssl:warn

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

 # For most configuration files from conf-available/, which are
 # enabled or disabled at a global level, it is possible to
 # include a line for only one particular virtual host. For example
 # following line enables the CGI configuration for this host only
 # after it has been globally disabled with "a2disconf".
 #Include conf-available/serve-cgi-bin.conf
```

Figura 5.113: Fichero /etc/apache2/sites-available/default-ssl.conf

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

A self-signed (snakeoil) certificate can be created by installing
the ssl-cert package. See
/usr/share/doc/apache2/README.Debian.gz for more info.
If both key and certificate are stored in the same file, on
SSLCertificateFile directive is needed.
SSLCertificateFile /etc/ssl/certs/ssl-cert-snakeoil.pem
SSLCertificateKeyFile /etc/ssl/private/ssl-cert-snakeoil.key

Server Certificate Chain:
Point SSLCertificateChainFile at a file containing the
concatenation of PEM encoded CA certificates which form the
certificate chain for the server certificate. Alternatively
```

Figura 5.114: Fichero /etc/apache2/sites-available/default-ssl.conf

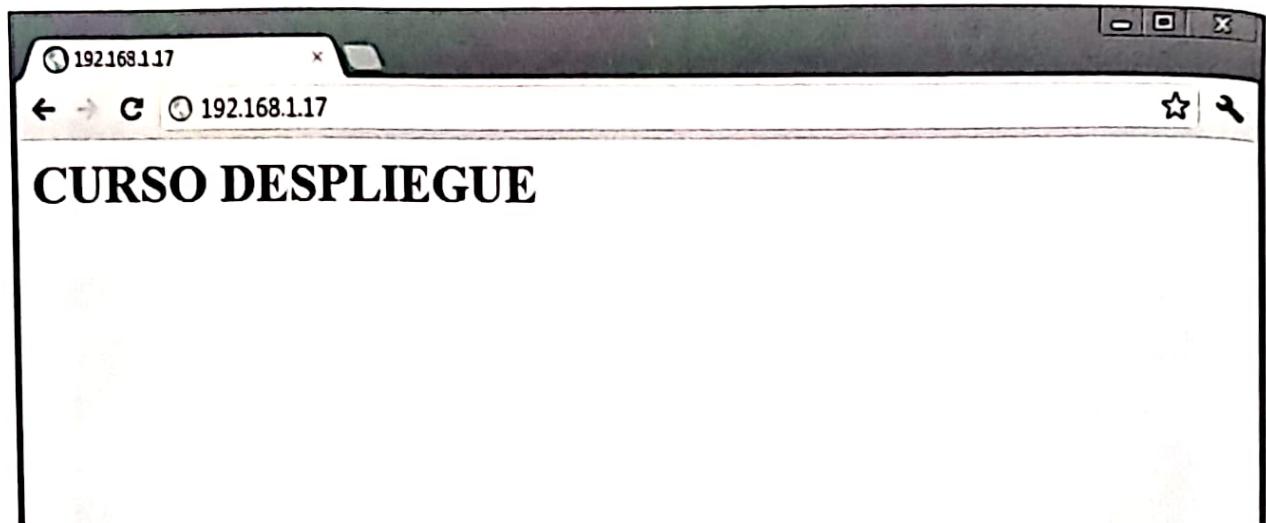


Figura 5.115: Conexión http

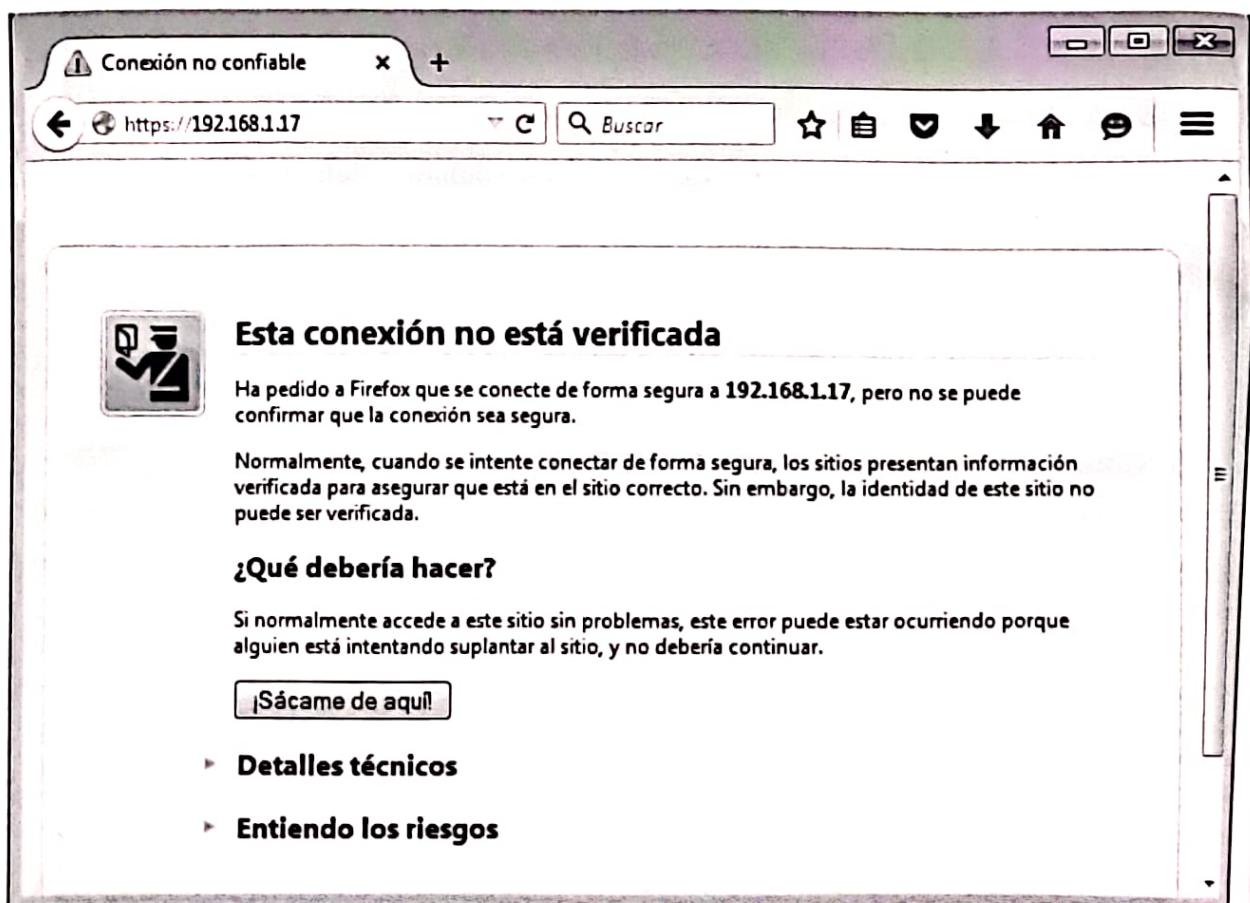


Figura 5.116: Conexión https

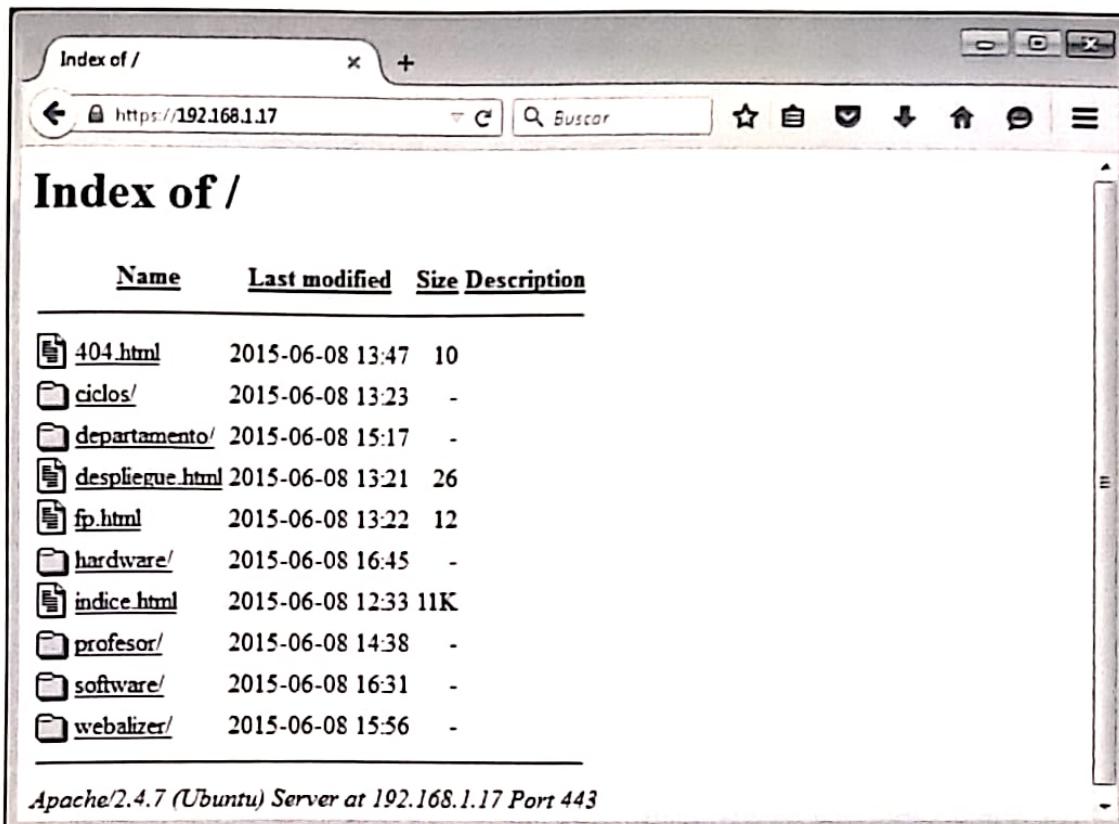


Figura 5.117: Conexión https

## 5.20. Servidor virtual HTTPS en Linux

Realiza la siguiente configuración en el servidor *Apache* instalado en **ServidorLinuxXX**.

- Deshabilita el servidor virtual *ssl* por defecto (*default-ssl*).
- Crea un certificado digital autofirmado con *openssl* para el dominio **seguro.dawXX.net**.
- Crea y habilita un servidor virtual *https* para el dominio **seguro.dawXX.net**
  - Directorio raíz **/var/www/html/seguro/**.
    - Se servirá el fichero **index.html** si no se indica ningún fichero en la URL.
    - Se mostrará un listado del directorio raíz si no se solicita ningún fichero.
    - Podrán acceder todos los usuarios.
  - El *log* de errores será **/var/log/apache2/seguro.error.log**.
  - El *log* de accesos será **/var/log/apache2/seguro.access.log**, con formato *combined*.

Prueba la configuración.

1. Configura el servidor DNS de **ServidorW2008XX** o **ServidorW2012XX** para que resuelva el nombre **seguro.dawXX.net**. La dirección IP asociada al nombre será la IP de **ServidorLinuxXX** es decir, 192.168.1.X7, Figura 5.118.

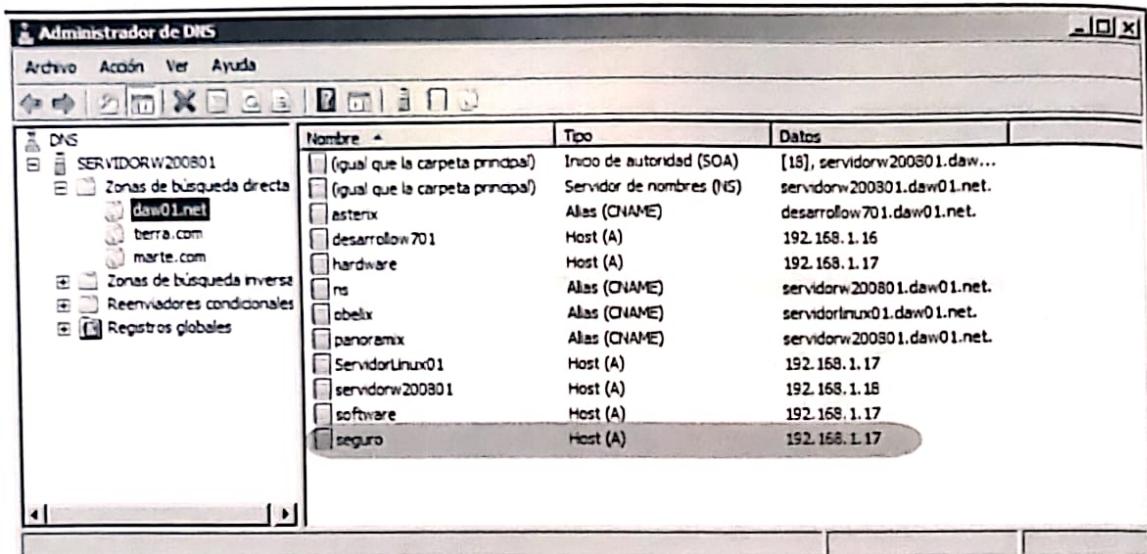


Figura 5.118: Configuración del servidor DNS en **ServidorWindowsXX**

2. Asegúrate que **DesarrolloW7XX** utiliza el servidor DNS que has configurado.
3. Inicia una sesión en **ServidorLinuxXX** con un usuario con privilegios de administración.
4. Crea el directorio `/var/www/html/seguro`.
5. Crea el fichero de texto `/var/www/html/seguro/index.html` con el contenido que quieras.
6. Crea un certificado digital autofirmado usando openssl.
  - 6.1. Sitúate en el directorio `home` del usuario con el que has iniciado sesión.
  - 6.2. Crea una clave privada RSA de 2048 bit, Figura 5.119.

```
openssl genrsa -out seguro.key 2048
```

```
a lumm0@ServidorLinux01:~$ openssl genrsa -out seguro.key 2048
Generating RSA private key, 2048 bit long modulus
.....+*
.....+*
e is 65537 (0x10001)
a lumm0@ServidorLinux01:~$
```

Figura 5.119: Creación de una clave privada

- 6.3. Genera una solicitud de certificado (CSR, *Certificate Signing Request*).

```
openssl req -new -key seguro.key -out seguro.csr
```

Introduce los datos del certificado, Figura 5.120

```
aumno@ServidorLinux01:~$ openssl req -new -key seguro.key -out seguro.csr
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.

Country Name (2 letter code) [AU]:ES
State or Province Name (full name) [Some-State]:Madrid
Locality Name (eg, city) []:Madrid
Organization Name (eg, company) [Internet Widgits Pty Ltd]:daw01
Organizational Unit Name (eg, section) []:daw01
Common Name (e.g. server FQDN or YOUR name) []:seguro.daw01.net
Email Address []:admin@daw01.net

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
aumno@ServidorLinux01:~$ _
```

Figura 5.120: Creación de la solicitud del certificado

Esta solicitud de certificado se la podrías enviar a una autoridad de certificación para que generase el certificado (CRT). En este caso lo vamos a firmar nosotros, vamos a crear un certificado autofirmado.

#### 6.4. Crea el certificado digital autofirmado usando la clave privada, Figura 5.121.

```
openssl x509 -req -days 365 -in seguro.csr -signkey seguro.key -out seguro.crt
```

```
aumno@ServidorLinux01:~$ openssl x509 -req -days 365 -in seguro.csr -signkey se
guo.key -out seguro.crt
Signature ok
subject=/C=ES/ST=Madrid/L=Madrid/O=daw01/OU=daw01/CN=seguro.daw01.net/emailAddre
ss=admin@daw01.net
Getting Private key
aumno@ServidorLinux01:~$ _
```

Figura 5.121: Creación del certificado digital autofirmado

#### 7. Copia la clave y el certificado en los directorios que utiliza por defecto *Apache* y configura los permisos adecuados.

```
sudo mv seguro.key /etc/ssl/private/
sudo mv seguro.crt /etc/ssl/certs/
sudo chown root:ssl-cert /etc/ssl/private/seguro.key
sudo chmod 640 /etc/ssl/private/seguro.key
sudo chown root:root /etc/ssl/certs/seguro.crt
```

#### 8. Crea el fichero `/etc/apache/site-available/seguro.conf` con las siguientes directivas, Figura 5.122.

```
<IfModule mod_ssl.c>
 <VirtualHost _default_:443>
 ServerName seguro.daw01.net

 DocumentRoot /var/www/html/seguro

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

 <Directory /var/www/html/seguro>
 Options Indexes FollowSymLinks
 AllowOverride None
 Require all granted
 </Directory>

 SSLEngine on
 SSLCertificateFile /etc/ssl/certs/seguro.crt
 SSLCertificateKeyFile /etc/ssl/private/seguro.key

 </VirtualHost>
</IfModule>
```

Figura 5.122: Fichero de configuración del servidor seguro

9. Deshabilita el servidor ssl por defecto.

```
sudo a2dissite default-ssl
```

10. Habilita el servidor virtual seguro.

```
sudo a2ensite seguro
```

11. Verifica que dentro del directorio /etc/apache2/sites-enabled se ha creado el enlace seguro.conf.
12. Reinicia el servidor para que los cambios tengan efecto.
13. Desde DesarrolloW7XX abre el navegador y establece una conexión a <https://seguro.dawXX.net>, Figuras 5.123, 5.124, 5.125 y 5.126.

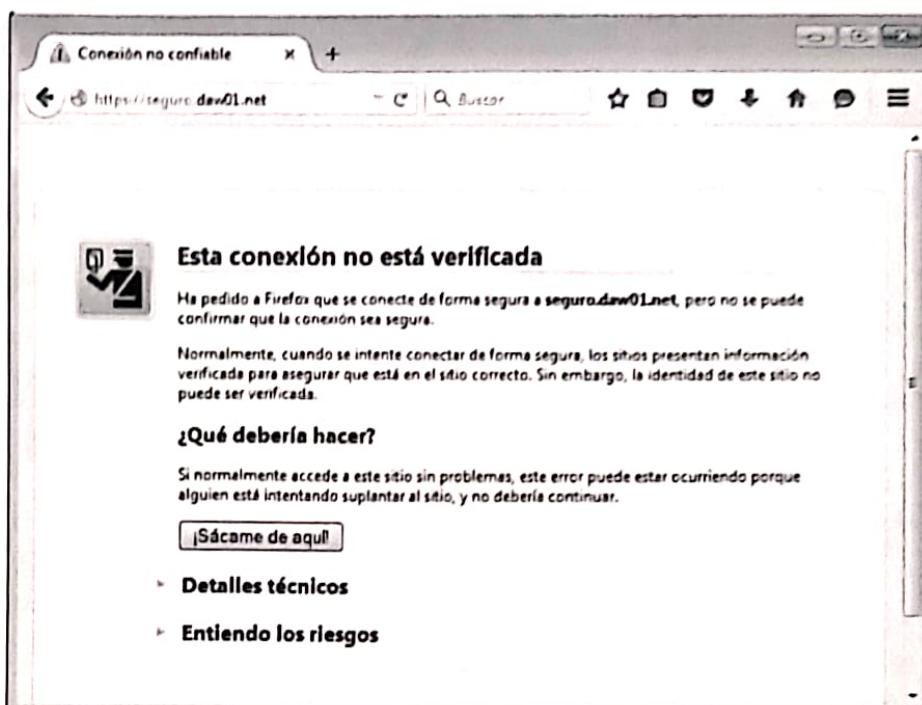


Figura 5.123: Conexión https

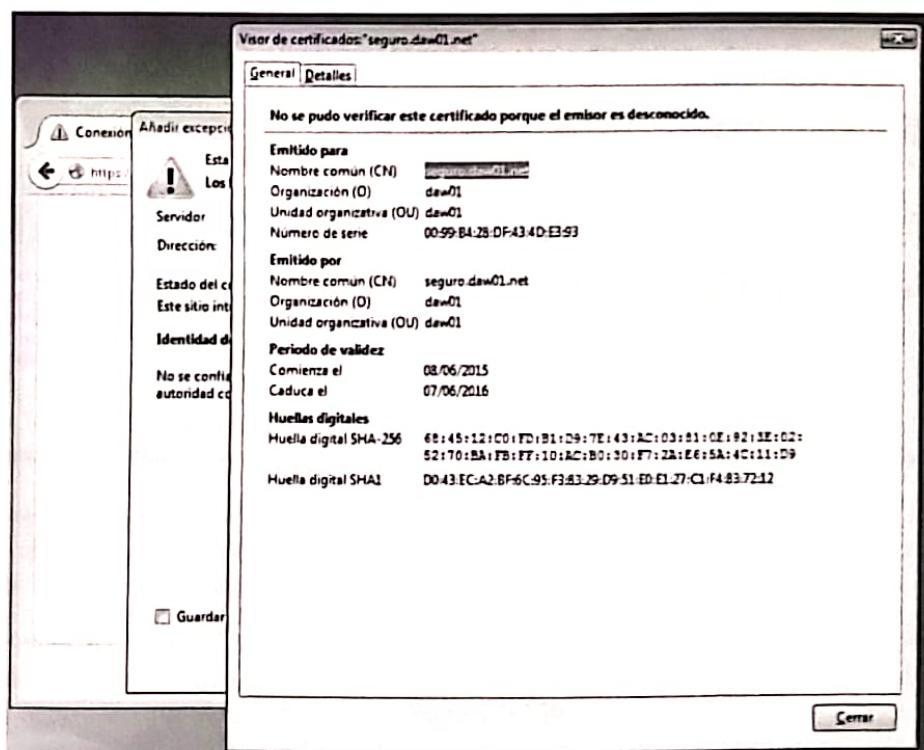


Figura 5.124: Conexión https

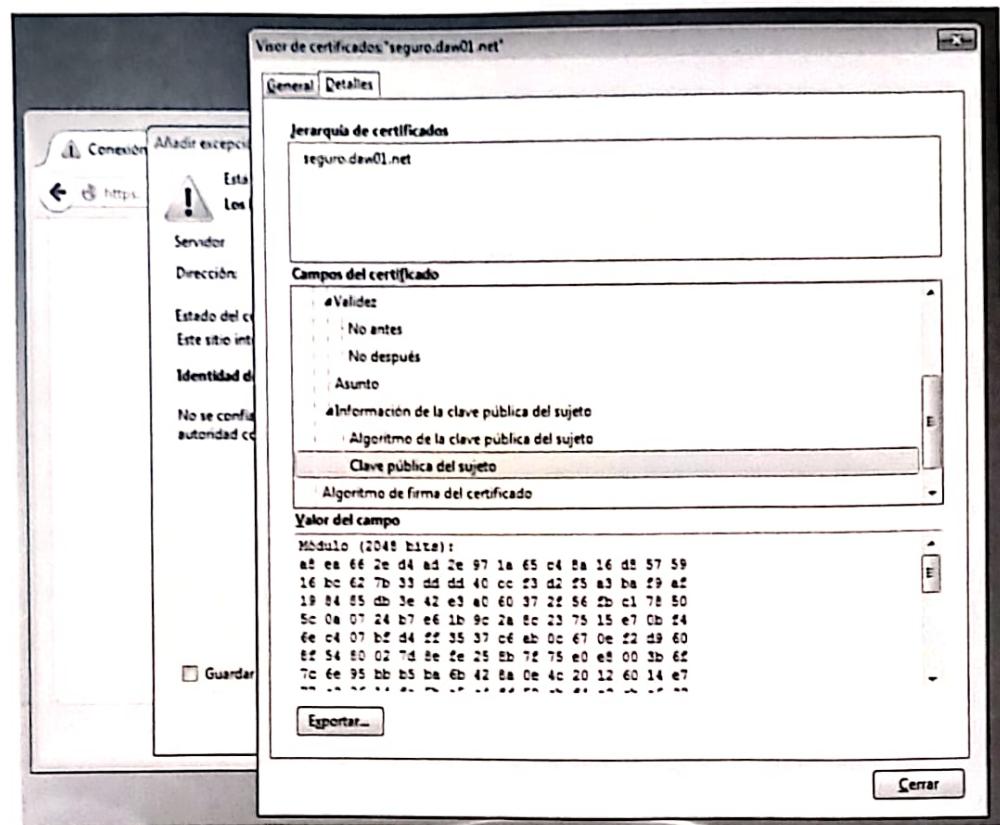


Figura 5.125: Conexión https

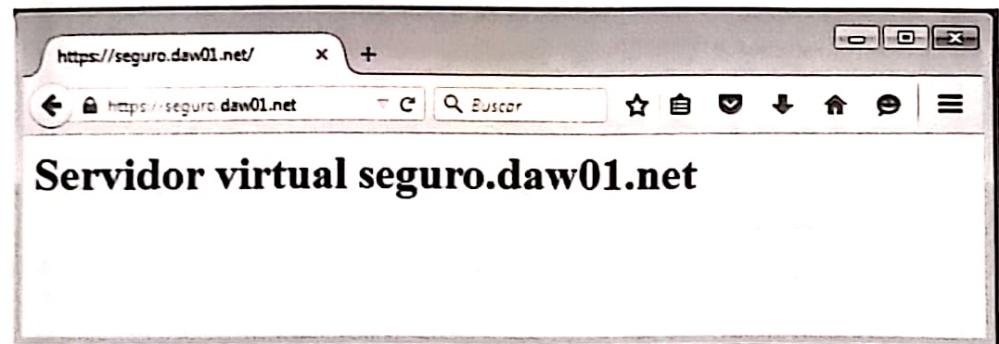


Figura 5.126: Conexión https

## 5.21. Servidor virtual HTTPS por defecto en Windows

Realiza la siguiente configuración en el servidor *Apache* instalado en **ServidorW2008XX**.

- Deshabilita los servidores virtuales creados en las prácticas anteriores.
- Habilita el modulo *mod\_ssl*.
- Habilita el servidor virtual *ssl* por defecto

Prueba la configuración.

1. Inicia una sesión en **ServidorW2008XX** con un usuario con privilegios de administración.
2. Deshabilita los servidores virtuales creados en prácticas anteriores.
- 2.1. Edita el fichero **C:\Program Files\Apache Software Foundation\Apache2.2\conf\httpd.conf** y comenta la directiva **Include** del fichero **conf/extra/httpd-vhost.conf**.
- 2.2. Reinicia el servidor para que los cambios tengan efecto.
3. Edita el fichero **C:\Program Files\Apache Software Foundation\Apache2.2\conf\httpd.conf** y habilita el módulo *mod\_ssl* eliminando el comentario de la directivas **LoadModule**, Figura 5.127.

```
#LoadModule mime_magic_module modules/mod_mime_magic.so
LoadModule negotiation_module modules/mod_negotiation.so
#LoadModule proxy_module modules/mod_proxy.so
#LoadModule proxy_ajp_module modules/mod_proxy_ajp.so
#LoadModule proxy_balancer_module modules/mod_proxy_balancer.so
#LoadModule proxy_connect_module modules/mod_proxy_connect.so
#LoadModule proxy_ftp_module modules/mod_proxy_ftp.so
#LoadModule proxy_http_module modules/mod_proxy_http.so
#LoadModule proxy_scgi_module modules/mod_proxy_scgi.so
#LoadModule reqtimeout_module modules/mod_reqtimeout.so
#LoadModule rewrite_module modules/mod_rewrite.so
LoadModule setenvif_module modules/mod_setenvif.so
#LoadModule spelling_module modules/mod_speling.so
#LoadModule ssl_module modules/mod_ssl.so
#LoadModule status_module modules/mod_status.so
#LoadModule substitute_module modules/mod_substitute.so
#LoadModule unique_id_module modules/mod_unique_id.so
LoadModule userdir_module modules/mod_userdir.so
#LoadModule usertrack_module modules/mod_usertrack.so
#LoadModule version_module modules/mod_version.so
```

Figura 5.127: Habilitar el módulo *mod\_ssl*

4. Habilita el servidor virtual *ssl* defecto (*default-ssl*) de *Apache*. Edita el fichero **C:\Program Files\Apache Software Foundation\Apache2.2\conf\httpd.conf** y eliminana el comentario de la directiva **Include** del fichero **conf/extra/httpd-ssl.conf**, Figura 5.128.
5. Si observas en el fichero **C:\Program Files\Apache Software Foundation\Apache2.2\conf\extra\httpd-ssl.conf** existen dos directivas para definir el certificado digital y la clave privada del servidor (que debemos crear), Figura 5.129.
6. Crea un certificado digital autofirmado usando *openssl*.
  - 6.1. Abre un terminal.
  - 6.2. Accede al directorio **C:\Program Files\Apache Software Foundation\Apache2.2\conf**.
  - 6.3. Ejecuta el comando **C:\Program Files\Apache Software Foundation\Apache2.2\bin\openssl**.

```
#Include conf/extra/httpd-vhosts.conf
Local access to the Apache HTTP Server Manual
#Include conf/extra/httpd-manual.conf

Distributed authoring and versioning (WebDAV)
#Include conf/extra/httpd-dav.conf

Various default settings
#Include conf/extra/httpd-default.conf

Secure (SSL/TLS) connections
#Include conf/extra/httpd-ssl.conf

Note: The following must be present to support
starting without SSL on platforms with no /dev/random equivalent
but a statically compiled-in mod_ssl.

<IfModule ssl_module>
SSLRandomSeed startup builtin
```

Figura 5.128: Habilitar el servidor virtual https

```
#SSLCipherSuite RC4-SHA:AES128-SHA:HIGH:MEDIUM:!aNULL:!MD5
#SSLHonorCipherOrder on

Server Certificate:
Point SSLCertificateFile at a PEM encoded certificate. If
the certificate is encrypted, then you will be prompted for a
pass phrase. Note that a kill -HUP will prompt again. Keep
in mind that if you have both an RSA and a DSA certificate you
can configure both in parallel (to also allow the use of DSA
ciphers, etc.)
SSLCertificateFile "C:/Program Files/Apache Software Foundation/Apache2.2/conf/server.crt"
#SSLCertificateFile "C:/Program Files/Apache Software Foundation/Apache2.2/conf/server-dsa.crt"

Server Private Key:
If the key is not combined with the certificate, use this
directive to point at the key file. Keep in mind that if
you've both a RSA and a DSA private key you can configure
both in parallel (to also allow the use of DSA ciphers, etc.)
SSLCertificateKeyFile "C:/Program Files/Apache Software Foundation/Apache2.2/conf/server.key"
#SSLCertificateKeyFile "C:/Program Files/Apache Software Foundation/Apache2.2/conf/server-dsa.key"
```

Figura 5.129: Fichero httpd-ssl.conf

```
C:\Program Files\Apache Software Foundation\Apache2.2\conf>"c:\Program Files\Apache Software Foundation\Apache2.2\bin\openssl.exe"
OpenSSL> genrsa -out server.key 2048
Loading 'screen' into random state - done
Generating RSA private key, 2048 bit long modulus
.....+
.....+
e is 65537 <0x10001>
OpenSSL> _
```

Figura 5.130: Creación de una clave privada

## 6.4. Crea una clave privada RSA de 2048 bit, Figura 5.130.

```
OpenSSL> genrsa -out server.key 2048
```

6.5. Genera una solicitud de certificado (CSR, *Certificate Signing Request*).

```
OpenSSL> req -config openssl.cnf -new -key server.key -out server.csr
```

Introduce los datos del certificado, Figura 5.131

```
OpenSSL> req -new -key server.key -out server.csr
Unable to load config info from /usr/local/ssl/openssl.cnf
error in req
OpenSSL> req -config openssl.cnf -new -key server.key -out server.csr
Loading 'screen' into random state - done
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.

Country Name <2 letter code> [AU]:ES
State or Province Name <full name> [Some-State]:Madrid
Locality Name <eg, city> []:Madrid
Organization Name <eg, company> [Internet Widgits Pty Ltd]:dau01.net
Organizational Unit Name <eg, section> []:dau01.net
Common Name <e.g. server FQDN or YOUR name> []:servidorwindows01.dau01.net
Email Address []:admin@dau01.net

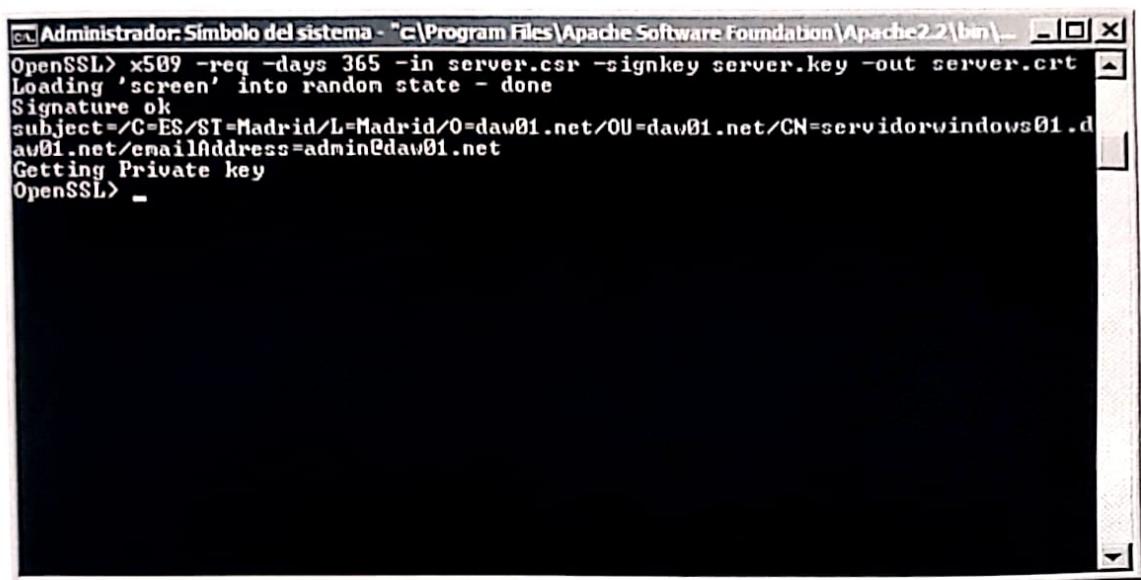
Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
OpenSSL>
```

Figura 5.131: Creación de la solicitud del certificado

Esta solicitud de certificado se la podrías enviar a una autoridad de certificación para que generase el certificado (CRT). En este caso lo vamos a firmar nosotros, vamos a crear un certificado autofirmado.

6.6. Crea el certificado digital autofirmado usando la clave privada, Figura 5.132.

```
openssl> x509 -req -days 365 -in server.csr -signkey server.key -out server.crt
```



```
Administrator: Símbolo del sistema - "c:\Program Files\Apache Software Foundation\Apache2.2\bin">_
OpenSSL> x509 -req -days 365 -in server.csr -signkey server.key -out server.crt
Loading 'screen' into random state - done
Signature ok
subject=/C=ES/ST=Madrid/L=Madrid/O=daw01.net/OU=daw01.net/CN=servidorwindows01.d
aw01.net/emailAddress=admin@daw01.net
Getting Private key
OpenSSL> _
```

Figura 5.132: Creación del certificado digital autofirmado

7. Reinicia el servidor para que los cambios tengan efecto.

8. Verifica que el servidor escucha en los puertos 80/TCP y 443/TCP.

```
netstat -a -p TCP -n
```

9. Desde DesarrolloW7XX abre el navegador y establece una conexión a <http://192.168.1.18>, Figura 5.133.

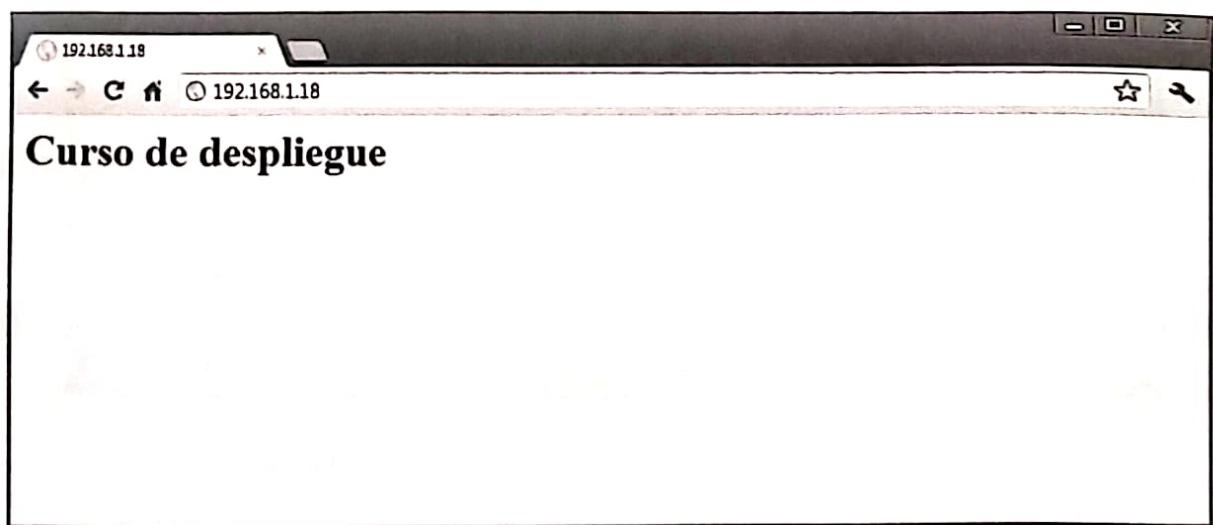


Figura 5.133: Conexión http

10. Desde DesarrolloW7XX abre el navegador y establece una conexión a `https://192.168.1.18`. Figuras 5.134 y 5.135.

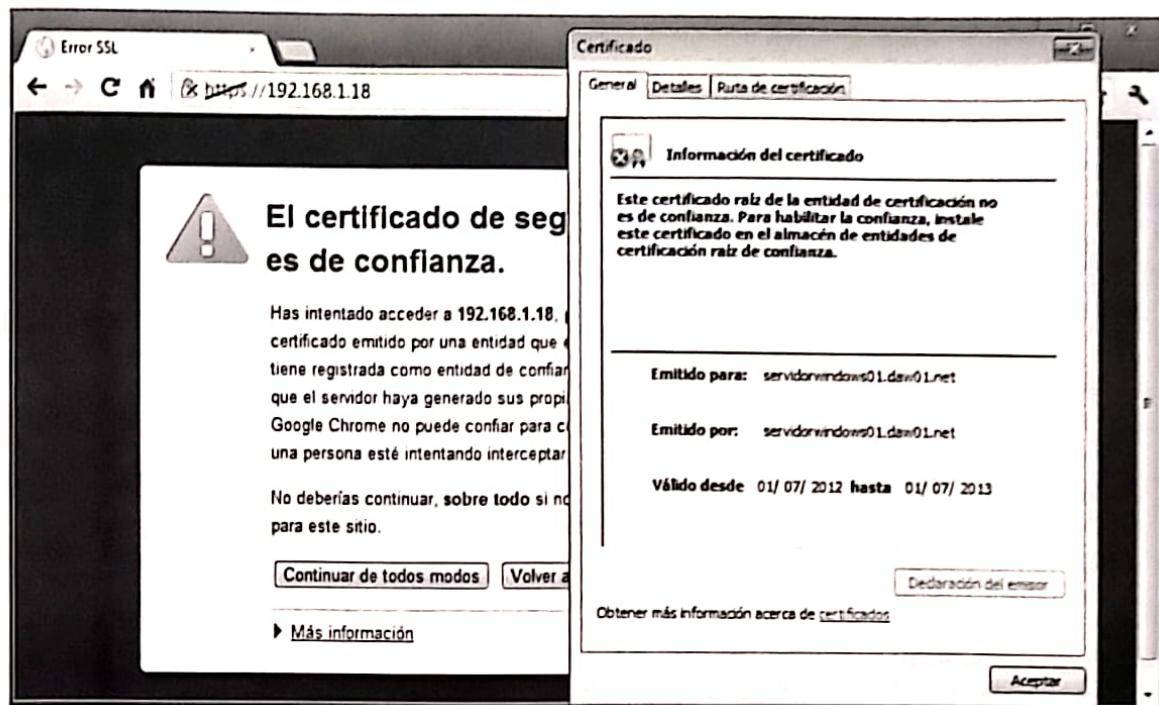


Figura 5.134: Conexión https

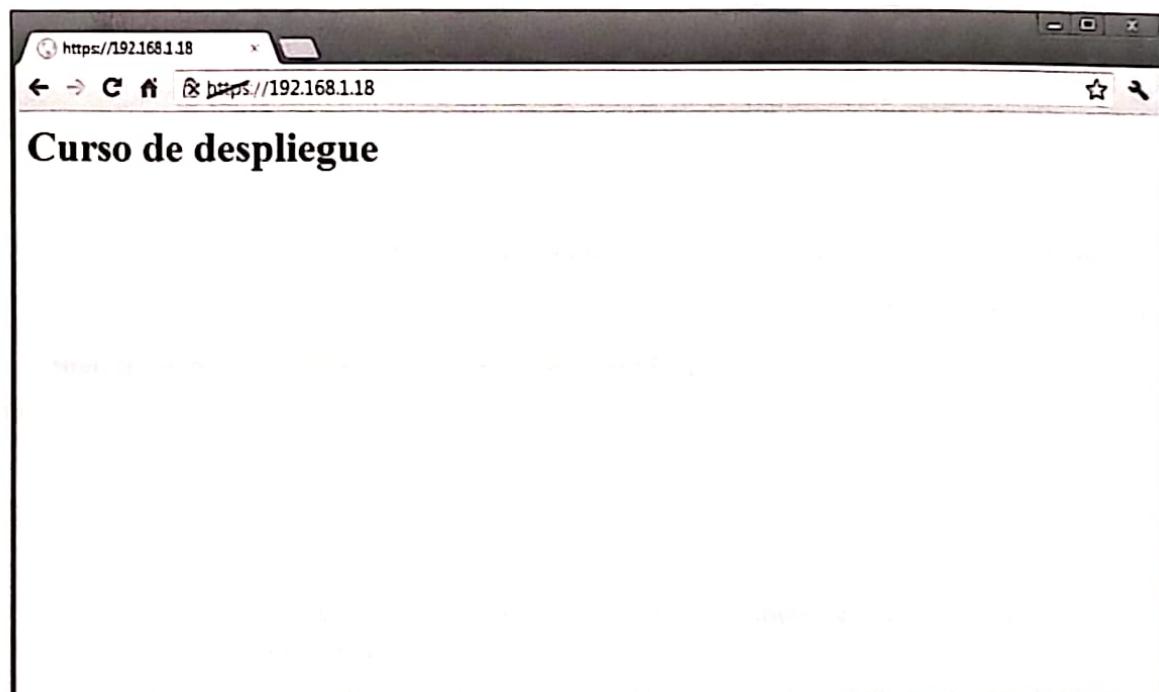


Figura 5.135: Conexión https

## 5.22. WebDav

En la máquina **ServidorLinuxXX** crea el directorio **/var/www/html/webdav** y configura el servidor virtual por defecto para que el directorio sea accesible desde clientes WebDav. Deberás configurar la autenticación HTTP *Digest* sobre el directorio **/var/www/html/webdav** para que solo puedan acceder los usuarios **admin1** y **admin2**. También tendrás que configurar los permisos adecuados para que los clientes puedan consultar, borrar, modificar, etc. ficheros del directorio.

### 1. Habilitar los módulos necesarios para WebDav

- 1.1. Inicia una sesión en **ServidorLinuxXX** con un usuario con privilegios de administrador.
- 1.2. Consulta el directorio **/etc/apache2/mods-available** y observa que entre los módulos disponibles para cargar están **dav** y **dav\_fs**.
- 1.3. Habilita los módulos ejecutando los siguientes comandos:

```
sudo a2enmod dav
sudo a2enmod dav_fs
```

- 1.4. Verifica que dentro del directorio **/etc/apache2/mods-enabled** se han creado enlaces simbólicos de los módulos **dav** y **dav\_fs** (ficheros **.conf** y **.load**) hacia **/etc/apache2/mod\_availables**
- 1.5. Consulta el fichero **/etc/apache2/mod\_availables/dav\_fs.conf** y observa su configuración por defecto, Figura 5.136.

```
DAVLockDB ${APACHE_LOCK_DIR}/DAVLock
```

Figura 5.136: Fichero **/etc/apache2/mod-available/dav\_fs.conf**

La directiva **DAVLockDB** define la ubicación de la base de datos que almacenará la información para controlar los bloqueos (control del acceso simultáneo a los mismos ficheros por parte de múltiples clientes).

- 1.6. Reinicia el servidor para que los cambios tengan efecto.

### 2. Configuración del directorio

- 2.1. Crea el directorio **/var/www/html/webdav** y dentro el fichero **prueba.html**.
- 2.2. Edita el fichero **/etc/apache/sites-available/000-default.conf** y añade las siguientes directivas, Figura 5.137.

```

<Directory /var/www/html/departamento>
 Options Indexes FollowSymLinks
 AllowOverride None
 AuthType Digest
 AuthName "informatica"
 AuthDigestProvider file
 AuthUserFile /etc/apache2/digest
 Require user admin1 admin2
</Directory>

<Directory /var/www/html/webdav>
 Dav On
 Options Indexes FollowSymLinks
 AllowOverride None
 AuthType Digest
 AuthName "informatica"
 AuthDigestProvider file
 AuthUserFile /etc/apache2/digest
 Require user admin1 admin2
</Directory>

```

Figura 5.137: Fichero /etc/apache/sites-available/000-default.conf

Como puedes observar se ha utilizado el fichero /etc/apache2/digest de usuarios y contraseñas creado en prácticas anteriores. Si no has realizado la práctica correspondiente deberías crearlo según se explica a continuación.

- Crea el fichero y añade el usuario **admin1** al dominio **informatica** (la opción -c es para crear el fichero).

```
sudo htdigest -c /etc/apache2/digest informatica admin1
```

- Añade el usuario **admin2** (no se usa la opción -c porque el fichero ya existe).

```
sudo htdigest /etc/apache2/digest informatica admin2
```

### 2.3. Reinicia el servidor para que los cambios tengan efecto.

## 3. Conexión desde un cliente WebDav

Para conectarse a través de WebDav es necesario utilizar un cliente específico. **Windows7** tiene integrado un cliente WebDav en sus explorador de archivos (tiene algunos problemas con conexiones sobre HTTPS).

- En **DesarrolloW7XX** accede a **Inicio, Equipo, botón secundario del ratón, Conectar a Unidad de Red**.
- Introduce los siguientes parámetros para establecer una conexión, Figura 5.138.

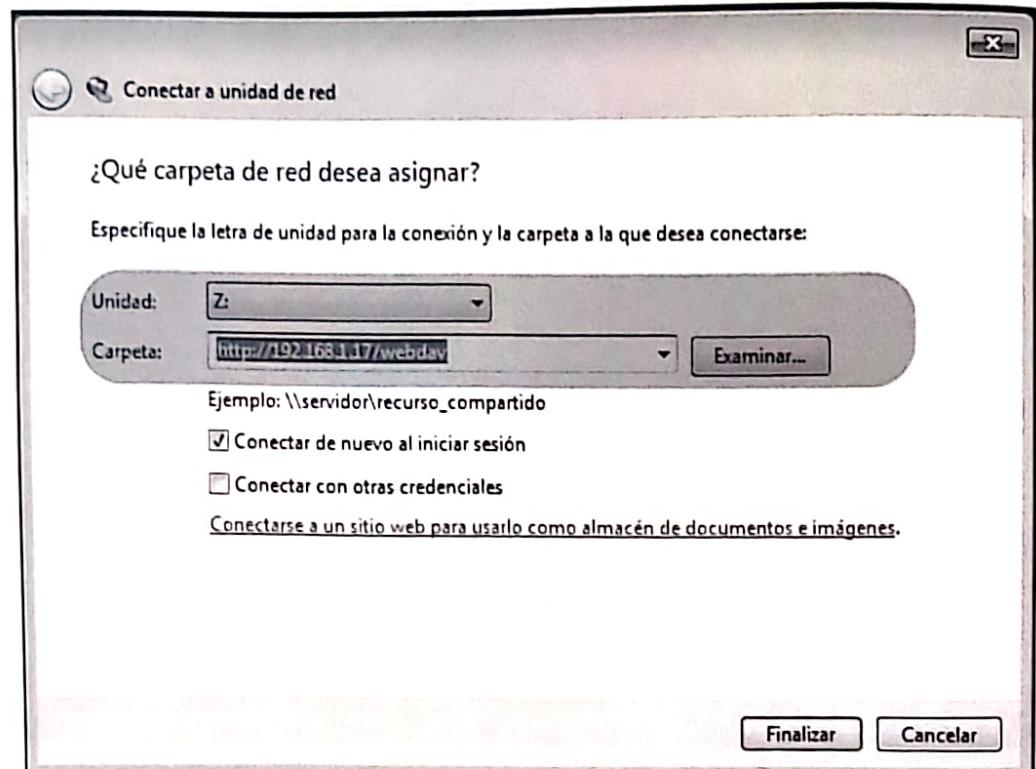


Figura 5.138: Configuración de la conexión WebDav desde DesarrrolloW7XX

3.3. Pincha en **Conectar** y establece una conexión como **admin1**, Figuras 5.139

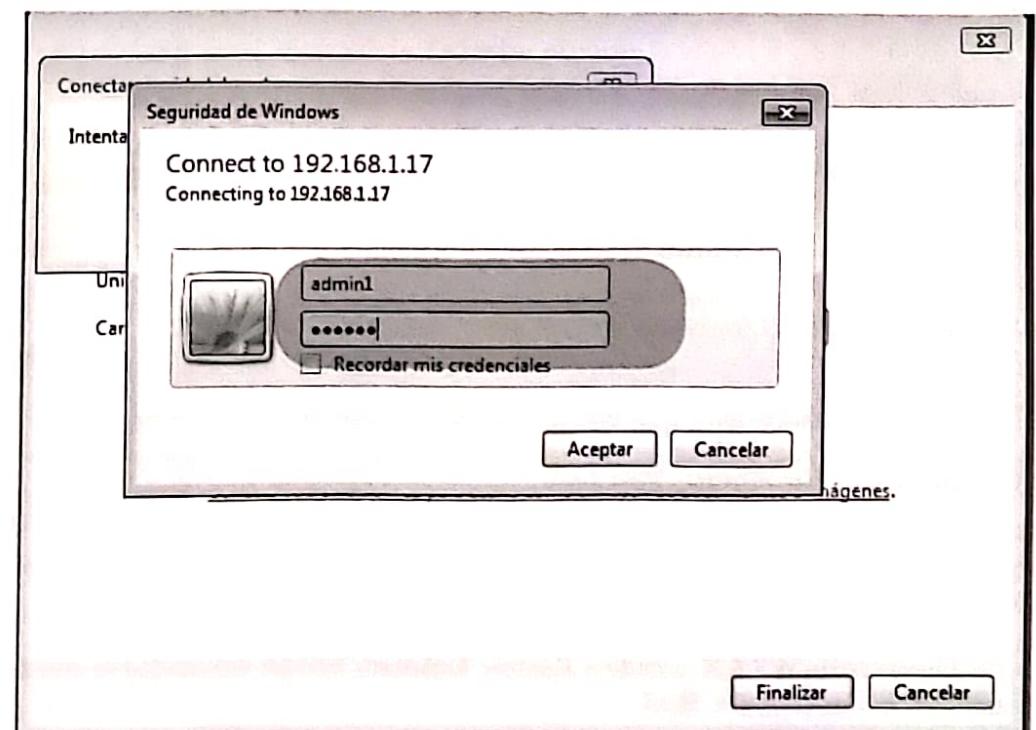


Figura 5.139: Conexión como *admin1*

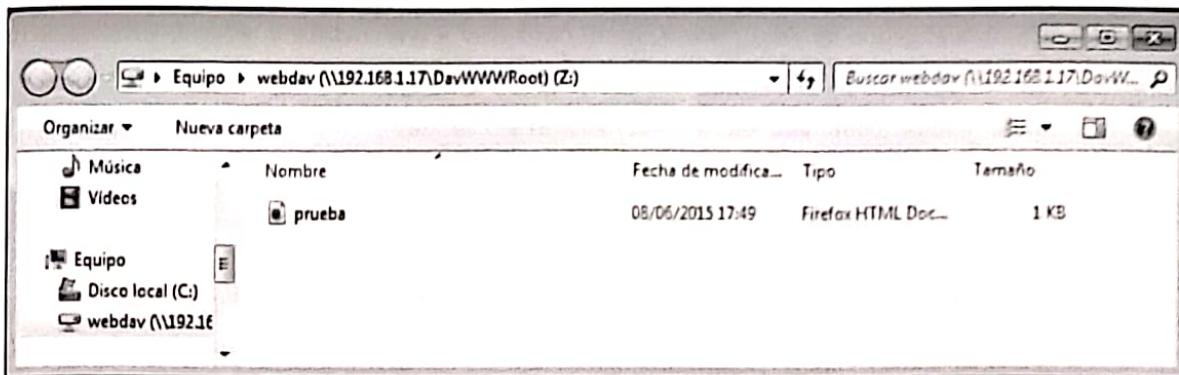


Figura 5.140: Conexión como admin1

3.4. Intenta crear un nuevo archivo. Observa que no tienes permisos.

#### 4. Configuración de los permisos adecuados

4.1. En **ServidorLinuxXX** consulta el fichero **/etc/apache2/envvars**. Observa el usuario y el grupo con los que se ejecuta *Apache*, Figura 5.141.

```
#!/bin/sh -e
#
envars - default environment variables for apache2ctl
#
this won't be correct after changing uid
unset HOME

for supporting multiple apache2 instances
if ["${APACHE_CONFDIR##*/etc/apache2-}" != "${APACHE_CONFDIR}"] ; then
 SUFFIX="-${APACHE_CONFDIR##*/etc/apache2-}"
else
 SUFFIX=
fi

Since there is no sane way to get the parsed apache2 config in scripts, some
settings are defined via environment variables and then used in apache2ctl,
/etc/init.d/apache2, /etc/logrotate.d/apache2, etc.
export APACHE_RUN_USER=www-data
export APACHE_RUN_GROUP=www-data
export APACHE_PID_FILE=/var/run/apache2$SUFFIX.pid
export APACHE_RUN_DIR=/var/run/apache2$SUFFIX
export APACHE_LOCK_DIR=/var/lock/apache2$SUFFIX
Only /var/log/apache2 is handled by /etc/logrotate.d/apache2.
export APACHE_LOG_DIR=/var/log/apache2$SUFFIX
```

Figura 5.141: Fichero /etc/apache2/envvars

4.2. Accede al directorio **/var/www/html**.

4.3. Consulta los permisos del directorio **/var/www/html/webdav**.

- 4.4. Cambia el grupo de /var/www/html/webdav a www-data y otórgale privilegios de escritura.

```
sudo chown root:www-data /var/www/html/webdav
sudo chmod 775 /var/www/html/webdav
```

- 4.5. Desde DesarrolloW7XX observa que ahora es posible crear un nuevo archivo, Figura 5.142.

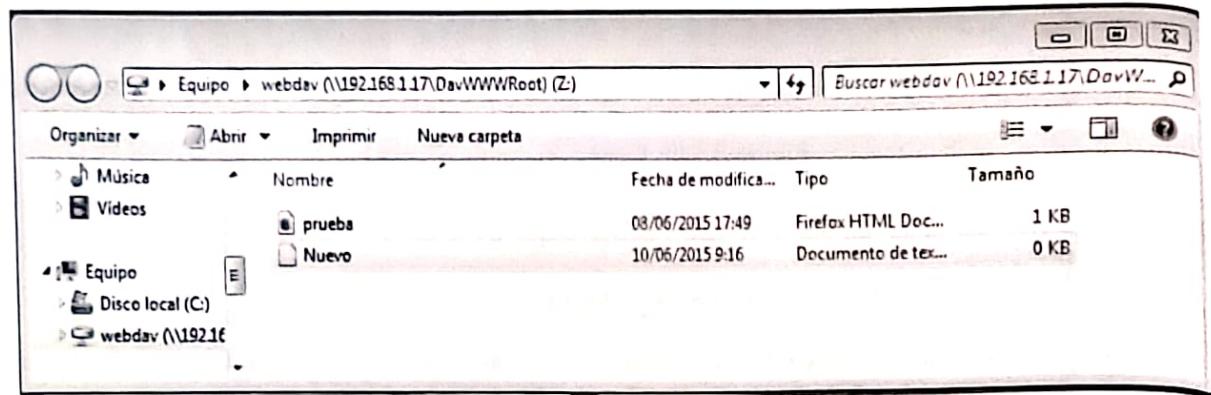


Figura 5.142: Nuevo archivo

- 4.6. Prueba a copiar, borrar, renombrar, etc.