

# **UT 5 – Administración** **Servidores Web**

## **P 5.2 – Servidor Apache Linux: Configuración y directivas**

**Cristina Moure Feijoó**

**2º DAW**

# Índice

Desarrollo de la práctica.....	2
1. Acceder a los ficheros de configuración de la carpeta /etc/apache2.....	2
2. Abre el fichero /etc/apache/apache2.conf y analiza su configuración. Observa que incluye con la directiva include a otros ficheros y directorios.....	2
Servidor Virtual por Defecto.....	3
3. Accede al directorio /etc/apache2/sites-available y comprueba que está creado el archivo default que contiene la configuración del servidor virtual por defecto.....	3
4. Accede a /etc/apache2/sites-enabled y comprueba que existe el fichero 000-default.conf que es un enlace simbólico a default (recuerda que los ficheros incluidos en /etc/apache2/sites-enabled se incluyen en /etc/apache2/apache2.conf en orden alfabético).....	4
Directivas.....	5
5. Consulta el fichero /etc/apache2/apache2.conf y comprueba cuál es el valor de las siguientes directivas: ServerRoot, User, Group y Timeout.....	5
6. ¿Se permiten conexiones persistentes? ¿Qué directiva define este comportamiento?.....	6
7. ¿Cuál es el fichero de errores? ¿Qué directiva lo define?.....	8
8. Consulta el fichero /etc/apache2/ports.conf y comprueba cual es el puerto en el que escucha las peticiones Apache. ¿En qué puerto escuchará también si se habilita el módulo mod_ssl?...9	
9. Consulta el fichero /etc/apache2/sites-available/default. Comprueba el valor de las directivas <VirtualHost> ... </VirtualHost>, DocumentRoot, ErrorLog y determina su funcionalidad.....	10
10. Consulta el fichero /etc/apache2/apache.conf. Comprueba el valor de la directiva <Directory> ... </Directory> y determina su funcionalidad.....	10

## Desarrollo de la práctica

### 1. Acceder a los ficheros de configuración de la carpeta /etc/apache2.

Primero vamos a situarnos en la carpeta apache2 ejecutando **cd /etc/apache2/**

Luego listamos el contenido de dicha carpeta con **ls** para ver que ahí se encuentran los ficheros de configuración.

```
alumno@alumnomv:/$ cd /etc/apache2/
alumno@alumnomv:/etc/apache2$ ls
apache2.conf  conf-enabled  magic         mods-enabled  sites-available
conf-available  envvars      mods-available  ports.conf    sites-enabled
```

2. Abre el fichero `/etc/apache2/apache2.conf` y analiza su configuración. Observa que incluye con la directiva `include` a otros ficheros y directorios.

Para abrir el fichero ejecutamos el comando: `cat /etc/apache2/apache.conf`

```
alumno@alumnov:~$ cat /etc/apache2/apache2.conf
```

Si leemos el fichero comprobamos que efectivamente con `include` incluye `mods-enabled`, `conf-enabled`, `ports.conf`, y `sites-enabled`.

```
# Include module configuration:
IncludeOptional mods-enabled/*.load
IncludeOptional mods-enabled/*.conf

# Include list of ports to listen on
Include ports.conf
```

```
# 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
```

## Servidor Virtual por Defecto

3. Accede al directorio `/etc/apache2/sites-available` y comprueba que está creado el archivo `default` que contiene la configuración del servidor virtual por defecto.

Con el comando `cd sites-available/` nos situamos en dicho directorio. Listamos su contenido con `ls` para comprobar que el archivo `default` está creado.

```
alumno@alumnov:/etc/apache2$ cd sites-available/
alumno@alumnov:/etc/apache2/sites-available$ ls
000-default.conf  default-ssl.conf
```

Ahora con el comando `cat 000.default.conf` abrimos el archivo para ver que es el que contiene la configuración del servidor virtual por defecto.

```

alumno@alumnov:/etc/apache2/sites-available$ cat 000-default.conf
<VirtualHost *:80>
    # The ServerName directive sets the request scheme, hostname and port that
    # 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) this
    # 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 ejemplo.com

    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html
    DirectoryIndex daw2.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
    # 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
</VirtualHost>

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

```

4. Accede a `/etc/apache2/sites-enabled` y comprueba que existe el fichero `000-default.conf` que es un enlace simbólico a `default` (recuerda que los ficheros incluidos en `/etc/apache2/sites-enabled` se incluyen en `/etc/apache2/apache2.conf` en orden alfabético).

Mediante los siguientes comandos nos situamos en `sites-enabled` y listamos su contenido. Vemos que también contiene en fichero `000-default.conf` que hemos abierto antes desde `sites-available`.

```

alumno@alumnov:/etc/apache2/sites-available$ cd ..
alumno@alumnov:/etc/apache2$ cd sites-enabled/
alumno@alumnov:/etc/apache2/sites-enabled$ ls
000-default.conf

```

Si abrimos de nuevo el archivo que contiene la configuración de apache (`apache.conf`) podemos comprobar que se incluye dicho archivo.

```

alumno@alumnov:/etc/apache2/sites-enabled$ cd ..

```

```
alumno@alumnov:/etc/apache2$ cat apache2.conf
```

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

## Directivas

5. Consulta el fichero `/etc/apache2/apache2.conf` y comprueba cuál es el valor de las siguientes directivas: `ServerRoot`, `User`, `Group` y `TimeOut`.

Ejecutamos el comando `cat` seguido de la ruta del fichero:

**cat `apache2.conf`** (o **cat `etc/apache/apache2.conf`** si nos encontramos en el directorio raíz).

```
alumno@alumnov:/etc/apache2$ cat apache2.conf
```

`ServerRoot`:

```
# Global configuration  
#  
  
#  
# ServerRoot: The top of the directory tree under which the server's  
# configuration, error, and log files are kept.  
#  
# NOTE! If you intend to place this on an NFS (or otherwise network)  
# mounted filesystem then please read the Mutex documentation (available  
# at <URL:http://httpd.apache.org/docs/2.4/mod/core.html#mutex>);  
# you will save yourself a lot of trouble.  
#  
# Do NOT add a slash at the end of the directory path.  
#  
#ServerRoot "/etc/apache2"
```

`User` y `Group`:

```
# These need to be set in /etc/apache2/envvars  
User ${APACHE_RUN_USER}  
Group ${APACHE_RUN_GROUP}
```

`TimeOut`:

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

Consulta la documentación de Apache y responde a las siguientes preguntas:

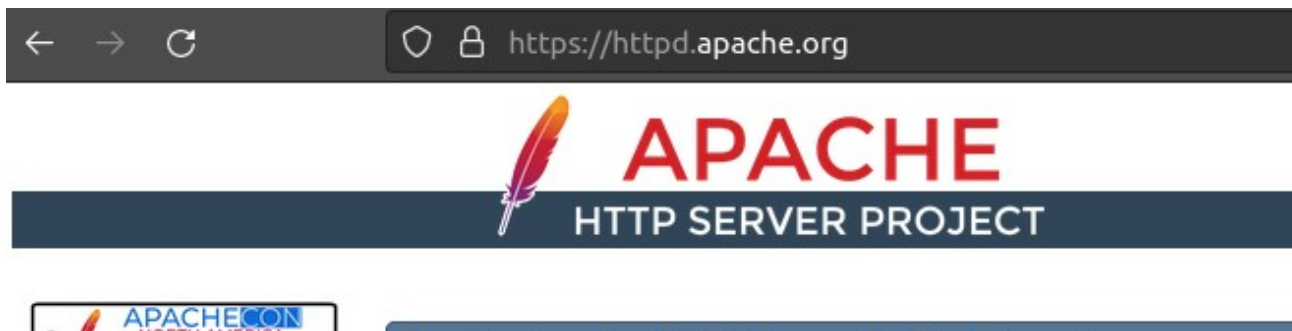
6. ¿Se permiten conexiones persistentes? ¿Qué directiva define este comportamiento?

Primero vamos a comprobar qué versión de apache tenemos instalada ejecutando **apache2 -v**

```
alumno@alumnov:/ $ apache2 -v
Server version: Apache/2.4.41 (Ubuntu)
Server built:   2022-06-14T13:30:55
```

Tenemos la versión 2.4.

La documentación la podemos encontrar en el siguiente enlace: <https://httpd.apache.org>



En la columna de la izquierda vemos que hay un apartado “Documentación”, pinchamos en la versión 2.4.

Documentation

- [Version 2.4](#)
- [Trunk \(dev\)](#)
- [Wiki](#)

Ahora para buscar las directivas pinchamos en “Módulos”.



Y luego en core

## ▣ Funcionalidad Básica y Módulos de MultiProcesamiento (MPM)

### core

Funcionalidades básicas del Servidor HTTP Apache que siempre están presentes.

Si bajamos encontraremos la directiva KeepAlive que es la que permite conexiones persistentes.

### ▣ Directiva KeepAlive

<u>Descripción:</u>	Enables HTTP persistent connections
<u>Sintaxis:</u>	KeepAlive On Off
<u>Valor por defecto:</u>	KeepAlive On
<u>Contexto:</u>	server config, virtual host
<u>Estado:</u>	Core
<u>Módulo:</u>	core

The Keep-Alive extension to HTTP/1.0 and the persistent connection feature of HTTP/1.1 provide long-lived HTTP sessions which allow multiple requests to be sent over the same TCP connection. In some cases this has been shown to result in an almost 50% speedup in latency times for HTML documents with many images. To enable Keep-Alive connections, set `KeepAlive On`.

For HTTP/1.0 clients, Keep-Alive connections will only be used if they are specifically requested by a client. In addition, a Keep-Alive connection with an HTTP/1.0 client can only be used when the length of the content is known in advance. This implies that dynamic content such as CGI output, SSI pages, and server-generated directory listings will generally not use Keep-Alive connections to HTTP/1.0 clients. For HTTP/1.1 clients, persistent connections are the default unless otherwise specified. If the client requests it, chunked encoding will be used in order to send content of unknown length over persistent connections.

When a client uses a Keep-Alive connection it will be counted as a single "request" for the `MaxConnectionsPerChild` directive, regardless of how many requests are sent using the connection.

#### Consulte también

- [MaxKeepAliveRequests](#)

También podemos ver la información de esta directiva en el archivo de configuración de apache:

```
alumno@alumnov:/etc/apache2$ cat apache2.conf
```

Y vemos que está activada, por lo que sí se permiten conexiones persistentes.

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

## 7. ¿Cuál es el fichero de errores? ¿Qué directiva lo define?

En la documentación de Apache podemos encontrar información sobre este fichero en el apartado “Ficheros de Log”.



# Apache HTTP Server Versión 2.4 Documentación

## Notas de la versión

- [Nuevas funcionalidades en Apache 2.3/2.4](#)
- [Nuevas funcionalidades en Apache 2.1/2.2](#)
- [Nuevas funcionalidades en Apache 2.0](#)
- [Actualizar a la versión 2.4 desde la 2.2](#)
- [Licencia Apache](#)

## Manual de Referencia

- [Compilar e Instalar](#)
- [Ejecutando Apache](#)

## Guía de Usuario

- [Empezando](#)
- [Enlazando Direcciones y Puertos](#)
- [Ficheros de Configuración](#)
- [Secciones de Configuración](#)
- [Almacenamiento de Contenido en Caché](#)
- [Negociación de Contenido](#)
- [Objetos Compartidos Dinámicamente \(DSO\)](#)
- [Variables de Entorno](#)
- [Ficheros de Log](#)



# APACHE

## HTTP SERVER PROJECT

Apache > HTTP Server > [Documentation](#) > [Version 2.4](#)

### Log Files

El fichero de errores es ErrorLog:

### Error Log

Related Modules	Related Directives
<a href="#">core</a>	<a href="#">ErrorLog</a>
	<a href="#">ErrorLogFormat</a>
	<a href="#">LogLevel</a>

The server error log, whose name and location is set by the [ErrorLog](#) directive, is the most important log file. **This is the place where Apache httpd will send diagnostic information and record any errors that it encounters in processing requests.** It is the first place to look when a problem occurs with starting the server or with the operation of the server, since it will often contain details of what went wrong and how to fix it.

Y la directiva que lo define es ErrorLogFormat:

The server error log, whose name and location is set by the [ErrorLog](#) directive, is the most important log file. This is the place where Apache httpd will send diagnostic information and record any errors that it encounters in processing requests. It is the first place to look when a problem occurs with starting the server or with the operation of the server, since it will often contain details of what went wrong and how to fix it.

The error log is usually written to a file (typically `error_log` on Unix systems and `error.log` on Windows and OS/2). On Unix systems it is also possible to have the server send errors to `syslog` or [pipe them to a program](#).

**The format of the error log is defined by the [ErrorLogFormat](#) directive,** with which you can customize what values are logged. A default is format defined if you don't specify one. A typical log message follows:



En el fichero de configuración encontramos también esta información:

```
alumno@alumnomv:/etc/apache2$ cat apache2.conf
```

```
# ErrorLog: The location of the error log file.
# If you do not specify an ErrorLog directive within a <VirtualHost>
# container, error messages relating to that virtual host will be
# logged here.  If you *do* define an error logfile for a <VirtualHost>
# container, that host's errors will be logged there and not here.
#
ErrorLog ${APACHE_LOG_DIR}/error.log
```

8. Consulta el fichero `/etc/apache2/ports.conf` y comprueba cual es el puerto en el que escucha las peticiones Apache. ¿En qué puerto escuchará también si se habilita el módulo `mod_ssl`?

Nos situamos en la carpeta de `apache2` y abrimos el fichero `ports.conf` ejecutando el comando **cat ports.conf** o **cat /etc/apache2/ports.conf** si nos encontramos en el directorio raíz.

```
alumno@alumnomv:/$ cd /etc/apache2/
alumno@alumnomv:/etc/apache2$ ls
apache2.conf  conf-enabled  magic          mods-enabled  sites-available
conf-available  envvars      mods-available  ports.conf    sites-enabled
alumno@alumnomv:/etc/apache2$ cat ports.conf
# 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
```

El puerto por el que escucha es el 80 y si habilitamos el módulo `mod_ssl` escuchará también por el 443.

9. Consulta el fichero `/etc/apache2/sites-available/default`. Comprueba el valor de las directivas `<VirtualHost> ... </VirtualHost>`, `DocumentRoot`, `ErrorLog` y determina su funcionalidad.

Lo consultamos con `cat 000-default.conf` o `cat /etc/apache2/sites-available/000-default.conf`

```
alumno@alumnov:/etc/apache2$ cd sites-available/
alumno@alumnov:/etc/apache2/sites-available$ ls
000-default.conf  default-ssl.conf
alumno@alumnov:/etc/apache2/sites-available$ cat 000-default.conf
<VirtualHost *:80>
    # The ServerName directive sets the request scheme, hostname and port that
    # 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) this
    # 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 ejemplo.com

    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html
    DirectoryIndex daw2.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
    # 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
</VirtualHost>

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

**VirtualHost** contiene las directivas que se aplican únicamente a un hostname específico o a una IP.

**DocumentRoot** indica el directorio desde el que va a servir archivos. Su valor es `/var/www/html`.

**ErrorLog** indica qué fichero tiene el registro de errores. Su valor actual es `error.log`.

10. Consulta el fichero `/etc/apache2/apache.conf`. Comprueba el valor de la directiva `<Directory> ... </Directory>` y determina su funcionalidad.

Lo consultamos con `cat apache2.conf` o `cat /etc/apache2/apache2.conf`

```
alumno@alumnov:/etc/apache2/sites-available$ cd ..
alumno@alumnov:/etc/apache2$ cat apache2.conf
```

```
# Sets the default security model of the Apache2 HTTPD server. It does
# not allow access to the root filesystem outside of /usr/share and /var/www.
# The former is used by web applications packaged in Debian,
# the latter may be used for local directories served by the web server. If
# your system is serving content from a sub-directory in /srv you must allow
# 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>
```

La directiva **Directory** contiene un grupo de directivas que se aplican al directorio especificado, sus subdirectorios y su contenido.

Contiene las siguientes directivas:

**FollowSymLinks:** controla la habilidad del servidor de seguir los enlaces simbólicos de un directorio..

**AllowOverride:** controla qué directivas se pueden situar en los ficheros .htaccess. En este caso está en None.

**Require all:** controla el acceso. Se encuentra en denied por lo que el acceso es denegado incondicionalmente.