

Servidor web Apache en Linux

Hamza Akdi

13/11/2025

Servicios de red e internet

Índice:

Índice:.....	2
Referencias	3
Introducción.....	4
1. Tabla de directorios utilizados:	5
2.Instalación y verificación del servicio	6
2.1 Instalación de Apache	6
2.2 Verificación del servicio y puertos.....	6
2.3 Análisis del archivo de configuración principal (apache2.conf)	8
3. Gestión del sitio web por defecto.....	10
4. Comprobación de los módulos habilitados	12
5. Añadir soporte PHP a Apache	15
5.1 Instalación e Integración.....	15
5.2 Verificación de funcionamiento	15
5.3 Creación de página de prueba	16
Comentarios	19

Referencias

Distribución Linux [Debian 11](#).

Manual oficial Apache.

Documentación sobre servicios en Debian y Ubuntu (systemctl).

Configuración de la red en Debian:

- <https://wiki.debian.org/es/NetworkConfiguration>
- <https://www.cyberciti.biz/faq/howto-configuring-network-interface-cards-on-debian/>

Monitorización de red:

- <https://wiki.debian.org/SystemMonitoring>
- <https://stackoverflow.com/questions/28917135/how-to-force-all-of-the-dhcp-clients-to-renew>

Documentación oficial de Apache:

- <https://httpd.apache.org/docs/2.4/es/>

Introducción

En esta práctica vamos a trabajar los siguientes aspectos:

- Instalación y configuración del servicio Apache en sistemas Debian (11, 12 o 13), entendiendo su estructura de ficheros y funcionamiento básico.
- Edición y análisis de los archivos de configuración principales, como apache2.conf, ports.conf y los ficheros de sitios dentro de /etc/apache2/sites-available/.
- Creación y personalización de páginas web básicas, modificando el contenido por defecto y gestionando el archivo de inicio mediante la directiva DirectoryIndex.
- Gestión de permisos y propiedad de archivos web, comprendiendo la importancia del usuario www-data.
- Configuración de páginas de error personalizadas (por ejemplo, error 404) y análisis de los ficheros de registro (access.log y error.log).
- Creación y activación de sitios virtuales (VirtualHost), simulando varios dominios o secciones independientes dentro del mismo servidor.
- **Modificación del archivo /etc/hosts** para definir nombres de dominio locales y acceder a los sitios configurados.

1. Tabla de directorios utilizados:

Directorio	Descripción / Función principal
<code>/etc/apache2/</code>	Directorio de configuración principal. Contiene todos los archivos de configuración.
<code>/etc/apache2/apache2.conf</code>	Archivo de configuración global. Define configuraciones básicas, la ubicación de logs y directorios web, e incluye otros archivos de configuración. Es el archivo más importante.
<code>/etc/apache2/ports.conf</code>	Define los puertos en los que Apache escucha las peticiones.
<code>/etc/apache2/sites-available/</code>	Contiene los archivos de configuración de todos los sitios web (VirtualHosts) creados.
<code>/etc/apache2/sites-enabled/</code>	Contiene enlaces simbólicos a los archivos de configuración de los sitios que están actualmente activos y sirviendo contenido .
<code>/etc/apache2/mods-available/</code>	Contiene los archivos de configuración de todos los módulos disponibles para Apache.
<code>/etc/apache2/mods-enabled/</code>	Contiene enlaces simbólicos a los módulos que están actualmente activos en el servidor.
<code>/var/www/</code>	Directorio raíz por defecto para alojar los archivos de los sitios web.
<code>/var/www/html/</code>	Ubicación por defecto del sitio web inicial (el sitio por defecto). El DocumentRoot por defecto apunta aquí.
<code>/var/log/apache2/</code>	Contiene los archivos de registro (logs) , como access.log (peticiones recibidas) y error.log (errores del servidor).

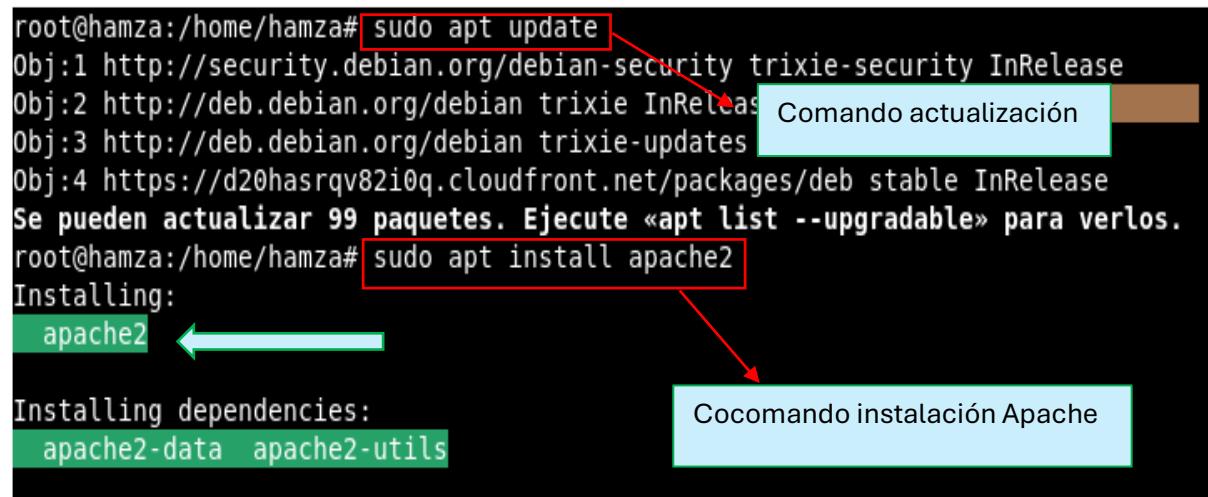
/usr/lib/apache2/	Directorio que contiene los binarios de los módulos (.so files) de Apache.
/etc/init.d/apache2 o /lib/systemd/system apache2.service	Scripts o unidades de servicio (systemctl) para controlar el servicio de Apache (iniciar, detener, reiniciar, verificar estado).

2. Instalación y verificación del servicio

Primero, instalaremos el servidor web y verificaremos su funcionamiento.

2.1 Instalación de Apache

Comenzaremos actualizando el sistema y los paquetes necesarios para instalar Apache2 en nuestro equipo.



```

root@hamza:/home/hamza# sudo apt update
Obj:1 http://security.debian.org/debian-security trixie-security InRelease
Obj:2 http://deb.debian.org/debian trixie InRelease
Obj:3 http://deb.debian.org/debian trixie-updates InRelease
Obj:4 https://d20hasrqv82i0q.cloudfront.net/packages/deb stable InRelease
Se pueden actualizar 99 paquetes. Ejecute «apt list --upgradable» para verlos.
root@hamza:/home/hamza# sudo apt install apache2
Installing:
apache2 ←
Installing dependencies:
apache2-data apache2-utils

```

Comando actualización

Cocomando instalación Apache

(Actualización de paquetes y descarga de Apache2)

2.2 Verificación del servicio y puertos

Una vez instalado, verificaremos que el servicio esté corriendo y en qué puertos escucha por defecto. Para ello, utilizaremos el siguiente comando:

```
root@hamza:/home/hamza# sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset:
             Active: active (running) since Mon 2025-11-17 09:00:42 CET; 5min ago
     Invocation: e6f1a12bf2cf4f40b6287e743c18ac6f
      Docs: https://httpd.apache.org/docs/2.4/
    Main PID: 4217 (apache2)
       Tasks: 55 (limit: 9336)
      Memory: 7.1M (peak: 7.9M)
        CPU: 93ms
       CGroup: /system.slice/apache2.service
                 └─4217 /usr/sbin/apache2 -k start
                   ├─4219 /usr/sbin/apache2 -k start
                   ├─4220 /usr/sbin/apache2 -k start
```

(Verificación del estado y servicio de Apache)

Una vez verificado que el servicio de Apache corre correctamente, iremos a ver en puerto corre. Para ello iremos a la siguiente ruta (**/etc/apache2/ports.conf**), en la cual podremos cambiar este puerto, que por defecto corre en el **puerto 80**.

```
GNU nano 8.4                               /etc/apache2/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>
```

Puerto de escucha por defecto

(Fichero de configuración de puertos)

Después, accederemos al archivo de configuración principal de Apache ubicado en **/etc/apache2/apache2.conf**.

¿Qué contiene este archivo y por qué es tan importante?

Contiene la **configuración global y directivas** que se aplican a todo el servidor de Apache.

Es tan importante, porque establece **configuraciones de seguridad**, define dónde Apache debe buscar sus módulos, dónde están los archivos de configuración de los sitios y los registros de errores.

```
GNU nano 8.4          /etc/apache2/apache2.conf
# This is the main Apache server configuration file. It contains the
# configuration directives that give the server its instructions.
# See http://httpd.apache.org/docs/2.4/ for detailed information about
# the directives and /usr/share/doc/apache2/README.Debian about Debian specific
# hints.
#
#
# Summary of how the Apache 2 configuration works in Debian:
# The Apache 2 web server configuration in Debian is quite different to
# upstream's suggested way to configure the web server. This is because Debian's
# default Apache2 installation attempts to make adding and removing modules,
# virtual hosts, and extra configuration directives as flexible as possible, in
# order to make automating the changes and administering the server as easy as
# possible.
```

(Archivo de configuración principal de Apache2)

2.3 Análisis del archivo de configuración principal (apache2.conf)

Explica qué elementos aparecen por defecto en su interior y cuál es su función dentro del funcionamiento general del servidor web.

Bloques <Directory>: Definen permisos de acceso y opciones de configuración para directorios específicos del sistema de archivos, como /var/www/, para controlar cómo el servidor gestiona el contenido.

```
<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/>
```

(Elemento Directory en fichero de configuración principal)

Directivas Include: Estas líneas le dicen a Apache que incorpore configuraciones de otros directorios, como la configuración de módulo y los archivos de sitios web.

```
GNU nano 8.4          /etc/apache2/apache2.conf
# error, crit, alert, emerg.
# It is also possible to configure the log level for particular modules, e.g.
# "LogLevel info ssl:warn"
#
LogLevel warn

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

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

(Directivas *Include*)

Elementos que defines el usuario y el grupo con el que se ejecutaran los procesos de Apache. Cruciales para seguridad y la gestión de permisos.

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

#
# HostnameLookups: Log the names of clients or just their IP addresses
# e.g., www.apache.org (on) or 204.62.129.132 (off).
# The default is off because it'd be overall better for the net if people
# had to knowingly turn this feature on, since enabling it means that
# each client request will result in AT LEAST one lookup request to the
# nameserver.
#
HostnameLookups Off
```

(Elementos de definición de usuarios y grupos)

Elemento de **registro de errores**:

```
# 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
```

(Registro de errores)

3. Gestión del sitio web por defecto

3.1 ¿Se puede modificar este sitio web por defecto? Para comprobarlo, revisa los archivos que se encuentran en la ruta /etc/apache2/sites-available/.

Si, se puede modificar. Se hace editando el archivo de configuración por defecto, que suele ser ‘000-default.conf’, en la ruta /etc/apache2/sites-available/.

```
root@hamza:/etc/apache2/sites-available# ls  
000-default.conf default-ssl.conf
```

3.2 Dentro de uno de estos archivos aparece la línea: DocumentRoot /var/www/html ¿Qué significa exactamente?

La directiva **DocumentRoot** le indica a Apache la ubicación en el sistema de archivos (el directorio raíz) donde se encuentran los archivos públicos del sitio web que debe servir cuando se accede a ese host virtual. En este caso, el sitio por defecto servirá los archivos que estén en /var/www/html/.

```
GNU nano 8.4                               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 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
```

(Indicación de Directiva DocumentRoot)

3.3 ¿Es posible modificar el contenido del directorio /var/www/html para que se muestren diferentes sitios web?

Sí, es posible. Además, podemos crear nuevos directorios dentro de /var/www/... y luego crear un nuevo archivo de VirtualHost en /etc/apache2/sites-available/ que apunte su DocumentRoot a esa nueva carpeta.

3.4 Personalización de la página principal

A continuación, añadiremos nuestro propio sitio html dentro del directorio mencionado en el apartado anterior y comprobaremos el acceso desde nuestro navegador web.

Primero, crearemos un nuevo fichero html (hamza.html), ya que ya hay uno por defecto (index.html).

```
root@hamza:/var/www/html# ls  
index.html  
root@hamza:/var/www/html# nano index.html  
root@hamza:/var/www/html# touch hamza.html  
root@hamza:/var/www/html# ls  
hamza.html index.html  
root@hamza:/var/www/html# nano hamza.html
```

Creación fichero html

Fichero html por defecto

Configuración html

Fichero de configuración html (hamza.html) configurado.

```
GNU nano 8.4  
hamza.html *  
<!DOCTYPE html>  
<html>  
<head>  
    <title>Sitio de Prueba</title>  
</head>  
<body>  
    <h1>¡Helloo! Este es mi sitio web en Apache.</h1>  
    <p>Todo correcto.</p>  
</body>  
</html>
```

Prueba desde el navegador web correcta.



3.5 ¿Es posible establecer este archivo como página principal al acceder a una web?
¿Cómo se haría?

Sí, es posible. Esto se hace con la directiva **DirectoryIndex**.

Para ello iremos a la ruta **/etc/apache2/sites-available** y accederemos a fichero de configuración **000-default.conf**. En este será donde haremos la siguiente configuración:

```
GNU nano 8.4                               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 www.example.com

    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html
    <Directory /var/www/html>
        DirectoryIndex hamza.html index.html index.php
    </Directory>
```

(Configuración página principal)

Verificación en navegador web accediendo solo con localhost, verificando así que se ha configurado correctamente.



(Prueba página principal desde navegador web)

4. Comprobación de los módulos habilitados

Apache permite activar o desactivar módulos según las necesidades del administrador o del sitio web. Estos módulos son pequeñas extensiones o componentes que añaden funcionalidades adicionales al servidor web, haciendo que sea más flexible y adaptable a diferentes escenarios.

4.1 Listar módulos activos

Para ver qué módulos están cargados y en funcionamiento en el servidor, usamos el siguiente comando:

```
root@hamza:/etc/apache2/sites-available# sudo apache2ctl -M
Loaded Modules:
core_module (static)
so_module (static)
watchdog_module (static)
http_module (static)
log_config_module (static)
logio_module (static)
version_module (static)
unixd_module (static)
access_compat_module (shared)
alias_module (shared)
auth_basic_module (shared)
authn_core_module (shared)
authn_file_module (shared)
authz_core_module (shared)
```

(Módulos cargados activos)

4.2 Identificación de módulos

```
alias_module (shared)
auth_basic_module (shared)
authn_core_module (shared)
authn_file_module (shared)
authz_core_module (shared)
authz_host_module (shared)
authz_user_module (shared)
autoindex_module (shared)
deflate_module (shared)
dir_module (shared)
env_module (shared)
filter_module (shared)
mime_module (shared)
mpm_event_module (shared)
negotiation_module (shared)
reqtimeout_module (shared)
setenvif_module (shared)
status_module (shared)
```

- **dir_module:** Define qué archivo se sirve automáticamente cuando se accede a un directorio.
- **authz_host_module:** Permite la autorización basada en la **dirección IP del cliente** o el nombre del host.
- **auth_basic_module:** Habilita la autenticación de usuario básica a través de HTTP.

4.3 Ubicación de los módulos disponibles y activados

La ubicación de estos módulos se organiza en dos directorios:

- **Módulos disponibles:** Se encuentran en la ruta **/etc/apache2/mods-available/**.
Qué contiene todos los archivos de configuración, carga de los módulos de apache que podemos usar.
- **Módulos activos:** Se encuentran en la ruta **/etc/apache2/mods-enabled/**. Qué contiene los módulos que están cargados y en funcionamiento en el servidor.

```
root@hamza:/etc/apache2/mods-available# ls
access_compat.load    dir.load          proxy_connect.load
actions.conf           dnssd.conf        proxy_express.load
actions.load           dnssd.load        proxy_fcgi.load
alias.conf             dump_io.load     proxy_fdpass.load
alias.load              echo.load         proxy_ftp.conf
allowmethods.load      env.load         proxy_ftp.load
asis.load               expires.load    proxy_hcheck.load
auth_basic.load        ext_filter.load proxy_html.conf
auth_digest.load       file_cache.load proxy_html.load
auth_form.load         filter.load     proxy_http2.load
authn_anon.load        headers.load   proxy_http.load
authn_core.load        heartbeat.load proxy.load
authn_dbd.load         heartmonitor.load proxy_scgi.load
```

(Directorio de módulos disponibles)

```
root@hamza:/etc/apache2/mods-enabled# ls
access_compat.load  authz_core.load  deflate.load  mime.load      reqtimeout.load
alias.conf          authz_host.load  dir.conf      mpm_event.conf  setenvif.conf
alias.load          authz_user.load  dir.load      mpm_event.load  setenvif.load
auth_basic.load     autoindex.conf  env.load     negotiation.conf status.conf
authn_core.load     autoindex.load  filter.load  negotiation.load status.load
authn_file.load     deflate.conf   mime.conf   reqtimeout.conf
```

(Directorio de módulos activos)

4.4 Comprobación de enlaces simbólicos

Los módulos activos en **mods-enabled** son enlaces simbólicos que apuntan a los archivos reales en **mods-available**

Este comando mostrará una lista donde cada elemento tendrá una flecha indicando que es un enlace simbólico al archivo de origen en el directorio.

```
root@hamza:/etc/apache2/mods-enabled# ls -l /etc/apache2/mods-enabled/
total 0
lrwxrwxrwx 1 root root 36 nov 17 09:00 access_compat.load -> ../mods-available/access_compat.load
lrwxrwxrwx 1 root root 28 nov 17 09:00 alias.conf -> ../mods-available/alias.conf
lrwxrwxrwx 1 root root 28 nov 17 09:00 alias.load -> ../mods-available/alias.load
lrwxrwxrwx 1 root root 33 nov 17 09:00 auth_basic.load -> ../mods-available/auth_basic.load
lrwxrwxrwx 1 root root 33 nov 17 09:00 authn_core.load -> ../mods-available/authn_core.load
lrwxrwxrwx 1 root root 33 nov 17 09:00 authn_file.load -> ../mods-available/authn_file.load
lrwxrwxrwx 1 root root 33 nov 17 09:00 authz_core.load -> ../mods-available/authz_core.load
```

5. Añadir soporte PHP a Apache

Para habilitar el procesamiento de archivos .php, necesitamos el módulo de PHP. Esto permitirá que el servidor procese archivos .php además de los .html.

5.1 Instalación e Integración

Con el siguiente comando, instalaremos el intérprete principal de PHP y el módulo de Apache, que permite procesar peticiones de archivos .php.

```
root@hamza:~# sudo apt install php libapache2-mod-php -y
Installing:
  libapache2-mod-php  php

Installing dependencies:
  libapache2-mod-php8.4  php8.4      php8.4-common  php8.4-readline
  php-common            php8.4-cli   php8.4-opcache

Paquetes sugeridos:
  php-pear

Summary:
  Upgrading: 0, Installing: 9, Removing: 0, Not Upgrading: 99
  Download size: 4.922 kB
```

(Instalación de PHP y módulo de Apache)

5.2 Verificación de funcionamiento

Una vez instalado, es necesario verificar que el módulo que acabamos de instalar esté cargado y activo en la configuración de Apache. Para ello comprobaremos que el módulo de PHP esté disponible y activado utilizando el siguiente comando:

```
root@hamza:~# sudo apache2ctl -M
Loaded Modules:
 core_module (static)
 so_module (static)
 watchdog_module (static)
 http_module (static)
 log_config_module (static)
 logio_module (static)
 version_module (static)
 unixd_module (static)
 access_compat_module (shared)
 alias_module (shared)
 auth_basic_module (shared)
 authn_core_module (shared)
 authn_file_module (shared)
 authz_core_module (shared)
 authz_host_module (shared)
 authz_user_module (shared)
 autoindex_module (shared)
 deflate_module (shared)
 dir_module (shared)
 env_module (shared)
 filter_module (shared)
 mime_module (shared)
 mpm_prefork_module (shared)
 negotiation_module (shared)
 php_module (shared)
```

Verificación módulo PHP instalado

(Listado y verificación de instalación del módulo PHP)

Este comando también verifica la disponibilidad del módulo PHP en Apache. Al filtrar con “grep” solo mostrará las líneas que contienen el texto "php".

```
root@hamza:~# ls /etc/apache2/mods-available/ | grep php
php8.4.conf
php8.4.load
```

(Verificación de disponibilidad filtrada)

- **php8.4.conf:** Es el archivo de configuración del módulo PHP, donde se definen directivas específicas de cómo debe funcionar PHP con Apache.
- **php8.4.load:** Es el archivo que contiene la directiva para cargar el binario del módulo PHP en la memoria de Apache.

5.3 Creación de página de prueba

Crearemos una página de prueba con código PHP para confirmar que el servidor está procesando el código correctamente y no sirviendo el archivo como texto plano. Para ello crearemos un archivo llamado “info.php” en el directorio raíz por defecto (**/var/www/html**).

Una vez en la ruta (**/var/www/html/info.php**) configuraremos el fichero de la siguiente manera:

```
GNU nano 8.4                               /var/www/html/info.php
<?php
    phpinfo();
?>
```

(Configuración de fichero de página de prueba)

Una vez configurado el fichero anterior, accederemos desde el navegador web a <http://localhost/info.php>. Si todo está correcto, Aparecerá una página extensa de HTML generada por PHP, detallando la versión, módulos cargados, variables, etc.



(Página de prueba generada por PHP)

Explica qué relación existe entre los módulos de Apache y el funcionamiento de PHP:

La relación entre los módulos de Apache y PHP se basa en la **delegación y extensión de funcionalidad**. Apache es un servidor diseñado para servir contenido estático, no tiene la capacidad de ejecutar código dinámico. El módulo PHP actúa como intérprete del servidor. Cuando Apache recibe una petición para un archivo .php, delega la tarea de procesamiento a este módulo. El módulo ejecuta el código PHP y devuelve ese resultado a Apache, que luego lo sirve al cliente como una respuesta HTTP.

¿Qué pasaría si el módulo PHP no estuviera habilitado?

Si el módulo de PHP no estuviera habilitado, el servidor Apache trataría el archivo info.php como un archivo de texto plano, ya que Apache no sabría cómo interpretar el código.

¿Qué tipo de error mostraría el navegador?

El navegador mostraría el código fuente PHP como texto plano (<?php phpinfo(); ?>), ya que Apache no sabría que debe ejecutarlo, sino simplemente servirlo como si fuera un archivo de texto.

¿Por qué Apache necesita delegar la interpretación del código PHP a un módulo externo?

Apache necesita delegar porque el código PHP **no es lenguaje nativo de Apache**. Para ejecutar lenguajes dinámicos como PHP, necesita que un **módulo externo** actúe como intérprete y procesador. Esto mantiene el núcleo de Apache.

[Desactiva y reactiva el módulo PHP para comprobar el comportamiento del servidor:](#)

Desactivar el módulo:

```
root@hamza:~# sudo a2dismod php8.4
Module php8.4 disabled.
To activate the new configuration, you need to run:
    systemctl restart apache2
```

(Desactivación del módulo php)

Una vez desactivado, reiniciaremos el servicio de Apache con el comando **systemctl restart apache2**.

Intento de acceso a info.php

Una vez aplicado desactivado el módulo, intentaremos acceder a info.php en el navegador. Mostrando el código fuente PHP como texto plano (<?php phpinfo(); ?>)



(info.php en texto plano en el navegador web)

Comentarios

Esta práctica ha permitido explorar en profundidad la instalación, configuración y gestión de un servidor web Apache en un entorno Debian.

Problemas solventados:

El navegador anterior (Firefox) estaba almacenando en caché una respuesta HTTP de la página en blanco o el error anterior (304 Not Modified) y se negaba a cargar la nueva página PHP, incluso después de reiniciar Apache.

Solución:

Usé un navegador diferente (Chrome), este forzó una nueva conexión limpia al servidor Apache, lo cual permitió que el PHP se ejecutara correctamente y mostrara la página phpinfo.