

Introducción	2
Cómo modificar los archivos de configuración	2
Configuración de Apache	3
Ubicación de las páginas web	3
Alias	4
Control de acceso en Apache	6
DocumentRoot	8

Introducción

Esta explicación sobre la configuración de Apache ha sido preparada para la versión XAMPP 8.1.6

Cada vez que se modifica alguno de los ficheros de configuración es necesario detener y reiniciar el servidor.

Cómo modificar los archivos de configuración

Antes de modificar cualquier archivo de configuración, recomiendo hacer una copia de seguridad del archivo de configuración actual.

Si al modificar algún archivo de configuración nos equivocamos e introducimos errores, es probable que el servidor no pueda arrancar. Si no somos capaces de encontrar el error, siempre podemos restaurar el archivo de configuración anterior y empezar de nuevo, sin necesidad de volver a instalar XAMPP.

Al modificar cualquier archivo de configuración, es conveniente documentar los cambios. Para ello, recomiendo hacer tres cosas:

- no modificar la línea del archivo original, sino añadirle el carácter de comentario al principio de la línea (almohadilla o punto y coma).
- añadir la línea modificada.
- añadir una línea de comentario indicando la fecha, el autor y el cambio realizado.

Por ejemplo, la siguiente línea es una línea del archivo de configuración de Apache:

```
DocumentRoot "C:/xampp/htdocs"
```

Tras la modificación, el archivo de configuración quedaría así (se ha resaltado el texto añadido):

```
# 1/10/2023 FMG: He cambiado el directorio raíz del  
servidor
```

```
# DocumentRoot "C:/xampp/htdocs"  
DocumentRoot "C:/www"
```

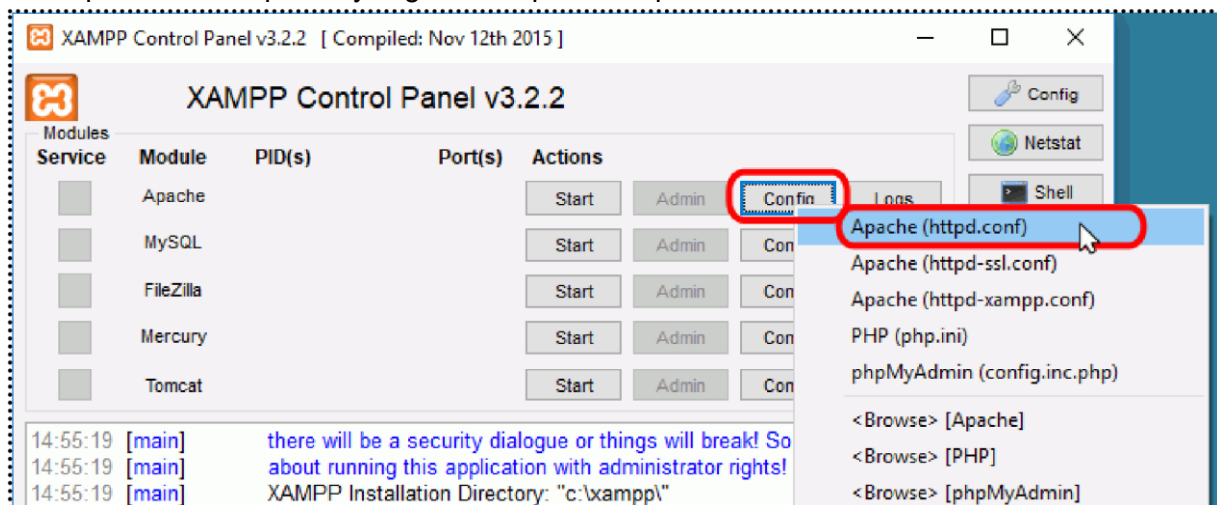
Configuración de Apache

El fichero de configuración de Apache es el archivo **httpd.conf**, un archivo de texto sin formato.

En httpd.conf las líneas comentadas empiezan por el carácter almohadilla (#). En Windows, al escribir una ruta de directorio en httpd.conf se puede utilizar tanto la barra como la contrabarra (es decir, se puede escribir tanto C:\xampp como C:/xampp).

En Windows el archivo **httpd.conf** se encuentra en el directorio **C:\xampp\apache\conf**.

Se puede abrir directamente el archivo httpd.conf haciendo clic en el botón "Config" correspondiente a Apache y eligiendo la primera opción:



Vamos a comentar algunas directivas de configuración de Apache. Antes de modificar cualquier archivo de configuración, se recomienda hacer una copia de seguridad del archivo de configuración.

Ubicación de las páginas web

Con la configuración inicial de XAMPP, al abrir en el navegador una dirección que empieza por **http://localhost/**, Apache busca el elemento solicitado a partir del directorio raíz del servidor, indicado por la directiva **DocumentRoot** del archivo httpd.conf:

- si se trata de un archivo:
 - si encuentra el archivo, lo sirve al navegador.
 - si no encuentra el archivo, envía un error 404 (objeto no localizado).
- si se trata de un directorio:
 - si encuentra el directorio, busca si existe alguno de los archivos indicados por la directiva **DirectoryIndex** (por ejemplo, en XAMPP, index.php index.pl index.cgi index.asp index.shtml index.html index.htm etc) :
 - si encuentra alguno de esos archivos, lo sirve al navegador.
 - si no encuentra ninguno de esos archivos, Apache envía al navegador el listado del directorio.
 - si no encuentra el directorio, envía un error 404 (objeto no localizado).

Esta configuración es adecuada para entornos de programación, pero en entornos de producción se suele modificar, por ejemplo, para no enviar el listado del directorio.

En el directorio raíz del servidor, XAMPP crea una serie de páginas que permiten administrar el servidor. Por eso al abrir en el navegador la dirección `http://localhost` se abre una página de administración propia de XAMPP.

A la hora de añadir otras páginas en el servidor, se puede hacer de varias formas:

- Añadir nuestras páginas en el directorio predeterminado de XAMPP. Tiene el inconveniente de que ese directorio no está donde se suelen guardar los documentos (en Windows, Documentos) y estamos mezclando nuestros ficheros con los de XAMPP.
- Cambiar el directorio raíz del servidor, modificando la directiva `DocumentRoot` (como se explica más adelante). Tiene el inconveniente de que no podríamos acceder a la página web de administración de XAMPP (aunque podríamos crear una directiva `Alias` para posibilitar el acceso).
- Crear directorios de usuario, mediante la directiva `UserDir`. Es una buena solución en un ordenador con varios usuarios de Windows sin perfil de administrador para que cada usuario tenga sus propias páginas separadas e independientes del resto. Cada usuario puede guardar sus páginas en un directorio (con el mismo nombre para todos los usuarios, pero cada uno en su propia carpeta de Documentos) y cada usuario sólo ve sus páginas.
- Crear uno o varios alias, mediante la directiva `Alias`. Es la mejor solución cuando sólo hay un usuario de Windows que utiliza XAMPP. También se puede utilizar cuando hay varios usuarios, pero entonces estos tienen que tener en cuenta los alias creados por los demás, puesto que los nombres de los alias no se pueden repetir.

Alias

La directiva `Alias` permite que el servidor abra páginas situadas en cualquier directorio, aunque no estén dentro del directorio establecido mediante la directiva `DocumentRoot`.

Para que Apache pueda acceder a esos directorios, en el archivo `httpd.conf` debe estar cargado el módulo `alias` mediante la directiva:

```
LoadModule alias_module modules/mod_alias.so
```

En XAMPP 8.1.6 esta directiva no está comentada de forma predeterminada, por lo que no es necesario modificarla, aunque conviene comprobar su valor.

Además, hay que añadir en el archivo `httpd.conf` una directiva `Alias` dentro de la directiva `<IfModule alias_module>` (que ya existe En XAMPP 8.1.6).

Como ejemplo, vamos a crear un par de alias. Supongamos que un usuario está siguiendo una clase de PHP en el módulo de desarrollo web entorno servidor y al mismo tiempo en su trabajo también escribe programas en PHP. Este usuario quiere guardar los archivos en

diferentes carpetas de Documentos, pero al mismo tiempo acceder fácilmente a los archivos sin tener que escribir direcciones muy largas:

Programas	Ruta en el navegador	Carpeta en el disco
Programas escritos en clase (Turno mañana)	http://localhost/dwes_manana/	Documentos > Clase > DWES > PHP
Programas en el trabajo	http://localhost/dwes_curro/	Documentos > Trabajo > PHP

Para ello, hay que añadir en el archivo httpd.conf dos directivas Alias dentro de la directiva <IfModule alias_module>. En esta directiva se pueden crear redirecciones, alias y alias de script. Para mantener el archivo bien organizado, se recomienda escribir los alias a continuación de las líneas de comentario relativas a los alias. El ejemplo siguiente muestra los dos alias del ejemplo anterior.

```
<IfModule alias_module>
# ...

# 2022-09-15. FMG. He añadido este Alias para los programas escritos
# en clase de DWES de PHP.
Alias /dwes_manana
"C:\Users\NombreDeUsuario\Documents\Clase\DWES\PHP"
<Directory "C:\Users\NombreDeUsuario\Documents\Clase\DWES\PHP">
    Options Indexes FollowSymLinks Includes ExecCGI
    AllowOverride All
    Require all granted
</Directory>

# 2022-09-15. FMG. He añadido este Alias para los programas escritos
# para el trabajo.
Alias /dwes_curro
"C:\Users\NombreDeUsuario\Documents\Trabajo\PHP"
<Directory "C:\Users\NombreDeUsuario\Documents\Trabajo\PHP">
    Options Indexes FollowSymLinks Includes ExecCGI
    AllowOverride All
    Require all granted
</Directory> # ...
</IfModule>
```

Como se comentará en el apartado de la directiva DocumentRoot, se debe incluir una directiva <Directory> para el directorio indicado en la directiva Alias.

Hay que tener cuidado con las erratas, ya que cualquier error impedirá que Apache se ponga en marcha.

Un alias puede abarcar varios pasos de la ruta, como muestra el ejemplo siguiente.

Programas	Ruta en el navegador	Carpeta en el disco
Programas escritos en el año anterior (2021)	http://localhost/dwes_20/php	Documentos > Clase > 21 > PHP

La directiva Alias sería la siguiente.

```
<IfModule alias_module>
# ...

# 2021-09-21. FMG. He añadido este Alias para los programas escritos
en clase de PHP del año anterior.
Alias /dwes_21/php
"C:\Users\NombreDeUsuario\Documents\Clase\21\PHP"
<Directory "C:\Users\NombreDeUsuario\Documents\Clase\21\PHP">
    Options Indexes FollowSymLinks Includes ExecCGI
    AllowOverride All
    Require all granted
</Directory>

# ...
</IfModule>
```

NOTA: Podría usarse "Require local" para ser un poco más restrictivos que con "all granted" ya que solo podríamos acceder a las páginas web desde la máquina local.

Control de acceso en Apache

En Apache 2.2, el control de acceso se realizaba mediante las directivas Order, Allow, Deny, and Satisfy.

En Apache 2.4, el control de acceso se realiza mediante el módulo mod_authz_host, que utiliza otras directivas, aunque el módulo mod_access_compat permite utilizar las directivas antiguas.

En el caso de XAMPP, la versión XAMPP 1.8.0 (que ya incluía Apache 2.4.2) utilizaba todavía las directivas habituales de Apache 2.2 order y deny. El fichero httpd.conf contenía una directiva Directory general que restringía el acceso general a los directorios utilizando las directivas Order y Deny

```
# 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.
#
<Directory />
    Options FollowSymLinks
    AllowOverride None
    Order deny,allow
    Deny from all
</Directory>
```

Por eso, cuando el usuario quería permitir el acceso a un directorio (por ejemplo, en un alias) debían utilizarse esas mismas directivas. Por ejemplo

```
<IfModule alias_module>
    Alias /mi-proyecto/mi-carpeta
    "C:\Users\NombreDeUsuario\Documents\PHP"
    <Directory "C:\Users\NombreDeUsuario\Documents\PHP">
        Options Indexes FollowSymLinks Includes ExecCGI
        AllowOverride All
        Order allow,deny
        Allow from all
    </Directory>

    ScriptAlias /cgi-bin/ "C:/xampp/cgi-bin/"
</IfModule>
```

Sin embargo, la versión de XAMPP 1.8.3 (que ya incluye Apache 2.4.4) utiliza ya la directiva Require. El fichero httpd.conf contiene una directiva Directory general que restringe el acceso general a los directorios utilizando la directiva Require

```
# Deny access to the entirety of your server's filesystem. You must
# explicitly permit access to web content directories in other #
<Directory> blocks below.
#
<Directory />
    AllowOverride none
    Require all denied
</Directory>
```

Por eso, cuando el usuario quiere permitir el acceso a un directorio (por ejemplo en un alias) debe utilizarse esa misma directiva. Por ejemplo

```
<IfModule alias_module>

    Alias /mi-proyecto/mi-carpeta
    "C:\Users\NombreDeUsuario\Documents\PHP"
    <Directory "C:\Users\NombreDeUsuario\Documents\PHP">
        Options Indexes FollowSymLinks Includes ExecCGI
        AllowOverride All
        Require all granted
    </Directory>

    ScriptAlias /cgi-bin/ "C:/xampp/cgi-bin/"
</IfModule>
```

DocumentRoot

Nota: En este curso no se recomienda el cambio de la directiva DocumentRoot, aunque se comente a continuación cómo hacerlo.

La directiva DocumentRoot especifica el directorio donde se encuentran las páginas web. Cuando en la barra de dirección del navegador se escriba una dirección que comience por `http://localhost/`, el servidor buscará los archivos a partir de ese directorio. La directiva DocumentRoot va acompañada de una directiva `<Directory>` que especifica qué servicios y características están permitidos o prohibidos en el directorio.

En XAMPP 7.0 el valor predeterminado de DocumentRoot es el siguiente:

```
#
# 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:/xampp/htdocs"
<Directory "C:/xampp/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/2.4/mod/core.html#options #
        for more information.
        #
        Options Indexes FollowSymLinks Includes ExecCGI

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

        #
        # Controls who can get stuff from this server.
        #
        Require all granted
</Directory>

```

Si se quiere cambiar DocumentRoot y utilizar, por ejemplo, el directorio C:\www, hay que modificar tanto la línea de DocumentRoot como el Directory correspondiente. El archivo de configuración debería quedar así (se ha resaltado el texto añadido):

```

#
# 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.
#
# 2022-09-15. FMG. He cambiado el directorio raíz del servidor
# DocumentRoot "C:/xampp/htdocs"
DocumentRoot "C:/www"

# 2022-09-15 FMG. He cambiado la directiva Directory de
DocumentRoot
# <Directory "C:/xampp/htdocs">
<Directory "C:/www"> #
        # 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/2.4/mod/core.html#options #
# for more information.
#
Options Indexes FollowSymLinks Includes ExecCGI

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

#
# Controls who can get stuff from this server.
#
Require all granted
</Directory>
```

Hay que tener cuidado con las erratas, ya que cualquier error impedirá que Apache se ponga en marcha.