

Configuración Apache y PHP

Introducción.....	1
Cómo modificar los archivos de configuración.....	1
Configuración de Apache.....	2
Ubicación de las páginas web.....	3
Alias.....	4
DocumentRoot.....	6
Configuración de PHP.....	7
El archivo php.ini.....	7
Directivas de configuración en php.ini.....	9
error_reporting.....	9
output_buffering.....	11
Short_open_tag.....	11
date.timezone.....	11
post_max_size y upload_max_filesize.....	12

Introducción

Esta explicación sobre la configuración de Apache ha sido preparada para aplicarla en cualquier versión 8.2.X e incluso 8.X de XAMPP.

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):

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

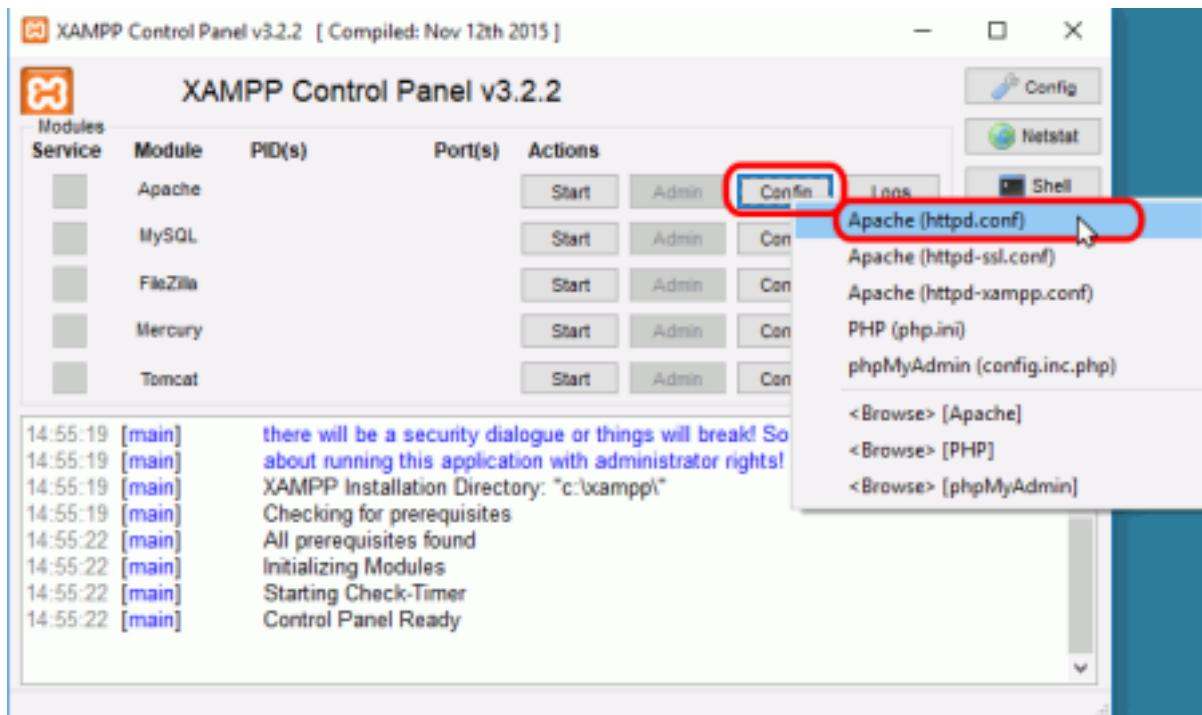
Configuración de Apache

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.

Yo soy más partidaria de **crear alias**, uno para cada alumno que utilice el ordenador.

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 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).

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	http://localhost/hlc/ Documentos > Clase > HLC > PHP

Programas en el trabajo	http://localhost/hlc_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. He añadido este Alias para los programas escritos #
#en clase de HLC de PHP.
Alias /hlc "C:\Users\NombreDeUsuario\Documents\Clase\HLC\PHP"
<Directory "C:\Users\NombreDeUsuario\Documents\Clase\HLC\PHP">
Options Indexes FollowSymLinks Includes ExecCGI
    AllowOverride All
    Require all granted
</Directory>

# 2022-09-15. He añadido este Alias para los programas escritos
para el trabajo.
Alias /hlc_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/hlc_21/php	Documentos > Clase > 21 > PHP

La directiva Alias sería la siguiente.

```

<IfModule alias_module>
# ...

# 2023-09-15. He añadido este Alias para los programas escritos
# en clase de PHP del año anterior.
Alias /hlc_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>

```

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 el valor predeterminado de DocumentRoot es el siguiente:

```

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. He cambiado el directorio raíz del servidor #
DocumentRoot "C:/xampp/htdocs"
DocumentRoot "C:/www"

# 2022-09-15 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
#
- - -
```

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

Configuración de PHP

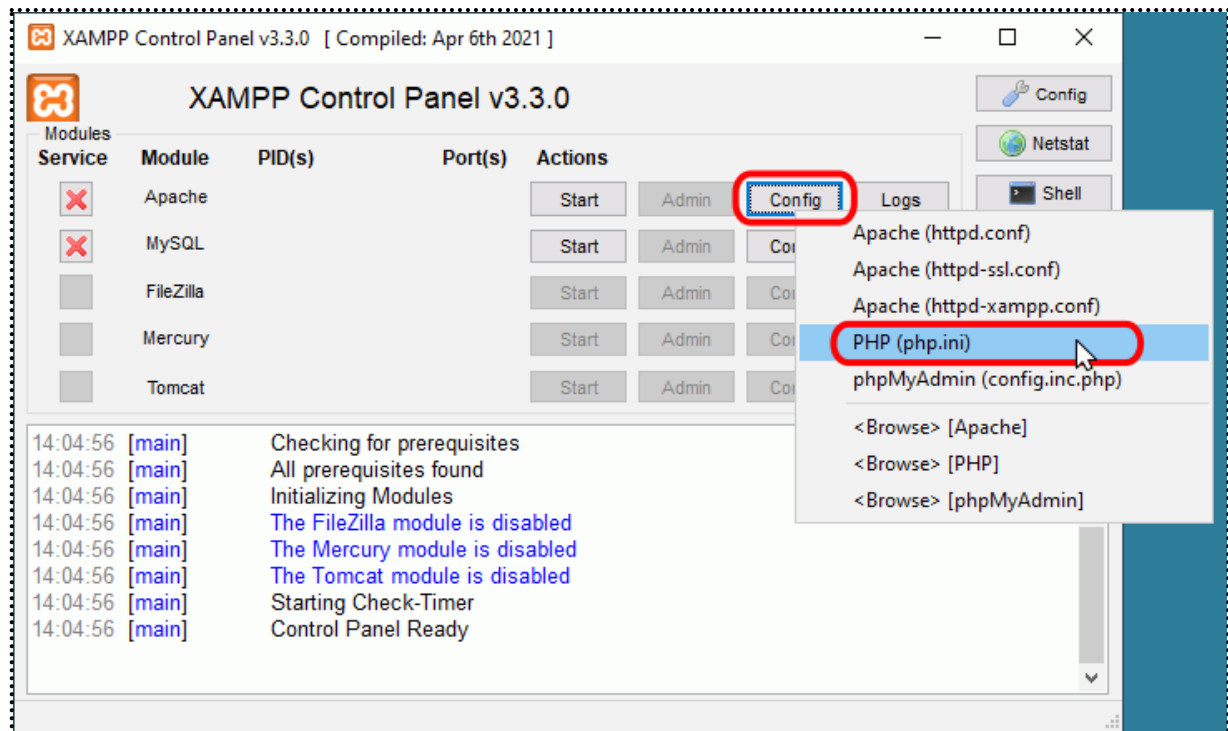
El archivo php.ini

El archivo de configuración de PHP es el archivo **php.ini**, un archivo de texto sin formato.

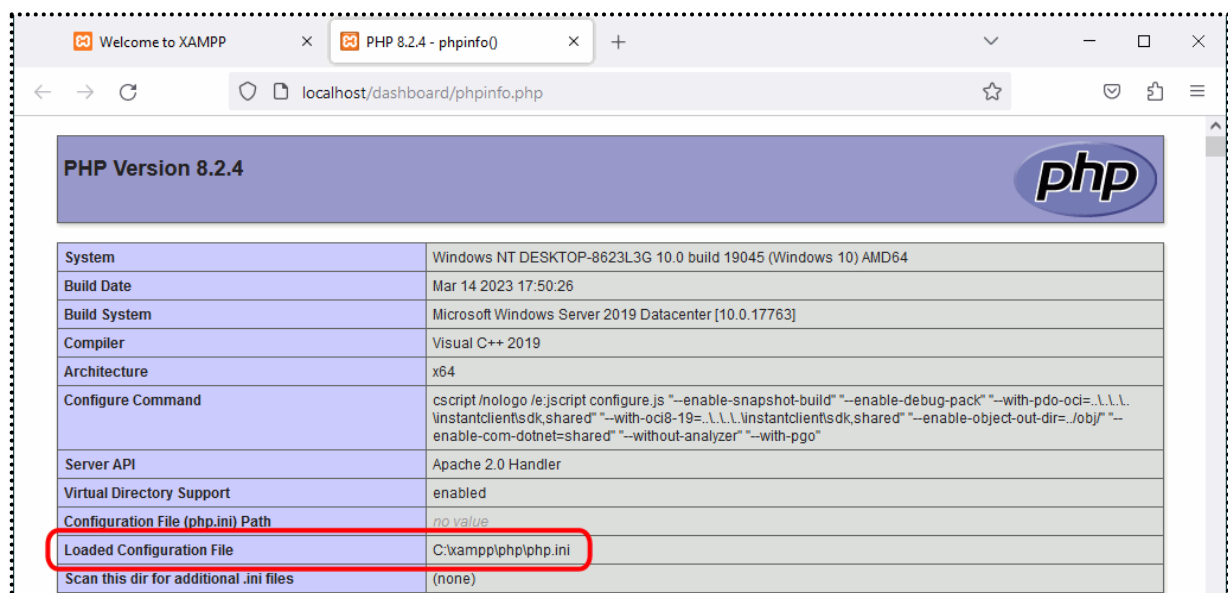
En php.ini las líneas comentadas empiezan por el carácter punto y coma (;).

En Windows, el archivo php.ini se encuentra en el directorio **C:\xampp\php\php.ini**

Se puede abrir directamente el archivo **php.ini** haciendo clic en el botón "Config" correspondiente a Apache y eligiendo opción correspondiente:



Nota: La ruta del archivo php.ini que está utilizando Apache se puede ver en el campo "Loaded Configuration File" de la primera tabla que muestra la función **phpinfo()**. En el menú de la página de administración de XAMPP 8.2 hay una opción **phpinfo()** que muestra la configuración de PHP.



Directivas de configuración en php.ini

Se comentan a continuación algunas directivas de configuración de PHP, aunque también se puede consultar el [manual de PHP](#). Antes de modificar cualquier archivo de configuración, se recomienda hacer una copia de seguridad del archivo de configuración original.

En el archivo de configuración php.ini, las líneas que comienzan por ; (punto y coma) son líneas comentadas, es decir, que no se tendrán en cuenta cuando PHP cargue el archivo. En el archivo de configuración se pueden encontrar bloques de varias líneas comentadas que explican el significado de una directiva y más adelante una línea sin comentar que establece el valor de la directiva. Un error de principiante bastante común es modificar el valor de la directiva en una línea comentada, lo que no sirve para nada.

error_reporting

La directiva [error_reporting](#) especifica el tipo de errores notificados por PHP. Por ejemplo:

Directiva / Valor	Significado
error_reporting=E_ALL & ~E_NOTICE & E_DEPRECATED	Se muestran todos los errores, menos los avisos y obsoletos.
error_reporting=E_ALL & ~E_DEPRECATED	Se muestran todos los errores, menos los obsoletos
error_reporting=E_ALL ; Valor recomendado en este curso	Se muestran todos los errores (Valor recomendado)

En un entorno de producción se suelen ocultar a los clientes los mensajes de error por motivos de seguridad (aunque se deben registrar en el archivo de logs), pero en un entorno de desarrollo es conveniente mostrar el mayor número de errores y avisos para corregirlos. Para mostrar todos los errores, la directiva **error_reporting** debe tomar el valor **E_ALL**

```
; 2023-09-11. He cambiado error_reporting para mostrar todos los errores
; error_reporting=E_ALL & ~E_DEPRECATED & ~E_STRICT
error_reporting
    Default Value: E_ALL
    Development Value: E_ALL
|    Production Value: E_ALL & ~E_DEPRECATED & ~E_STRICT
```

El inconveniente de utilizar una configuración tan exigente es que algunos paquetes de terceros (bibliotecas, CMS, etc.) pueden dejar de funcionar y a veces no queda más remedio que volver a un valor menos estricto para poderlos utilizar.

En XAMPP 8.2 esta directiva viene predeterminada con el valor E_ALL & ~E_DEPRECATED & ~E_STRICT, por lo que **se recomienda modificarla**:

Los dieciséis posibles valores de esta directiva, que se pueden combinar con el operador lógico & (Y), son los siguientes:

Tipo de error	Descripción
E_ALL	Todos los errores y avisos.
E_ERROR	Errores fatales detectados en tiempo de ejecución.
E_WARNING	Advertencias (errores no fatales) detectados en tiempo de ejecución.
E_PARSE	Errores de sintaxis detectados en tiempo de compilación.
E_NOTICE	Avisos detectados en tiempo de ejecución (pueden deberse a errores involuntarios o a errores intencionados, por ejemplo, utilizar una variable no inicializada, pero utilizarla teniendo en cuenta que se inicializará automáticamente a una cadena vacía).
E_STRICT	Avisos detectados en tiempo de ejecución, al activarlos PHP sugiere cambios en el código para que una mejor interoperabilidad y compatibilidad en el futuro.
E_RECOVERABLE_ERROR	Error recuperable.
E_DEPRECATED	Advertencias sobre código obsoleto que no funcionará en futuras versiones de PHP.
E_CORE_ERROR	Errores fatales detectados durante el arranque inicial de PHP.
E_CORE_WARNING	Advertencias (errores no fatales) detectados durante el arranque inicial de PHP.
E_COMPILE_ERROR	Errores fatales detectados en tiempo de compilación.
E_COMPILE_WARNING	Advertencias (errores no fatales) detectados en tiempo de compilación.
E_USER_ERROR	Mensajes de error generados por el usuario.
E_USER_WARNING	Advertencias generadas por el usuario.
E_USER_NOTICE	Avisos generados por el usuario.

E_USER_DEPRECATED	Advertencias generadas por el usuario sobre código que no funcionará en futuras versiones de PHP.
-------------------	---

output_buffering

La directiva `output_buffering` especifica si la salida del programa se envía a medida que se va generando o se utiliza un buffer intermedio para reducir el tráfico de salida del servidor.

Directiva / Valor	Significado
<code>output_buffering=4096</code>	Se utiliza un buffer de 4096 bytes.
<code>output_buffering=0</code> ; Valor recomendado en este curso	No se utiliza buffer.

En un entorno de producción se suele utilizar un buffer de salida, pero este puede enmascarar errores de programación (básicamente, la creación de contenido de la página antes del envío de cabeceras, como se comenta en la lección sobre cabeceras HTTP). Por eso es conveniente desactivar el buffer y poder detectar ese tipo de errores al escribir los programas.

```
output_buffering = 0 ; Valor recomendado en este curso
```

En XAMPP 8.2 esta directiva viene predeterminada con el valor 4096, por lo que **se recomienda modificarla**.

Al desactivar el buffer, dependiendo de la forma de compilación del intérprete PHP, se pueden encontrar problemas con los archivos UTF-8 con marca de orden de bytes (BOM),

Short_open_tag

Los fragmentos de código PHP se deben identificar mediante las etiquetas `<?php ?>`. La directiva [short_open_tag](#) especifica si también se admiten las etiquetas `<? ... ?>`.

Directiva / Valor	Significado
<code>short_open_tag=On</code>	Se admite el uso de <code><? ... ?></code> en vez de <code><?php ... ?></code>
<code>short_open_tag=Off</code>	No se admite el uso de <code><? ... ?></code> en vez de <code><?php ... ?></code> (Valor recomendado)

Se recomienda que esta directiva tome el valor **Off** y que se utilice siempre la etiqueta `<?php` para empezar los fragmentos de código php.

date.timezone

Las [funciones relacionadas con fechas y horas](#) se configuran mediante [varias directivas](#).

La directiva **date.timezone** establece la zona horaria predeterminada para todas las funciones de fecha y hora.

Al instalar XAMPP 8.2, el instalador asigna un valor estimado de la zona horaria (por eso se añade al final del fichero de configuración por lo que hay dos apartados [Date] en el fichero). En el caso de España, XAMPP 8.2 asigna el valor **Europe/Berlin** que se puede cambiar al valor Europe/Madrid (hasta ahora no era necesario, puesto que entre Berlín y Madrid no hay diferencias horarias, pero eso puede cambiar en el futuro). Se puede consultar [la lista completa de zonas horarios permitidas](#) en el manual de php.

post_max_size y upload_max_filesize

La directiva [post_max_size](#) especifica el tamaño máximo de los datos que pueden enviarse al servidor.

La directiva [upload_max_filesize](#) especifica el tamaño máximo de un fichero que se envíe al servidor.

Lógicamente, [post_max_size](#) debe ser superior a [upload_max_filesize](#). Si el valor se expresa sin unidades, se interpreta como bytes. Se pueden utilizar las unidades K (kilobytes), M (megabytes) y G (Gigabytes).

En XAMPP 8.2 esta directiva viene predeterminada con el **valor 40M**, por lo que **no** es necesario modificarla, aunque conviene comprobar su valor.