

# Cómo instalar el servidor Apache en Linux

Por: Sandra Manzano Martínez

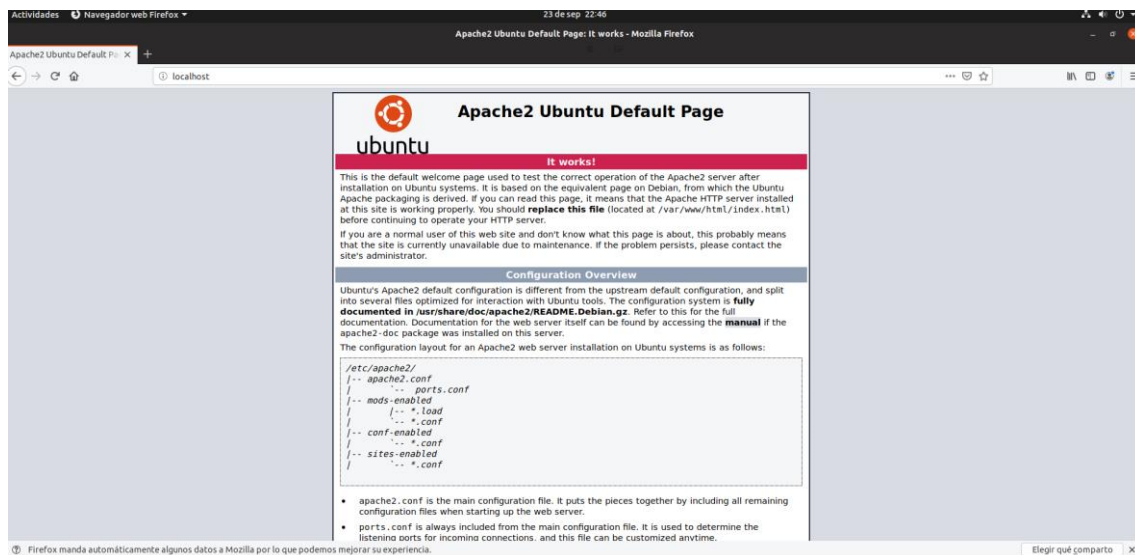
Abrimos el terminal e instalamos php con el comando: **sudo apt install php**

```
usuario@usuario-VirtualBox: ~  
usuario@usuario-VirtualBox:~$ sudo apt install php  
[sudo] contraseña para usuario:  
Leyendo lista de paquetes... Hecho  
Creando árbol de dependencias  
Leyendo la información de estado... Hecho  
Se instalarán los siguientes paquetes adicionales:  
  apache2 apache2-bin apache2-data apache2-utils libapache2-mod-php7.2 libapr1  
  libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap liblua5.2-0 php-common  
  php7.2 php7.2-cli php7.2-common php7.2-json php7.2-opcache php7.2-readline  
Paquetes sugeridos:  
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom php-pear  
Se instalarán los siguientes paquetes NUEVOS:  
  apache2 apache2-bin apache2-data apache2-utils libapache2-mod-php7.2 libapr1  
  libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap liblua5.2-0 php  
  php-common php7.2 php7.2-cli php7.2-common php7.2-json php7.2-opcache  
  php7.2-readline  
0 actualizados, 18 nuevos se instalarán, 0 para eliminar y 2 no actualizados.  
Se necesita descargar 5.748 kB de archivos.  
Se utilizarán 25,2 MB de espacio de disco adicional después de esta operación.  
¿Desea continuar? [S/n]
```

Nos preguntará si deseamos continuar, presionamos la tecla **s** y **enter**.

Finalizará la instalación automáticamente. Para comprobar que se ha instalado bien, abrimos el navegador web y escribimos **localhost/**

A continuación, presionando **enter**, nos debe de aparecer la página de **Apache 2**.



Con esto sería suficiente, pero vamos a activar una directiva de apache para no tener que trabajar con sudo en nuestra carpeta personal.

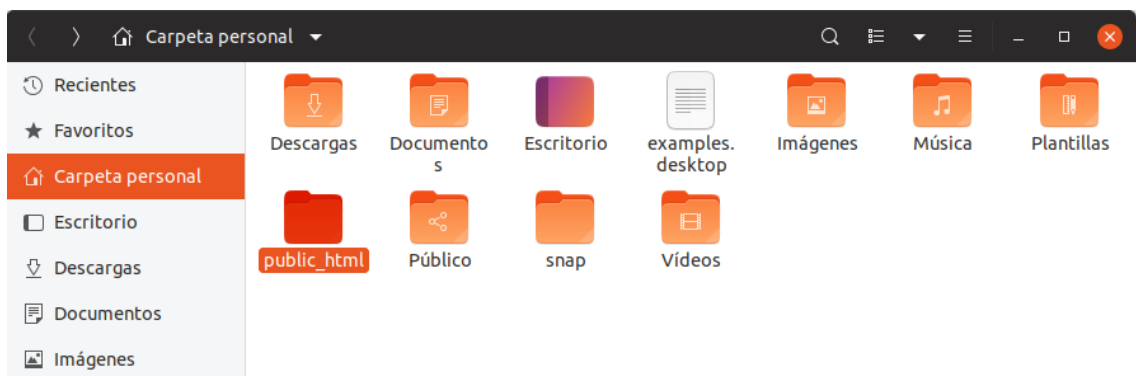
Para ello, debemos de hacer que nuestra carpeta personal se convierta en carpeta web.

Debemos de activar el módulo userdir que por defecto viene deshabilitado, con el comando:  
**sudo a2enmod userdir**

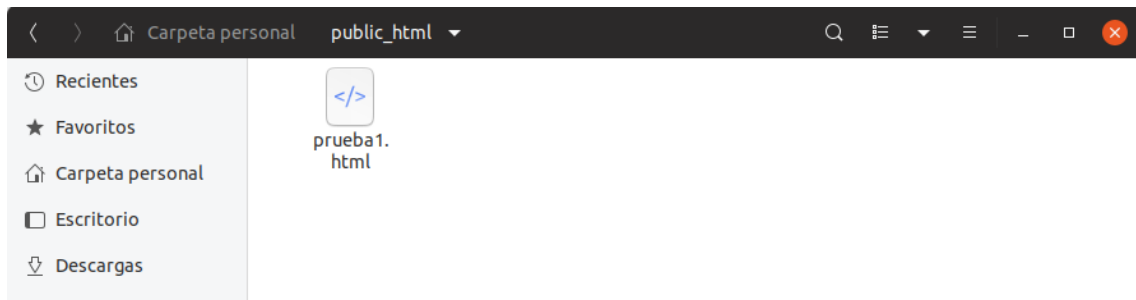
Y reiniciamos la máquina apache: **systemctl restart apache2**

```
usuario@usuario-VirtualBox:~$ sudo a2enmod userdir
Enabling module userdir.
To activate the new configuration, you need to run:
systemctl restart apache2
usuario@usuario-VirtualBox:~$ systemctl restart apache2
usuario@usuario-VirtualBox:~$
```

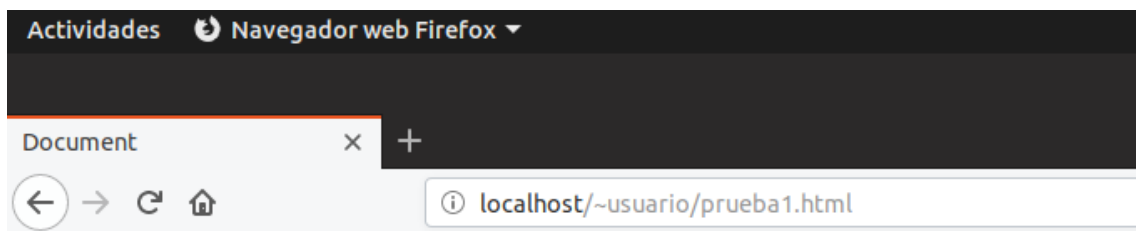
A continuación, nos vamos a nuestra carpeta **home** y creamos una carpeta llamada **public\_html**



Creamos un archivo html para comprobar que funcione bien.



Y la abrimos en el navegador.



Hola Mundo

El siguiente paso es habilitar php en la carpeta web que hemos creado.

Por tanto, debemos de irnos a la carpeta mods-enabled: **cd /etc/apache2/mods-enabled**

Y editar uno de los ficheros con el comando **sudo gedit php7.2.config**

Este comando nos abra el archivo solicitado con el editor de texto gedit.

Nos pedira la contraseña del usuario.

```
usuario@usuario-VirtualBox:~$ cd /etc/apache2/mods-enabled
usuario@usuario-VirtualBox:/etc/apache2/mods-enabled$ sudo gedit php7.2.config
[sudo] contraseña para usuario:
```



```
php7.2.conf
/etc/apache2/mods-enabled

<FilesMatch ".+\.ph(ar|p|tml)$">
    SetHandler application/x-httpd-php
</FilesMatch>
<FilesMatch ".+\.phps$">
    SetHandler application/x-httpd-php-source
    # Deny access to raw php sources by default
    # To re-enable it's recommended to enable access to the files
    # only in specific virtual host or directory
    Require all denied
</FilesMatch>
# Deny access to files without filename (e.g. '.php')
<FilesMatch "^\.ph(ar|p|ps|tml)$">
    Require all denied
</FilesMatch>

# Running PHP scripts in user directories is disabled by default
#
# To re-enable PHP in user directories comment the following lines
# (from <IfModule ...> to </IfModule>.) Do NOT set it to On as it
# prevents .htaccess files from disabling it.
<IfModule mod_userdir.c>
    <Directory /home/*/public_html>
        php_admin_flag engine Off
    </Directory>
</IfModule>
```

Texto plano ▾ Anchura del tabulador: 8 ▾ Ln 21, Col 1 ▾ INS

Una vez abierto, debemos de comentar con # las líneas que se indican en la imagen:

(Comentamos toda la línea de <IfModule></IfModule>

```
#<IfModule mod_userdir.c>
#    <Directory /home/*/public_html>
#        php_admin_flag engine Off
#    </Directory>
#</IfModule>
```


Guardamos los cambios realizados y volvemos a reiniciar la máquina apache.

```
usuario@usuario-VirtualBox:/etc/apache2/mods-enabled$ systemctl restart apache2
```

Comprobamos que funciona php. Para ello, creamos un archivo php en nuestra carpeta web y lo abrimos.

phpinfo()
+
localhost/~usuario/info.php

PHP Version 7.2.19-0ubuntu0.19.04.2



System	Linux usuario-VirtualBox 5.0.0-29-generic #31-Ubuntu SMP Thu Sep 12 13:05:32 UTC 2019 x86_64
Build Date	Aug 13 2019 11:45:23
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.2/apache2
Loaded Configuration File	/etc/php/7.2/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/7.2/apache2/conf.d
Additional .ini files parsed	/etc/php/7.2/apache2/conf.d/10-opcache.ini, /etc/php/7.2/apache2/conf.d/10-pdo.ini, /etc/php/7.2/apache2/conf.d/20-calendar.ini, /etc/php/7.2/apache2/conf.d/20-ctype.ini, /etc/php/7.2/apache2/conf.d/20-exif.ini, /etc/php/7.2/apache2/conf.d/20-fileinfo.ini, /etc/php/7.2/apache2/conf.d/20-ftp.ini, /etc/php/7.2/apache2/conf.d/20-gettext.ini, /etc/php/7.2/apache2/conf.d/20-iconv.ini, /etc/php/7.2/apache2/conf.d/20-xml.ini, /etc/php/7.2/apache2/conf.d/20-xmlrpc.ini, /etc/php/7.2/apache2/conf.d/20-zip.ini, /etc/php/7.2/apache2/conf.d/20-zlib.ini, /etc/php/7.2/apache2/conf.d/20-sockets.ini, /etc/php/7.2/apache2/conf.d/20-sysmsg.ini, /etc/php/7.2/apache2/conf.d/20-syssem.ini, /etc/php/7.2/apache2/conf.d/20-sysshm.ini, /etc/php/7.2/apache2/conf.d/20-tokenizer.ini
PHP API	20170718
PHP Extension	20170718
Zend Extension	320170718
Zend Extension Build	API320170718.NTS
PHP Extension Build	API20170718.NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled
Zend Memory Manager	enabled
Zend Multibyte Support	disabled
IPv6 Support	enabled
DTrace Support	available, disabled
Registered PHP Streams	https, ftps, compress.zlib, php, file, glob, data, http, ftp, phar

Por último, vamos a habilitar que php nos avise de los errores de nuestro código cuando abramos los archivos en el navegador.

Nos vamos a la carpeta apache2: **cd php/7.2/apache2/**

Y con el comando **ls**, listamos los archivos y directorios que se encuentran dentro.

```
usuario@usuario-VirtualBox:/etc$ cd php/7.2/apache2/  
usuario@usuario-VirtualBox:/etc/php/7.2/apache2$
```

```
usuario@usuario-VirtualBox:/etc/php/7.2/apache2$ ls
conf.d  php.ini
```

El archivo que nos concierne es **php.ini**.

Como vamos a modificarlo, vamos a crear una copia de seguridad por si ocurre algún error.

Para hacer una copia de ese archivo debemos de poner el siguiente comando:

**sudo cp php.ini php.ini.copia** (sudo copiar archivo original a archivo copia)

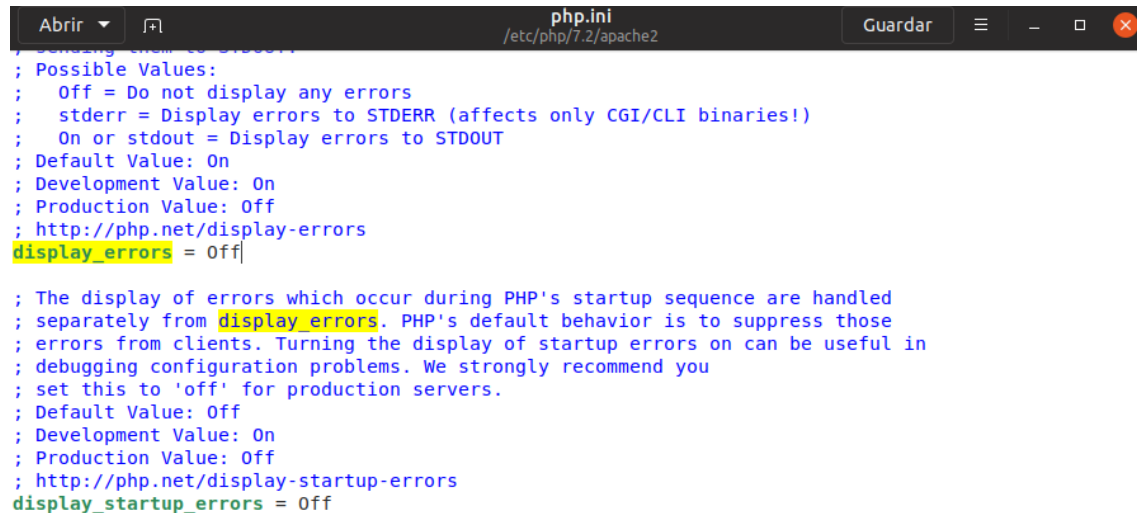
Listamos de nuevo para comprobar que se ha realizado correctamente la copia.

```
usuario@usuario-VirtualBox:/etc/php/7.2/apache2$ sudo cp php.ini php.ini.copia
usuario@usuario-VirtualBox:/etc/php/7.2/apache2$ ls
conf.d  php.ini  php.ini.copia
```

Y ahora, abrimos el archivo php.ini con gedit:

```
usuario@usuario-VirtualBox:/etc/php/7.2/apache2$ sudo gedit php.ini
```

Nos aparecerá lo siguiente:

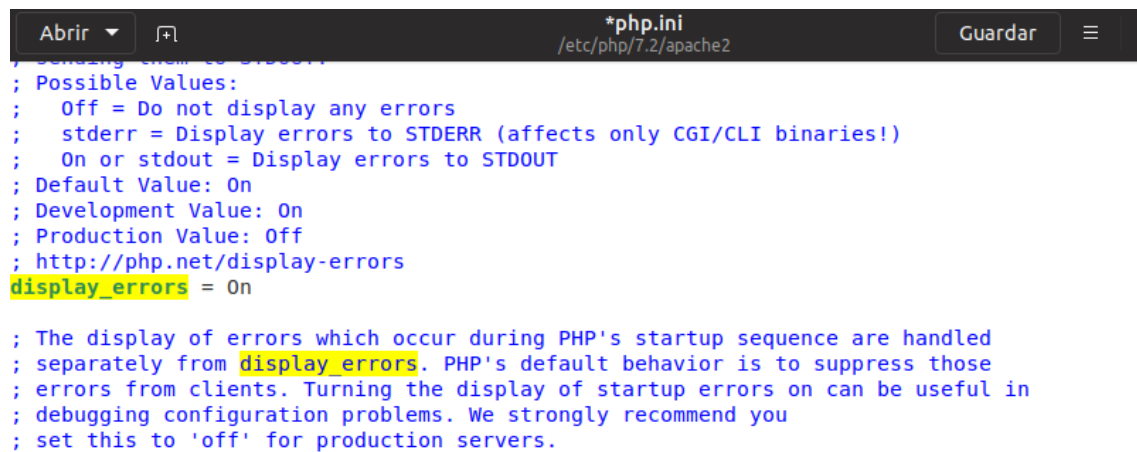


```
Abrir  [icon]  php.ini  Guardar  [icon]  [icon]  [icon]
/etc/php/7.2/apache2

; Sending them to STDERR.
; Possible Values:
;   Off = Do not display any errors
;   stderr = Display errors to STDERR (affects only CGI/CLI binaries!)
;   On or stdout = Display errors to STDOUT
; Default Value: On
; Development Value: On
; Production Value: Off
; http://php.net/display-errors
display_errors = Off

; The display of errors which occur during PHP's startup sequence are handled
; separately from display_errors. PHP's default behavior is to suppress those
; errors from clients. Turning the display of startup errors on can be useful in
; debugging configuration problems. We strongly recommend you
; set this to 'off' for production servers.
; Default Value: Off
; Development Value: On
; Production Value: Off
; http://php.net/display-startup-errors
display_startup_errors = Off
```

Debemos de hacer búsqueda de **display\_errors** y debemos activarlo cambiando **Off** por **On**



```
Abrir  [icon]  *php.ini  Guardar  [icon]  [icon]  [icon]
/etc/php/7.2/apache2

; Sending them to STDERR.
; Possible Values:
;   Off = Do not display any errors
;   stderr = Display errors to STDERR (affects only CGI/CLI binaries!)
;   On or stdout = Display errors to STDOUT
; Default Value: On
; Development Value: On
; Production Value: Off
; http://php.net/display-errors
display_errors = On

; The display of errors which occur during PHP's startup sequence are handled
; separately from display_errors. PHP's default behavior is to suppress those
; errors from clients. Turning the display of startup errors on can be useful in
; debugging configuration problems. We strongly recommend you
; set this to 'off' for production servers.
; Default Value: Off
; Development Value: On
; Production Value: Off
; http://php.net/display-startup-errors
display_startup_errors = Off
```

Guardamos y reiniciamos apache:

```
usuario@usuario-VirtualBox:/etc/php/7.2/apache2$ systemctl restart apache2
```

Comprobamos que efectivamente, nos muestra los errores cuando los hay.



Nos avisa de que en la línea 14 hay un error. Efectivamente, no le pusimos ; a nuestro echo.



¡Y listo!