



Herramientas opcionales

Instalación apache y configuración de la página web

Lo primero, obviamente, es configurar apache. Si no lo tenemos instalado, se debe instalar.

```
root@servidor:/# apt install apache2
```

Tras ello, instalamos PHP.

```
root@servidor:/# apt install php libapache2-mod-php php-mysql
```

Revisamos la versión de PHP con el siguiente comando.

```
root@servidor:/# php -v
PHP 8.1.2-1ubuntu2.14 (cli) (built: Aug 18 2023 11:41:11) (NTS)
Copyright (c) The PHP Group
Zend Engine v4.1.2, Copyright (c) Zend Technologies
    with Zend OPcache v8.1.2-1ubuntu2.14, Copyright (c), by Zend Technologies
```

Ahora debemos ir al archivo 000-default.conf en /etc/apache2/sites-enabled para poder configurar el document root donde cambiaremos /var/www/html por el directorio donde vayamos a poner nuestra página HTML. En nuestro caso lo alojaremos en el directorio /trabajo.

```
root@servidor:/etc/apache2/sites-enabled# nano 000-default.conf
```



```
GNU nano 6.2                                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 /trabajo

# 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

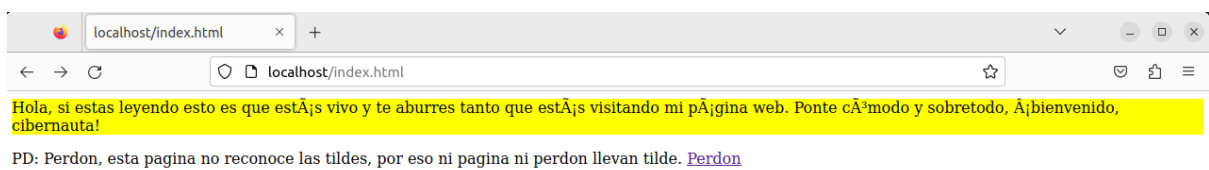
# For most configuration files from conf-available/, which are
# enabled or disabled at a global level, it is possible to
# include a line for only one particular virtual host. For example the
# following line enables the CGI configuration for this host only
# after it has been globally disabled with "a2disconf".
#Include conf-available/serve-cgi-bin.conf
</VirtualHost>

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

Dentro del directorio /trabajo crearemos un index.html que será el fichero donde configuremos nuestra página web. Lo editaremos con el comando nano y pondremos “Hola mundo” para ver si lo reconoce a buscarlo en el navegador.

```
root@servidor: /trabajo
GNU nano 6.2                                index.html
<html>
<head>
</head>
<body>
<p style="background-color:yellow">Hola, si estas leyendo esto es que estás vivo y te aburres tanto que estás visitando mi página web. Ponte cómodo y sobretodo, ¡bienvenido, cibernauta!</p>
<p>PD: Perdon, esta pagina no reconoce las tildes, por eso ni pagina ni perdon llevan tilde.</p>
<p>Perdon</p>
</body>
</html>
```

En el buscador del navegador de nuestro ordenador podremos buscar la página a través de “localhost/”, “localhost/index.html” o en base a nuestra dirección IP local.



Una vez nos aparece, podremos configurar la página web a nuestro gusto.



Instalación PHP y configuración Xdebug

En el apartado interior ya hemos instalado PHP. Ahora en nuestro directorio de trabajo crearemos un documento php llamado info.php

```
root@servidor:/trabajo# touch info.php
```

Dentro de éste configuraremos PHP.

```
GNU nano 6.2 info.php *
<?php
phpinfo();
?>
```

Para saber si está bien configurado en el buscador ponemos localhost/info.php y nos deberá aparecer la página por defecto de php.

PHP Version 8.1.2-1ubuntu2.14	
System	Linux servidor 6.2.0-34-generic #34~22.04.1-Ubuntu SMP PREEMPT_DYNAMIC Thu Sep 7 13:12:03 UTC 2 x86_64
Build Date	Aug 18 2023 11:41:11
Build System	Linux
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/8.1/apache2
Loaded Configuration File	/etc/php/8.1/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/8.1/apache2/conf.d
Additional .ini files parsed	/etc/php/8.1/apache2/conf.d/10-mysqlnd.ini, /etc/php/8.1/apache2/conf.d/10-opcache.ini, /etc/php/8.1/apache2/conf.d/10-pdo.ini, /etc/php/8.1/apache2/conf.d/20-calendar.ini, /etc/php/8.1/apache2/conf.d/20-ctype.ini, /etc/php/8.1/apache2/conf.d/20-exif.ini, /etc/php/8.1/apache2/conf.d/20-ffi.ini, /etc/php/8.1/apache2/conf.d/20-fileinfo.ini, /etc/php/8.1/apache2/conf.d/20-ftp.ini, /etc/php/8.1/apache2/conf.d/20-gettext.ini, /etc/php/8.1/apache2/conf.d/20-iconv.ini, /etc/php/8.1/apache2/conf.d/20-mysqli.ini, /etc/php/8.1/apache2/conf.d/20-pdo_mysql.ini, /etc/php/8.1/apache2/conf.d/20-phar.ini, /etc/php/8.1/apache2/conf.d/20-posix.ini, /etc/php/8.1/apache2/conf.d/20-readline.ini, /etc/php/8.1/apache2/conf.d/20-shmop.ini, /etc/php/8.1/apache2/conf.d/20-sockets.ini, /etc/php/8.1/apache2/conf.d/20-sysvmsg.ini, /etc/php/8.1/apache2/conf.d/20-sysvsem.ini, /etc/php/8.1/apache2/conf.d/20-sysvshm.ini, /etc/php/8.1/apache2/conf.d/20-tokenizer.ini
PHP API	20210902
PHP Extension	20210902
Zend Extension	420210902
Zend Extension Build	API420210902.NTS
PHP Extension Build	API20210902.NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled
Zend Memory Manager	enabled
Zend Multibyte Support	disabled

Para configurar el Xdebug deberíamos buscar la versión de PHP con `php -v`, como hemos visto anteriormente. En base a la versión instalaremos el xdebug con el siguiente comando:

```
root@servidor:/trabajo# php -v
PHP 8.1.2-1ubuntu2.14 (cli) (built: Aug 18 2023 11:41:11) (NTS)
Copyright (c) The PHP Group
Zend Engine v4.1.2, Copyright (c) Zend Technologies
    with Zend OPcache v8.1.2-1ubuntu2.14, Copyright (c), by Zend Technologies
root@servidor:/trabajo# apt install php8.1-xdebug
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  php8.1-xdebug
0 upgraded, 1 newly installed, 0 to remove and 3 not upgraded.
Need to get 621 kB of archives.
After this operation, 1.863 kB of additional disk space will be used.
Get:1 http://es.archive.ubuntu.com/ubuntu jammy/universe amd64 php8.1-xdebug amd64 3.1.2+2.9.8+2.8.1+2.5.5-4 [621 kB]
Fetched 621 kB in 1s (771 kB/s)
Selecting previously unselected package php8.1-xdebug.
(Reading database ... 199811 files and directories currently installed.)
Preparing to unpack .../php8.1-xdebug_3.1.2+2.9.8+2.8.1+2.5.5-4_amd64.deb ...
Unpacking php8.1-xdebug (3.1.2+2.9.8+2.8.1+2.5.5-4) ...
Setting up php8.1-xdebug (3.1.2+2.9.8+2.8.1+2.5.5-4) ...
Processing triggers for php8.1-cli (8.1.2-1ubuntu2.14) ...
Processing triggers for libapache2-mod-php8.1 (8.1.2-1ubuntu2.14) ...
root@servidor:/trabajo#
```

Tras instalarlo, recargamos nuestro servicio apache.

```
root@servidor:/trabajo# systemctl reload apache2
```

Para revisar si se ha instalado y configurado xdebug, vamos a nuestro buscador e introduciendo `localhost/info.php` podemos revisarlo bajando la barra.

PHP 8.1.2-1ubuntu2.14 - php: x

localhost/info.php

Tokenizer Support enabled

xdebug

Version 3.1.2

[Support Xdebug on Patreon, GitHub, or as a business](#)

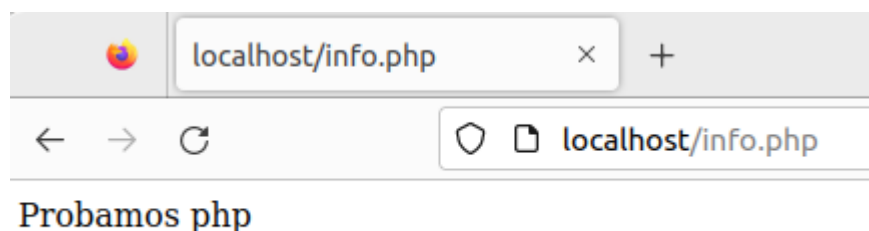
Enabled Features (through 'xdebug.mode' setting)		
Feature	Enabled/Disabled	Docs
Development Helpers	✓ enabled	Docs
Coverage	✗ disabled	Docs
GC Stats	✗ disabled	Docs
Profiler	✗ disabled	Docs
Step Debugger	✗ disabled	Docs
Tracing	✗ disabled	Docs

Optional Features	
Compressed File Support	no
Clock Source	clock_gettime

Directive	Local Value	Master Value
xdebug.auto_trace	(setting renamed in Xdebug 3)	(setting renamed in Xdebug 3)
xdebug.cli_color	0	0
xdebug.client_discovery_header	no value	no value
xdebug.client_host	localhost	localhost
xdebug.client_port	9003	9003

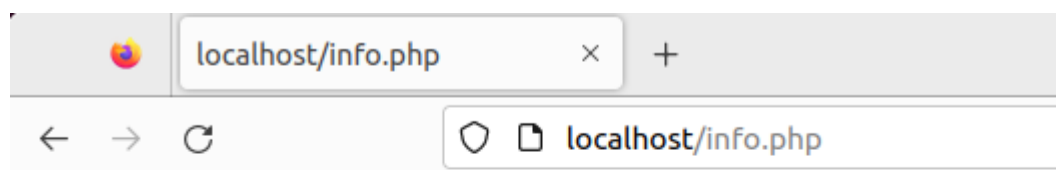
Configuramos el info.php para probar si funciona php.

```
GNU nano 6.2 info.php
<?php
echo "Probamos php";
?>
```



Funciona perfectamente. Ahora meteremos texto al azar sin etiquetas.

```
GNU nano 6.2 info.php *
<?php
sddjdjds
echo "Probamos php";
dsjfdsjsd
?>
```



Nos dará error ya que no aparecerá nada ya que habrá un error en el documento info.php. Recordemos que el xdebug sirve para que te marque, leyendo el documento de arriba hacia abajo, los diferentes fallos que puedan afectar al documento. Para que el xdebug actúe debemos configurarlo en la siguiente ruta:

/etc/php/8.1/apache2/php.ini

Dentro configuraremos el archivo php.ini como queramos nosotros configurarlo. Nos iremos al final del documento y escribiremos lo siguiente:

- xdebug.mode=debug (que trabaje en modo debug)

- xdebug.client_host=localhost (que responda las peticiones localhost)
- xdebug.client_port=9003 (por defecto que se conecte por el puerto)
- xdebug.idekey="VSCODE" (le especificamos el ide que vamos a usar)

```
GNU nano 6.2 /etc/php/8.1/apache2/php.ini *
opcache.file_update_protection=2
; Absolute path used to store shared lockfiles (for *nix only).
opcache.lockfile_path=/tmp

[curl]
; A default value for the CURLOPT_CAINFO option. This is required to be an
; absolute path.
curl.cainfo =

[openssl]
; The location of a Certificate Authority (CA) file on the local filesystem
; to use when verifying the identity of SSL/TLS peers. Most users should
; not specify a value for this directive as PHP will attempt to use the
; OS-managed cert stores in its absence. If specified, this value may still
; be overridden on a per-stream basis via the "cafile" SSL stream context
; option.
openssl.cafile=

; If openssl.cafile is not specified or if the CA file is not found, the
; directory pointed to by openssl.capath is searched for a suitable
; certificate. This value must be a correctly hashed certificate directory.
; Most users should not specify a value for this directive as PHP will
; attempt to use the OS-managed cert stores in its absence. If specified,
; this value may still be overridden on a per-stream basis via the "capath"
; SSL stream context option.
openssl.capath=

[ffi]
; FFI API restriction. Possible values:
; "preload" - enabled in CLI scripts and preloaded files (default)
; "false" - always disabled
; "true" - always enabled
ffi.enable=preload

; List of headers files to preload, wildcard patterns allowed.
ffi.preload=

xdebug.mode=debug
xdebug.client_host=localhost
xdebug.client_port=9003
xdebug.idekey="VSCODE"
```

Hecho esto, pulsamos Ctrl + W y buscamos "display_errors = Off". Al hacerlo nos llevará a la línea y en ella deberemos poner On en lugar de Off, tanto en display_errors como en display_startup_errors, que también estará en Off.

```
display_errors = On

; The display of errors which occur during PHP's startup sequence are handled
; separately from display_errors. We strongly recommend you set this to 'off'
; for production servers to avoid leaking configuration details.
; Default Value: On
; Development Value: On
; Production Value: Off
; https://php.net/display-startup-errors
display_startup_errors = On
```



Tras guardar el documento con Ctrl + x, reiniciamos el servidor con `systemctl restart apache2`.

Al reiniciarlo e intentar acceder a `localhost/php.ini`, donde no nos aparecía nada, solamente la pantalla en blanco, ahora nos aparecerá el error en el documento php.



Nos dice que en la línea 3 de documento `info.php` dentro del directorio `/trabajo` hay un error de sintaxis, cosa que no aparecía anteriormente y nos ayudará en el futuro a poder corregir nuestros errores al desarrollar una página.

Instalación SSL

SSL es una herramienta o software que proporciona una certificación lets encrypt (abierta, automatizada y gratuita) que se instala en los servidores web para habilitar el protocolo HTTPS, protocolo principal usado para enviar datos entre un navegador web y un sitio web, aumentando así la seguridad de los datos.

Para hacerlo en nuestro servidor ejecutaremos el siguiente comando:

```
root@servidor:/home/pablo# apt install certbot python3-certbot-apache
```

Como hemos estado trabajando con el servicio apache, para activarlo en nuestro servidor y dominio ejecutaremos el siguiente comando:

```
root@servidor:/home/pablo# certbot --apache
```

Nos pedirá el correo del dominio con el que lo hemos registrado.

```
Enter email address (used for urgent renewal and security notices)
(Enter 'c' to cancel):
```

Diremos yes, no, y luego introduciremos el nombre de nuestro dominio. Para ver si se ha activado o no, introduciremos el comando "systemctl status certbot.timer". Al ver que está activo, reiniciaremos el servidor web, apache (systemctl restart apache2).

```
root@servidor:/home/pablo# systemctl status certbot.timer
● certbot.timer - Run certbot twice daily
   Loaded: loaded (/lib/systemd/system/certbot.timer; enabled; vendor preset:
   Active: active (waiting) since Wed 2023-10-18 21:46:06 CEST; 4min 19s ago
   Trigger: Thu 2023-10-19 07:15:34 CEST; 9h left
   Triggers: ● certbot.service

oct 18 21:46:06 servidor systemd[1]: Started Run certbot twice daily.
lines 1-7/7 (END)
```


Instalación y uso de GIT

Lo primero que debemos hacer es instalar el software git. Como siempre, apt install git.

Una vez lo hayamos instalado deberemos darnos de alta en la web www.github.com.

Ya registrados debemos darnos de alta debemos crear una clave ssh para identificarnos:

```
root@servidor:/home/pablo# ssh-keygen -C "pmspablo@hotmail.es"
```

Nos pedirá un fichero donde guardar la clave. En mi caso elegiré "clave", para no complicarme y acordarme. También su contraseña.

```
Enter file in which to save the key (/root/.ssh/id_rsa): clave
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in clave
Your public key has been saved in clave.pub
The key fingerprint is:
SHA256:4WdE9xIbaI105oJ8NXEYeyN31Ud2i5qacCbxAU91gKo pmspablo@hotmail.es
The key's randomart image is:
+---[RSA 3072]---+
|      . . *X* . . *|
|      +oX+o=.o=|
|      . ooX+. * . o..|
|      oo=O++O+|
|      .+S=oo|
|      E  =oo|
|      o|
|      |
+-----[SHA256]-----+
```

Con cat clave.pub podemos ver la clave. Con ésta debemos irnos a www.github.com, y en la parte superior derecha, donde el icono de nuestro perfil, iremos a "Settings". Dentro nos iremos a "SSH and GPG keys", en el menú de la izquierda.

SSH and GPG keys



Le daremos a “new SSH key”, le pondremos un nombre, en mi caso asir.com y la clave que nos ha generado en el archivo clave.pub la pegaremos en el recuadro de abajo.

Add new SSH Key

Title

Key type

Authentication Key ↕

Key

Begins with 'ssh-rsa', 'ecdsa-sha2-nistp256', 'ecdsa-sha2-nistp384', 'ecdsa-sha2-nistp521', 'ssh-ed25519', 'sk-ecdsa-sha2-nistp256@openssh.com', or 'sk-ssh-ed25519@openssh.com'

Add SSH key

Podremos ahora clonar nuestro repositorio de GitHub. O bien con SSH, o bien con HTTPS. En mi caso lo haré con HTTPS. ¿Cómo? Con el siguiente comando.

```
pablo@servidor:~/pablo/pablo$ git clone https://github.com/pablotepalote/pablo.git
Cloning into 'pablo'...
warning: You appear to have cloned an empty repository.
```

La sintaxis de la ruta es comando git + parámetro clone + dirección https donde ponemos el dominio de github seguido por el nombre de usuario y por último el nombre de nuestro repositorio git.

En caso de haberlo hecho con SSH la sintaxis sería: `git clone git@github.com:pablotepalote/pablo.git`

Una vez clonado, empezaremos a trabajar en el documento HTML de nuestra carpeta. Si no lo tenemos lo creamos.

```
pablo@servidor:~/pablo/pablo$ ls
hola.html  pablo
pablo@servidor:~/pablo/pablo$ echo nuevas líneas de prueba >> hola.html
```

En mi caso ya lo tenía creado el documento hola.html, por lo que le he comentado nuevamente con un echo. Para poder subir nuestro archivo debemos ejecutar los comandos `git add .` o `git add hola.html`, el documento html creado previamente. Es preferible `git add .` debido a que añadiríamos todos los documentos al buffer, no sólo el archivo hola.html.

```
pablo@servidor:~/pablo/pablo$ git add hola.html
```

Debemos realizar ahora el comentario en la subida con el siguiente comando:

`Git commit -m "nuevo comentario"`

Al hacerlo nos saldrá el paso más importante para la subida a github: identificarnos.

```
pablo@servidor:~/pablo/pablo$ git commit -m "comentario nuevo"
Author identity unknown

*** Please tell me who you are.

Run

  git config --global user.email "you@example.com"
  git config --global user.name "Your Name"

to set your account's default identity.
Omit --global to set the identity only in this repository.
```

Solamente debemos rellenarlo con el usuario de nuestra cuenta de github y en la siguiente orden con nuestro correo electrónico de github.

```
pablo@servidor:~/pablo/pablo$ git config --global user.email 'pablote@pablote.es'
pablo@servidor:~/pablo/pablo$ git config --global user.name "pablote@pablote"
```

Una vez nos hayamos registrado podremos ya comentarlo con el comentario expuesto anteriormente.

```
pablo@servidor:~/pablo/pablo$ sudo git commit -m "nuevo comentario"
[main b947901] nuevo comentario
1 file changed, 1 insertion(+)
```

Por último sólo quedaría el push del repositorio a la rama principal de nuestro repositorio Git con el comando `"git push -u origin main"` donde nos pedirá nuestros credenciales. Y listo.

