

# GUÍA INSTALACIÓN LAMP



Jefa de Equipo: Cristina Moure  
Integrantes: Guillermo Moreno, Adrián Alonso, Patryk Piasecki

## Índice

¿Qué es LAMP?.....	2
Requisitos del sistema.....	2
Recomendación en el proceso de Instalación.....	2
<b>Opción Mysql versión 7.4.3:.....</b>	<b>3</b>
Instalación de Apache.....	3
Instalación de PHP.....	6
Instalación de MySQL.....	7
Comprobación de Funcionamiento del Servidor.....	8
<b>Opción Mysql versión 8.1.10 de MaridDB:.....</b>	<b>11</b>
Instalación de Apache.....	11
Configuración de PHP:.....	13
Comprobación de MySQL.....	15
Comprobación de Funcionamiento del Servidor.....	16
Conclusión.....	17
Bibliografía.....	17

### ¿Qué es LAMP?

LAMP es el acrónimo usado para describir un sistema de infraestructura de internet que usa las siguientes herramientas:

- Linux, el sistema operativo; En algunos casos también se refiere a LDAP.
- Apache, el servidor web;
- MySQL/MariaDB, el gestor de bases de datos;
- PHP, el lenguaje de programación.

Por lo que la combinación de estas tecnologías es usada principalmente para definir la infraestructura de un servidor web, utilizando un paradigma de programación para el desarrollo del sistema.

Dicho esto, vamos a necesitar estas herramientas y por ello instalar sus respectivos paquetes

### Requisitos del sistema

- Requisitos mínimos del Hardware:
  - 128 MB de RAM
  - CPU de 800 MHz
  - 1 Gb de Disco duro SSD
- Tener instalado la versión de Ubuntu 20.04 LTS

### Recomendación en el proceso de Instalación

**En esta guía podemos instalar LAMP, MySQL versión 7.4.3 o la versión 8.1.10 usando MariaDB (aunque hay muchas versiones de pila LAMP que aún no son compatibles con una versión tan avanzada), se recomienda no instalar ambos ya que podría causar fallos en el servidor que a continuación vamos a instalar paso a paso.**

**Son formas parecidas de instalar, pero con pequeñas diferencias.**

## Opción Mysql versión 7.4.3:

### Instalación de Apache

Lo primero será actualizar paquetes con el siguiente comando y nos pediría la contraseña:

**sudo apt-get update**

```
alumno@alumnov:~$ sudo apt-get update
[sudo] contraseña para alumno:
```

Una vez terminado nos saltará el siguiente mensaje

```
Leyendo lista de paquetes... Hecho
alumno@alumnov:~$
```

Instalaremos apache con el siguiente comando

**sudo apt install apache2**

```
alumno@alumnov:~$ sudo apt install apache2
```

Nos pedirá confirmación y le damos que si

```
¿Desea continuar? [S/n]
```

Una vez instalado configuraremos el firewall con el siguiente comando

**sudo ufw app list**

```
Procesando aplicaciones para cada regla
alumno@alumnov:~$ sudo ufw app list
```

Veremos que nos sale lo siguiente

```
alumno@alumnov:~$ sudo ufw app list
Aplicaciones disponibles:
  Apache
  Apache Full
  Apache Secure
  CUPS
```

Y abre estos puertos

- Apache: este perfil abre solo el puerto 80 (tráfico web normal no cifrado).
- Apache Full: este perfil abre los puertos 80 (tráfico web normal no cifrado) y 443 (tráfico TLS/SSL cifrado).
- Apache Secure: este perfil abre solo el puerto 443 (tráfico TLS/SSL cifrado).

## Instalación LAMP

Y como solo usaremos apache abriremos el primero de la siguiente manera:

**sudo ufw allow in "Apache"**

```
alumno@alumnomv:~$ sudo ufw allow in "Apache"
Reglas actualizadas
Reglas actualizadas (v6)
```

Ponemos los siguiente para habilitar el cortafuegos:

**sudo ufw enable**

```
alumno@alumnomv:~$ sudo ufw enable
El cortafuegos está activo y habilitado en el arranque del sistema
```

Y comprobamos que este toco correcto:

**sudo ufw status**

```
alumno@alumnomv:~$ sudo ufw status
Estado: activo

Hasta          Acción         Desde
-----
Apache         ALLOW          Anywhere
Apache (v6)    ALLOW          Anywhere (v6)
```

Iniciaremos el servicio con el siguiente comando

**sudo service apache2 restart**

```
alumno@alumnomv:~$ sudo service apache2 restart
```

Comprobamos el estado de nuestro servidor con el siguiente comando

```
alumno@alumnomv:~$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2022-09-15 12:56:22 CEST; 5min ago
     Docs: https://httpd.apache.org/docs/2.4/
   Process: 34579 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
   Main PID: 34583 (apache2)
     Tasks: 6 (limit: 9450)
    Memory: 9.4M
   CGroup: /system.slice/apache2.service
           └─34583 /usr/sbin/apache2 -k start
             └─34585 /usr/sbin/apache2 -k start
               └─34586 /usr/sbin/apache2 -k start
                 └─34587 /usr/sbin/apache2 -k start
                   └─34588 /usr/sbin/apache2 -k start
                     └─34589 /usr/sbin/apache2 -k start

sep 15 12:56:22 alumnomv systemd[1]: Starting The Apache HTTP Server...
sep 15 12:56:22 alumnomv apachectl[34582]: AH00558: apache2: Could not reliably determine the server's fully qualified domain name
sep 15 12:56:22 alumnomv systemd[1]: Started The Apache HTTP Server.
lines 1-19/19 (END)
```

Con lo siguiente veremos la ip de nuestro servidor:

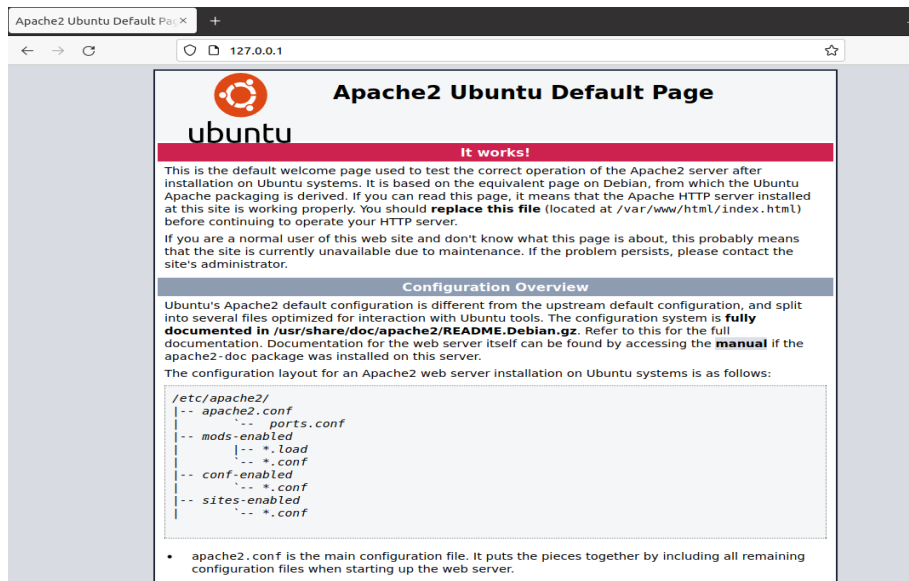
**hostname -I**

```
alumno@alumnomv:~$ hostname -I
10.33.30.2
```

## Instalación LAMP

Y para comprobar que este iniciado vamos a Google y ponemos lo siguiente <https://10.33.30.2>, también podemos con <https://127.0.0.1>

Y nos saldrá lo siguiente



## Instalación de PHP

Instalaremos los siguientes paquetes:

**sudo apt install php**

**sudo apt install php-mysql**

```
alumno@alumnov:~$ sudo apt install php
```

```
alumno@alumnov:~$ sudo apt install php-mysql
```

Para confirmar que todo está correcto ver la versión de php pondremos lo siguiente:

**php -v**

```
alumno@alumnov:~$ php -v
PHP 7.4.3 (cli) (built: Aug 17 2022 13:29:56) ( NTS )
Copyright (c) The PHP Group
Zend Engine v3.4.0, Copyright (c) Zend Technologies
    with Zend OPcache v7.4.3, Copyright (c), by Zend Technologies
alumno@alumnov:~$
```

## Instalación de MySQL

Pondremos el siguiente comando para instalar:

```
sudo apt-get install mysql-server
```

```
alumno@alumnov:~$ sudo apt-get install mysql-server
```

Después de la confirmación vemos que se instala sin fallos, lo siguiente será crear la base de datos con este comando

```
mysql -u root -p
```

```
alumno@alumnov:~$ mysql -u root -p
Enter password: 
```

Y nos saldrá que se ha creado correctamente

```
alumno@alumnov:~$ sudo mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 14
Server version: 8.0.30-0ubuntu0.20.04.2 (Ubuntu)

Copyright (c) 2000, 2022, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> 
```

Crearemos la base de datos con el siguiente parámetro

```
mysql> create database webdata;
Query OK, 1 row affected (0,05 sec)

mysql> 
```

Le daremos los siguientes parámetros

```
mysql> grant all on database.*to"admin" identified by "password";
```



## Comprobación de Funcionamiento del Servidor

Crearemos el directorio para nuestra web:

**sudo mkdir /var/www/servidor**

```
alumno@alumnomv:~$ sudo mkdir /var/www/servidor
```

hará referencia a su usuario de sistema actual de la siguiente forma:

**sudo chown -R \$USER:\$USER /var/www/servidor**

```
alumno@alumnomv:~$ sudo chown -R $USER:$USER /var/www/servidor
```

Crearemos un archivo con la configuración básica:

**sudo nano /etc/apache2/sites-available/servidor.conf**

```
alumno@alumnomv:~$ sudo nano /etc/apache2/sites-available/servidor.conf
```

Y metemos la siguiente información en su interior:

```
GNU nano 4.8 /etc/apache2/sites-available/servidor.conf Modificado
<VirtualHost *:80>
    ServerName servidor
    ServerAlias www.servidor
    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/servidor
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

Habilitamos el sitio con el siguiente comando:

**sudo a2ensite servidor**

```
alumno@alumnomv:~$ sudo a2ensite servidor
Enabling site servidor.
To activate the new configuration, you need to run:
    systemctl reload apache2
```

Deshabilitaremos el sitio web predeterminado por apache con lo siguiente:

**sudo a2dissite 000-default**

```
alumno@alumnomv:~$ sudo a2dissite 000-default
Site 000-default disabled.
To activate the new configuration, you need to run:
    systemctl reload apache2
```

## Instalación LAMP

Verificaremos que no tenga errores:

**sudo apache2ctl configtest**

```
alumno@alumnov:~$ sudo apache2ctl configtest
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using
127.0.1.1. Set the 'ServerName' directive globally to suppress this message
Syntax OK
```

Y cargaremos apache:

**sudo systemctl reload apache2**

```
alumno@alumnov:~$ sudo systemctl reload apache2
```

Crearemos un html para probar que funcione el host virtual:

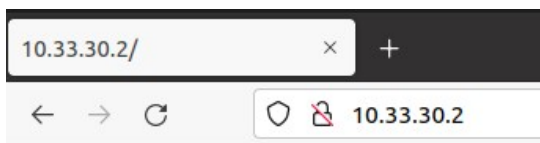
**nano /var/www/servidor/index.html**

```
alumno@alumnov:~$ nano /var/www/servidor/index.html
```

Y le metemos lo siguiente contenido:

```
GNU nano 4.8 /var/www/servidor/index.html
<h1>Funciona</h1>
<p>Inicio de la pagina.</p>
```

Y en la web se verá así:



# Funciona

Inicio de la pagina.

Probaremos el funcionamiento de php con el siguiente comando:

**nano /var/www/servidor/info.php**

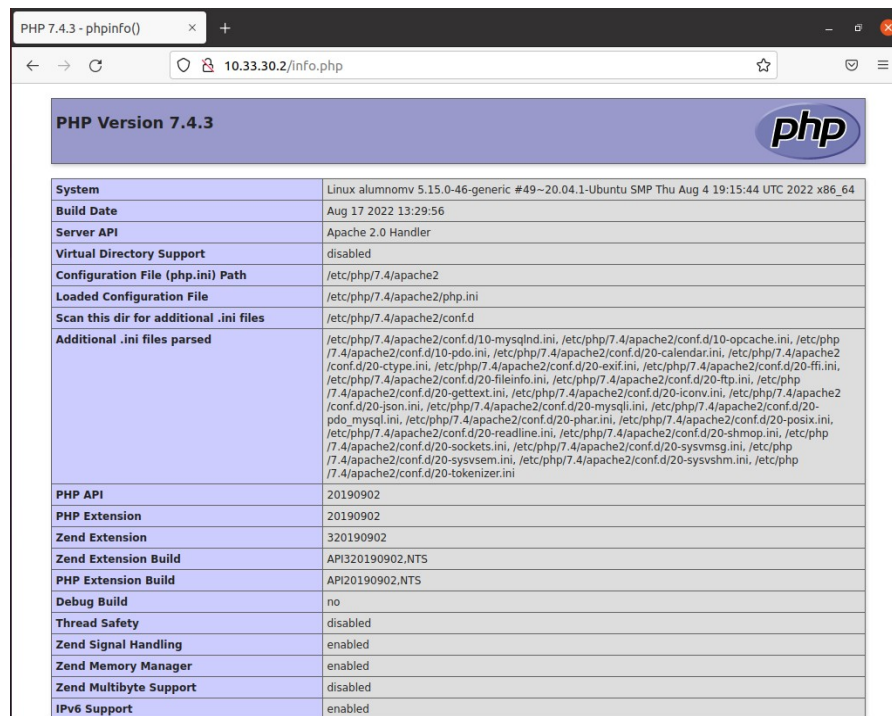
```
alumno@alumnov:~$ nano /var/www/servidor/info.php
```

Dentro metemos lo siguiente:

```
GNU nano 4.8 /var/www/servidor/info.php
<?php
phpinfo();
```

## Instalación LAMP

Y si vamos al buscador y ponemos la siguiente dirección veremos la información de php:



PHP Version 7.4.3	
System	Linux alumnovm 5.15.0-46-generic #49~20.04.1-Ubuntu SMP Thu Aug 4 19:15:44 UTC 2022 x86_64
Build Date	Aug 17 2022 13:29:56
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.4/apache2
Loaded Configuration File	/etc/php/7.4/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/7.4/apache2/conf.d
Additional .ini files parsed	/etc/php/7.4/apache2/conf.d/10-mysqlnd.ini, /etc/php/7.4/apache2/conf.d/10-opcache.ini, /etc/php/7.4/apache2/conf.d/10-pdo.ini, /etc/php/7.4/apache2/conf.d/20-calendar.ini, /etc/php/7.4/apache2/conf.d/20-ctype.ini, /etc/php/7.4/apache2/conf.d/20-exif.ini, /etc/php/7.4/apache2/conf.d/20-ffi.ini, /etc/php/7.4/apache2/conf.d/20-fileinfo.ini, /etc/php/7.4/apache2/conf.d/20-ftp.ini, /etc/php/7.4/apache2/conf.d/20-gettext.ini, /etc/php/7.4/apache2/conf.d/20-iconv.ini, /etc/php/7.4/apache2/conf.d/20-jison.ini, /etc/php/7.4/apache2/conf.d/20-mysqli.ini, /etc/php/7.4/apache2/conf.d/20-pdo_mysql.ini, /etc/php/7.4/apache2/conf.d/20-phar.ini, /etc/php/7.4/apache2/conf.d/20-posix.ini, /etc/php/7.4/apache2/conf.d/20-readline.ini, /etc/php/7.4/apache2/conf.d/20-shmop.ini, /etc/php/7.4/apache2/conf.d/20-sockets.ini, /etc/php/7.4/apache2/conf.d/20-sysvmsg.ini, /etc/php/7.4/apache2/conf.d/20-sysvsem.ini, /etc/php/7.4/apache2/conf.d/20-tokenizer.ini
PHP API	20190902
PHP Extension	20190902
Zend Extension	320190902
Zend Extension Build	API320190902.NTS
PHP Extension Build	API20190902.NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled
Zend Memory Manager	enabled
Zend Multibyte Support	disabled
IPv6 Support	enabled

## Opción Mysql versión 8.1.10 de MariaDB:

### Instalación de Apache

Lo primero será actualizar paquetes con el siguiente comando y nos pedirá la contraseña:

**sudo apt update**

```
alumno@alumnov:~$ sudo apt-get update
[sudo] contraseña para alumno:
```

Una vez terminado nos saltará el siguiente mensaje

```
Leyendo lista de paquetes... Hecho
alumno@alumnov:~$
```

Instalaremos todos los paquetes necesarios en un solo comando en este caso. Si usamos exclusivamente los repositorios de Ubuntu 22.04 LTS lo haremos así:

**sudo apt install -y apache2 libapache2-mod-php mariadb-server php-mysql**

```
alumno@alumnov:~/Escritorio$ sudo apt install -y apache2 libapache2-mod-php mariadb-server php-mysql
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
```

Nos pedirá confirmación y le damos que si

```
¿Desea continuar? [S/n]
```

Podemos comprobar fácilmente el estado de estos dos servicios con el comando:

**systemctl status apache2 mariadb -n 0**

```
alumno@alumnov:~/Escritorio$ systemctl status apache2 mariadb -n 0
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2022-09-15 12:26:38 CEST; 2min 59s ago
     Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 11127 (apache2)
    Tasks: 6 (limit: 9450)
   Memory: 10.5M
   CGroup: /system.slice/apache2.service
           └─11127 /usr/sbin/apache2 -k start
             └─11130 /usr/sbin/apache2 -k start
               └─11131 /usr/sbin/apache2 -k start
                 └─11132 /usr/sbin/apache2 -k start
                   └─11133 /usr/sbin/apache2 -k start
                     └─11134 /usr/sbin/apache2 -k start

● mariadb.service - MariaDB 10.3.34 database server
   Loaded: loaded (/lib/systemd/system/mariadb.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2022-09-15 12:25:52 CEST; 3min 45s ago
     Docs: man:mysqld(8)
```

## Instalación LAMP

Si tienes activado el firewall UFW y quieres acceder desde red a las aplicaciones que instales sobre la pila LAMP será necesario permitir las conexiones estándar al servicio web:

**sudo ufw allow http**

```
alumno@alumnov:~/Escritorio$ sudo ufw allow http
[sudo] contraseña para alumno:
Reglas actualizadas
Reglas actualizadas (v6)
```

Y también las conexiones seguras:

**sudo ufw allow https**

```
alumno@alumnov:~/Escritorio$ sudo ufw allow https
Reglas actualizadas
Reglas actualizadas (v6)
```

## Configuración de PHP:

Para configurar php trabajaremos en primer lugar sobre el archivo de configuración del módulo del servicio web:

**sudo nano /etc/php/8.1/apache2/php.ini**

```
GNU nano 4.8 /etc/php/8.1/apache2/php.ini
[PHP]
;;;;;;;;;;;;;;;;;;;;;;;;
; About php.ini        ;
;;;;;;;;;;;;;;;;;;;;;;;;
```

Un ajuste que siempre conviene realizar es el de la zona horaria del servidor, para lo que buscamos la directiva “date.timezone”:

Buscamos el comando con Ctrl+W: **;date.timezone** y en este caso añadimos “Europe/Madrid” que es nuestra zona horaria.

```
;date.timezone = Europe/Madrid
```

Por otro lado, la configuración por defecto está orientada en cuanto al tratamiento de errores para un entorno de producción. Si estamos configurando un servidor para desarrollo entonces habría que hacer algunos cambios que nos faciliten la depuración de errores, para lo que buscaremos las siguientes directivas:

**display\_errors = Off**

**display\_startup\_errors = Off**

**error\_reporting = E\_ALL &~E\_DEPRECATED &~E\_STRICT**

Y las dejaremos así.

```
; display_errors
;   Default Value: On
;   Development Value: On
;   Production Value: On

; display_startup_errors
;   Default Value: On
;   Development Value: On
;   Production Value: On

; error_reporting
;   Default Value: E_ALL
;   Development Value: E_ALL
;   Production Value: E_ALL
```

De esta forma se mostrarán mensajes de error y aviso en las aplicaciones web que facilitarán el desarrollo, algo que no interesaría mostrar en un entorno de producción a los usuarios finales.

## Instalación LAMP

Una vez guardado con Ctrl+O y cerrado el archivo aplicamos los cambios recargando la configuración del servicio web:

**sudo systemctl reload apache2**

```
alumno@alumnov:~/Escritorio$ sudo systemctl reload apache2
```

Existe otro archivo de configuración para la ejecución de aplicaciones en consola:

**sudo nano /etc/php/8.1/cli/php.ini**

En el cual también cambiaremos el **date.timezone** como antes para que no haya discrepancias.

## Comprobación de MySQL

El servicio de bases de datos viene listo para usar sin necesidad de configuraciones adicionales. El único usuario que se incluye es el administrador root my para conectar al servicio desde consola habrá que usar sudo:

**sudo mysql**

```
alumno@alumnov:~/Escritorio$ sudo mysql
[sudo] contraseña para alumno:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 36
Server version: 10.3.34-MariaDB-0ubuntu0.20.04.1 Ubuntu 20.04

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> █
```

Para salir escribir **exit**.



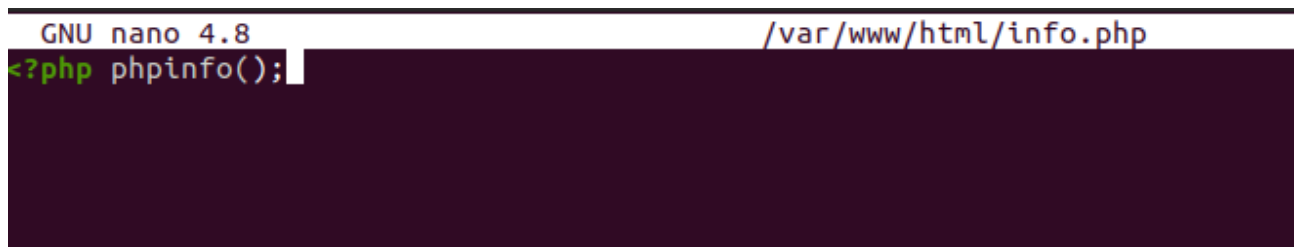
## Comprobación de Funcionamiento del Servidor

Para probar el funcionamiento de la pila LAMP bastará con crear un pequeño script en PHP de sencillo acceso vía web.

Creemos el archivo en un lugar que pueda ser accedido por el servicio web:

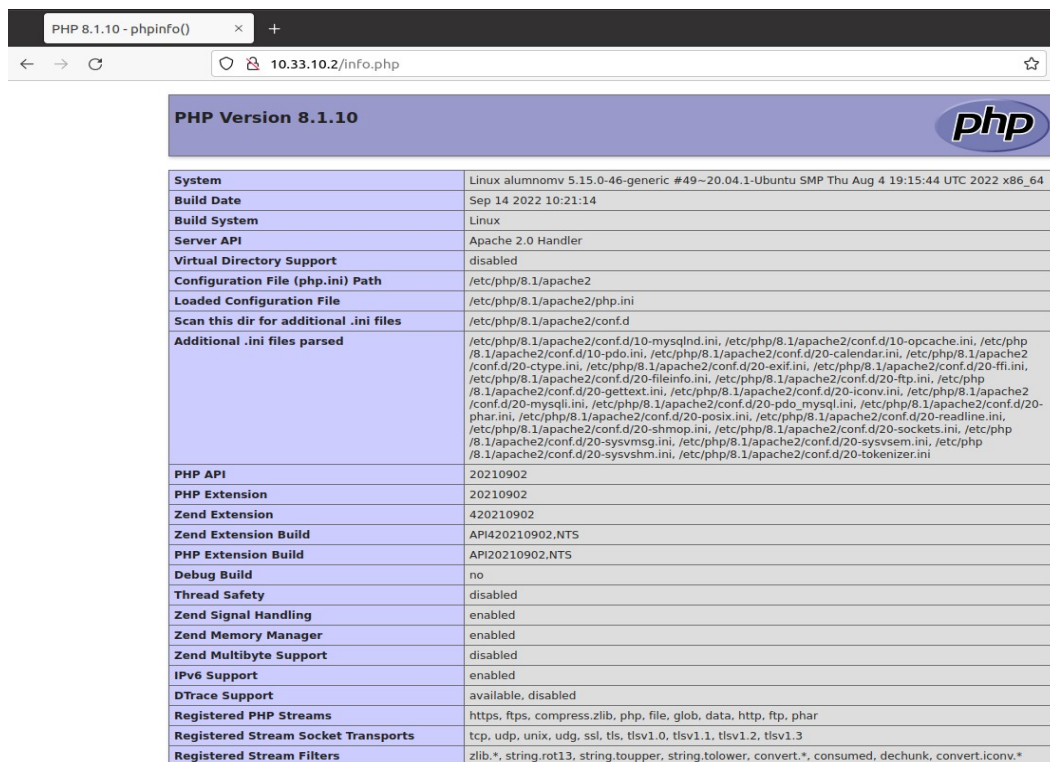
**sudo nano /var/www/html/info.php**

Y dentro el contenido: `<?php phpinfo();`



```
GNU nano 4.8 /var/www/html/info.php
<?php phpinfo();
```

Comprobaremos escribiendo en el navegador nuestra ip/dominio de la máquina virtual o real (En este caso 10.33.XX.2) + /info.php.



PHP Version 8.1.10	
<b>System</b>	Linux alumnomv 5.15.0-46-generic #49~20.04.1-Ubuntu SMP Thu Aug 4 19:15:44 UTC 2022 x86_64
<b>Build Date</b>	Sep 14 2022 10:21:14
<b>Build System</b>	Linux
<b>Server API</b>	Apache 2.0 Handler
<b>Virtual Directory Support</b>	disabled
<b>Configuration File (php.ini) Path</b>	/etc/php/8.1/apache2
<b>Loaded Configuration File</b>	/etc/php/8.1/apache2/php.ini
<b>Scan this dir for additional .ini files</b>	/etc/php/8.1/apache2/conf.d
<b>Additional .ini files parsed</b>	/etc/php/8.1/apache2/conf.d/10-mysqld.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
<b>PHP API</b>	20210902
<b>PHP Extension</b>	20210902
<b>Zend Extension</b>	420210902
<b>Zend Extension Build</b>	API420210902.NTS
<b>PHP Extension Build</b>	API20210902.NTS
<b>Debug Build</b>	no
<b>Thread Safety</b>	disabled
<b>Zend Signal Handling</b>	enabled
<b>Zend Memory Manager</b>	enabled
<b>Zend Multibyte Support</b>	disabled
<b>IPv6 Support</b>	enabled
<b>DTrace Support</b>	available, disabled
<b>Registered PHP Streams</b>	https, ftps, compress.zlib, php, file, glob, data, http, ftp, phar
<b>Registered Stream Socket Transports</b>	tcp, udp, unix, udg, ssl, tls, tlsv1.0, tlsv1.1, tlsv1.2, tlsv1.3
<b>Registered Stream Filters</b>	zlib.*, string.rot13, string.toupper, string.tolower, convert.*, consumed, dechunk, convert.iconv.*

## Conclusión

Ha sido una instalación sencilla de realizar, con unos pasos no muy complicados que nos servirá para montar un servidor LAMP.

## Bibliografía

<https://www.digitalocean.com/community/tutorials/how-to-install-the-apache-web-server-on-ubuntu-20-04-es>

[https://comoinstalar.me/como-instalar-lamp-en-ubuntu-20-04-lts/#Configurar\\_el\\_firewall\\_UFW\\_de\\_Ubuntu\\_2004\\_para\\_la\\_pila\\_LAMP](https://comoinstalar.me/como-instalar-lamp-en-ubuntu-20-04-lts/#Configurar_el_firewall_UFW_de_Ubuntu_2004_para_la_pila_LAMP)

<https://blog.infranetworking.com/servidor-lamp/>

<https://www.digitalocean.com/community/tutorials/how-to-install-linux-apache-mysql-php-lamp-stack-on-ubuntu-20-04-es#paso-1-instalar-apache-y-actualizar-el-firewall>