

# 1 GUÍA INSTALACIÓN SERVIDOR LAMP

## MATERIAL

- Máquina Virtual Ubuntu 22.04 Desktop.
- Virtualbox
- Ordenador con S.O. Windows 10.

### 1.1 Instalación Servidor Web - APACHE

1. Instalación del servidor web Apache desde la terminal de Ubuntu.

Para instalar el servidor Web – Apache, desde la terminal (ctrl+alt+T) lanzamos el siguiente comando:

**sudo apt-get install apache2**

Nos avisará del espacio necesario si deseamos continuar, a los que contestamos:

**S**

Se inicia el proceso de instalación.

2. Comprobación del funcionamiento del servidor Web desde la terminal y desde el navegador.

#### Comprobación desde Terminal

Para la comprobación de que el Servidor Web está activo tenemos dos opciones:

**sudo systemctl status apache2**

ó

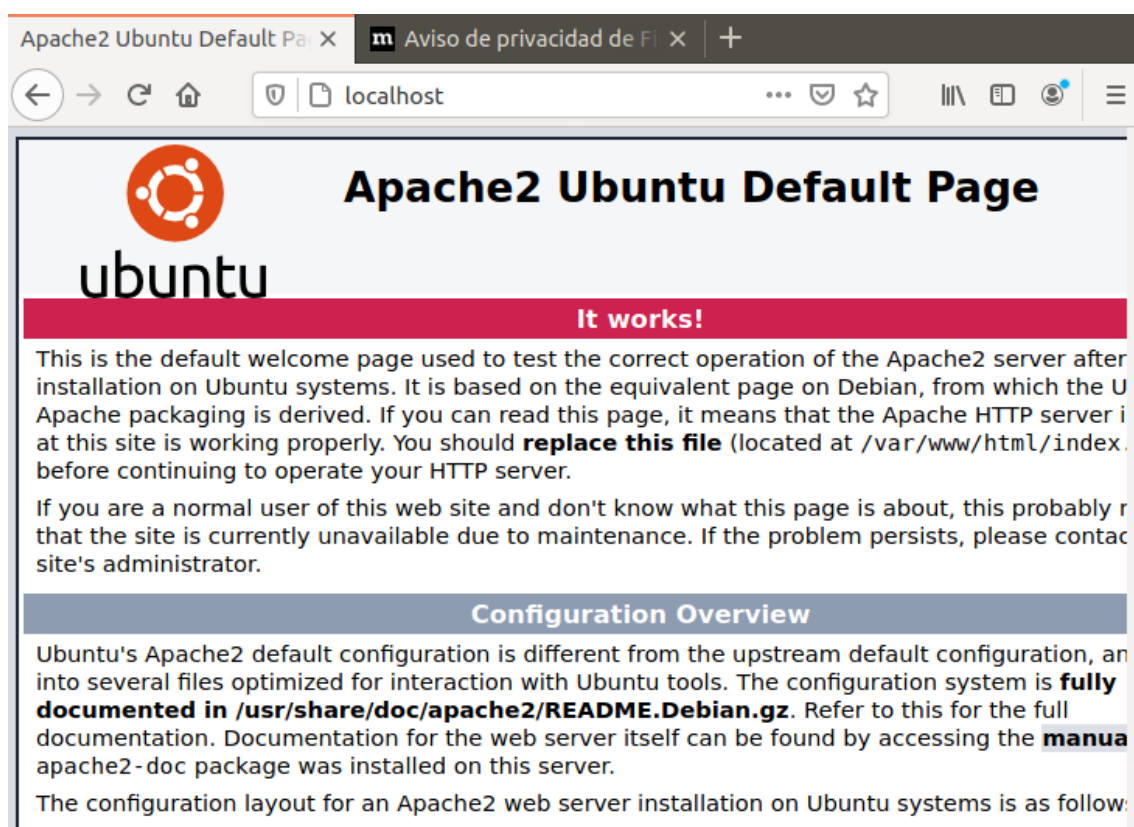
**/etc/init.d/apache2 status**

Deberemos ver para las dos opciones un Mensaje de status Activo con un círculo de color verde en el caso de que esté operativo.

```
informatica@informatica-VirtualBox:~$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset:
  Drop-In: /lib/systemd/system/apache2.service.d
           └─apache2-systemd.conf
   Active: active (running) since Mon 2020-09-28 18:27:51 WEST; 1min 19s ago
     Main PID: 3942 (apache2)
        Tasks: 55 (limit: 2329)
      CGroup: /system.slice/apache2.service
              └─3942 /usr/sbin/apache2 -k start
                └─3944 /usr/sbin/apache2 -k start
                  └─3945 /usr/sbin/apache2 -k start
```

### Comprobación desde el navegador

En este caso nos dirigimos al navegador del Ubuntu virtualizado y en la parte de la dirección URL escribimos la IP de la máquina o localhost o 127.0.0.1 y si está correcto debería mostrarnos la página index.html de Apache, también podemos probar desde el navegador del equipo principal y debería de dar el mismo resultado (Recuerda que tienes que poner en la URL en este caso la IP de la MV).



3. Cambiar el puerto por el que escucha el servidor Web al puerto 85.

Por defecto Apache y los servidores Web escuchan por el puerto 80. En este caso vamos a cambiar el puerto de escucha por defecto por el 85, para ello hacemos lo siguiente.

Editamos el fichero **ports.conf** :

**sudo gedit /etc/apache2/ports.conf**

```
# If you just change the port or add more ports here, you will likely also
# have to change the VirtualHost statement in
# /etc/apache2/sites-enabled/000-default.conf

Listen 80

<IfModule ssl_module>
    Listen 443
</IfModule>

<IfModule mod_gnutls.c>
    Listen 443
</IfModule>

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

Modificamos la línea de Listen 80, sustituyendo el 80 por en este caso el 85.

```
# If you just change the port or add more ports here, you will likely also
# have to change the VirtualHost statement in
# /etc/apache2/sites-enabled/000-default.conf

Listen 85|

<IfModule ssl_module>
    Listen 443
</IfModule>

<IfModule mod_gnutls.c>
    Listen 443
</IfModule>

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

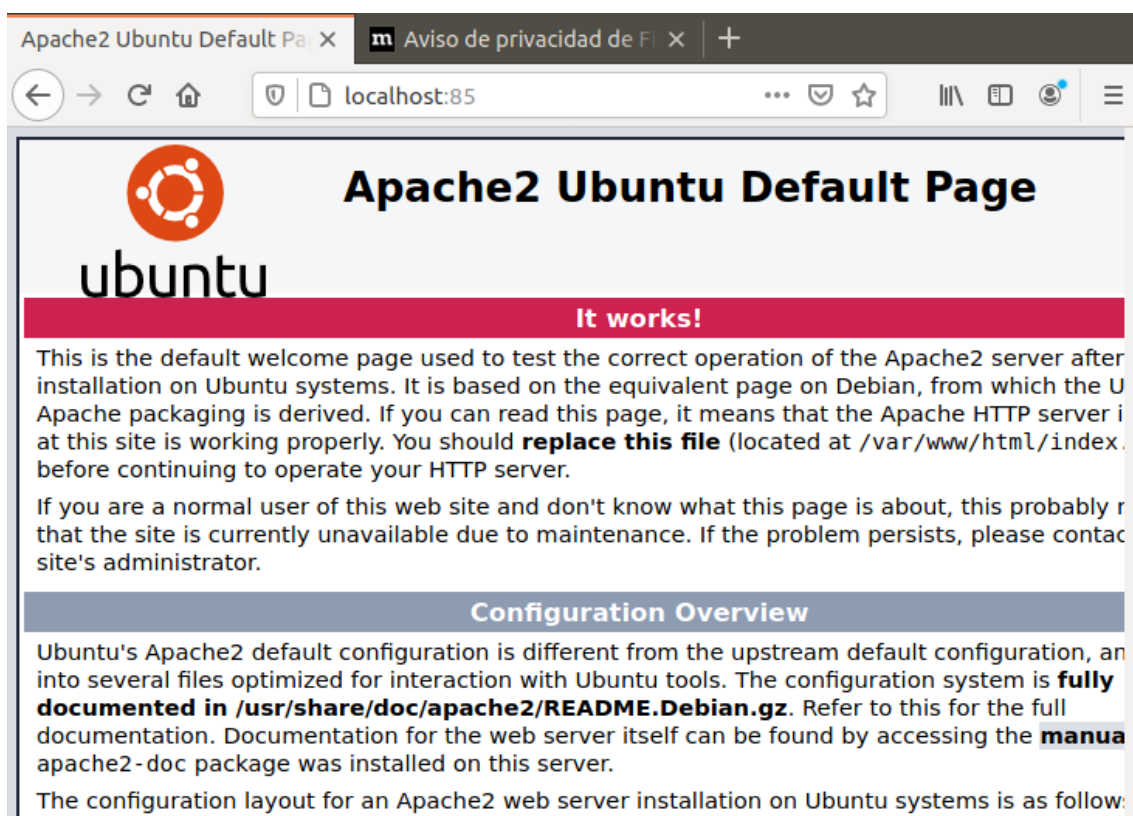
Guardamos el cambio en el gedit y reiniciamos el servicio MySQL.

**sudo systemctl restart apache2**

ó

**/etc/init.d/apache2 restart**

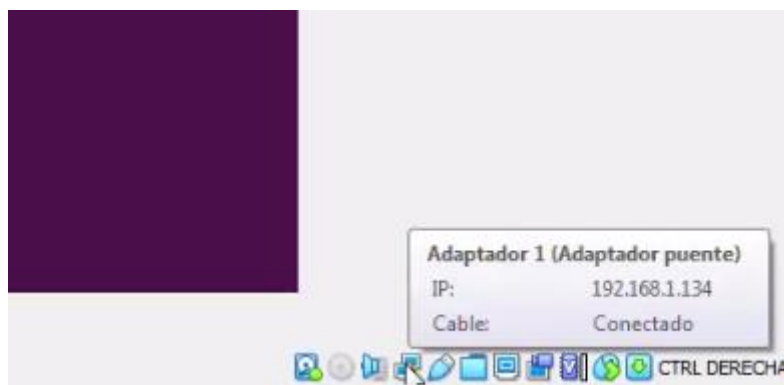
Si recargamos las páginas que accedimos anteriormente nos dará fallo, por que intentará ir al puerto por defecto, 80 y lo hemos cambiado al 85. Por lo que deberemos actualizar la URL para que vaya al referido puerto.



4. Comprobar acceso al servidor Web desde el navegador del equipo principal.

Para realizar esta comprobación debemos obtener la IP de la máquina virtual en la que hemos instalado Apache (Servidor Web). Disponemos de dos maneras de obtenerla:

1. Situando el ratón sobre los monitores que aparecen en la parte inferior derecha de la MV, donde se nos desplegará una ventana con la IP de la máquina.

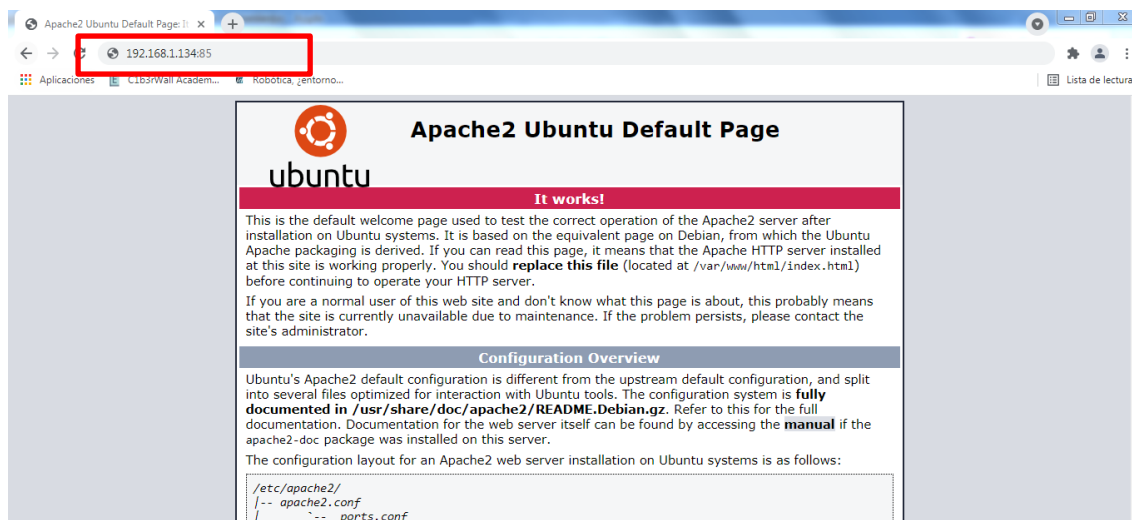


2. Accediendo a la terminal y usando el comando **ip a**.

```

informatica@informatica-VirtualBox: ~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:f8:11:2d brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.134/24 brd 192.168.1.255 scope global dynamic noprefixroute enp0s3
        valid_lft 83975sec preferred_lft 83975sec
    inet6 fe80::a389:db76:a991:8c39/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
3: ham0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1404 qdisc fq_codel state UNKNOWN group default qlen 1000
    link/ether 7a:79:19:26:74:60 brd ff:ff:ff:ff:ff:ff
    inet 25.33.155.42/8 brd 25.255.255.255 scope global ham0
        valid_lft forever preferred_lft forever
    inet6 2620:9b::1921:9b2a/96 scope global
        valid_lft forever preferred_lft forever
    inet6 fe80::7879:19ff:fe26:7460/64 scope link
        valid_lft forever preferred_lft forever
informatica@informatica-VirtualBox: ~$
  
```

Nos anotamos la IP obtenida (en este caso 192.168.1.134) y para acceder a la página del servidor web debemos de poner la en la barra de direcciones del navegador del equipo principal. Por defecto las páginas web están en el puerto 80, en el caso de que hayamos cambiado el puerto donde responde nuestro servidor web debemos indicarlo en la barra de direcciones de la siguiente forma **IP\_servidor\_web:num\_puerto**.



Para controlar el tráfico entrante y saliente en una máquina, con el fin de evitar accesos no autorizados o robo de datos se emplean los cortafuegos, estas reglas se aplican sobre los puertos que no son más que las puertas de entrada y salida para interconectar las las aplicaciones/servicios (software) de la máquina con el mundo exterior (Internet).

En Ubuntu el firewall o cortafuegos viene por defecto deshabilitado, si lo queremos habilitar lanzamos el siguiente comando desde el terminal:

**sudo ufw enable**

Para deshabilitarlo nuevamente usamos el comando:

**sudo ufw disable**

En la configuración del mismo se puede impedir el acceso a un determinado puerto o varios puertos.

## 1.2 Instalación SGBD- MySQL

### 1. Instalación de MySQL.

Para instalar MySQL ejecutamos desde terminal (ctrl+alt+T) la siguiente orden:

**sudo apt-get install mysql-server**

Esto instalará MySQL, pero no le pedirá que cree una contraseña ni que haga ningún otro cambio de configuración. Dado a que esto deja su instalación de MySQL insegura, vamos a abordarlo a continuación.

Nos saldrá un mensaje como con Apache informando del espacio que ocupará la instalación a lo que contestaremos que **S**.

Para verificar que está funcionando podemos comprobar el estado del servicio.

### **sudo systemctl status mysql**

Si está operativo nos deberá salir en pantalla el círculo del servicio MySQL en verde como que está activo.

```
● mysql.service - MySQL Community Server
   Loaded: loaded (/lib/systemd/system/mysql.service; enabled; vendor preset: e
   Active: active (running) since Mon 2020-09-28 18:46:34 WEST; 3min 38s ago
     Main PID: 6079 (mysqld)
        Tasks: 27 (limit: 2329)
      CGroup: /system.slice/mysql.service
             └─6079 /usr/sbin/mysqld --daemonize --pid-file=/run/mysqld/mysqld.pi
```

## 2. Configurar MySQL

Para las instalaciones recientes, habrá que ejecutar el script de seguridad que viene incluido. Esto cambia algunas de las opciones predeterminadas menos seguras para cosas como inicios de sesión root remotos y usuarios de ejemplo. Para las versiones antiguas de MySQL, también deberá inicializar el directorio de datos manualmente, pero ahora esto se hace automáticamente.

Ejecute el script de seguridad:

### **sudo mysql\_secure\_installation**

Esto hará que pase por una serie de indicaciones en las que puede hacer algunos cambios en las opciones de seguridad de su instalación de MySQL. La primera indicación le preguntará si quiere configurar el plugin de Validación de Contraseña, la que puede usarse para probar la solidez de su contraseña de MySQL. Independientemente de lo que seleccione, la siguiente indicación será establecer una contraseña para el usuario **root** de MySQL. Ingrese y luego confirme una contraseña segura de su elección.



Desde este punto, puede presionar **Y** y luego **ENTER** para aceptar las configuraciones predeterminadas para todas las preguntas siguientes. Esto eliminará algunos usuarios anónimos y la base de datos de prueba, deshabilitará los inicios de sesión root remotos y cargará estas nuevas reglas para que MySQL respete los cambios que haya realizado inmediatamente. Para iniciar el directorio de datos de MySQL, debe usar `mysql_install_db` para versiones anteriores a 5.7.6, y `mysqld --initialize` para la versión 5.7.6 y posteriores. No obstante, si instaló MySQL desde la distribución Debian, como se describió en la parte de instalación, el directorio de datos se inició automáticamente; no es necesario que haga nada. Si trata de ejecutar el comando igual, se le mostrará el siguiente error:

Output
<pre>mysqld: Can't create directory '/var/lib/mysql/' (Errcode: 17 - File exists) . . . 2018-04-23T13:48:00.572066Z 0 [ERROR] Aborting</pre>

Note que, si bien estableció una contraseña para el usuario **root** de MySQL, este usuario no está configurado para autenticarse con una contraseña al conectarse al shell de MySQL. Si quiere, puede ajustar esta configuración siguiendo las siguientes instrucciones.

### Ajustar la autenticación y los privilegios del usuario

Para los sistemas Ubuntu que estén usando MySQL 5.7 (y las versiones posteriores), el usuario **root** de MySQL está configurado, de forma predeterminada, para autenticarse usando el plugin **auth\_socket** en vez de una contraseña. En muchos casos, esto permite que la seguridad y usabilidad sea mayor pero también puede complicar las cosas cuando deba permitir que un programa externo (tal como phpMyAdmin) tenga acceso al usuario.

Deberá cambiar su método de autenticación de **auth\_socket** a **mysql\_native\_password** para usar una contraseña para conectarse a MySQL como **root**. Para hacerlo, abra la indicación de MySQL desde su terminal (ctrl+alt+T):

**sudo mysql -u root -p**

Posteriormente, consulte cuál método de autenticación usa cada una de sus cuentas de usuario de MySQL usando el siguiente comando:

**mysql> SELECT user,authentication\_string,plugin,host FROM mysql.user;**



```

+-----+-----+-----+-----+
| user | authentication_string | plugin | host |
+-----+-----+-----+-----+
| root | *THISISNOTAVALIDPASSWORDTHATCANBEUSEDHERE | auth_socket | localhost |
| mysql.session | *THISISNOTAVALIDPASSWORDTHATCANBEUSEDHERE | mysql_native_password | localhost |
| mysql.sys | *THISISNOTAVALIDPASSWORDTHATCANBEUSEDHERE | mysql_native_password | localhost |
| debian-sys-maint | *CC744277A401A7D25BE1CA89AFF17BF607F876FF | mysql_native_password | localhost |
+-----+-----+-----+-----+
4 rows in set (0.00 sec)

```

En este ejemplo, puede ver que el usuario **root** verdaderamente se autentica usando el **plugin auth\_socket**. Para configurar la cuenta **root** para autenticarse usando una contraseña, ejecute el siguiente comando **ALTER USER**. Asegúrese de cambiar **password** (contraseña) a una contraseña segura de su elección y sepa que este comando cambiará la contraseña de **root** que estableció en el Paso 2:

**SOLUCIÓN PROBLEMA POLÍTICAS DE SEGURIDAD:**

```

mysql>SHOW VARIABLES LIKE 'validate_password%';
mysql>SET GLOBAL validate_password_length=4;
mysql>SET GLOBAL validate_password_number_count=0;
mysql>SET GLOBAL validate_password_policy=LOW;

```

Con el primer comando mostramos las reglas de validación de la contraseña. Mientras con los SET GLOBAL fijamos los parámetros, para el primero la longitud de la contraseña, con el segundo el número mínimo de dígitos y con el policy el nivel de seguridad.

```

mysql>ALTER USER 'root'@'localhost' IDENTIFIED WITH
mysql_native_password BY 'password';

```

Luego, ejecute **FLUSH PRIVILEGES** (purgar privilegios), lo que le dice al servidor que vuelva a cargar las tablas grant e implemente sus nuevos cambios:

```

mysql>FLUSH PRIVILEGES;

```

Vuelva a verificar los métodos de autenticación que usa cada uno de sus usuarios para confirmar que **root** ya no se autentica usando el **auth\_socket**:

**Mysql> SELECT user,authentication\_string,plugin,host FROM mysql.user;**

```

+-----+-----+-----+-----+
| user | authentication_string | plugin | host |
+-----+-----+-----+-----+
| root | *3636DACC8616D997782ADD0839F92C1571D6D78F | mysql_native_password | localhost |
| mysql.session | *THISISNOTAVALIDPASSWORDTHATCANBEUSEDHERE | mysql_native_password | localhost |
| mysql.sys | *THISISNOTAVALIDPASSWORDTHATCANBEUSEDHERE | mysql_native_password | localhost |
| debian-sys-maint | *CC744277A401A7D25BE1CA89AFF17BF607F876FF | mysql_native_password | localhost |
+-----+-----+-----+-----+
4 rows in set (0.00 sec)

```

En este resultado de ejemplo, puede ver que, ahora, el usuario **root** de MySQL se autentica usando una contraseña. Una vez que confirme esto en su propio servidor, puede salir del shell de MySQL:

```
mysql> exit
```

### Crear un usuario

Alternativamente, para otras personas puede adaptarse mejor a su flujo de trabajo si se conectan a MySQL con un usuario dedicado. Para crear tal usuario, vuelva a abrir el shell de MySQL nuevamente:

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

Desde ese punto, cree un nuevo usuario y use una contraseña sólida:

```
mysql> CREATE USER 'nombre_usuario'@'localhost' IDENTIFIED BY 'password_usuario';
```

Luego, dele a su nuevo usuario los privilegios adecuados. Por ejemplo, puede concederle al usuario privilegios a todas las tablas dentro de la base de datos, así como autoridad para agregar, cambiar y eliminar privilegios de usuario, mediante este comando:

```
mysql> GRANT ALL PRIVILEGES ON *.* TO 'nombre_usuario'@'localhost' WITH GRANT OPTION;
```

Note que, en este punto, no necesita volver a ejecutar el comando **FLUSH PRIVILEGES**. Solamente necesita este comando al modificar las tablas grant utilizando declaraciones como **INSERT**, **UPDATE** o **DELETE**. Dado a que creó un usuario nuevo en vez de modificar uno existente, no es necesario que use **FLUSH PRIVILEGES** aquí.

Después de esto, salga del Shell de MySQL:

```
mysql> exit
```

### Probar MySQL

Independientemente de cómo lo instaló, MySQL debería haber empezado a ejecutarse automáticamente. Para probar esto, verifique su estado.

```
systemctl status mysql.service
```

Verá un resultado parecido al de abajo:

```

Output

● mysql.service - MySQL Community Server
   Loaded: loaded (/lib/systemd/system/mysql.service; enabled; vendor preset: en
   Active: active (running) since Wed 2018-04-23 21:21:25 UTC; 30min ago
   Main PID: 3754 (mysqld)
     Tasks: 28
    Memory: 142.3M
       CPU: 1.994s
   CGroup: /system.slice/mysql.service
           └─3754 /usr/sbin/mysqld
  
```

Si MySQL no se está ejecutando, puede iniciarlo usando **sudo systemctl start mysql**.

Para una verificación adicional, puede tratar de conectarse a la base de datos usando la herramienta **mysqladmin**, que es un cliente que le permite ejecutar comandos administrativos. Por ejemplo, este comando dice que se conecte a MySQL como **root** (-u root), pida una contraseña (-p) y devuelva la versión.

**sudo mysqladmin -p -u root version**

Debería ver un resultado parecido a este:

```

Output

mysqladmin Ver 8.42 Distrib 5.7.21, for Linux on x86_64
Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

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

Server version      5.7.21-1ubuntu1
Protocol version    10
Connection          Localhost via UNIX socket
UNIX socket         /var/run/mysqld/mysqld.sock
Uptime:             30 min 54 sec

Threads: 1  Questions: 12  Slow queries: 0  Opens: 115  Flush tables: 1  Open tabl
  
```

Esto quiere decir que MySQL está funcionando.

## 1.3 Instalación PHP

### 1. Instalación PHP.

Para instalar PHP ejecutamos la siguiente instrucción:

### **sudo apt-get -y install php**

Donde el parámetro -y ya le decimos que acepte todas las preguntas que nos haga durante el proceso de instalación.

2. Comprobar que está instalado PHP desde terminal y hacer una prueba de funcionamiento.

Al finalizar la descarga e instalación, PHP y el módulo para el servidor web quedan perfectamente integrados y listos para usar.

Cómo saber la versión de PHP instalada:

Si quieres saber cuál es la versión de PHP que se acaba de instalar en tu sistema Ubuntu 18.04 puedes usar el comando php en consola:

**php -v**

```
~$ php -v
PHP 7.2.24-0ubuntu0.18.04.1 (cli) (built: Oct 28 2019 12:07:07) ( NTS )
Copyright (c) 1997-2018 The PHP Group
Zend Engine v3.2.0, Copyright (c) 1998-2018 Zend Technologies
    with Zend OPcache v7.2.24-0ubuntu0.18.04.1, Copyright (c) 1999-2018, by Zend Technologies
```

Para comprobar el funcionamiento de PHP directamente en consola crearemos un pequeño script con nuestro editor de texto favorito:

### **nano test.php**

Su contenido podría ser el siguiente:

```
<?php echo "Hola, mundo\n\n";
```

Para ejecutar este script en consola usaremos el comando php y observaremos la salida que produce:

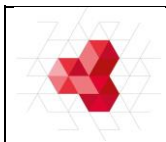
```
~$ php test.php
Hola, mundo

~$
```

Podemos ver que PHP funciona correctamente.

3. Comprobar que está integrado PHP con Apache (Servidor Web).

Tras instalar PHP en Ubuntu 18.04 y realizar, si es necesario, la integración con el servidor web, podemos comprobar que todo funciona creando una sencilla página web en PHP ubicada en la raíz de documentos de la página web por defecto del servidor:



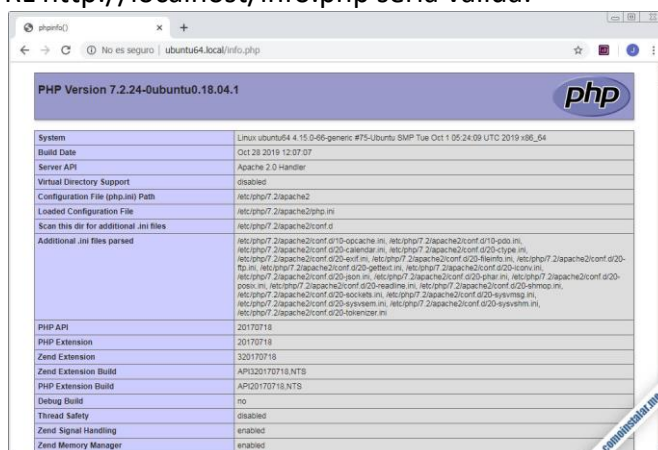
**sudo nano /var/www/html/info.php**

El contenido será la función `phpinfo()`:

```
<?php phpinfo();
```

Ahora, a través de un navegador introduciremos como URL la dirección IP, el nombre de máquina o su dominio si lo tiene, añadiendo al final la ruta `/info.php`.

En este ejemplo, el servidor es accesible mediante el dominio `localhost`, así que la URL `http://localhost/info.php` sería válida:



#### 4. Configurar PHP y aplicar cambios.

Aunque gran parte de la configuración ya viene hecha, para configurar PHP en Ubuntu 18.04 LTS buscaremos los archivos de configuración bajo el directorio `/etc/php/7.x/` (donde `x` depende la versión de PHP instalada en nuestro sistema Ubuntu 18.04).

El archivo de configuración de PHP para el módulo de Apache es `/etc/php/7.x/apache2/php.ini`:

**sudo nano /etc/php/7.2/apache2/php.ini**

Vamos a ver algunos ajustes iniciales que podríamos hacer en `php.ini`.

##### Zona horaria

Editando la configuración de PHP en Ubuntu 18.04 deberíamos configurar la zona horaria del servidor, buscando la cláusula `date.timezone`:

```
...
[Date]
; Defines the default timezone used by the date functions
; http://php.net/date.timezone
;date.timezone =
...
```

Para conocer qué valores admite `date.timezone` podemos seguir el enlace <https://www.php.net/manual/es/timezones.php>. Una vez conocido el valor que necesitamos, descomentamos `date.timezone` y añadimos ese valor:

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

### Tratamiento de mensajes de error

Es importante tener en cuenta el comportamiento de PHP ante los errores. Por defecto PHP viene con valores para entornos de producción, por lo que por motivos de seguridad no se informa sobre los errores que se produzcan en los scripts PHP. Podemos comprobarlo buceando en php.ini:

```
...
; Default Value: E_ALL & ~E_NOTICE & ~E_STRICT & ~E_DEPRECATED
; Development Value: E_ALL
; Production Value: E_ALL & ~E_DEPRECATED & ~E_STRICT
; http://php.net/error-reporting
error_reporting = E_ALL & ~E_DEPRECATED & ~E_STRICT
...
; Default Value: On
; Development Value: On
; Production Value: Off
; http://php.net/display-errors
display_errors = Off
...
; Default Value: Off
; Development Value: On
; Production Value: Off
; http://php.net/display-startup-errors
display_startup_errors = Off
...
```

- Si el servidor Ubuntu 18.04 en el que estamos trabajando es para desarrollo, prácticas, etc. entonces puede ser más conveniente asignar valores de desarrollo, tal y como indica la ayuda en los comentarios:

```
...
error_reporting = E_ALL
...
display_errors = On
...
display_startup_errors = On
...
```

### Aplicar los cambios

Todos estos cambios y cualquier otro que necesites realizar en tu configuración no entrarán en vigor hasta la próxima vez que se cargue la nueva configuración.

En el caso de Apache recargamos la configuración del servidor web:

**sudo systemctl reload apache2**