

Instalación y Configuración de un Servidor de Aplicaciones Web en Windows y Linux

Eloy Celdrán Madrid

Contenido

Introducción	1
WAMP	1
LAMP	1
Instalación WAMP.....	2
Instalación de LAMP.....	6
Comparación y reflexión final.....	11
Diferencias entre la instalación en Windows y Linux.....	11
Sistema más cómodo para la gestión.....	11
Ajustes para producción segura	11

Introducción

WAMP

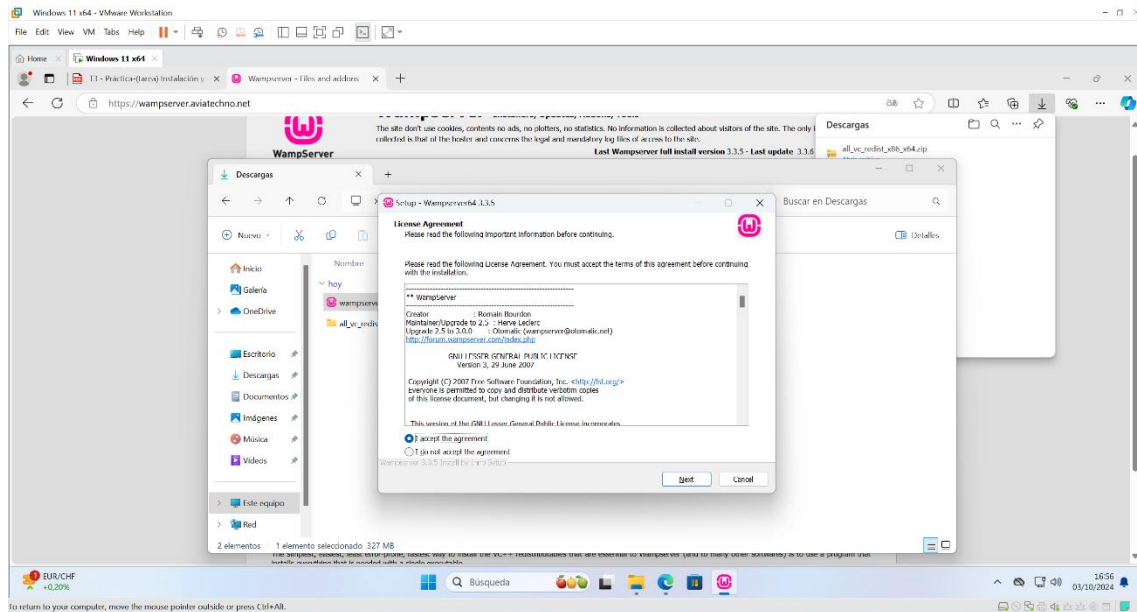
Me resulta especialmente atractivo por su facilidad de uso en entornos Windows. Su configuración es sencilla y me permite tener un entorno de desarrollo local rápidamente. Considero que otras opciones multiplataforma como XAMPP y AMPPS son muy populares considero que esta opción me da un mayor control y personalización

LAMP

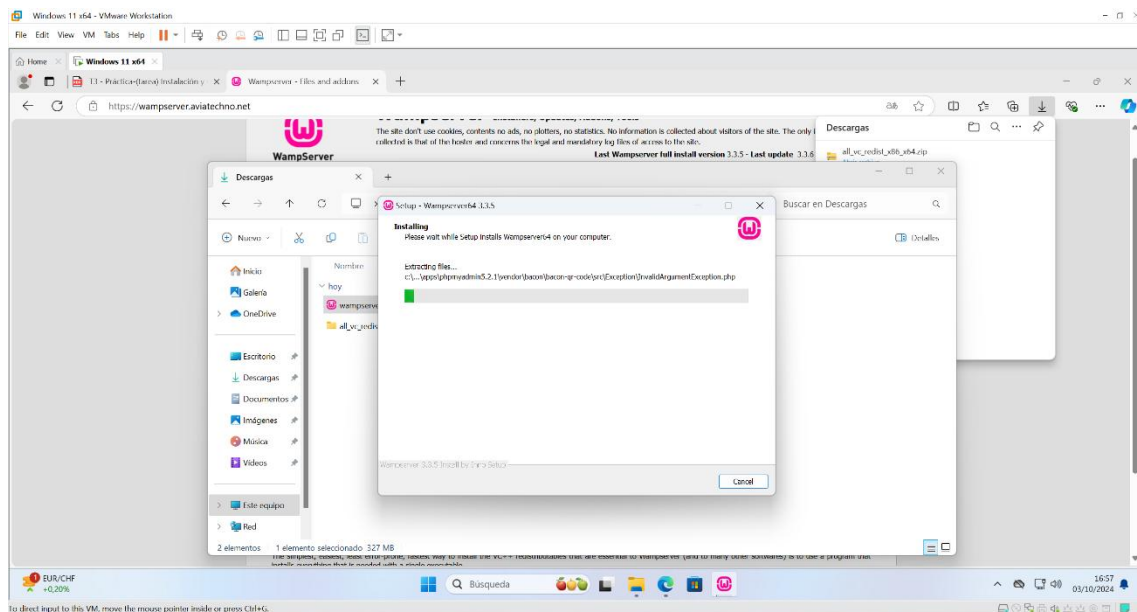
Mayor personalización y control, simplemente, lo tienes que instalar todo uno a uno, cosa que no es difícil para mí, es más si lo instalas todo parte por parte eres consciente de lo que estas instalando y no le estas dando siguiente y siguiente, también me pareció el más escalable de todos.

Instalación WAMP

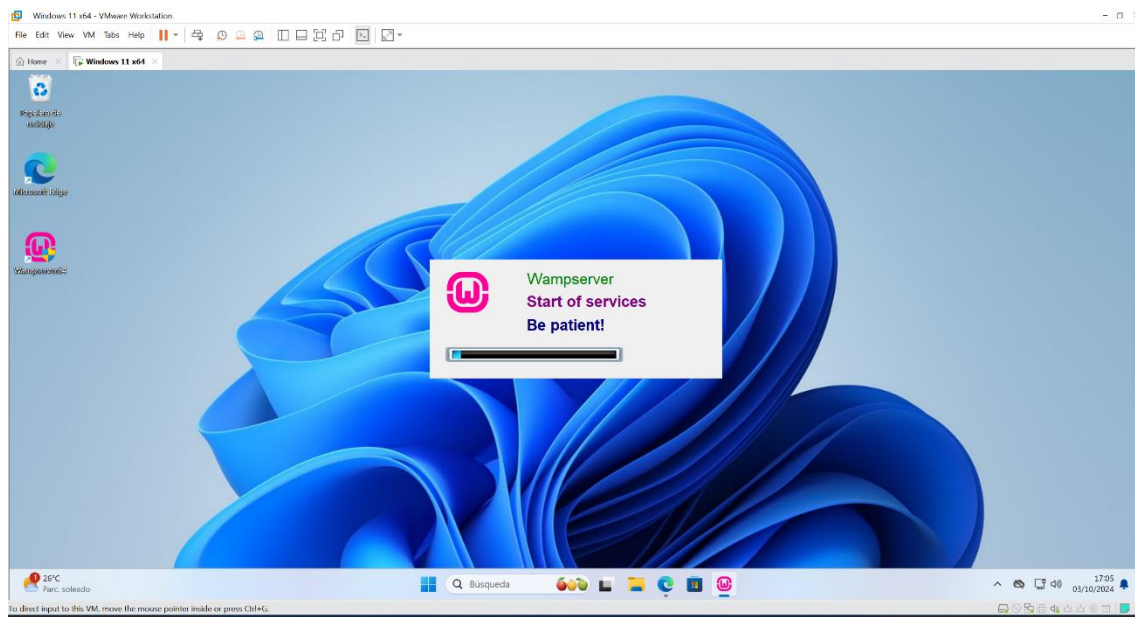
Deberemos irnos a su pagina web como hacemos habitualmente para instalarnos cualquier aplicación y ejecutamos el .exe, una vez ejecutado aceptamos las condiciones.



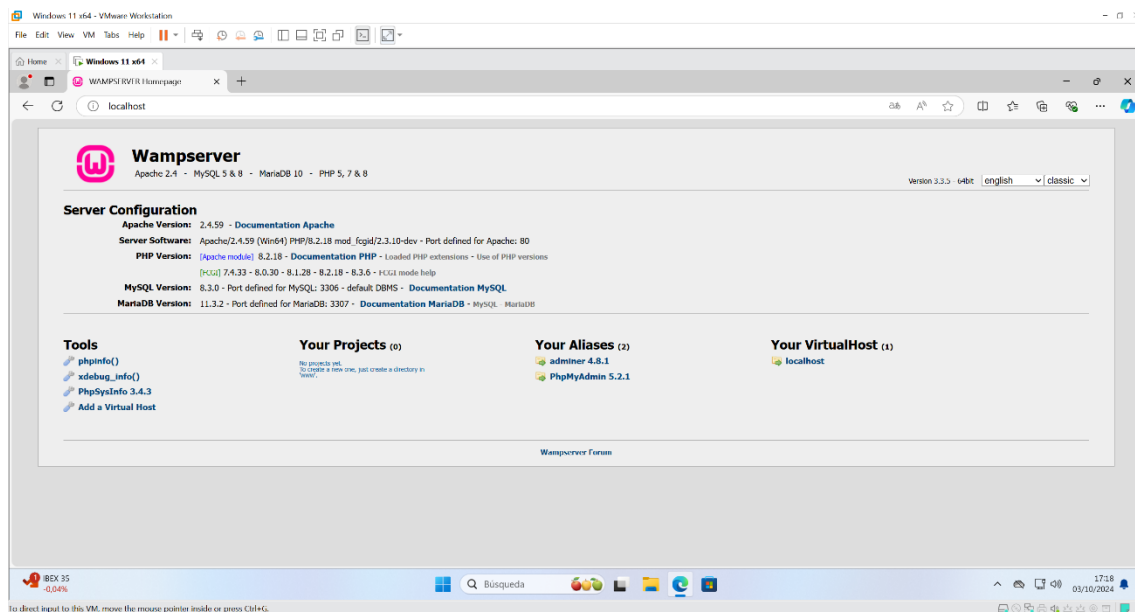
Y comenzará la instalación.



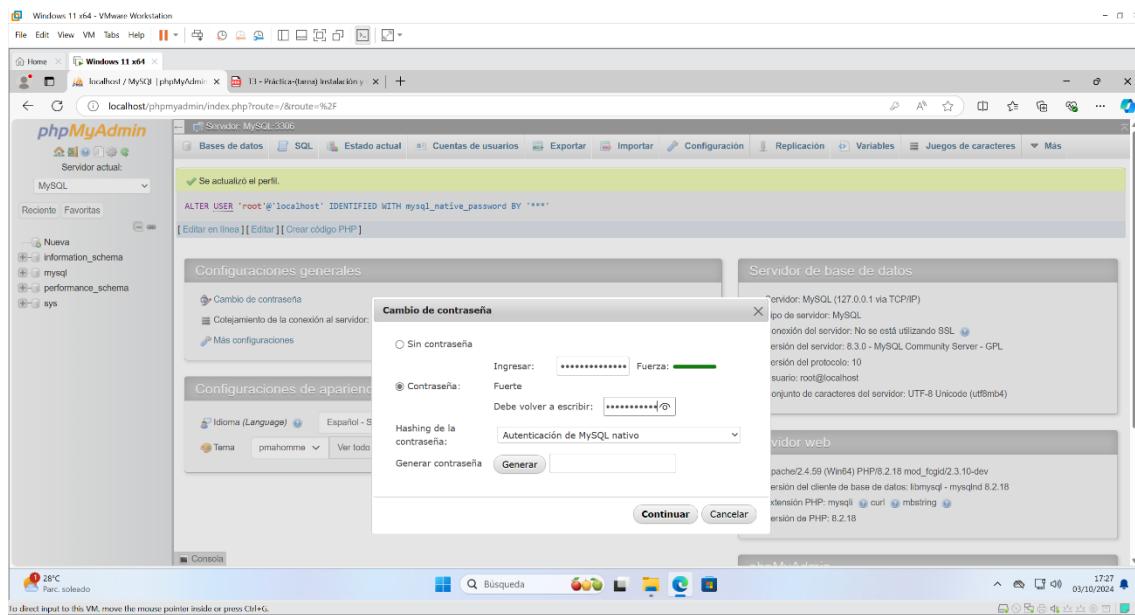
Una vez instalado nos mostrará un mensaje de que los servicios se están iniciando



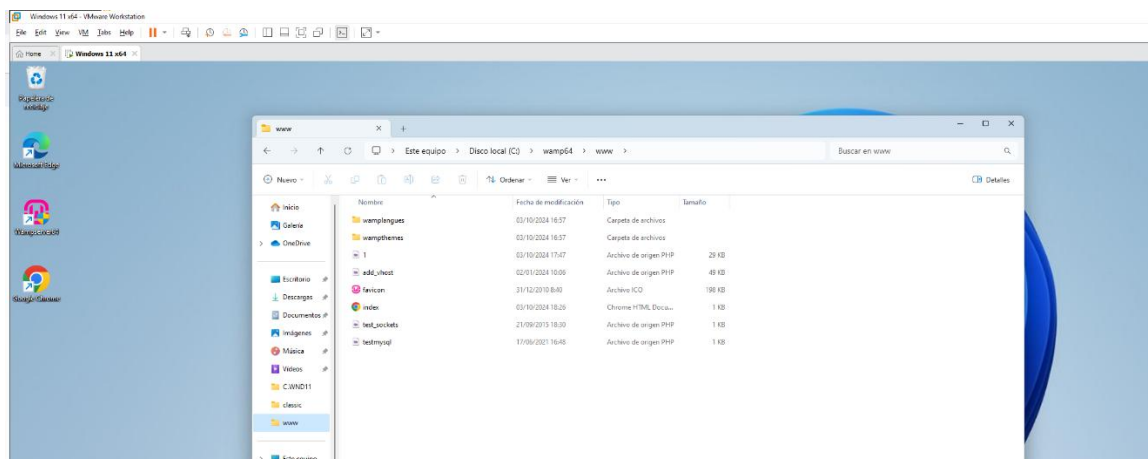
Nos vamos a localhost y con esta pantalla principal podemos comprobar que se ha instalado correctamente.



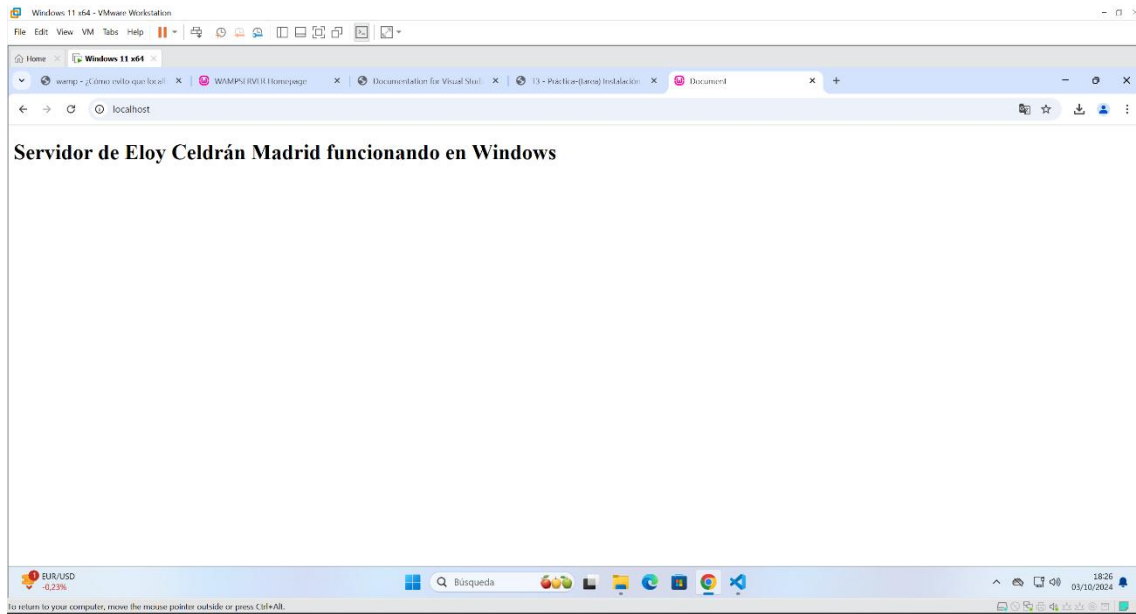
Cambiamos la contraseña de nuestro admin.



Ahora deberemos cambiar la pagina principal que nos muestra cuando buscamos localhost, deberemos movernos a la carpeta de instalación de wamp una vez dentro a “www”, hay un archivo que se llama index.php, esto es la pantalla que nos sale actualmente en localhost, deberemos cambiarle el nombre para que no sea la pantalla “predefinida” que muestre, y crearemos otro archivo que se llamara “index.html” que será nuestra predefinida, ya que cuando busquemos localhost el servicio buscará un archivo llamado “index” para mostrarlo como pantalla principal.



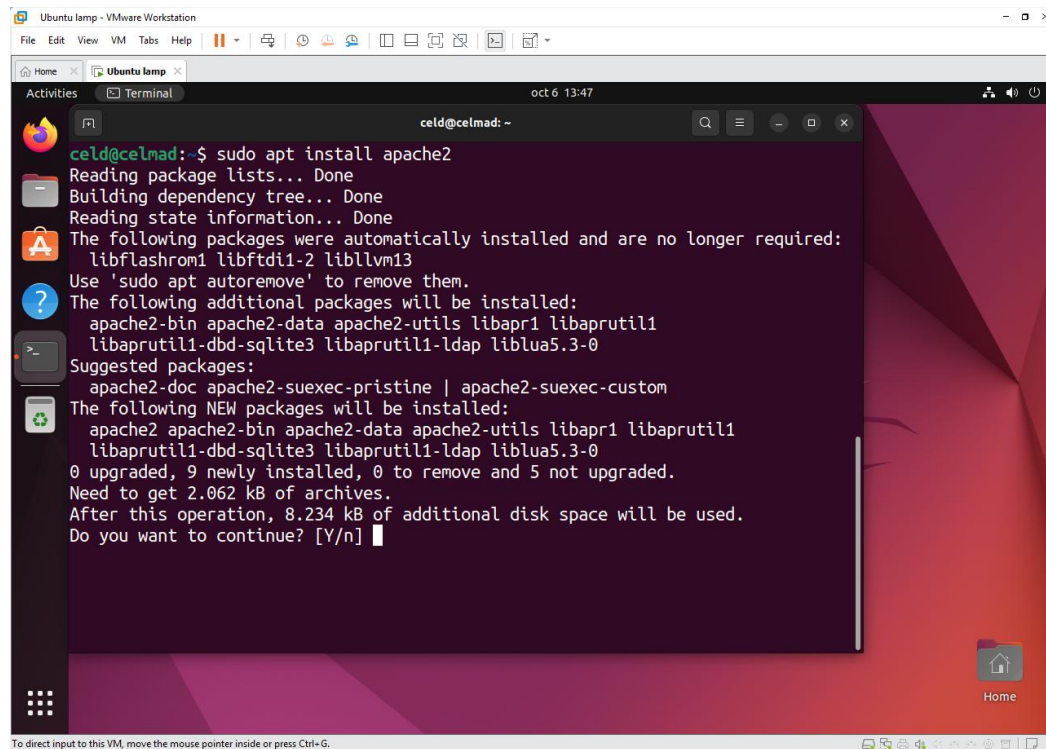
Con nuestro nuevo archivo creado y el otro renombrado podemos comprobar que ya ha cambiado nuestra pantalla principal de localhost.



Instalación de LAMP

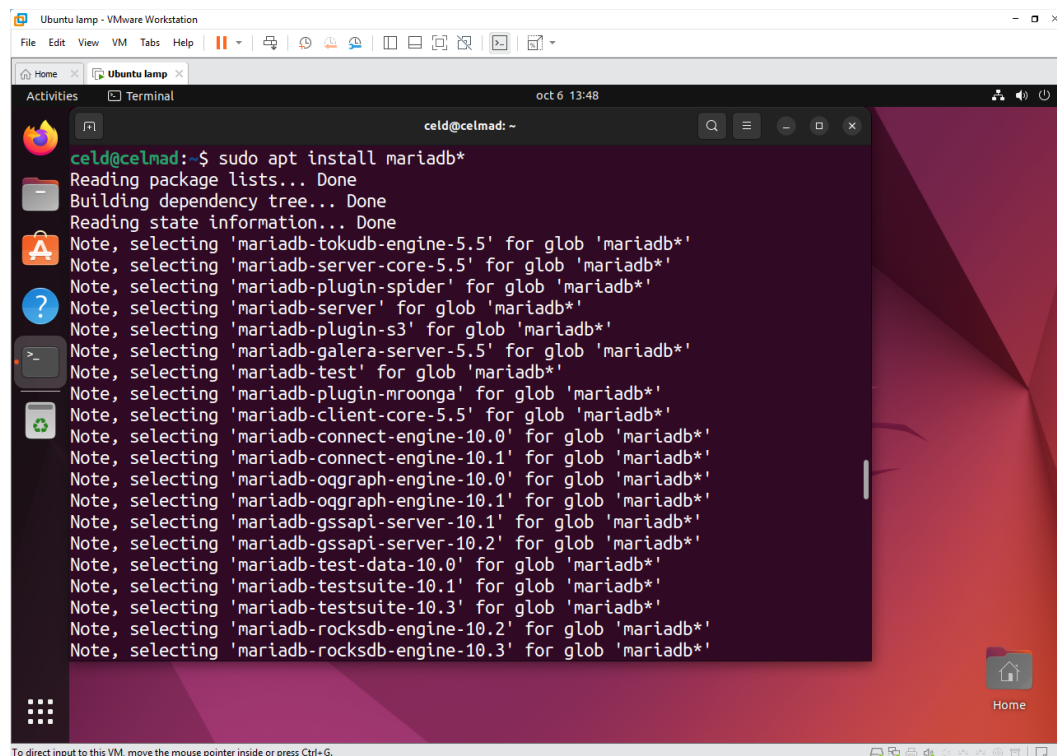
LAMP es Apache, MySQL y PHP por lo que empezaremos en orden.

Primero instalaremos apache, con el comando “sudo apt install apache2”



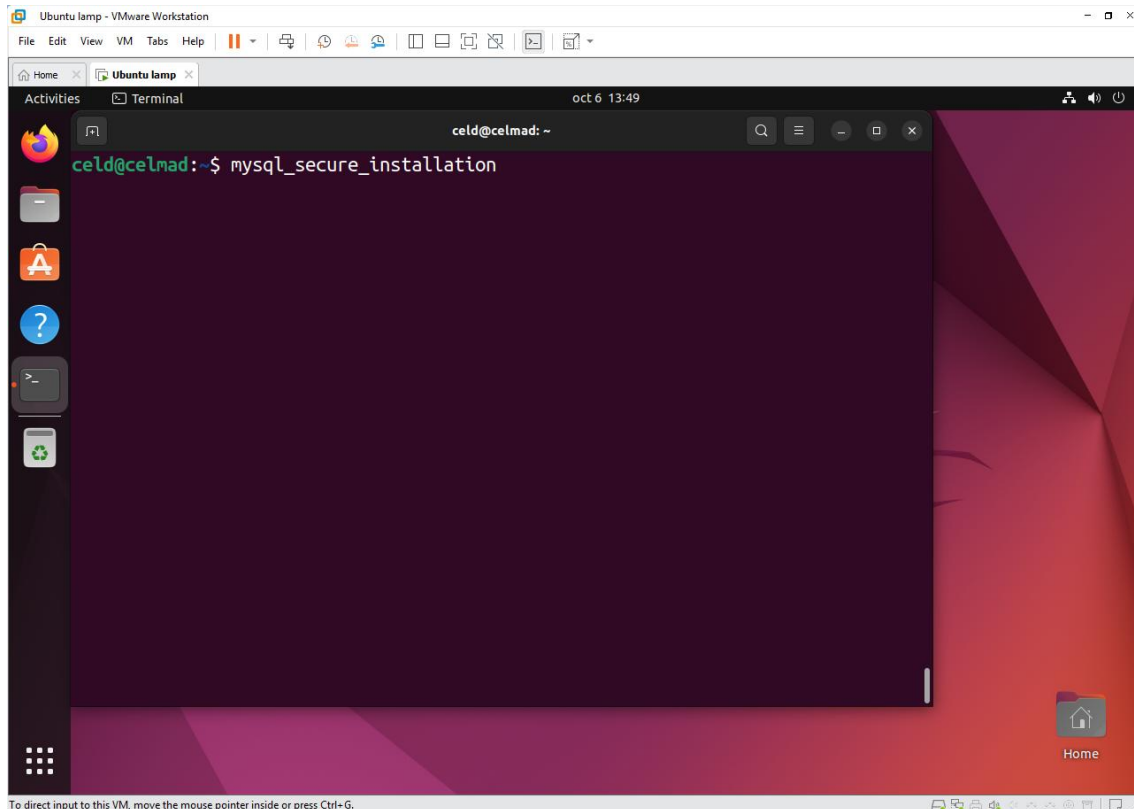
```
cel@celmad:~$ sudo apt install apache2
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libflashrom1 libftdi1-2 liblvm13
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils libapr1 libaprutil1
  libaprutil1-dbd-sqlite3 libaprutil1-ldap liblua5.3-0
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils libapr1 libaprutil1
  libaprutil1-dbd-sqlite3 libaprutil1-ldap liblua5.3-0
0 upgraded, 9 newly installed, 0 to remove and 5 not upgraded.
Need to get 2.062 kB of archives.
After this operation, 8.234 kB of additional disk space will be used.
Do you want to continue? [Y/n]
```

Una vez instalado, seguiremos en orden e instalaremos mariadb con el comando “sudo apt install mariadb*”

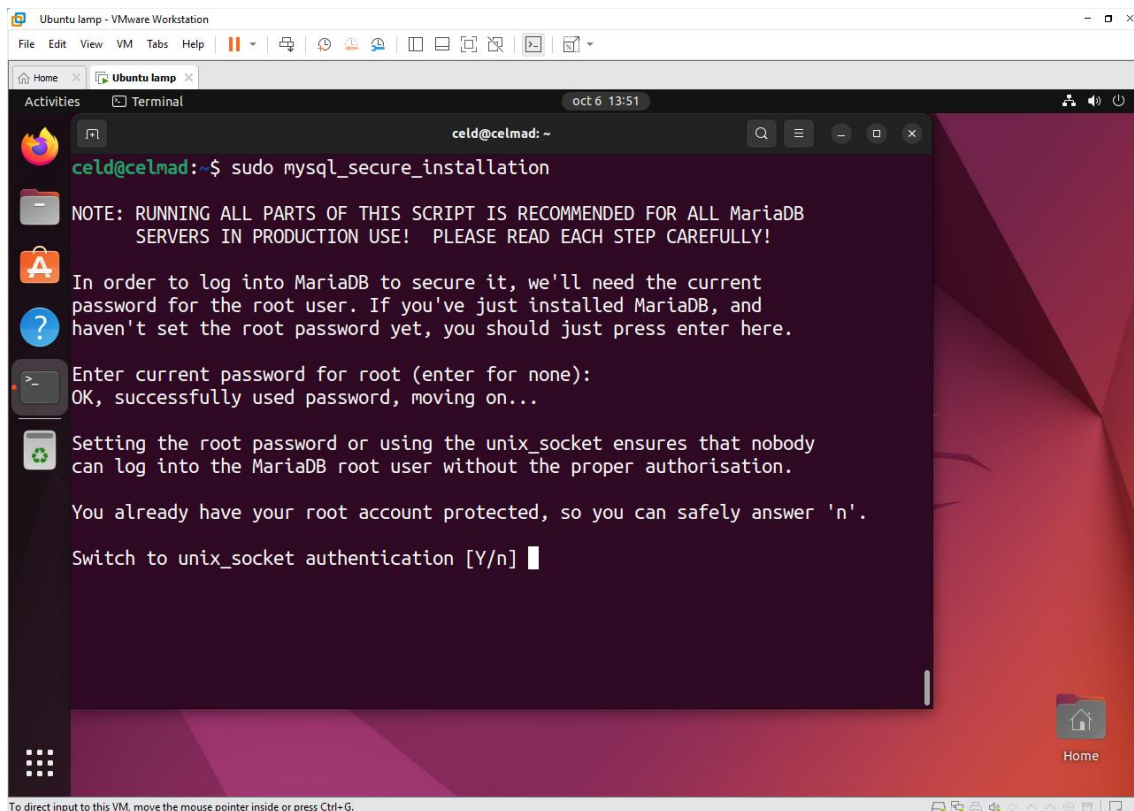


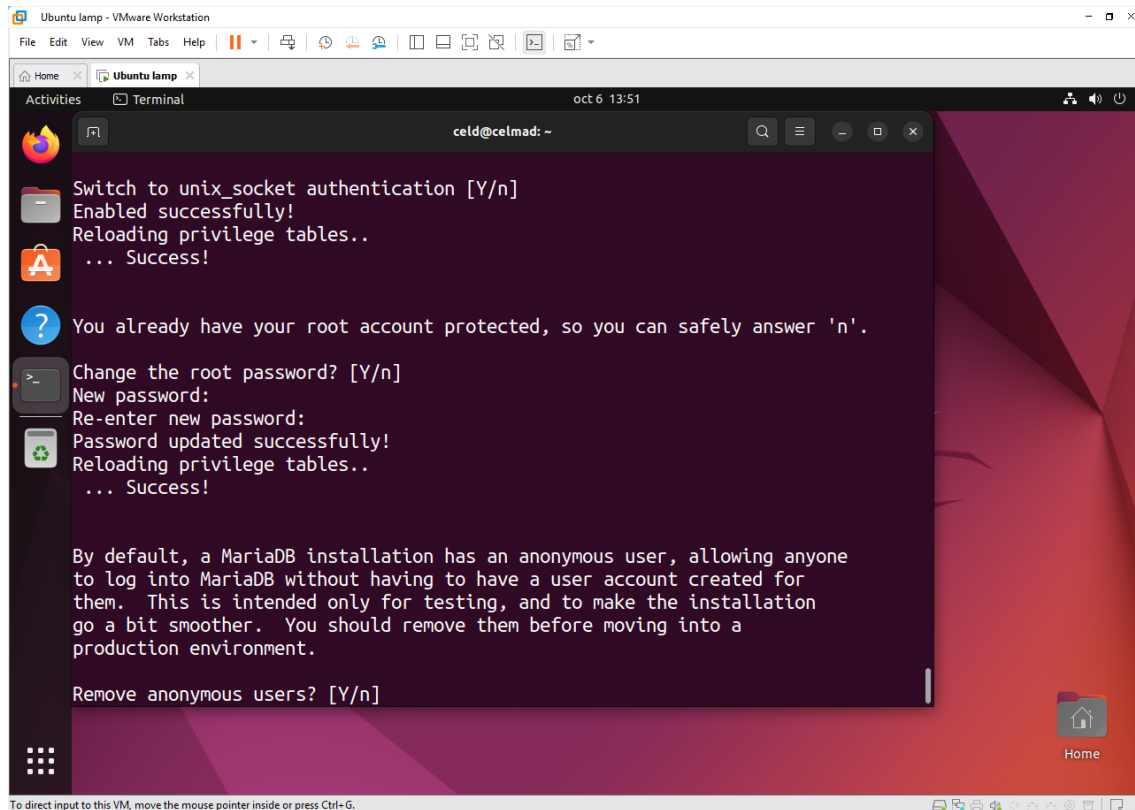
```
cel@celmad:~$ sudo apt install mariadb*
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Note, selecting 'mariadb-tokudb-engine-5.5' for glob 'mariadb*'
Note, selecting 'mariadb-server-core-5.5' for glob 'mariadb*'
Note, selecting 'mariadb-plugin-spider' for glob 'mariadb*'
Note, selecting 'mariadb-server' for glob 'mariadb*'
Note, selecting 'mariadb-plugin-s3' for glob 'mariadb*'
Note, selecting 'mariadb-galera-server-5.5' for glob 'mariadb*'
Note, selecting 'mariadb-test' for glob 'mariadb*'
Note, selecting 'mariadb-plugin-mroonga' for glob 'mariadb*'
Note, selecting 'mariadb-client-core-5.5' for glob 'mariadb*'
Note, selecting 'mariadb-connect-engine-10.0' for glob 'mariadb*'
Note, selecting 'mariadb-connect-engine-10.1' for glob 'mariadb*'
Note, selecting 'mariadb-ocgraph-engine-10.0' for glob 'mariadb*'
Note, selecting 'mariadb-ocgraph-engine-10.1' for glob 'mariadb*'
Note, selecting 'mariadb-gssapi-server-10.1' for glob 'mariadb*'
Note, selecting 'mariadb-gssapi-server-10.2' for glob 'mariadb*'
Note, selecting 'mariadb-test-data-10.0' for glob 'mariadb*'
Note, selecting 'mariadb-testsuite-10.1' for glob 'mariadb*'
Note, selecting 'mariadb-testsuite-10.3' for glob 'mariadb*'
Note, selecting 'mariadb-rocksdb-engine-10.2' for glob 'mariadb*'
Note, selecting 'mariadb-rocksdb-engine-10.3' for glob 'mariadb*'
Do you want to continue? [Y/n]
```


Ahora una vez instalado mysql, escribiremos el comando “mysql_secure_installation” es un script que inicia la configuración de mysql

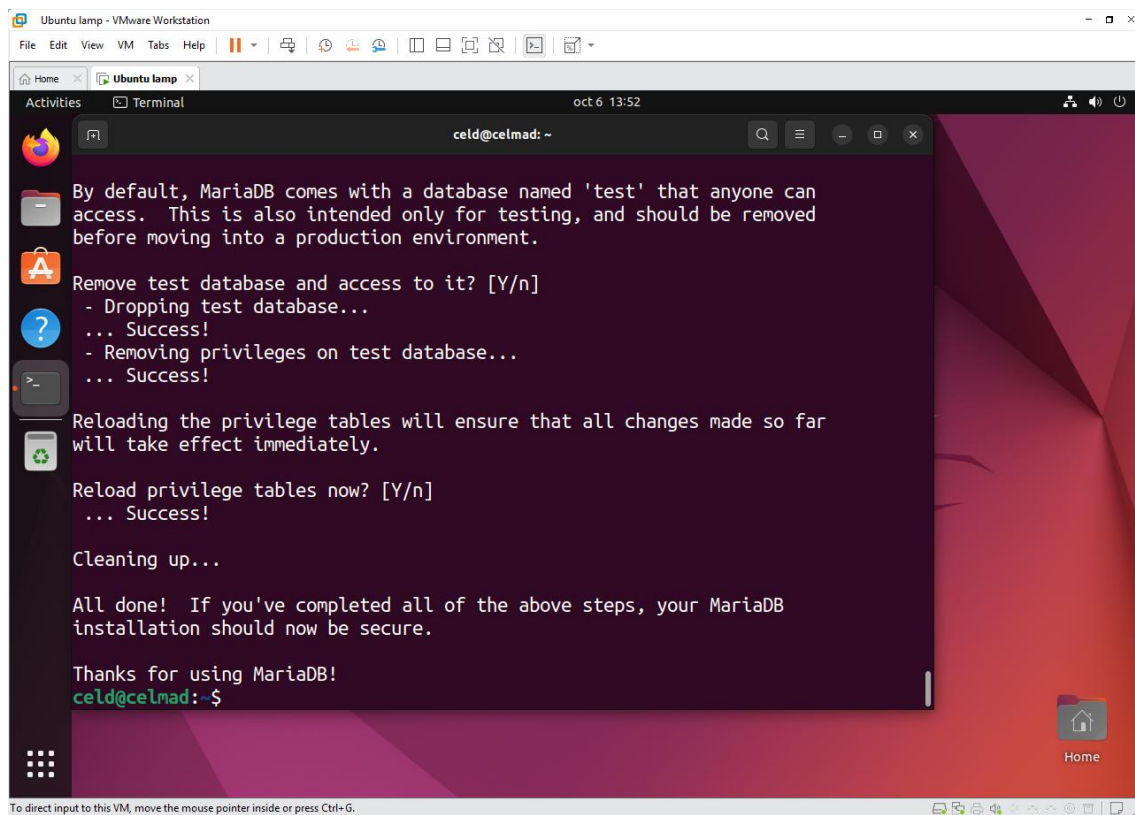


Durante el proceso nos preguntará varias veces sobre opciones y lo que queremos hacer.

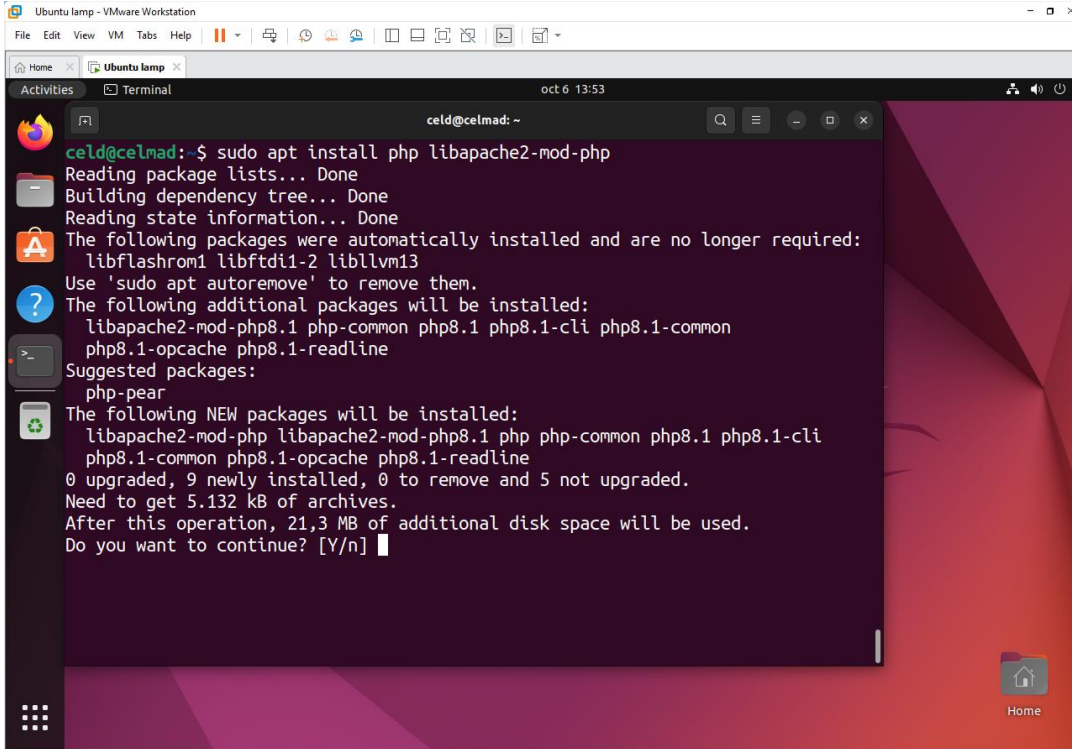




Una vez contestado todo se finalizará la instalación.



Ahora seguiremos con php escribiendo el comando “sudo apt install php libapache2-mod-php”

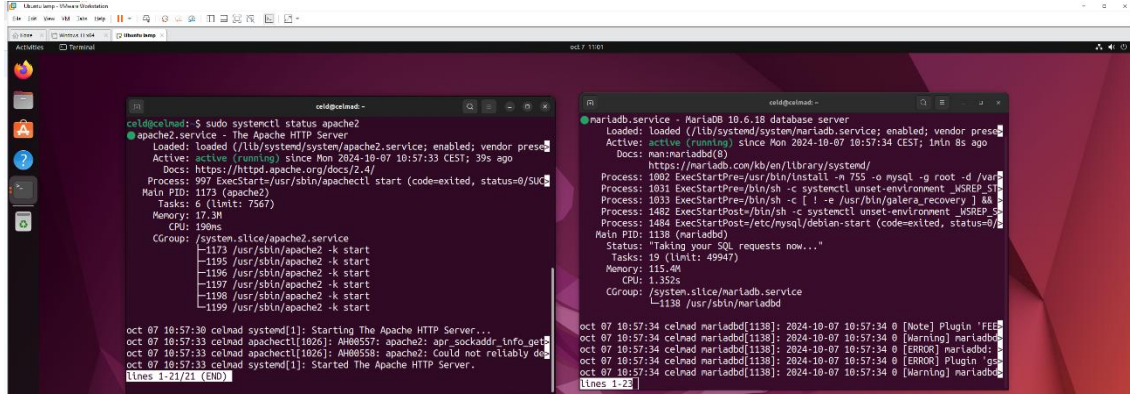


```

celld@celmad: ~
celld@celmad:~$ sudo apt install php libapache2-mod-php
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libflashrom1 libftdi1-2 libllvm13
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  libapache2-mod-php8.1 php-common php8.1 php8.1-cli php8.1-common
  php8.1-opcache php8.1-readline
Suggested packages:
  php-pear
The following NEW packages will be installed:
  libapache2-mod-php libapache2-mod-php8.1 php php-common php8.1 php8.1-cli
  php8.1-common php8.1-opcache php8.1-readline
0 upgraded, 9 newly installed, 0 to remove and 5 not upgraded.
Need to get 5.132 kB of archives.
After this operation, 21,3 MB of additional disk space will be used.
Do you want to continue? [Y/n]
  
```

Con ese simple comando ya se nos instalará completamente.

Ahora comprobaremos el estado de mariadb y apache2 para poder seguir realizando los pasos, comprobaremos el estado con el comando “sudo systemctl status apache2” y “sudo systemctl status mariadb”



```

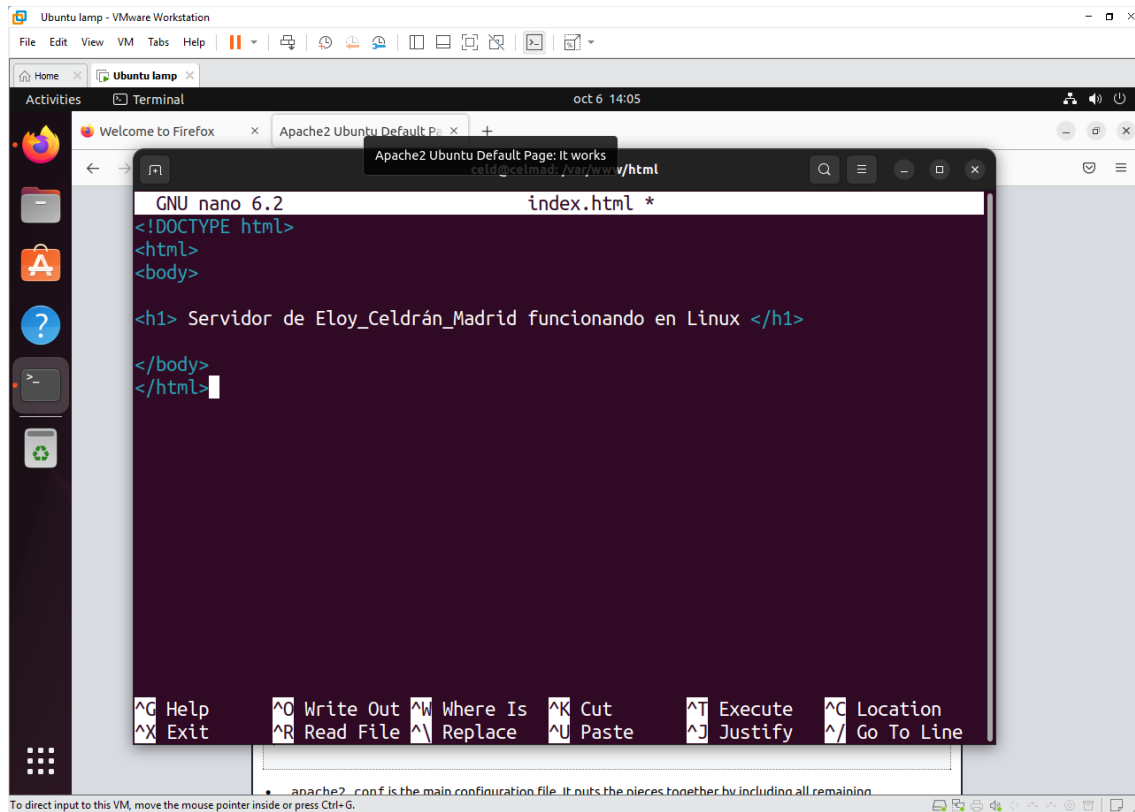
celld@celmad:~$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor prese
   Active: active (running) since Mon 2024-10-07 10:57:33 CEST; 39s ago
     Docs: https://httpd.apache.org/docs/2.4/
   Process: 997 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUK
 Main PID: 1173 (apache2)
    Tasks: 6 (limit: 7567)
     Memory: 17.3M
        CPU: 190ms
   CGroup: /system.slice/apache2.service
           └─1173 /usr/sbin/apache2 -k start
           └─1195 /usr/sbin/apache2 -k start
           └─1196 /usr/sbin/apache2 -k start
           └─1197 /usr/sbin/apache2 -k start
           └─1198 /usr/sbin/apache2 -k start
           └─1199 /usr/sbin/apache2 -k start

oct 07 10:57:38 celmad systemd[1]: Starting The Apache HTTP Server...
oct 07 10:57:33 celmad apachectl[1026]: AH00557: apache2: apr_sockaddr_info_get
oct 07 10:57:33 celmad apachectl[1026]: AH00558: apache2: Could not reliably de
oct 07 10:57:33 celmad systemd[1]: Started The Apache HTTP Server.
lines 1-21/21 (END)

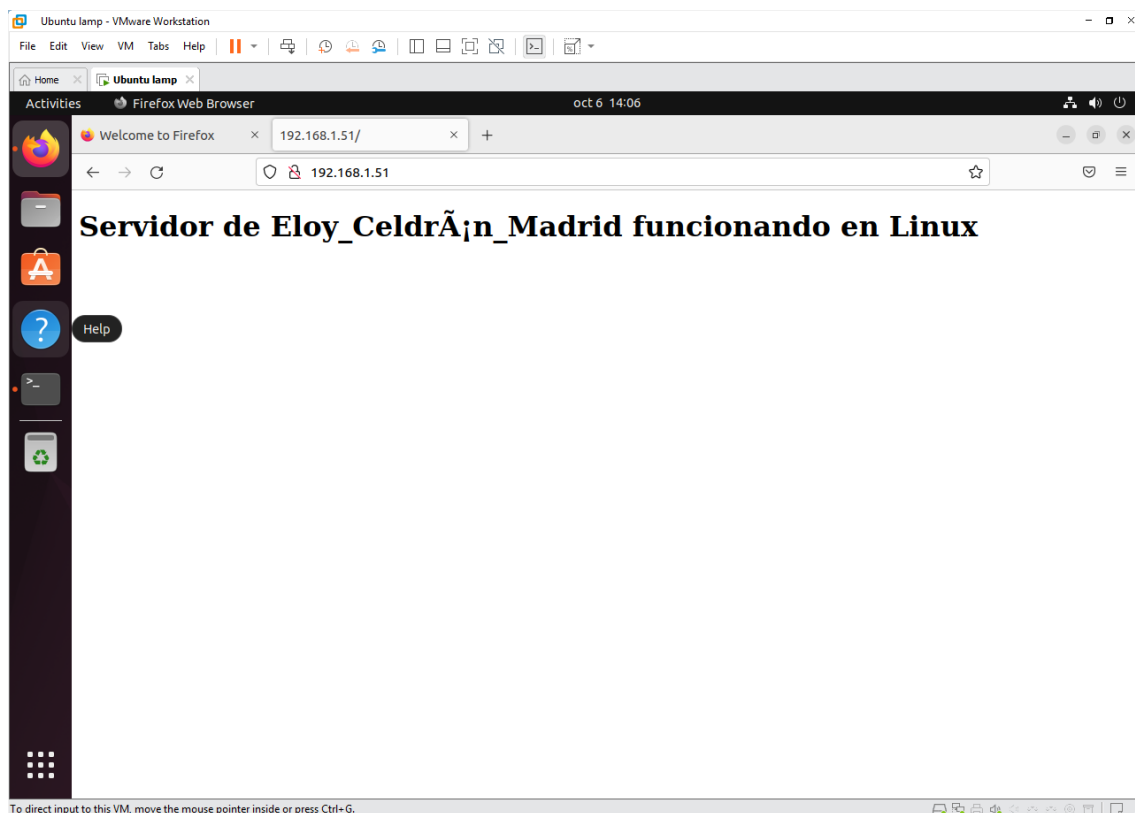
celld@celmad:~$ sudo systemctl status mariadb
● mariadb.service - MariaDB 10.6.18 database server
   Loaded: loaded (/lib/systemd/system/mariadb.service; enabled; vendor prese
   Active: active (running) since Mon 2024-10-07 10:57:34 CEST; 1min 8s ago
     Docs: https://mariadb.com/kb/en/library/systemd/
   Process: 1002 ExecStartPre=/usr/bin/install -m 755 -o mysql -g root -d /var
   Process: 1031 ExecStartPre=/bin/sh -c systemctl unset-environment _MSREP_S
   Process: 1033 ExecStartPre=/bin/sh -c [ ! -e /usr/bin/galera_recovery ] &&
   Process: 1482 ExecStartPost=/bin/sh -c systemctl unset-environment _MSREP_S
   Process: 1484 ExecStartPost=/etc/mysql/debian-start (code=exited, status=0/
 Main PID: 1138 (mariadb)
   Status: "Taking your SQL requests now..."
    Tasks: 19 (limit: 49947)
     Memory: 115.4M
        CPU: 1.352s
   CGroup: /system.slice/mariadb.service
           └─1138 /usr/sbin/mariadb

oct 07 10:57:34 celmad mariadb[1138]: 2024-10-07 10:57:34 0 [Note] Plugin 'FEE
oct 07 10:57:34 celmad mariadb[1138]: 2024-10-07 10:57:34 0 [Warning] mariadb
oct 07 10:57:34 celmad mariadb[1138]: 2024-10-07 10:57:34 0 [ERROR] mariadb
oct 07 10:57:34 celmad mariadb[1138]: 2024-10-07 10:57:34 0 [ERROR] Plugin '
oct 07 10:57:34 celmad mariadb[1138]: 2024-10-07 10:57:34 0 [Warning] mariadb
lines 1-23
  
```

Ahora una vez instalado todo tendremos que irnos a la carpeta /var/www/html y crearemos un nuevo archivo llamado index.html que será la pagina que mostrará cuando accedamos al localhost



Una vez escrito el código y guardado en la ruta mencionada, escribimos localhost/nuestra Ip y nos mostrará el código html que hemos escrito.



Comparación y reflexión final

Diferencias entre la instalación en Windows y Linux

- **Linux** es enfocado a línea de comandos, todo se realiza a través de la terminal por lo que también ofrece mayor control y flexibilidad.
- **Windows** te ofrece una interfaz gráfica para la instalación y gestión de componentes instalados, pero al ser un paquete tienes limitaciones a la hora de instalar paquetes y mayores limitaciones que en Linux.

Sistema más cómodo para la gestión

- **Linux** para mi gusto requiere de mayor curva de aprendizaje ya que necesitas saber de comandos y necesitas saber moverte en Linux, introduciendo rutas, cambiando nombres, etc, no es como en Windows que con clicks lo haces todo, pero aceptando todo eso es mucho más personalizable y puedes gestionar cada parámetro de lo que instales.
- **Windows** es mucho mas intuitivo y la interfaz gráfica facilita la gestión de las tareas contando con eso es menos personalizable

Ajustes para producción segura

Estas medidas se pueden aplicar tanto para Linux como para Windows

- **Fortalecer las contraseñas:** Utilizar contraseñas seguras y únicas para las bases de datos, usuarios del sistema y cualquier otra cuenta.
- **Deshabilitar funciones innecesarias:** Desactivar módulos PHP que no se utilicen para reducir la superficie de ataque.
- **Mantener software actualizado:** Aplicar regularmente parches de seguridad para corregir vulnerabilidades.
- **Configurar el firewall correctamente:** Bloquear el acceso a puertos innecesarios y permitir solo el tráfico necesario.
- **Copias de seguridad periódicas:** Proteger los datos del servidor ante posibles pérdidas.

Tanto LAMP como WAMP pueden ser excelentes opciones para un servidor de producción, siempre y cuando se implementen las medidas de seguridad adecuadas. La elección entre uno u otro dependerá de tus preferencias personales y de los requisitos específicos de tu proyecto.