

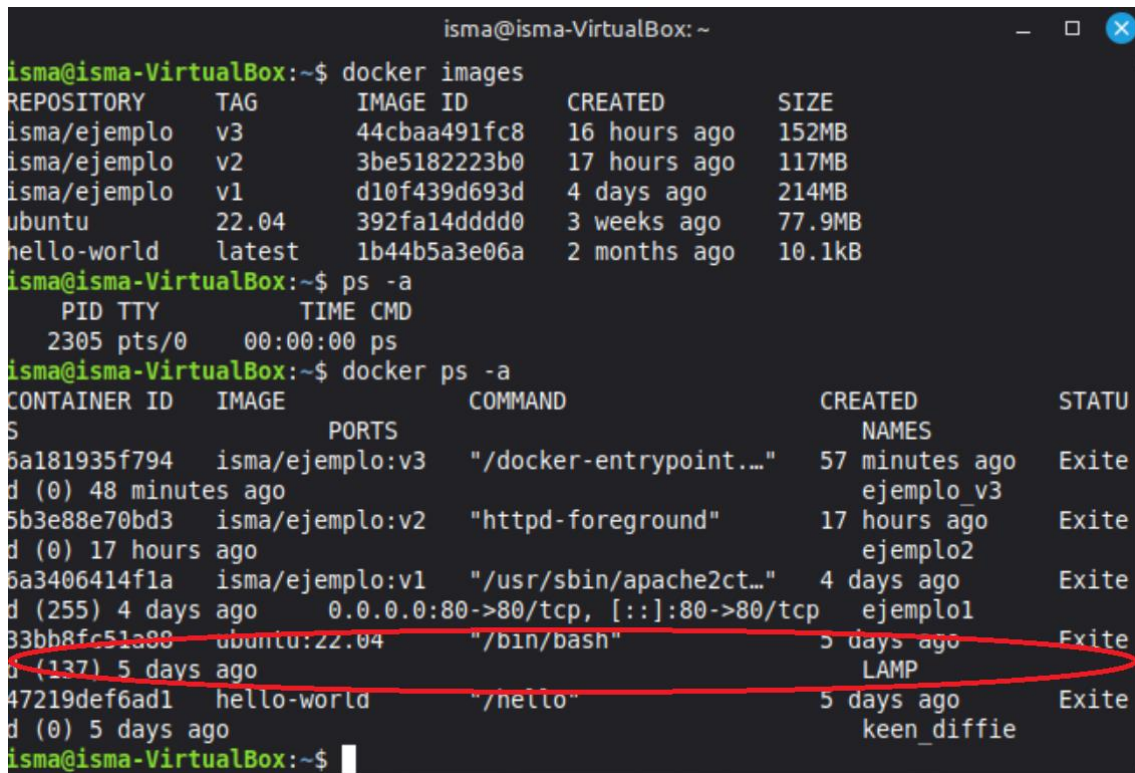
ADMINISTRACIÓN DE SISTEMAS OPERATIVOS (DOCKER LAMP + WORDPRESS)

ISMAEL BERMÚDEZ DEL RÍO

CREACIÓN DE UN DOCKER, INSTALANDO LAMP + WORDPRESS.

1. Primero crearemos un contenedor con la imagen base “Ubuntu” con la etiqueta “22.04”, lo llamaremos LAMP y lo exponremos el puerto 80 dentro del puerto 8080 de nuestro sistema. Al crearlo se crea una lista de “Shell” para instalar los programas necesarios. Lo haremos con el siguiente comando:

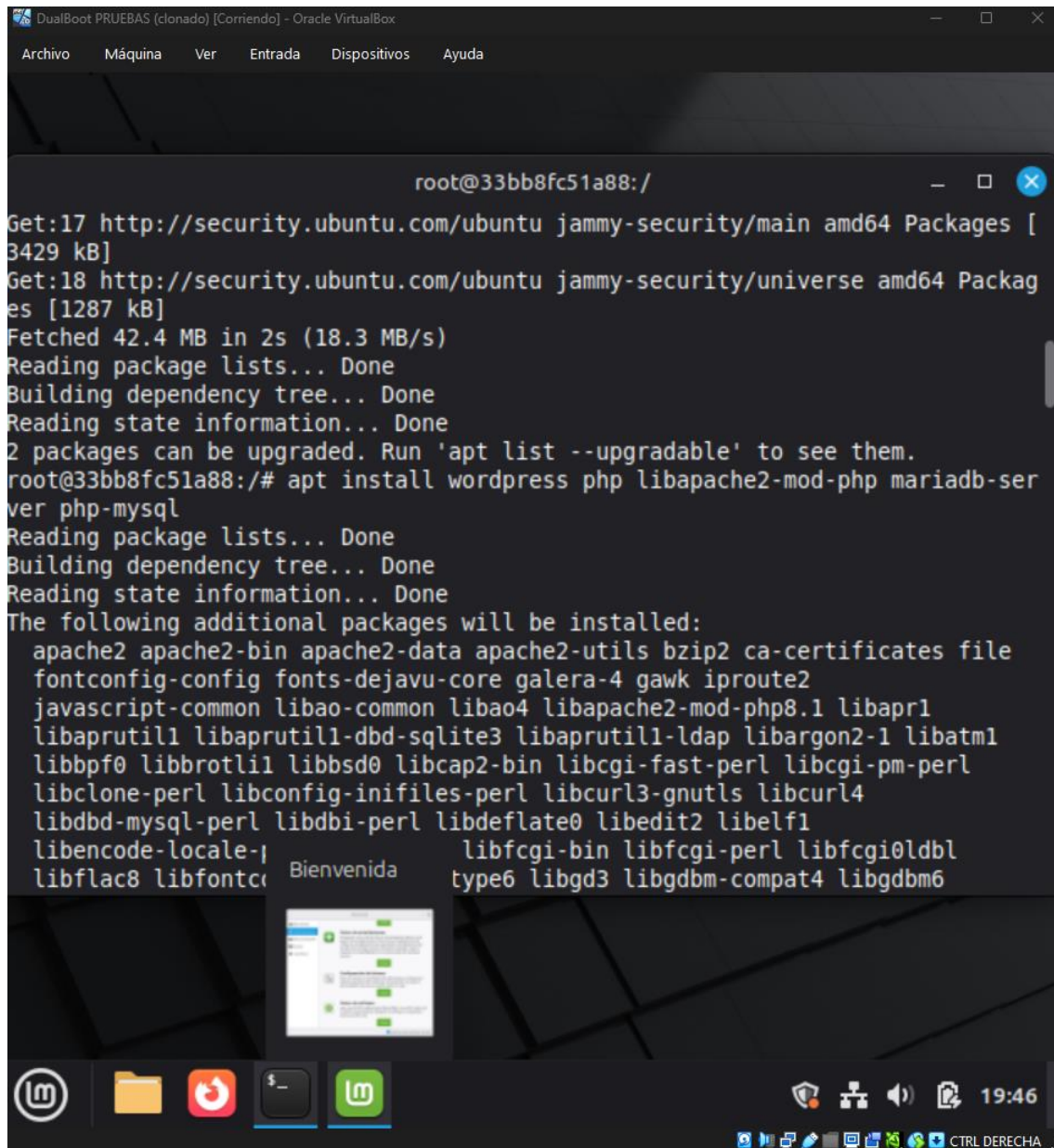
```
docker run -it -p 8080:80 -name LAMP Ubuntu:22.04 /bin/bash
```



```
isma@isma-VirtualBox: ~  
isma@isma-VirtualBox:~$ docker images  
REPOSITORY          TAG             IMAGE ID        CREATED         SIZE  
isma/ejemplo         v3              44cbaa491fc8    16 hours ago    152MB  
isma/ejemplo         v2              3be5182223b0    17 hours ago    117MB  
isma/ejemplo         v1              d10f439d693d    4 days ago      214MB  
ubuntu               22.04           392fa14dddd0    3 weeks ago     77.9MB  
hello-world          latest          1b44b5a3e06a    2 months ago    10.1kB  
isma@isma-VirtualBox:~$ ps -a  
PID TTY          TIME CMD  
2305 pts/0        00:00:00 ps  
isma@isma-VirtualBox:~$ docker ps -a  
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS  
6a181935f794   isma/ejemplo:v3   "/docker-entrypoint..." 57 minutes ago Exited  
d (0) 48 minutes ago  
5b3e88e70bd3   isma/ejemplo:v2   "httpd-foreground"        17 hours ago   Exited  
d (0) 17 hours ago  
6a3406414f1a   isma/ejemplo:v1   "/usr/sbin/apache2ct..." 4 days ago     Exited  
d (255) 4 days ago    0.0.0.0:80->80/tcp, [::]:80->80/tcp ejemplo1  
83bb8fc51a88   ubuntu:22.04     "/bin/bash"                5 days ago     Exited  
d (137) 5 days ago  
47219def6ad1   hello-world      "/hello"                   5 days ago     Exited  
d (0) 5 days ago  
isma@isma-VirtualBox:~$
```

- En la imagen veremos que está creado el contenedor con la base de Ubuntu pero el nombre de la imagen lo hemos llamado “LAMP”.

2. Luego instalaremos todo lo necesario para tener el Wordpress junto el LAMP, primero será actualizar la lista de los paquetes de repositorio con “apt update”, luego instalamos los paquetes necesarios para instalar LAMP + Wordpress, para instalar ejecutaremos “apt install wordpress php libapache2-mod-php mariadb-server php-mysql”



```
root@33bb8fc51a88:/  
Get:17 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [3429 kB]  
Get:18 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [1287 kB]  
Fetched 42.4 MB in 2s (18.3 MB/s)  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
2 packages can be upgraded. Run 'apt list --upgradable' to see them.  
root@33bb8fc51a88:/# apt install wordpress php libapache2-mod-php mariadb-server php-mysql  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
The following additional packages will be installed:  
  apache2 apache2-bin apache2-data apache2-utils bzip2 ca-certificates file  
  fontconfig-config fonts-dejavu-core galera-4 gawk iproute2  
  javascript-common libao-common libao4 libapache2-mod-php8.1 libapr1  
  libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap libargon2-1 libatm1  
  libbpf0 libbrotli1 libbsd0 libcap2-bin libcgi-fast-perl libcgi-pm-perl  
  libclone-perl libconfig-inifiles-perl libcurl3-gnutls libcurl4  
  libdbd-mysql-perl libdbi-perl libdeflate0 libedit2 libelf1  
  libencode-locale-perl libfcgi-bin libfcgi-perl libfcgi0ldbl  
  libflac8 libfontconfig1 libgdbm-compat4 libgdbm6  
Bienvenida
```

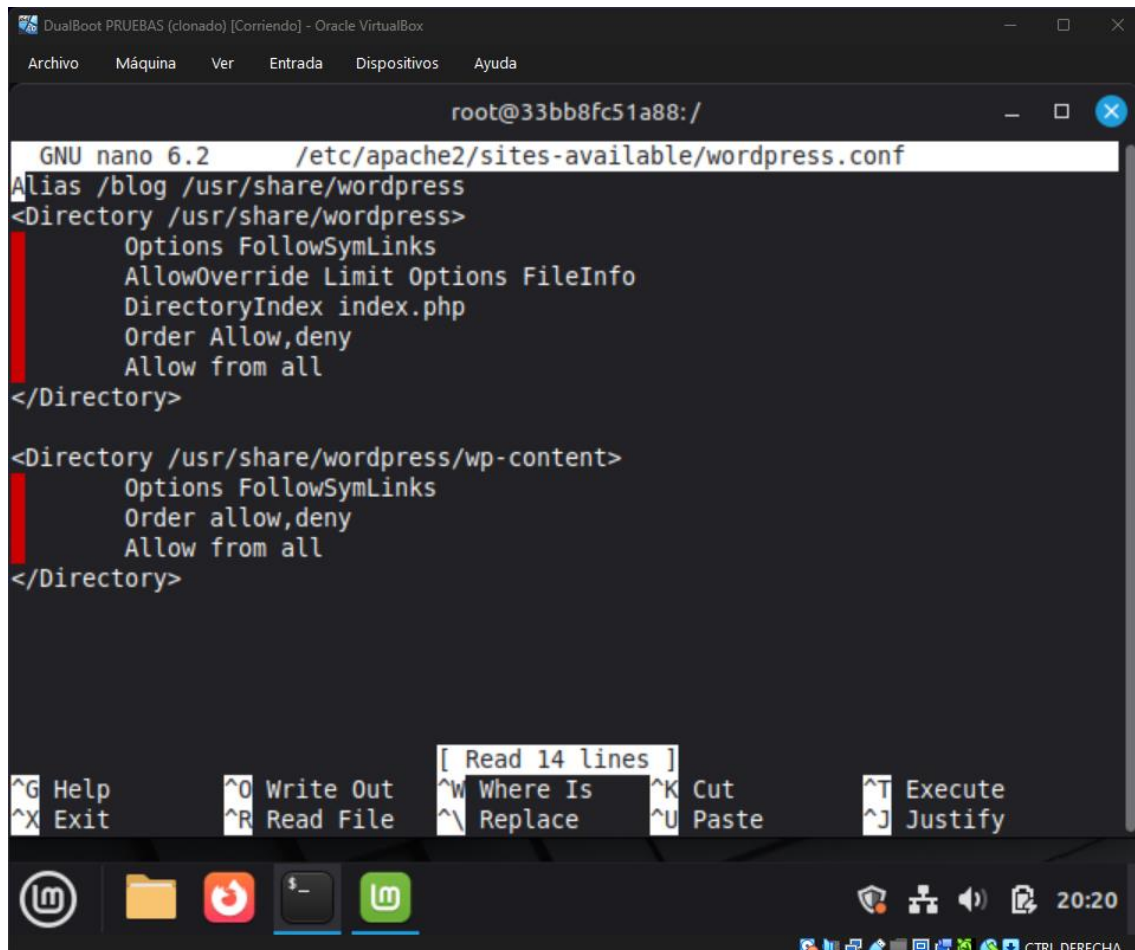
- Después podemos lanzar el servicio de apache con el comando:

“service apache2 start”

3. Antes de poder configurar el apache para trabajar con Wordpress instalaremos dentro del entorno Ubuntu que esta alojado en nuestro docker, para esto utilizamos el comando:

“apt install nano”

- Luego de haber instalado el editor de texto, nos basamos a la configuración de Apache para trabajar con Wordpress, para esto creamos un fichero de configuración del sitio en Apache **“/etc/apache2/sites-available/wordpress.conf”** lo que haremos es configurar el acceso al sitio Wordpress. Dentro del fichero pondremos:



```
DualBoot PRUEBAS (clonado) [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda

root@33bb8fc51a88: /
GNU nano 6.2 /etc/apache2/sites-available/wordpress.conf
Alias /blog /usr/share/wordpress
<Directory /usr/share/wordpress>
    Options FollowSymLinks
    AllowOverride Limit Options FileInfo
    DirectoryIndex index.php
    Order Allow,deny
    Allow from all
</Directory>

<Directory /usr/share/wordpress/wp-content>
    Options FollowSymLinks
    Order allow,deny
    Allow from all
</Directory>
```

[Read 14 lines]

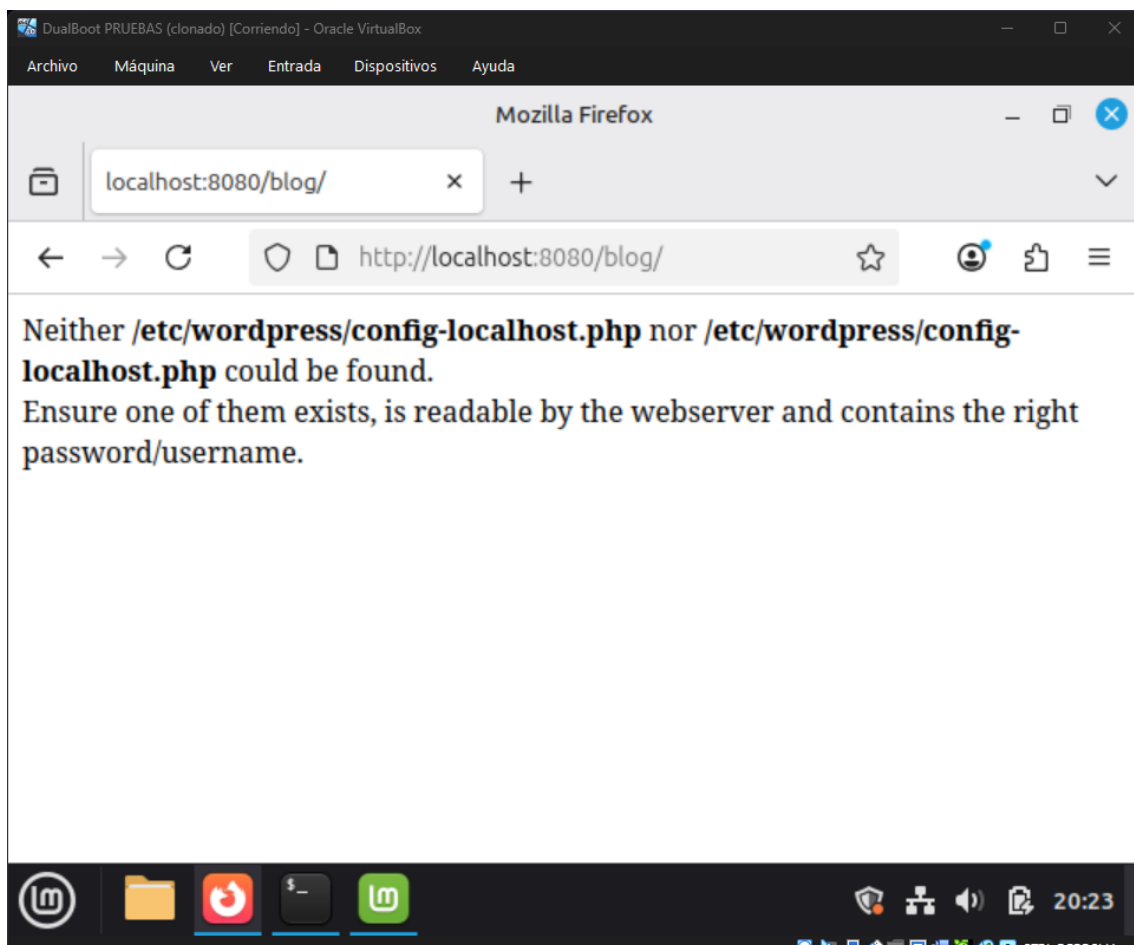
^G Help	^O Write Out	^W Where Is	^K Cut	^T Execute
^X Exit	^R Read File	^_ Replace	^U Paste	^J Justify

20:20

- Cuando hayamos creado el fichero cargaremos el sitio, habilitar la “URL rewriting” y recargaremos Apache con los comandos:

“a2ensite wordpress” “a2enmode rewrite” y service apache2 restart”

- Con esto podemos probar con la URL <http://localhost:8080/blog/>, pero veremos un error porque no hemos configurado aún el fichero de la ruta **“/etc/wordpress/config-localhost.php”**.



- Ahora deberemos de poner en marcha el servicio de MariaDB server, con el comando:

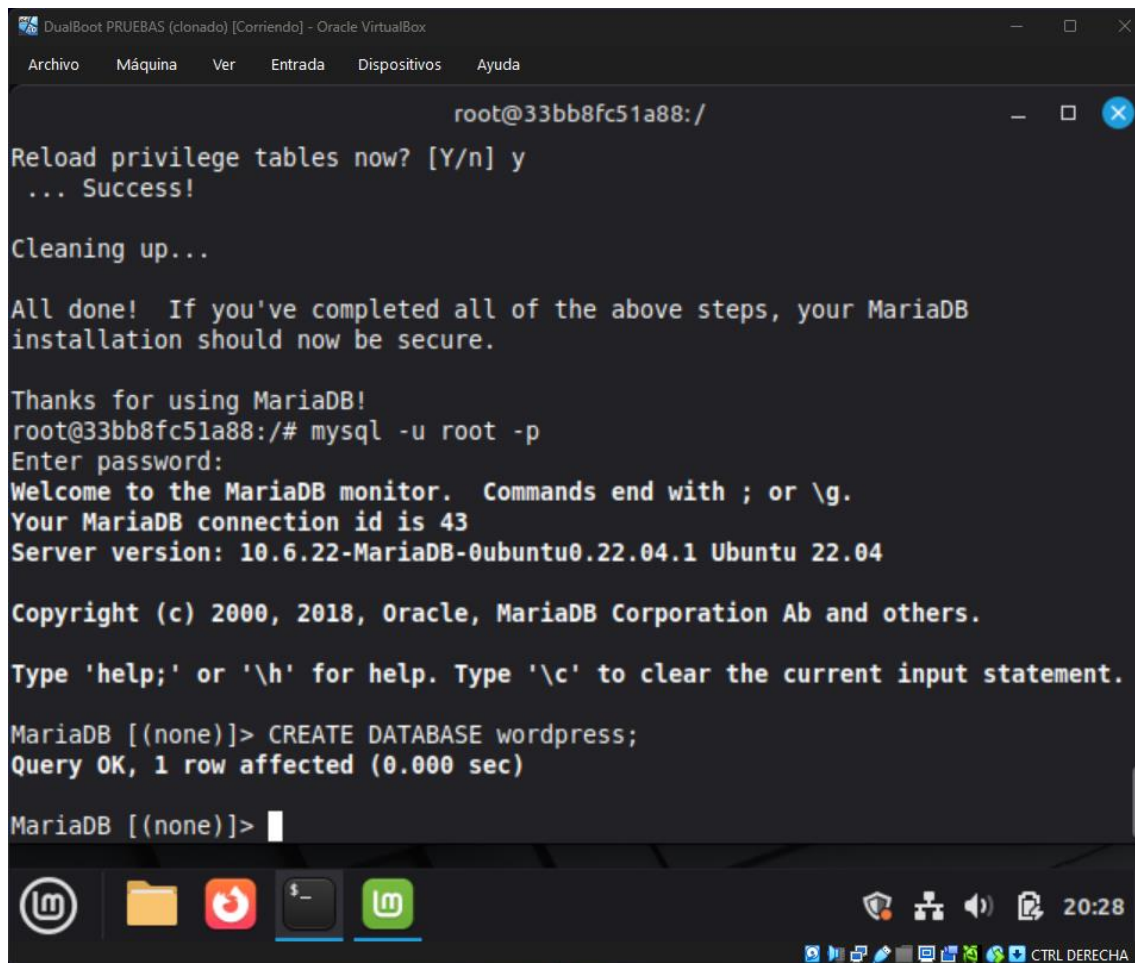
“service mariadb start”

- Después aplicaremos para ejecutar el comando para generar un password de root de MySQL Server de forma segura y otras opciones. El comando que ejecutamos es el siguiente:

“mysql_secure_installation”

- Después de la instalación accederemos a la base de datos con el cliente MySQL con el comando:

“mysql -u -root -p”



```
root@33bb8fc51a88:/
Reload privilege tables now? [Y/n] y
... Success!

Cleaning up...

All done! If you've completed all of the above steps, your MariaDB
installation should now be secure.

Thanks for using MariaDB!
root@33bb8fc51a88:/# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 43
Server version: 10.6.22-MariaDB-0ubuntu0.22.04.1 Ubuntu 22.04

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

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

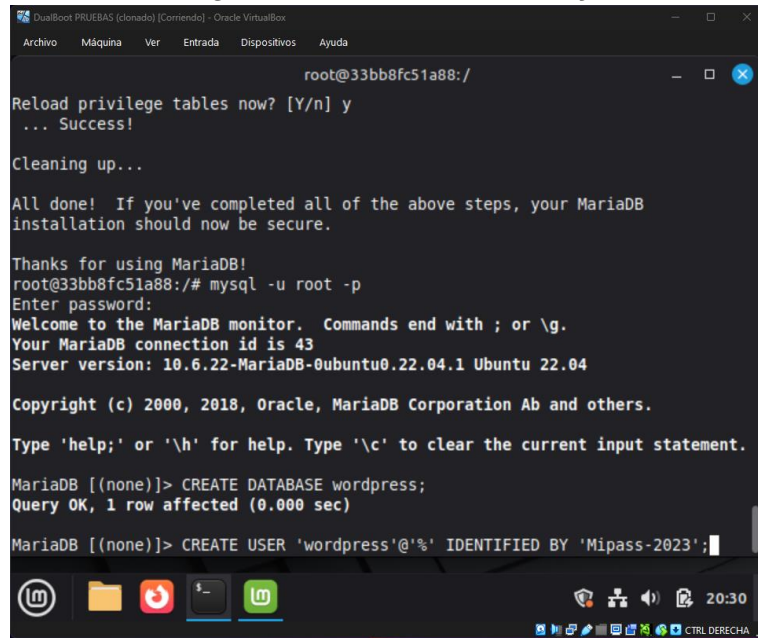
MariaDB [(none)]> CREATE DATABASE wordpress;
Query OK, 1 row affected (0.000 sec)

MariaDB [(none)]>
```

- Luego creamos la base de datos “Wordpress” con CREATE DATABASE wordpress;

- Tras ello creamos el usuario de “Worpress” (con contraseña “MiPass-2023”) y le damos los permisos totales en la base de datos “wordpress”.

CREATE USER 'wordpress'@'%' IDENTIFIED BY 'MyPass-2023';



The screenshot shows a terminal window titled "DualBoot PRUEBAS (clonado) [Corriendo] - Oracle VirtualBox". The prompt is "root@33bb8fc51a88:/". The user enters "y" to reload privilege tables, which is successful. The terminal then shows "Cleaning up..." and a message: "All done! If you've completed all of the above steps, your MariaDB installation should now be secure." It then says "Thanks for using MariaDB!" and shows the command "root@33bb8fc51a88:/# mysql -u root -p". The user enters a password, and the terminal shows "Welcome to the MariaDB monitor. Commands end with ; or \g. Your MariaDB connection id is 43. Server version: 10.6.22-MariaDB-0ubuntu0.22.04.1 Ubuntu 22.04". It then shows the copyright notice and instructions. The user enters "CREATE DATABASE wordpress;" and the terminal shows "Query OK, 1 row affected (0.000 sec)". Finally, the user enters "CREATE USER 'wordpress'@'%' IDENTIFIED BY 'Mipass-2023';" and the terminal shows the command being entered.

```
root@33bb8fc51a88:/
Reload privilege tables now? [Y/n] y
... Success!

Cleaning up...

All done! If you've completed all of the above steps, your MariaDB
installation should now be secure.

Thanks for using MariaDB!
root@33bb8fc51a88:/# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 43
Server version: 10.6.22-MariaDB-0ubuntu0.22.04.1 Ubuntu 22.04

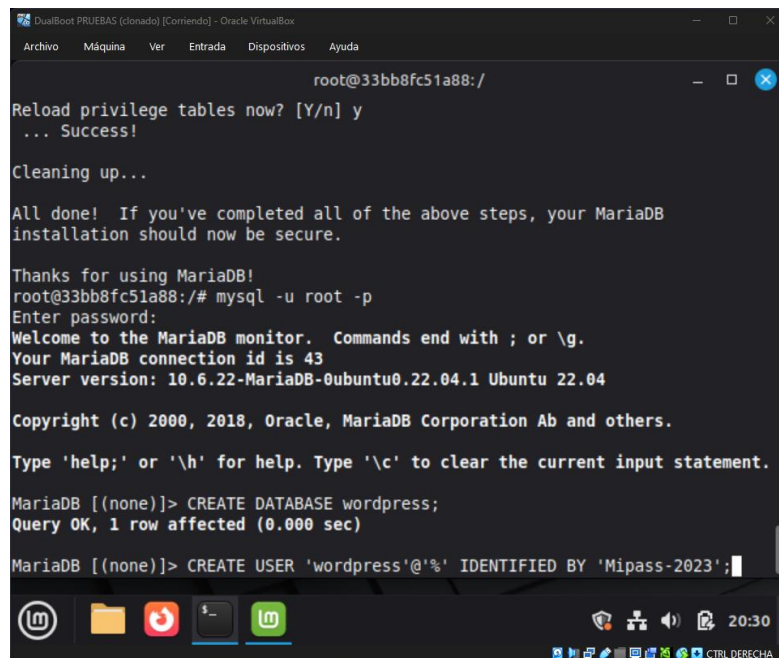
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

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

MariaDB [(none)]> CREATE DATABASE wordpress;
Query OK, 1 row affected (0.000 sec)

MariaDB [(none)]> CREATE USER 'wordpress'@'%' IDENTIFIED BY 'Mipass-2023';
```

GRANT ALL PRIVILEGES ON wordpress.* TO 'wordpress'@'%' WITH GRANT OPTION;



The screenshot shows a terminal window titled "DualBoot PRUEBAS (clonado) [Corriendo] - Oracle VirtualBox". The prompt is "root@33bb8fc51a88:/". The user enters "y" to reload privilege tables, which is successful. The terminal then shows "Cleaning up..." and a message: "All done! If you've completed all of the above steps, your MariaDB installation should now be secure." It then says "Thanks for using MariaDB!" and shows the command "root@33bb8fc51a88:/# mysql -u root -p". The user enters a password, and the terminal shows "Welcome to the MariaDB monitor. Commands end with ; or \g. Your MariaDB connection id is 43. Server version: 10.6.22-MariaDB-0ubuntu0.22.04.1 Ubuntu 22.04". It then shows the copyright notice and instructions. The user enters "CREATE DATABASE wordpress;" and the terminal shows "Query OK, 1 row affected (0.000 sec)". Finally, the user enters "CREATE USER 'wordpress'@'%' IDENTIFIED BY 'Mipass-2023';" and the terminal shows the command being entered.

```
root@33bb8fc51a88:/
Reload privilege tables now? [Y/n] y
... Success!

Cleaning up...

All done! If you've completed all of the above steps, your MariaDB
installation should now be secure.

Thanks for using MariaDB!
root@33bb8fc51a88:/# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 43
Server version: 10.6.22-MariaDB-0ubuntu0.22.04.1 Ubuntu 22.04

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)]> CREATE DATABASE wordpress;
Query OK, 1 row affected (0.000 sec)

MariaDB [(none)]> CREATE USER 'wordpress'@'%' IDENTIFIED BY 'Mipass-2023';
```

```
DualBoot PRUEBAS (clonado) [Corriendo] - Oracle VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda

root@33bb8fc51a88:/

All done! If you've completed all of the above steps, your MariaDB
installation should now be secure.

Thanks for using MariaDB!
root@33bb8fc51a88:/# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 43
Server version: 10.6.22-MariaDB-0ubuntu0.22.04.1 Ubuntu 22.04

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)]> CREATE DATABASE wordpress;
Query OK, 1 row affected (0.000 sec)

MariaDB [(none)]> CREATE USER 'wordpress'@'%' IDENTIFIED BY 'Mipass-2023';
Query OK, 0 rows affected (0.003 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON wordpress.* TO 'wordpress'@'%' WITH
GRANT OPTION;
```

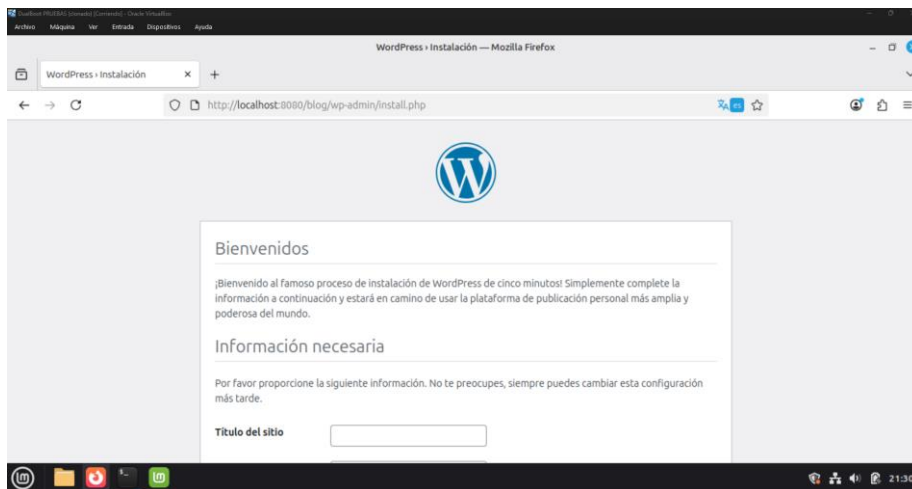
- Ahora utilizaremos el editor de texto de la consola, crearemos o editaremos si existiera el fichero de configuración Wordpress, “/etc/wordpress/config-localhost.php”, quedando de esta manera:

```
DualBoot PRUEBAS (clonado) [Corriendo] - Oracle VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda

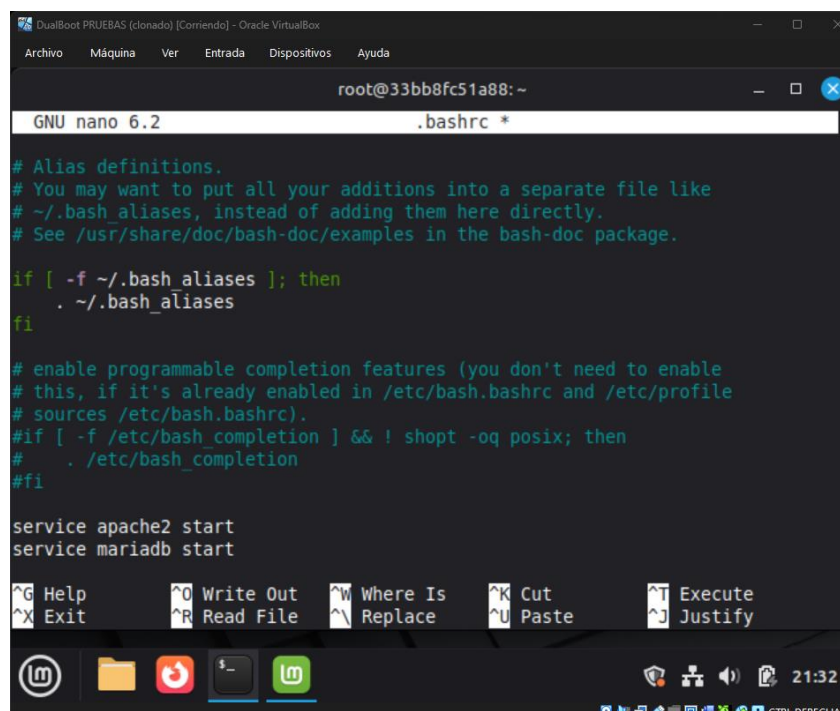
root@33bb8fc51a88:/

GNU nano 6.2 /etc/wordpress/config-localhost.php *
<?php
define('DB_NAME', 'wordpress');
define('DB_USER', 'wordpress');
define('DB_PASSWORD', 'MiPass-2023');
define('DB_HOST', 'localhost');
define('DB_COLLATE', 'utf8_general_ci');
define('WP_CONTENT_DIR', '/usr/share/wordpress/wp-content');
?>
```

- Si ha ido bien podremos acceder en el navegador al Wordpress desde la URL `http://localhost:8080/blog`.



4. Ahora crearemos un bash para que el docker se inicie automaticamente, para esto accederemos en /root y editaremos el fichero **".bashrc"** con el proposito que al iniciar shell lancen los servicios de Apache y MySQL:



- Ya por último para parar el docker iniciamos el comando:

docker stop LAMP

- Si queremos iniciar de nuevo con el comando: **docker start LAMP**.