

ADMINISTRACIÓN DE SISTEMAS OPERATIVOS, WORDPRESS + MARIADB

CASO PRÁCTICO WORDPRESS + MARIADB

1. En esta prácticaaremos correr CMS Wordpress. Para esto usaremos una red donde se conectarán dos contenedores: uno estará utilizando Apache + PHP, junto con una versión de Wordpress, mientras que el segundo contenedor contendrá un servidor de bases de datos MariaDB. También realizaremos una migración de versión del contenedor MariaDB.

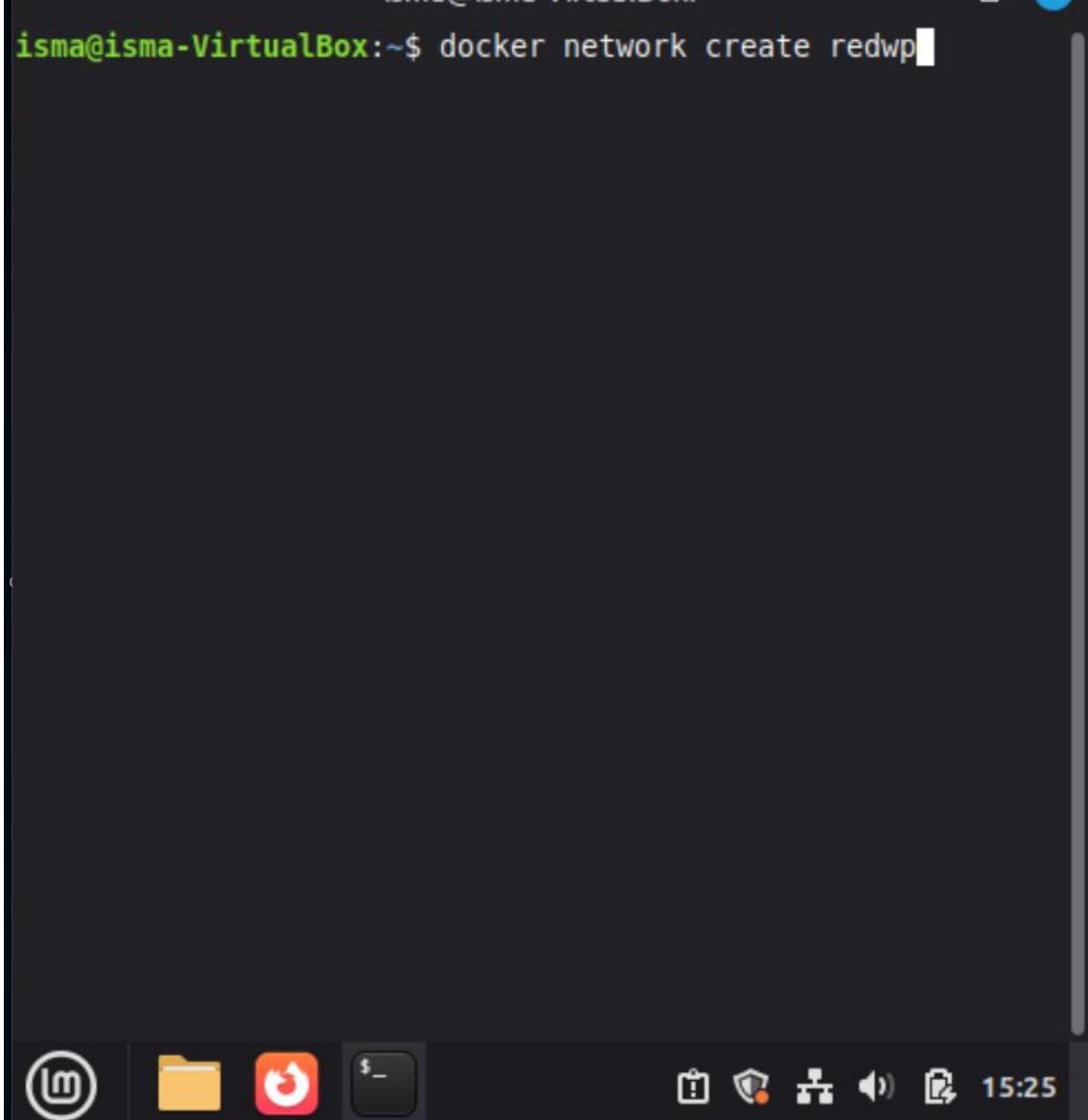
2. Vamos crear una red con el siguiente comando:

```
docker network create redwp
```

DualBoot PRUEBAS (clonado) [Corriendo] - Oracle VirtualBox

Archivo Máquina Ver Entrada Dispositivos Ayuda

```
isma@isma-VirtualBox:~$ docker network create redwp
```

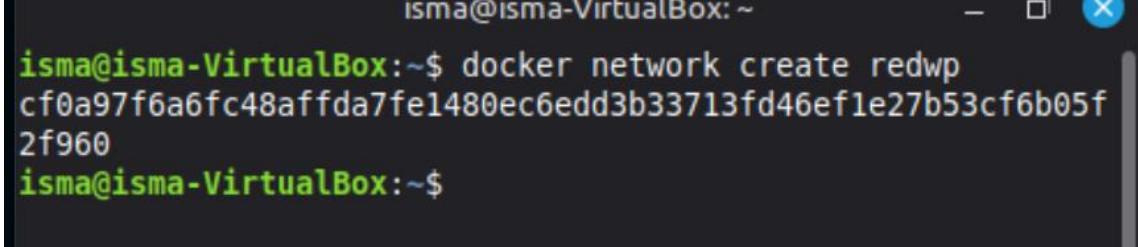


The screenshot shows a standard Ubuntu desktop environment running inside Oracle VM VirtualBox. At the top is the Unity interface with a menu bar. Below it is a terminal window titled 'isma@isma-VirtualBox:~\$'. The user has run the command 'docker network create redwp' and is awaiting a response. The desktop icons at the bottom include the Dash, Home, File Explorer, and Terminal. A system tray icon is visible on the right.

DualBoot PRUEBAS (clonado) - Oracle VirtualBox

Archivo Máquina Ver Entrada Dispositivos Ayuda

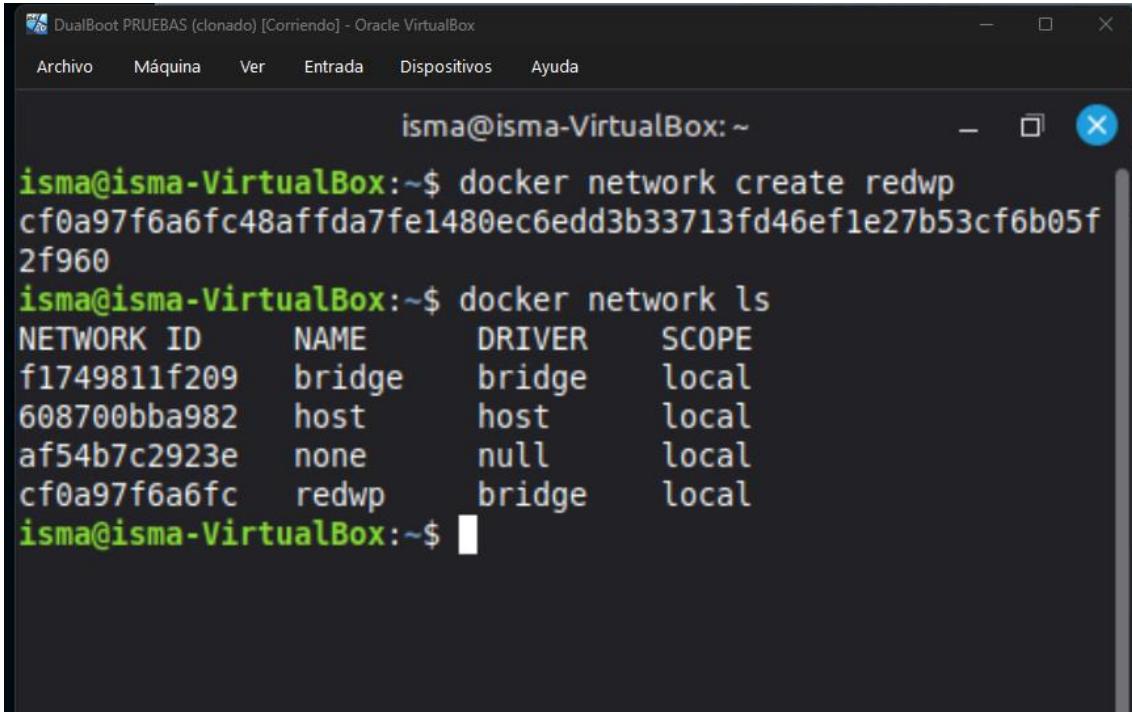
```
isma@isma-VirtualBox:~$ docker network create redwp
cf0a97f6a6fc48affda7fe1480ec6edd3b33713fd46ef1e27b53cf6b05f
2f960
isma@isma-VirtualBox:~$
```



This screenshot shows the same Ubuntu desktop environment after the command has been executed. The terminal window now displays the output of the 'docker network create redwp' command, which shows the generated network identifier 'cf0a97f6a6fc48affda7fe1480ec6edd3b33713fd46ef1e27b53cf6b05f2f960'. The desktop icons and system tray are visible at the bottom.

- Para asegurarnos que está creado vamos utilizar el comando:

```
docker network ls
```



The screenshot shows a terminal window titled "DualBoot PRUEBAS (clonado) [Corriendo] - Oracle VirtualBox". The window has a menu bar with "Archivo", "Máquina", "Ver", "Entrada", "Dispositivos", and "Ayuda". The title bar shows the user "isma@isma-VirtualBox: ~". The terminal content is as follows:

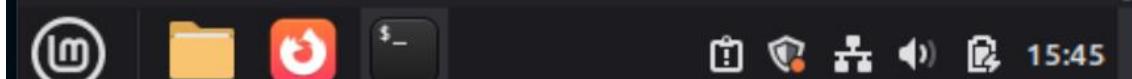
```
isma@isma-VirtualBox:~$ docker network create redwp
cf0a97f6a6fc48affda7fe1480ec6edd3b33713fd46ef1e27b53cf6b05f
2f960
isma@isma-VirtualBox:~$ docker network ls
NETWORK ID      NAME      DRIVER      SCOPE
f1749811f209    bridge    bridge      local
608700bba982    host      host       local
af54b7c2923e    none      null       local
cf0a97f6a6fc    redwp     bridge      local
isma@isma-VirtualBox:~$
```

3. Creación del contenedor MariaDB.

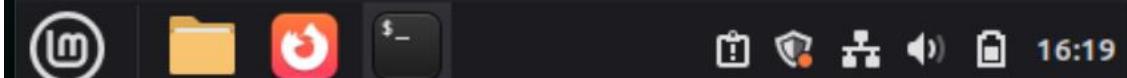
- Para crear la base de datos nos basaremos en el siguiente comando:

```
docker run --name nuestromariadb \
--network redwp \
-v /home/sergi/mariadbdata:/var/lib/mysql \
-e MARIADB_ROOT_PASSWORD=cefireroot \
-e MARIADB_USER=cefireuser \
-e MARIADB_PASSWORD=cefirepass \
-e MARIADB_DATABASE=cefiiredb \
-d mariadb:10.6
```

```
isma@isma-VirtualBox:~$ docker run --name nuestromariadb \
--network redwp \
-v /home/sergi/mariadbdata:/var/lib/mysql \
-e MARIADB_ROOT_PASSWORD=cefireroot \
-e MARIADB_USER=cefireuser \
-e MARIADB_PASSWORD=cefirepass \
-e MARIADB_DATABASE=cefiredb \
-d mariadb:10.7
Unable to find image 'mariadb:10.7' locally
10.7: Pulling from library/mariadb
99803d4b97f3: Downloading 3.535MB/28.58MB
b8bc823a83fd: Download complete
16685f710f5d: Downloading 1.851MB/7.059MB
b5660ff63058: Download complete
8b13582f0741: Download complete
7d98acc143e6: Waiting
6e4fbbea63e0: Waiting
d3da17671559: Waiting
```



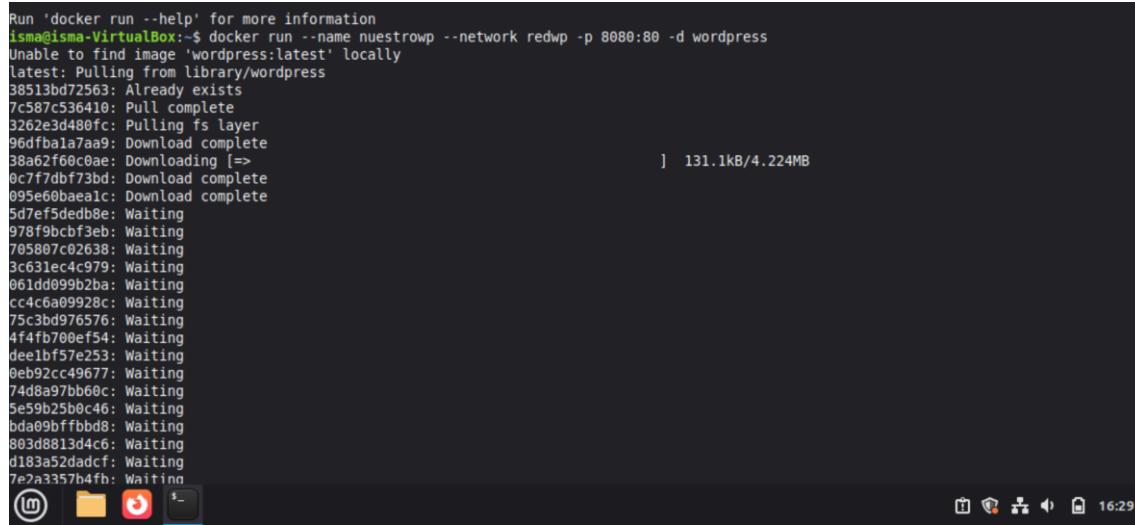
```
isma@isma-VirtualBox:~ - □ X
Run 'docker run --help' for more information
isma@isma-VirtualBox:~$ docker run --name nuestromariadb \
--network redwp \
-v /home/sergi/mariadbdata:/var/lib/mysql \
-e MARIADB_ROOT_PASSWORD=cefireroot \
-e MARIADB_USER=cefireuser \
-e MARIADB_PASSWORD=cefirepass \
-e MARIADB_DATABASE=cefiredb \
-d mariadb:10.7
Unable to find image 'mariadb:10.7' locally
10.7: Pulling from library/mariadb
99803d4b97f3: Pull complete
b8bc823a83fd: Pull complete
16685f710f5d: Pull complete
b5660ff63058: Pull complete
8b13582f0741: Pull complete
7d98acc143e6: Pull complete
6e4fbbea63e0: Pull complete
d3da17671559: Pull complete
Digest: sha256:9a48ac9f196f3d4fd6fea2cab59a49df9e7ca459bf14
b2f7b85a0e38a5454571
Status: Downloaded newer image for mariadb:10.7
5955e860b7887c4256419285ec7faf2bb96b72f2b8eb2f79029a6176363
ba51d
isma@isma-VirtualBox:~$
```



The screenshot shows a terminal window on an Ubuntu desktop environment. The terminal window has a dark background and white text. It displays a command-line session where the user is running a Docker container named 'nuestromariadb'. The container is based on the 'mariadb:10.7' image, which is being pulled from the 'library/mariadb' repository. The terminal shows the progress of the pull, including intermediate image names like '99803d4b97f3', 'b8bc823a83fd', etc., and their 'Pull complete' status. It also shows the final digest and status message. Below the terminal, the standard Ubuntu desktop taskbar is visible, featuring icons for the Dash, Home, File Explorer, Terminal, and a system tray with network, battery, and volume indicators. The current time is shown as 16:19.

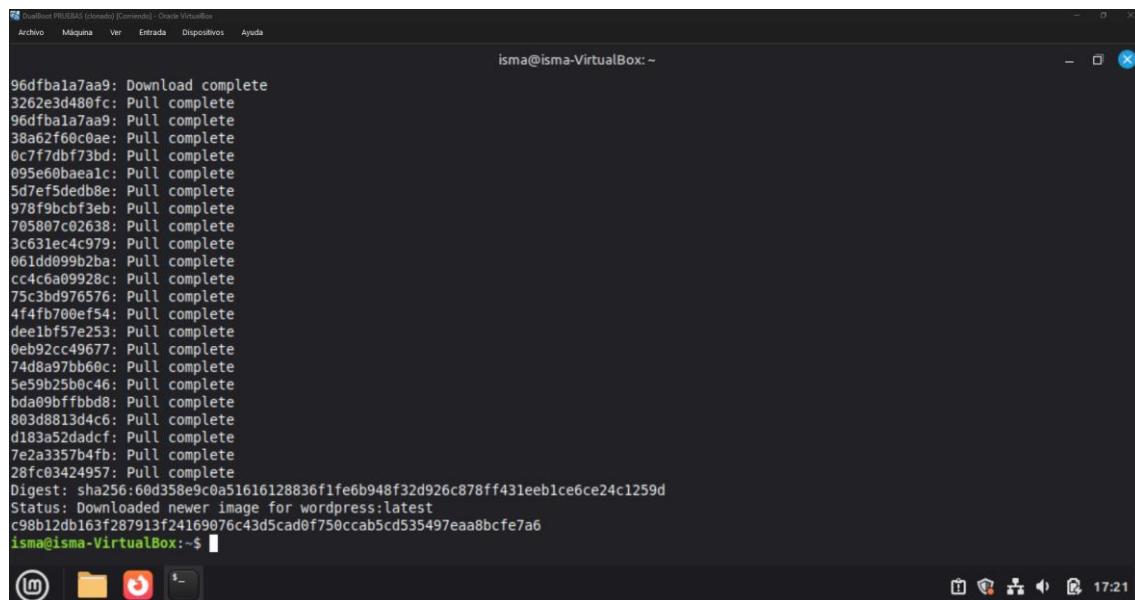
4. Ahora vamos crear el contenedor que contiene Apache + PHP, tanto Wordpress preinstalado, para crearlo será:

```
docker run --name nuestrowp --network redwp -p 8080:80 -d  
wordpress
```



```
Run 'docker run --help' for more information
isma@isma-VirtualBox:~$ docker run --name nuestrowp --network redwp -p 8080:80 -d wordpress
Unable to find image 'wordpress:latest' locally
latest: Pulling from library/wordpress
38513bd2563: Already exists
7c587c536410: Pull complete
3262e3d480fc: Pulling fs layer
96dfba1a7aa9: Download complete
38a62f60c0ae: Downloading [=>          ] 131.1kB/4.224MB
0c7ff7dbf73bd: Download complete
095e60bae1c: Download complete
5d7ef5dedb8e: Waiting
978f9bcfbf3eb: Waiting
705807c02638: Waiting
3c631ec4c979: Waiting
061dd099b2ba: Waiting
cc4c6a09928c: Waiting
75c3bd976576: Waiting
4f4fb700ef54: Waiting
dee1bf57e253: Waiting
0eb92cc49677: Waiting
74d8a97bb60c: Waiting
5e59b25b0c46: Waiting
bda09bffffbd8: Waiting
803d8813d4c6: Waiting
d183a52dadcf: Waiting
7ea3357b4fb: Waiting
isma@isma-VirtualBox:~$
```

- Despues de un rato termina de descargar todo lo necesario:



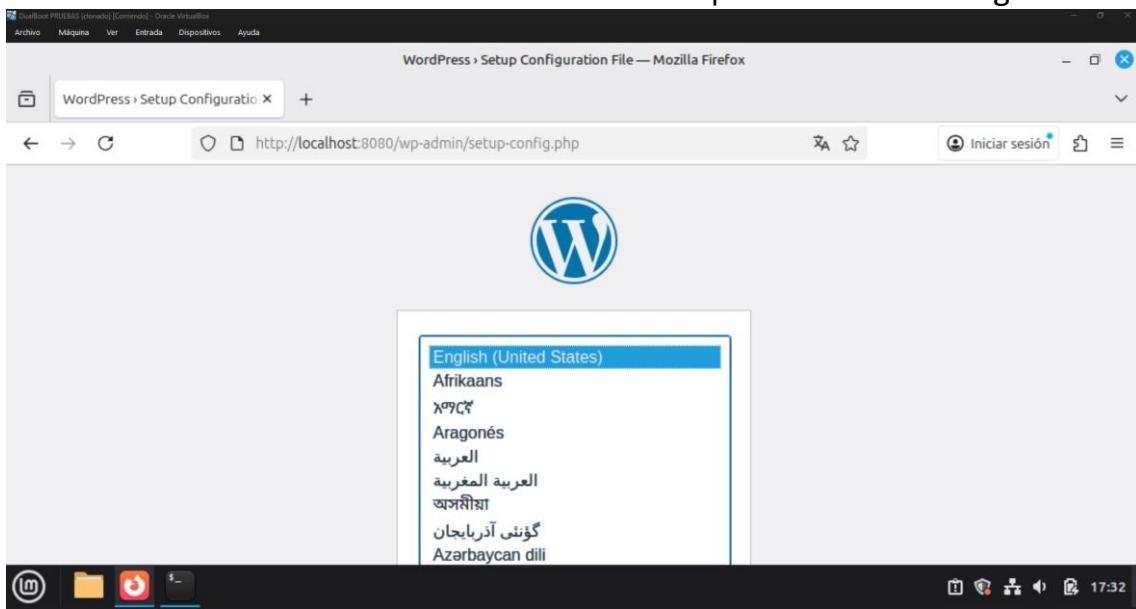
```
96dfba1a7aa9: Download complete
3262e3d480fc: Pull complete
96dfba1a7aa9: Pull complete
38a62f60c0ae: Pull complete
0c7ff7dbf73bd: Pull complete
095e60bae1c: Pull complete
5d7ef5dedb8e: Pull complete
978f9bcfbf3eb: Pull complete
705807c02638: Pull complete
3c631ec4c979: Pull complete
061dd099b2ba: Pull complete
cc4c6a09928c: Pull complete
75c3bd976576: Pull complete
4f4fb700ef54: Pull complete
dee1bf57e253: Pull complete
0eb92cc49677: Pull complete
74d8a97bb60c: Pull complete
5e59b25b0c46: Pull complete
bda09bffffbd8: Pull complete
803d8813d4c6: Pull complete
d183a52dadcf: Pull complete
7ea3357b4fb: Pull complete
28fc03424957: Pull complete
Digest: sha256:60d358e9c0a51616128836f1fe6b948f32d926c878ff431eeb1ce6ce24c1259d
Status: Downloaded newer image for wordpress:latest
c98b12db163f287913f24169076c43d5cad0f750ccab5cd535497eaa8bcfe7a6
isma@isma-VirtualBox:~$
```

- Hemos reiniciado nuestro sistema y hemos arrancado los dos contenedores creados:

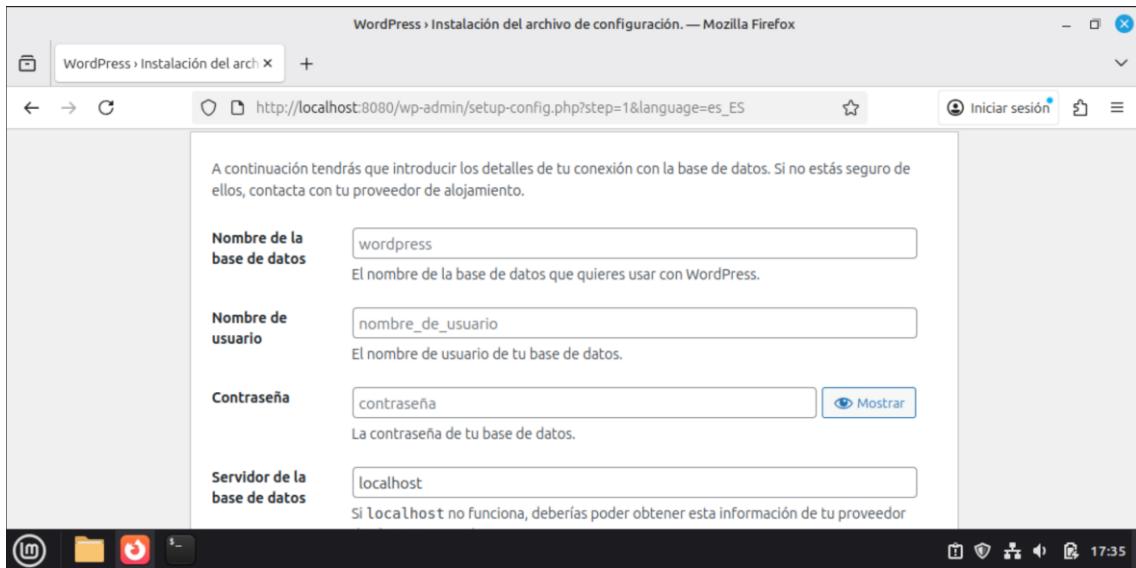


```
isma@isma-VirtualBox:~$ docker start nuestromariadb
nuestromariadb
isma@isma-VirtualBox:~$ docker start nuestrowp
nuestrowp
isma@isma-VirtualBox:~$
```

- Vamos continuar la instalación de Wordpress en el navegador:



- Seleccionamos el idioma que queramos, en este caso será el español, y ahora tendremos que configurar, el nombre del servidor y demás parámetros:



- Tendremos que seguir con la instalación e indicar el sistema de base de datos, así que usaremos como nombre de host, el nombre que le dimos al servidor MariaDB.

El usuario y nombre de base de datos lo utilizaremos con las variables proporcionadas al crear los contenedores en este caso en MariaDB. A continuación dejo una tabla a llenar y la imagen como sería:

Campo	Valor
Nombre de la base de datos	cefiredb
Nombre de usuario	cefireuser
Contraseña	cefirepass
Servidor de la base de datos	nuestromariadb
Prefijo de tabla	wp_

WordPress > Instalación del archivo de configuración. — Mozilla Firefox

WordPress > Instalación del arch: x +

http://localhost:8080/wp-admin/setup-config.php?step=1&language=es_ES

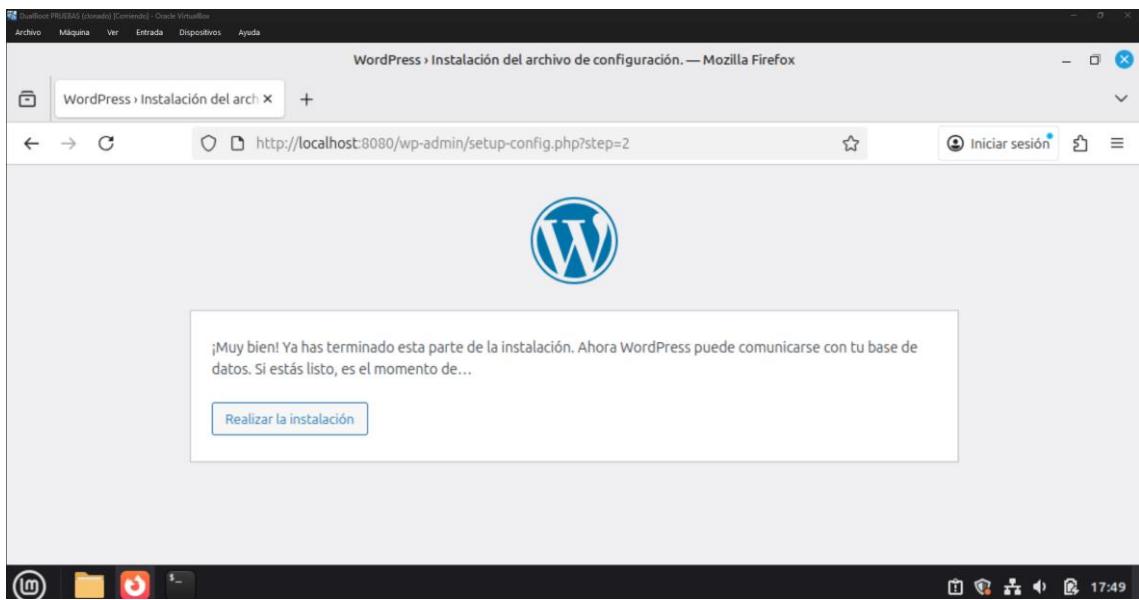
Nombre de la base de datos: cefiredb
El nombre de la base de datos que quieras usar con WordPress.

Nombre de usuario: cefireuser
El nombre de usuario de tu base de datos.

Contraseña: cefirepass Ocultar
La contraseña de tu base de datos.

Servidor de la base de datos: nuestromariadb
Si localhost no funciona, deberías poder obtener esta información de tu proveedor de alojamiento web.

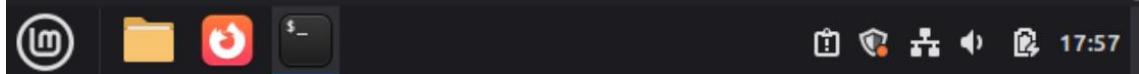
Prefijo de tabla: wp_
Si quieras ejecutar varias instalaciones de WordPress en una sola base de datos cambia



- Ya podremos dar a “Realización la instalación”.

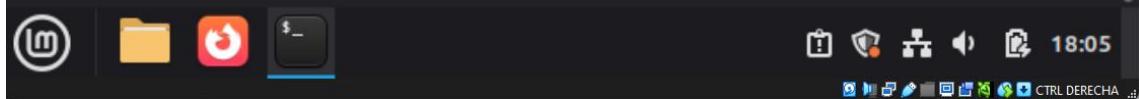
5. Ahora migraremos el Contenedor MariaDB de 10.6 a 10.7, primero pararemos el docker “**docker stop nuestromariadb**”, luego eliminamos con “**docker rm nuestromariadb**”

```
isma@isma-VirtualBox:~$ docker stop nuestromariadb  
nuestromariadb  
isma@isma-VirtualBox:~$ docker rm nuestromariadb
```

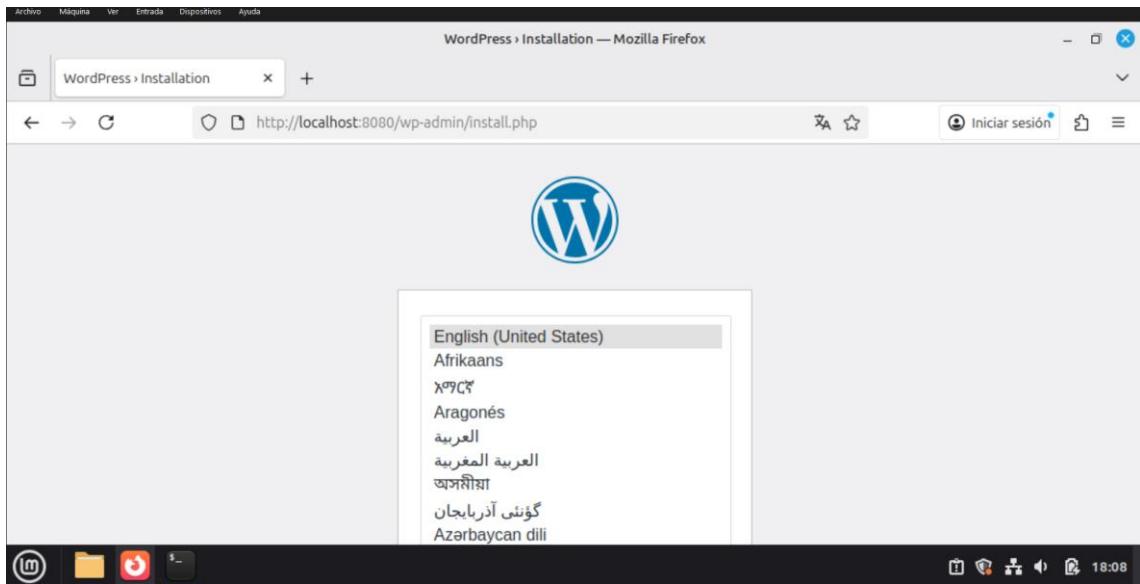


- Una vez que hemos parado y eliminado, podremos recordar que la información no la hemos perdido pues la información como la configuración y base de datos se había quedado almacenada en la carpeta del anfitrión “**/home/sergi/mariadbdata**”. Al crear un nuevo contenedor, simplemente deberemos mapear esos datos:

```
isma@isma-VirtualBox:~$ docker run --name nuestromariadb --network redwp -v /h  
ome/sergi/mariadbdata:/var/lib/mysql -e MARIADB_ROOT_PASSWORD=cefireroot -e MA  
RIADB_USER=cefireuser -e MARIADB_PASSWORD=cefirepass -e MARIADB_DATABASE=cefir  
edb -d mariadb:10.7  
7cfbab97f098d892222d4ee54d6f97bf969aa9e826d728b8c92b90289c83023f  
isma@isma-VirtualBox:~$
```



- Volvemos entrar en <http://localhost:8080/>



- Vemos que funciona bien.