

# Ejercicio practico UD9 :

## Introducción a Docker



Alex Godoy

## Índice

Introducción.....	3
Objetivos.....	3
Material utilizado del centro:.....	4
Desarrollo:.....	5
Actividades : Introducción a Docker.....	5
Parte II : Introducción a Docker (II).....	5
Parte III : Introducción a Docker (II).....	10
Tabla de tiempos:.....	16
Problemas encontrados , solución y/o sugerencias.....	17
Conclusión.....	18
Bibliografía / webgrafia.....	19

## Introducción

Docker es una herramienta que nos permite desarrollar, enviar y ejecutar aplicaciones dentro de contenedores. Estos contenedores crean un entorno aislado y controlado, ideal para ejecutar software de manera eficiente y segura.

Configurar Docker y hacer nuestros contenedores iniciales es una parte esencial al aprender sobre Sistemas Informáticos. Mediante este ejercicio, sentaremos los cimientos para gestionar herramientas y tecnologías más sofisticadas que usaremos en el futuro.

## Objetivos

- **Verificar la instalación de Docker:** Verificar que Docker esté instalado correctamente en tu ordenador y que puedas usarlo a través de la terminal.
- **Conocer la versión de Docker:** Verificar el funcionamiento correcto de Docker ejecutando un contenedor básico que muestre un mensaje, evidenciando así la capacidad de Docker para descargar imágenes desde Docker Hub y ejecutar contenedores sin inconvenientes.
- **Ejecutar un contenedor básico:** Verificar el funcionamiento correcto de Docker al correr un contenedor básico que muestre un mensaje, demostrando así la capacidad de Docker para descargar imágenes de Docker Hub y ejecutar contenedores sin dificultades.
- **Familiarizarse con comandos básicos de Docker:** Ejercitar la utilización de comandos simples de Docker (tales como 'docker version' y 'docker run hello-world') es esencial para poder llevar a cabo tareas más complejas con contenedores en el futuro.

## Material utilizado del centro:

- Ordenador completo en funcionamiento que cada alumno se responsabiliza.

Indicar características principales del equipo (benchmark):

COMPONENTES DE TU EQUIPO	VALOR
Marca y modelo de Procesador	Intel Pentium CPU G4400 @ 3.30GHz
Capacidad Memoria RAM	4GB RAM, 2400 MHz
Tipos de dispositivos de almacenamiento y capacidad	HDD 1TB

- Programas necesarios para realizar las prácticas:

1. VirtualBox (ejecución MV con el SOLubuntu)
2. Spectacle (para capturar pantallas)

Material que debe aportar el alumno:

- Memoria USB formateada de mínimo 32GB en caso de necesidad
- Disco duro SSD / HDD mínimo 500GB

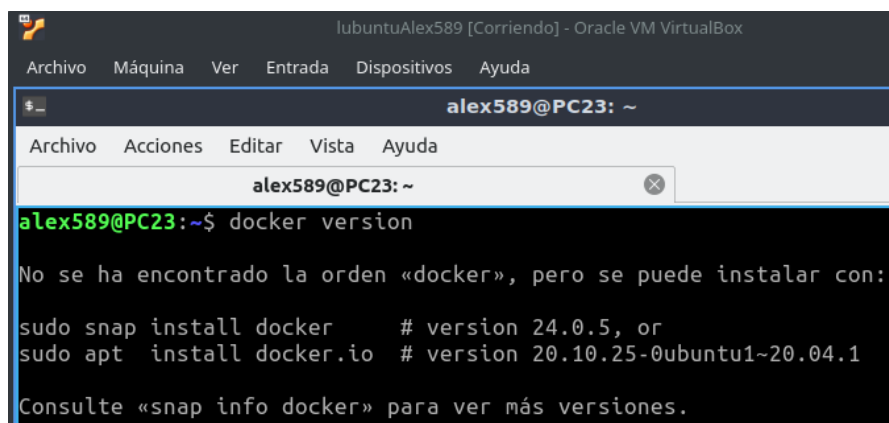
## Desarrollo:

### Actividades : Introducción a Docker

#### Parte II : Introducción a Docker (II)

1. Ejecuta en tu máquina con la instalación completa de Docker el siguiente comando:

**docker version**



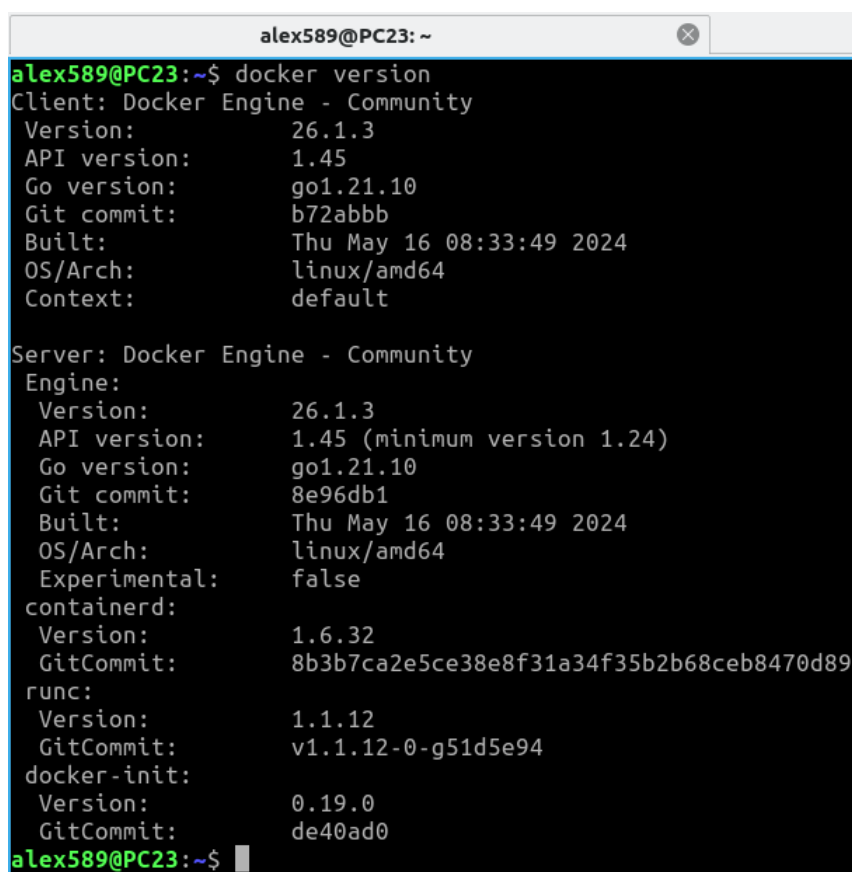
```
alex589@PC23:~$ docker version

No se ha encontrado la orden «docker», pero se puede instalar con:

sudo snap install docker      # version 24.0.5, or
sudo apt install docker.io    # version 20.10.25-0ubuntu1~20.04.1

Consulte «snap info docker» para ver más versiones.
```

Figura 1: Podemos observar que el docker no esta instalado



```
alex589@PC23:~$ docker version
Client: Docker Engine - Community
 Version:           26.1.3
 API version:       1.45
 Go version:        go1.21.10
 Git commit:        b72abbb
 Built:             Thu May 16 08:33:49 2024
 OS/Arch:           linux/amd64
 Context:           default

Server: Docker Engine - Community
 Engine:
  Version:           26.1.3
  API version:       1.45 (minimum version 1.24)
  Go version:        go1.21.10
  Git commit:        8e96db1
  Built:             Thu May 16 08:33:49 2024
  OS/Arch:           linux/amd64
  Experimental:      false
 containerd:
  Version:           1.6.32
  GitCommit:        8b3b7ca2e5ce38e8f31a34f35b2b68ceb8470d89
 runc:
  Version:           1.1.12
  GitCommit:        v1.1.12-0-g51d5e94
 docker-init:
  Version:           0.19.0
  GitCommit:        de40ad0
alex589@PC23:~$
```

Figura 2: Podemos observar que el docker esta instalado

## Instalacion del docker

### 1. Actualizar el índice de paquetes:

`sudo apt update`

```
alex589@PC23:~$ sudo apt update
[sudo] contraseña para alex589:
Obj:1 http://archive.ubuntu.com/ubuntu focal InRelease
```

### 2. Instalar dependencias necesarias para permitir que apt utilice un repositorio sobre HTTPS:

`sudo apt install apt-transport-https ca-certificates curl software-properties-common`

```
alex589@PC23:~$ sudo apt install apt-transport-https ca-certificates curl software-properties-common
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  libcurl4 python3-software-properties software-properties-qt
Se instalarán los siguientes paquetes NUEVOS:
  apt-transport-https curl
```

### 3. Añadir la clave GPG oficial de Docker:

`curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -`

```
alex589@PC23:~$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
OK
alex589@PC23:~$
```

### 4. Añadir el repositorio de Docker a las fuentes de apt:

`sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"`

```
alex589@PC23:~$ sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"
Obj:1 http://archive.ubuntu.com/ubuntu focal InRelease
Obj:2 http://archive.ubuntu.com/ubuntu focal-updates InRelease
Obj:3 http://archive.ubuntu.com/ubuntu focal-backports InRelease
Obj:4 http://security.ubuntu.com/ubuntu focal-security InRelease
Des:5 https://download.docker.com/linux/ubuntu focal InRelease [57,7 kB]
Des:6 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages [43,6 kB]
Descargados 101 kB en 1s (160 kB/s)
Leyendo lista de paquetes... Hecho
```

### 5. Actualizar el índice de paquetes una vez más:

`sudo apt update`

```
alex589@PC23:~$ sudo apt update
```

### 6. Instalar la versión comunitaria de Docker:

`sudo apt install docker-ce`

```
alex589@PC23:~$ sudo apt install docker-ce
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
```

### 7. Reiniciar tu sistema:

`sudo reboot`





## Parte III : Introducción a Docker (II)

### 1. Captura final del navegador accediendo a la interfaz gráfica mediante el cliente NoVNC. "CASO PRACTICO 3"

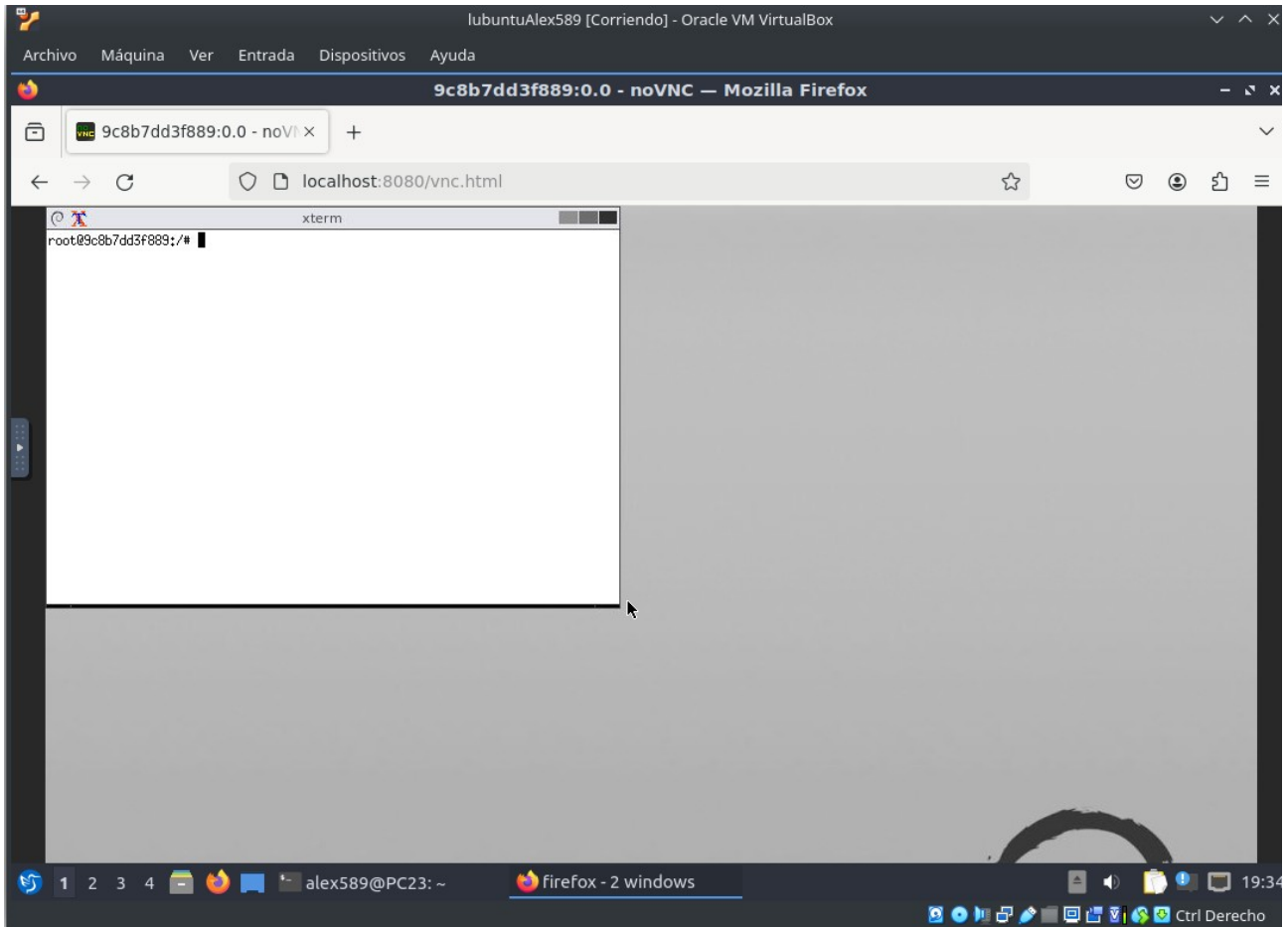


Figura 4: Podemos observar el navegador accediendo a la interfaz gráfica mediante el cliente NoVNC.

## Parte IV : Introducción a Docker (IV)

Tendrás que entregar en la tarea un documento que incluya un texto donde se indique la URL de tu repositorio público creado y después de esto, capturas demostrando la realización del otro caso práctico seleccionado.

### 1. Captura Creando imagen con APP en Nodo. "CASO PRACTICO 4"

```
alex589@PC23:~$ docker login
Log in with your Docker ID or email address to push and pull images from Docker Hub. If you
have a Docker ID, head over to https://hub.docker.com/ to create one.
You can log in with your password or a Personal Access Token (PAT). Using a limited-scope
token is better security and is required for organizations using SSO. Learn more at https://docs.docker.com/go/access-tokens/

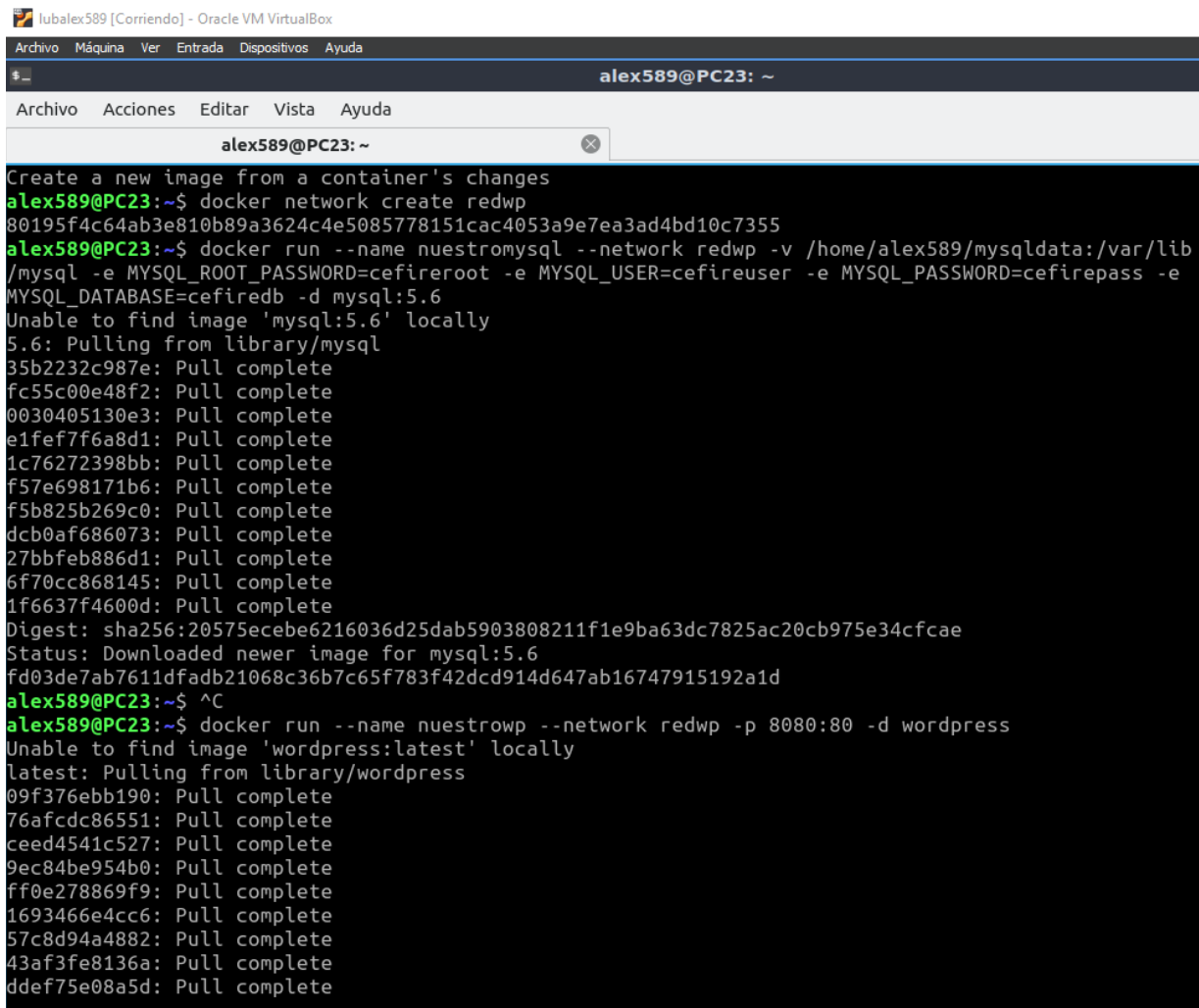
Username: herrerosalex
Password:
WARNING! Your password will be stored unencrypted in /home/alex589/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
```

Figura 5: Accediendo al docker

## Parte V : Introducción a Docker (V)

1. Adjunta una captura donde se muestre que Wordpress funciona y otra captura donde se pueda observar que los dos contenedores están funcionando (con "docker ps"). "CASO PRACTICO 5"

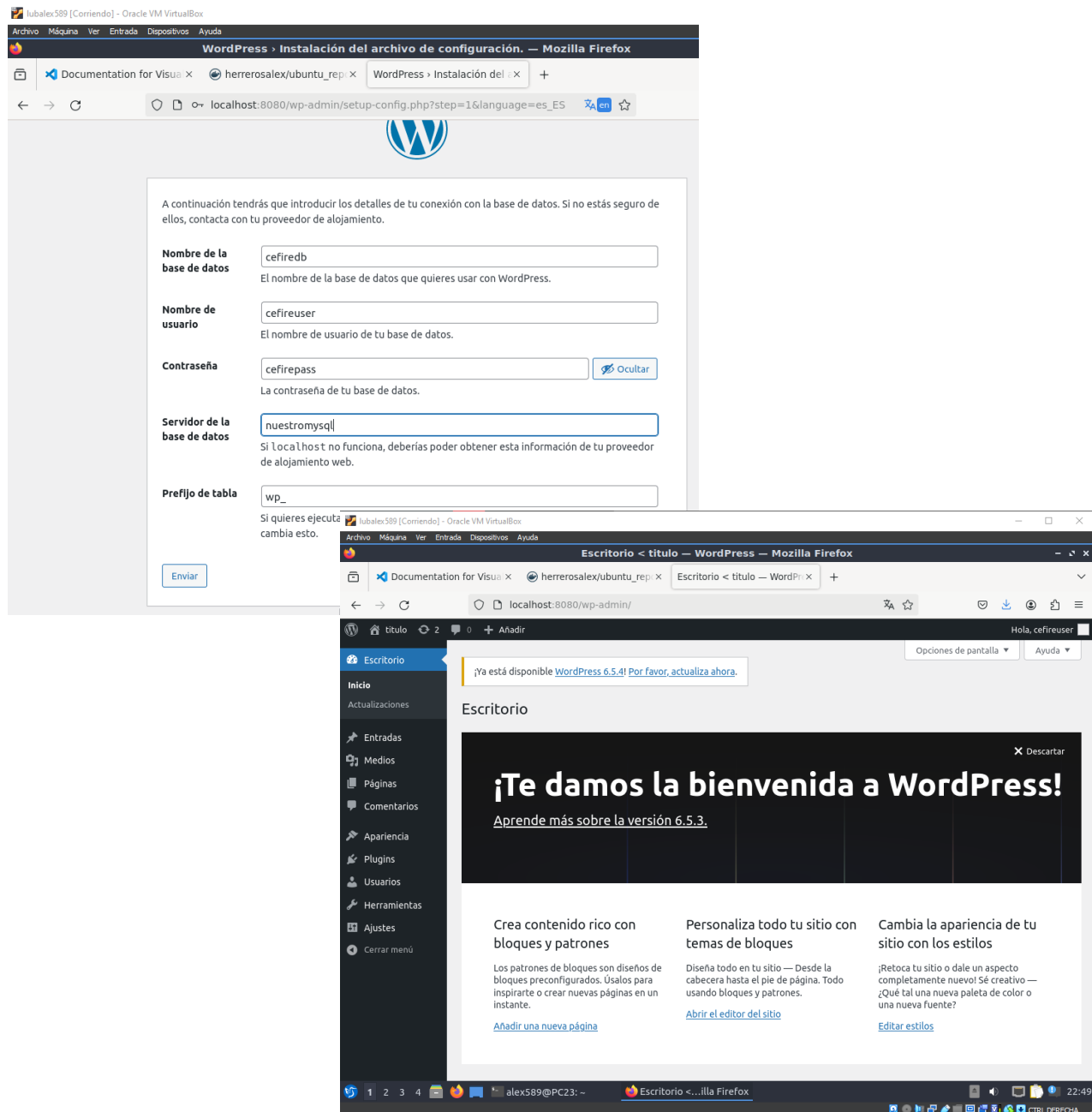


```

|ubalex589 [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
alex589@PC23: ~
Archivo  Acciones  Editar  Vista  Ayuda
alex589@PC23: ~
Create a new image from a container's changes
alex589@PC23:~$ docker network create redwp
80195f4c64ab3e810b89a3624c4e5085778151cac4053a9e7ea3ad4bd10c7355
alex589@PC23:~$ docker run --name nuestromysql --network redwp -v /home/alex589/mysqldata:/var/lib
/mysql -e MYSQL_ROOT_PASSWORD=cefireroot -e MYSQL_USER=cefireuser -e MYSQL_PASSWORD=cefirepass -e
MYSQL_DATABASE=cefiredb -d mysql:5.6
Unable to find image 'mysql:5.6' locally
5.6: Pulling from library/mysql
35b2232c987e: Pull complete
fc55c00e48f2: Pull complete
0030405130e3: Pull complete
e1fef7f6a8d1: Pull complete
1c76272398bb: Pull complete
f57e698171b6: Pull complete
f5b825b269c0: Pull complete
dcb0af686073: Pull complete
27bbfeb886d1: Pull complete
6f70cc868145: Pull complete
1f6637f4600d: Pull complete
Digest: sha256:20575e3e3e6216036d25dab5903808211f1e9ba63dc7825ac20cb975e34cfcae
Status: Downloaded newer image for mysql:5.6
fd03de7ab7611dfadb21068c36b7c65f783f42dcd914d647ab16747915192a1d
alex589@PC23:~$ ^C
alex589@PC23:~$ docker run --name nuestrowp --network redwp -p 8080:80 -d wordpress
Unable to find image 'wordpress:latest' locally
latest: Pulling from library/wordpress
09f376ebb190: Pull complete
76afcdc86551: Pull complete
ceed4541c527: Pull complete
9ec84be954b0: Pull complete
ff0e278869f9: Pull complete
1693466e4cc6: Pull complete
57c8d94a4882: Pull complete
43af3fe8136a: Pull complete
ddef75e08a5d: Pull complete

```

Figura 6: Podemos observar que los contenedores están funcionando

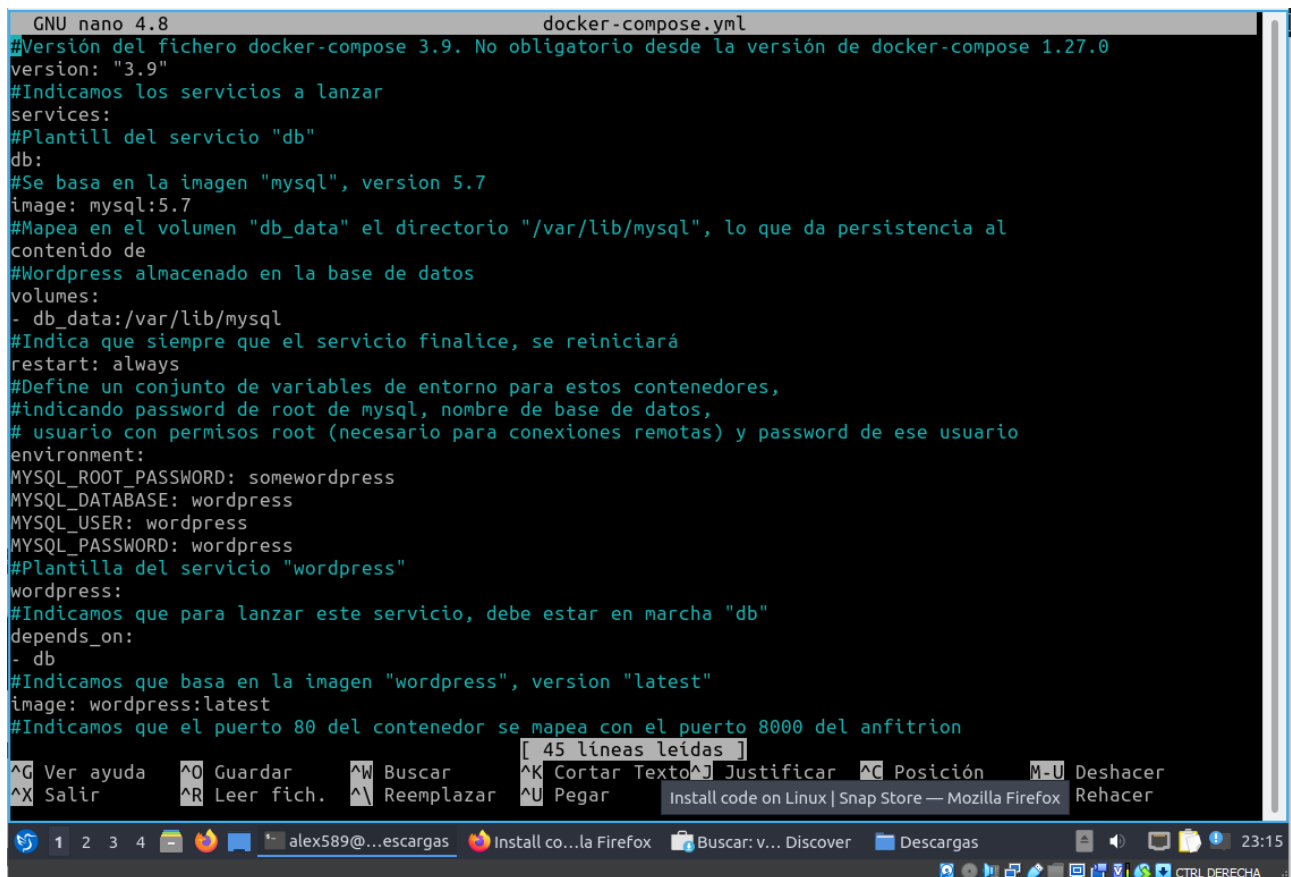


```
alex589@PC23:~$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS
c3290ff73e4d   wordpress     "docker-entrypoint.s..." 6 minutes ago  Up 6 minutes  0.0.0.0:8080->80/tcp, :::8080->80/tcp
fd03de7ab761   mysql:5.6     "docker-entrypoint.s..." 8 minutes ago  Up 7 minutes  3306/tcp
nuestrormysql
```

Figura 7: Imagen con el resultado

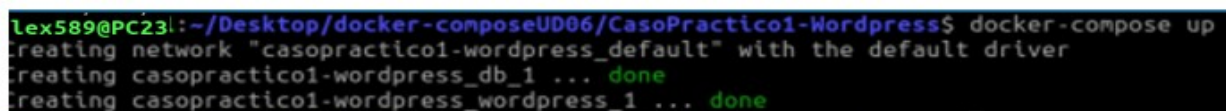
## Parte VI : Introducción a Docker (VI)

### 1. Adjunta una captura donde se muestre que Wordpress este con Docker Compose "CASO PRACTICO 6"



```
GNU nano 4.8 docker-compose.yml
#Versión del fichero docker-compose 3.9. No obligatorio desde la versión de docker-compose 1.27.0
version: "3.9"
#Indicamos los servicios a lanzar
services:
#Plantilla del servicio "db"
db:
#Se basa en la imagen "mysql", version 5.7
image: mysql:5.7
#Mapea en el volumen "db_data" el directorio "/var/lib/mysql", lo que da persistencia al
contenido de
#Wordpress almacenado en la base de datos
volumes:
- db_data:/var/lib/mysql
#Indica que siempre que el servicio finalice, se reiniciará
restart: always
#Define un conjunto de variables de entorno para estos contenedores,
#indicando password de root de mysql, nombre de base de datos,
# usuario con permisos root (necesario para conexiones remotas) y password de ese usuario
environment:
MYSQL_ROOT_PASSWORD: somewordpress
MYSQL_DATABASE: wordpress
MYSQL_USER: wordpress
MYSQL_PASSWORD: wordpress
#Plantilla del servicio "wordpress"
wordpress:
#Indicamos que para lanzar este servicio, debe estar en marcha "db"
depends_on:
- db
#Indicamos que basa en la imagen "wordpress", version "latest"
image: wordpress:latest
#Indicamos que el puerto 80 del contenedor se mapea con el puerto 8000 del anfitrión
[ 45 líneas leídas ]
^G Ver ayuda ^O Guardar ^W Buscar ^K Cortar Texto ^J Justificar ^C Posición M-U Deshacer
^X Salir ^R Leer fich. ^\ Reemplazar ^U Pegar Install code on Linux | Snap Store — Mozilla Firefox Rehacer
```

Figura 8: Podemos observar el archivo creado



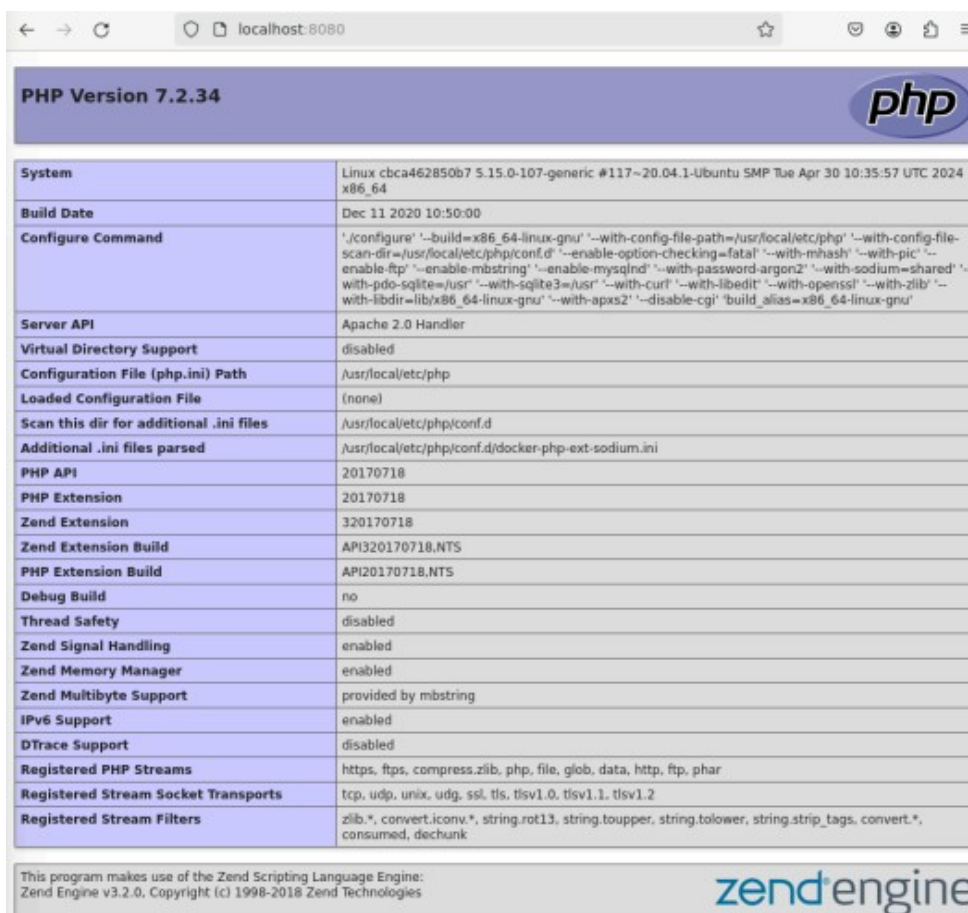
```
alex589@PC23: ~/Desktop/docker-composeUD06/CasoPractico1-Wordpress$ docker-compose up
Creating network "casopractico1-wordpress_default" with the default driver
Creating casopractico1-wordpress_db_1 ... done
Creating casopractico1-wordpress_wordpress_1 ... done
```

Figura 9: Ejecutamos el comando y se puede observar como ha funcionado correctamente

## Parte VII : Introducción a Docker (VII)

### 1. Adjunta una captura donde se muestre que “Visual Studio Code” está conectado a un contenedor. "CASO PRACTICO 7"

```
alex589@PC2:~$ docker run -d --name servidordesarrollo -p 8080:80 php:7.2-apache
Unable to find image 'php:7.2-apache' locally
7.2-apache: Pulling from library/php
6ec7b7d162b2: Pull complete
4b606474d60c: Pull complete
afb30f0cd8e0: Pull complete
3bb2e8051594: Pull complete
4c761b44e2cc: Pull complete
c2199db96575: Pull complete
1b9a9381eea8: Pull complete
fd07bbc59d34: Pull complete
72b73ab27698: Pull complete
983308f4f0d6: Pull complete
6c13f026e6da: Pull complete
e5e6cd163689: Pull complete
5c5516e56582: Pull complete
154729f6ba86: Pull complete
Digest: sha256:4dc0f0115acf8c2f0df69295ae822e49f5ad5fe849725847f15aa0e5802b55f8
Status: Downloaded newer image for php:7.2-apache
cbca462850b7b1618c189d253c3aefaa8a46c7ec8a8a92608c81b3778ce9e25b
```



PHP Version 7.2.34

System	Linux cbca462850b7 5.15.0-107-generic #117~20.04.1-Ubuntu SMP Tue Apr 30 10:35:57 UTC 2024 x86_64
Build Date	Dec 11 2020 10:50:00
Configure Command	'./configure' '--build=x86_64-linux-gnu' '--with-config-file-path=/usr/local/etc/php' '--with-config-file-scan-dir=/usr/local/etc/php/conf.d' '--enable-option-checking=fatal' '--with-mhash' '--with-pic' '--enable-ftp' '--enable-mbstring' '--enable-mysqlnd' '--with-password-argon2' '--with-sodium=shared' '--with-pdo-sqlite=/usr' '--with-sqlite3=/usr' '--with-curl' '--with-libedit' '--with-openssl' '--with-zlib' '--with-libdir=lib/x86_64-linux-gnu' '--with-apxs2' '--disable-cgi' 'build_alias=x86_64-linux-gnu'
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/usr/local/etc/php
Loaded Configuration File	(none)
Scan this dir for additional .ini files	/usr/local/etc/php/conf.d
Additional .ini files parsed	/usr/local/etc/php/conf.d/docker-php-ext-sodium.ini
PHP API	20170718
PHP Extension	20170718
Zend Extension	320170718
Zend Extension Build	API320170718.NTS
PHP Extension Build	API(20170718.NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled
Zend Memory Manager	enabled
Zend Multibyte Support	provided by mbstring
IPv6 Support	enabled
DTrace Support	disabled
Registered PHP Streams	https, ftps, compress.zlib, php, file, glob, data, http, ftp, phar
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, tls, tlsv1.0, tlsv1.1, tlsv1.2
Registered Stream Filters	zlib.*, convert.iconv.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, dechunk

This program makes use of the Zend Scripting Language Engine:  
Zend Engine v3.2.0, Copyright (c) 1998-2018 Zend Technologies

zend engine

Figura 10: Podemos observar que se ha conectado perfectamente

Tabla de tiempos:

ACTIVIDADES	TIEMPO ESTIMADO	TIEMPO REAL
1	5 HORAS	6 HORAS

## Problemas encontrados , solución y/o sugerencias

Algún ejercicio me ha tocado hacerlo varias veces ya que no me salia bien y / o el comando no lo ponía bien, con los documentos del profesor me he podido ir apañando para hacer la actividad.



## Conclusión

La actividad me ha resultado entretenida, ya que he aprendido sobre los Docker y las funciones, aunque algún ejercicio se me ha resistido y he tenido que repetir alguno para que me funcionara, pero buscando en los apuntes lo he podido solucionar.

Ha sido interesante ya que estaba relacionada con la UD9.

## **Bibliografia / webgrafia**

-Documentación aportada por el profesor del modulo.