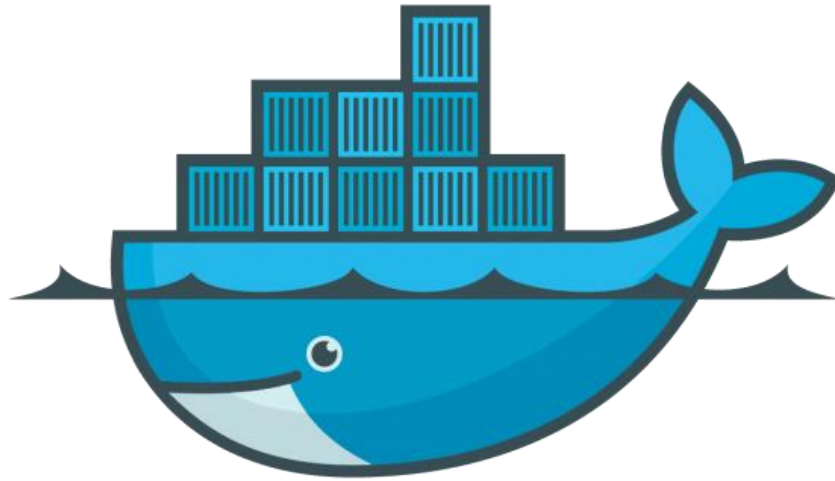


21 DE SEPTIEMBRE DE 2021



docker

UD01 – INTRODUCCIÓN A DOCKER

ACTIVIDADES EVALUABLES 01

VICENTE IZQUIERDO FORMENT

IES SERRA PERENXISA
2º DAMS

Índice

Actividad 01-----2

Parte 03-----2

Parte 04-----3

Parte 05-----4

Parte 06-----5

Parte 07-----8

Actividad 02-----9

Actividad 03-----9

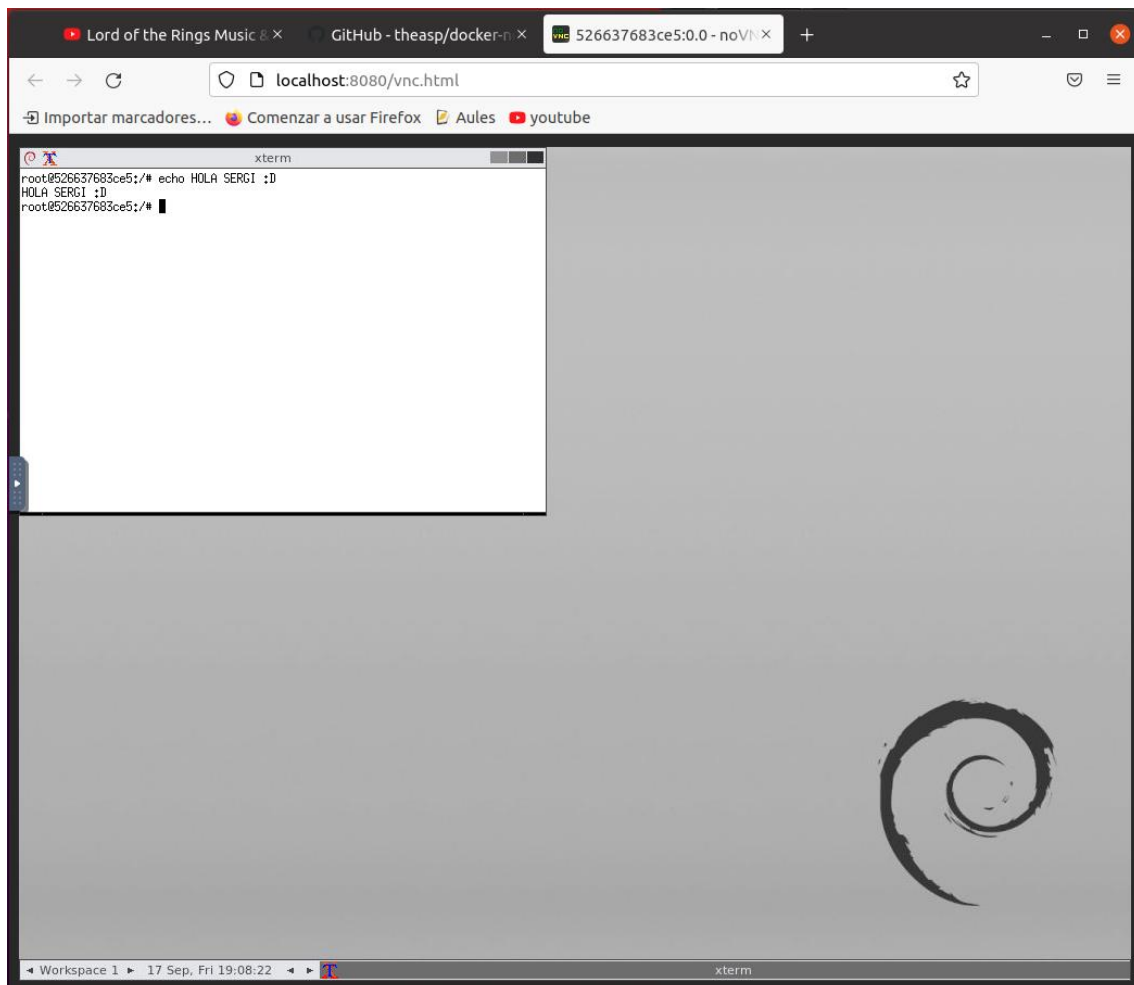
Actividad 01.- Casos prácticos parte 03 – parte 07:

3.01 - Adjunta una captura donde se observe que estás dentro del contenedor y se han creado 10 carpetas en “/root”. Tras ello, adjunta una captura donde se observe que se ha ejecutado el script propuesto para la corrección automática.

```
vicente@vicente-desktop:~$ docker exec -it entrega bash
root@d45b711da9d5:/# ls /root
01 02 03 04 05 06 07 08 09 10
root@d45b711da9d5:/#
```

```
vicente@vicente-desktop:~$ docker start entrega
entrega
vicente@vicente-desktop:~$ sh ./comprobar.sh
PRACTICA OK
```

3.03 - Captura final del navegador accediendo a la interfaz gráfica mediante el cliente NoVNC.



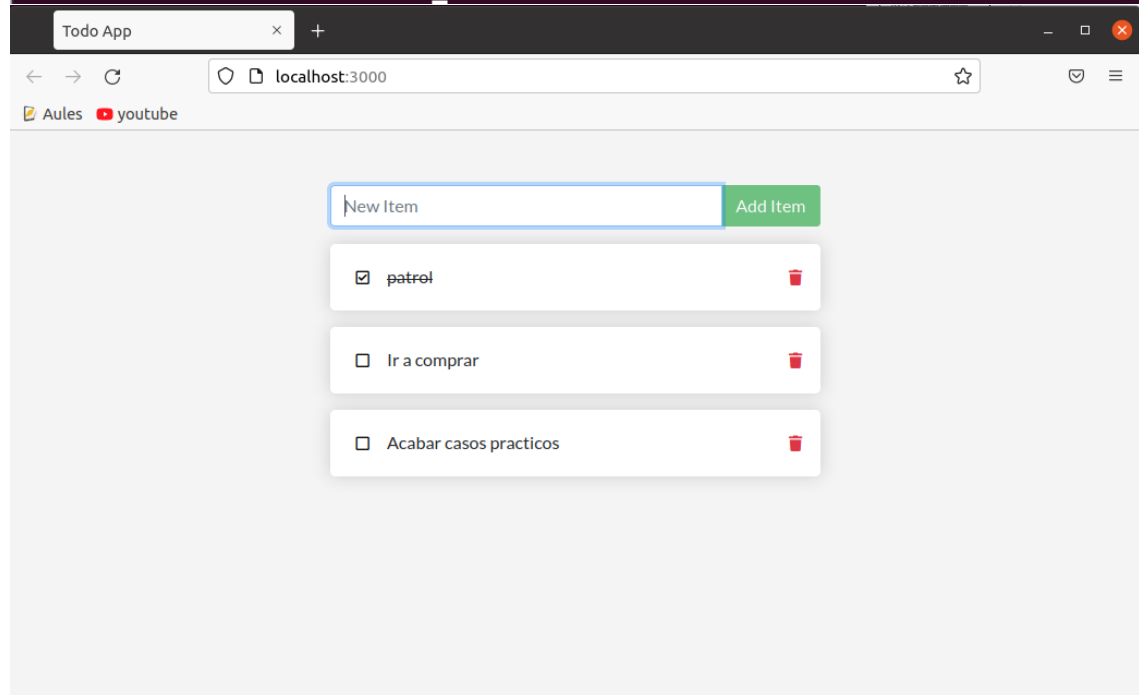
4.01 - En esta unidad debes elegir uno de los 3 casos prácticos, preparar la imagen con el “*Dockerfile*” y compartirla públicamente en tu repositorio de “*Docker Hub*”.

URL:

<https://hub.docker.com/repository/docker/vif30/ud1entregables>

He elegido el caso práctico 02, en el que creamos una imagen con una aplicación a la que accedemos a través de <http://localhost:3000>.

```
vicente@vicente-desktop:~$ docker run -dp 3000:3000 vif30/ud1entregables  
f5cfff242b57e0dbf11cef51bff797ae3a58c9dee978c602676f1597ba971980f
```

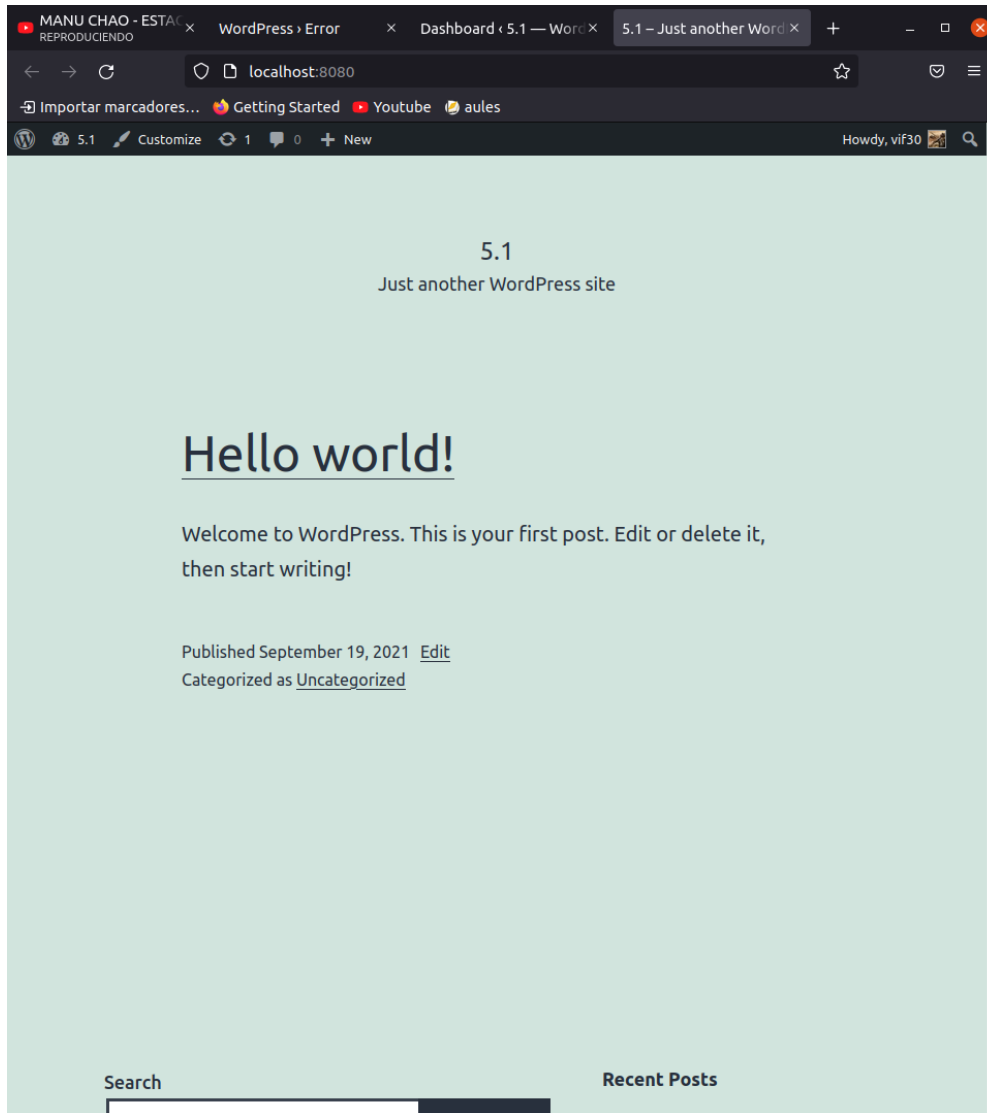


5.01 - Adjunta una captura donde se muestre que Wordpress funciona y otra captura donde se pueda observar que los dos contenedores están funcionando (con “*docker ps*”).

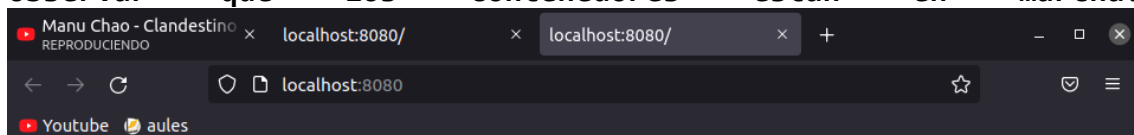
```

vicente@vicente-desktop:~$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                               NAMES
45dafef4dfb7   mysql:5.7      "docker-entrypoint.s..." 36 seconds ago Up 34 seconds 3306/tcp, 33060/tcp               nuestromysql
3c9d5806e55b   wordpress     "docker-entrypoint.s..." 4 minutes ago  Up 4 minutes  0.0.0.0:8080->80/tcp, :::8080->80/tcp nuestrowp

```



5.02 - Adjunta una captura donde se muestre que el balanceo de carga funciona correctamente y otra captura donde se pueda observar que los contenedores están en marcha.



Servido desde volumen, por cualquiera de nuestros servidores

```

vicente@vicente-desktop:~/Documentos/UD1/CAS02/UD05-CasoPractico02-HAProxy/HAProxy/apachevolumen$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                               NAMES
30762f0fab96   httpd:2.4      "httpd-foreground"      50 seconds ago Up 49 seconds 80/tcp               ap02
b6a3d517f564   miapachevolumen "httpd-foreground"      2 minutes ago  Up 2 minutes  80/tcp               ap01
f544b1f5a18e   mihaproxy      "docker-entrypoint.s..." 7 minutes ago  Up 7 minutes  0.0.0.0:8080->80/tcp, :::8080->80/tcp ha01

```

5.03 - Adjunta una captura donde se muestre como se ha accedido a MySQL y se ha hecho la consulta de prueba de rendimiento propuestas.

```
vicente@vicente-desktop:~$ docker exec -it mysqlcontmpfs bash
root@7f6557d9c81a:/# mysql -u root
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 1
Server version: 5.6.51 MySQL Community Server (GPL)

Copyright (c) 2000, 2021, 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.

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

mysql> USE test;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> SELECT SQL_NO_CACHE * FROM posts;
```

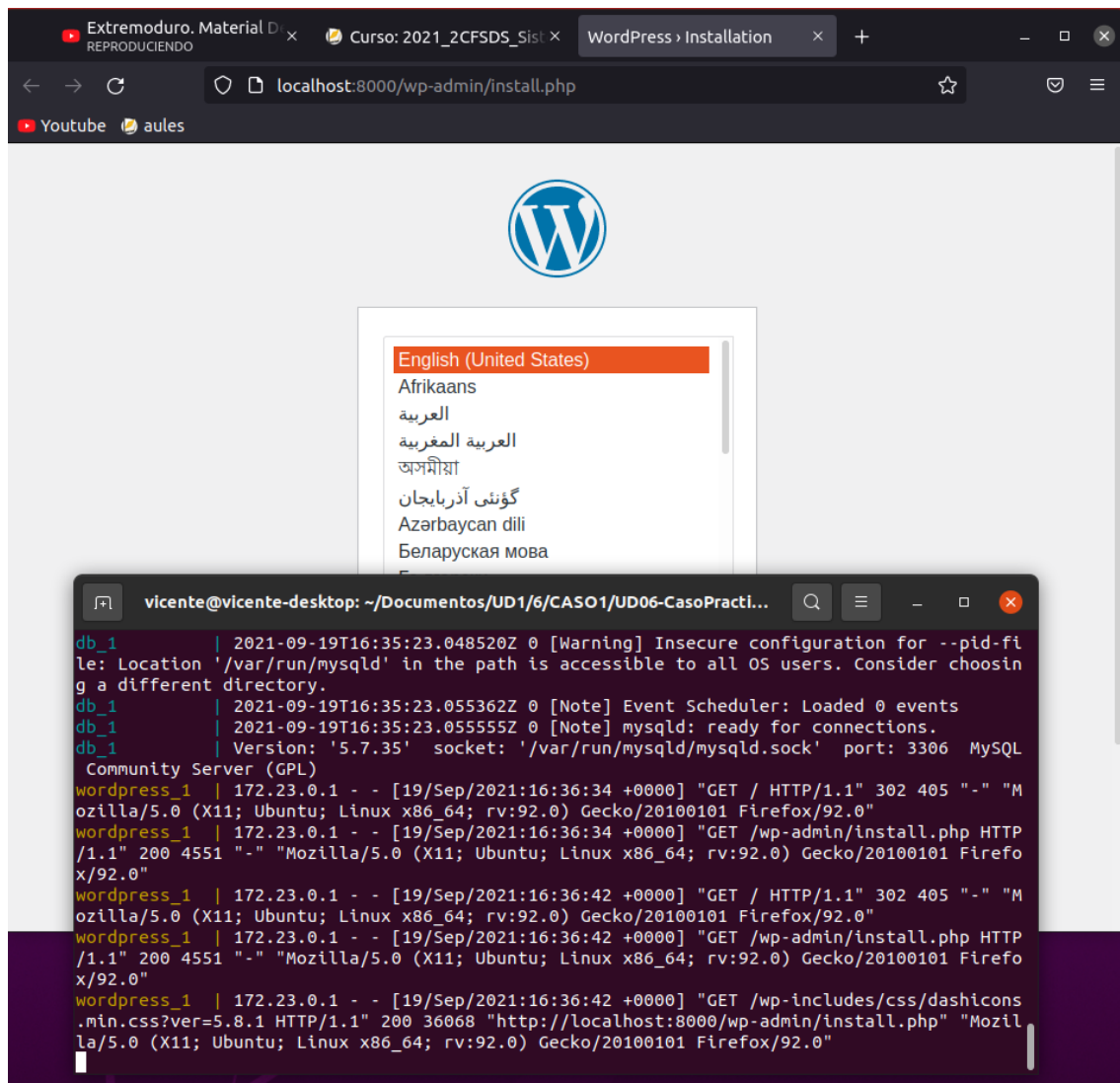
[illegible]

6.01 - Adjunta una captura donde se muestre que has lanzado “*Docker Compose*” y Wordpress funciona tras ello.

```

vicente@vicente-desktop:~/Documentos/UD1/6/CAS01/UD06-CasoPractico01-Wordpress/CasoPractico1-Wordpress$ docker-compose up
Creating network "casopractico1-wordpress_default" with the default driver
Creating volume "casopractico1-wordpress_db_data" with default driver
Creating casopractico1-wordpress_db_1 ... done
Creating casopractico1-wordpress_wordpress_1 ... done
Attaching to casopractico1-wordpress_db_1, casopractico1-wordpress_wordpress_1
db_1      | 2021-09-19 16:35:14+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 5.7.35-1debian10 started
wordpress_1 | WordPress not found in /var/www/html - copying now...
db_1      | 2021-09-19 16:35:14+00:00 [Note] [Entrypoint]: Switching to dedicated user 'mysql'
db_1      | 2021-09-19 16:35:14+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 5.7.35-1debian10 started
db_1      | 2021-09-19 16:35:14+00:00 [Note] [Entrypoint]: Initializing database files

```



6.02 – Adjunta una captura donde se muestre que has lanzado “*Docker Compose*”, creado la aplicación Django y que dicha aplicación está funcionando.

```
vicente@vicente-desktop:~/Documentos/UD1/6/CASO2/UD06-CasoPractico02-Django/CasoPractico2-Django$ docker-compose build
db uses an image, skipping
Building web
Sending build context to Docker daemon 5.12kB

Step 1/5 : FROM python:3
3: Pulling from library/python
955615a668ce: Pulling fs layer
2756ef5f69a5: Pulling fs layer
911ea9f2bd51: Pulling fs layer
```

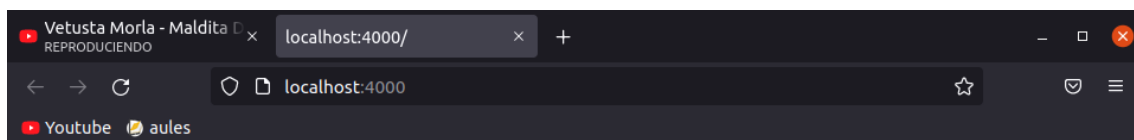
```
vicente@vicente-desktop:~/Documentos/UD1/6/CASO2/UD06-CasoPractico02-Django/CasoPractico2-Django$ docker-compose pull
Pulling db ... done
Pulling web ... done
vicente@vicente-desktop:~/Documentos/UD1/6/CASO2/UD06-CasoPractico02-Django/CasoPractico2-Django$ docker-compose up -d
Creating network "casopractico2-django_default" with the default driver
Creating casopractico2-django_db_1 ... done
Creating casopractico2-django_web_1 ... done
vicente@vicente-desktop:~/Documentos/UD1/6/CASO2/UD06-CasoPractico02-Django/CasoPractico2-Django$
```

6.03 - Adjunta una captura donde se muestre que con “*Docker Compose*” has lanzado el servicio escalando los servidores Apache y ver que se sirven correctamente.

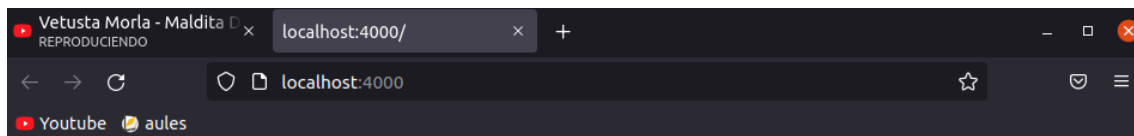
```
vicente@vicente-desktop:~/Documentos/UD1/6/CAS03/CasoPractico3-EscaladoProxyNginxYApache$ docker-compose up -d
Creating network "casopractico3-escaladoproxynginxyapache_default" with the default driver
Building apache
Sending build context to Docker daemon  3.072kB

Step 1/2 : FROM php:7.2-apache
7.2-apache: Pulling from library/php
6ec7b7d162b2: Pulling fs layer
db606474d60c: Pulling fs layer
```

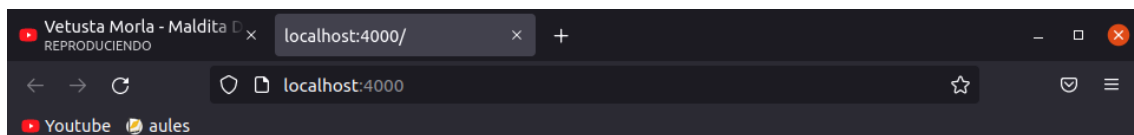
```
vicente@vicente-desktop:~/Documentos/UD1/6/CAS03/CasoPractico3-EscaladoProxyNginxYApache$ docker-compose up -d --scale apache=4
Creating network "casopractico3-escaladoproxynginxyapache_default" with the default driver
Creating casopractico3-escaladoproxynginxyapache_apache_1 ... done
Creating casopractico3-escaladoproxynginxyapache_apache_2 ... done
Creating casopractico3-escaladoproxynginxyapache_apache_3 ... done
Creating casopractico3-escaladoproxynginxyapache_apache_4 ... done
Creating casopractico3-escaladoproxynginxyapache_nginxproxy_1 ... done
```



Servido por: Servidor con IP 172.28.0.2 y hostname 16794fb540bb



Servido por: Servidor con IP 172.28.0.5 y hostname 7f36d809bbc8



Servido por: Servidor con IP 172.28.0.3 y hostname 7c2af083df48

7.01 - Adjunta una captura donde se muestre que “Visual Studio Code” está conectado a un contenedor.

The screenshot displays a web browser window at the top showing the PHP Version 7.2.34 page. Below the browser, a Visual Studio Code editor window is open, connected to a container named 'php:7.2-apache (servidordesarrollo)'. The editor shows the file 'index.php' with the following code:

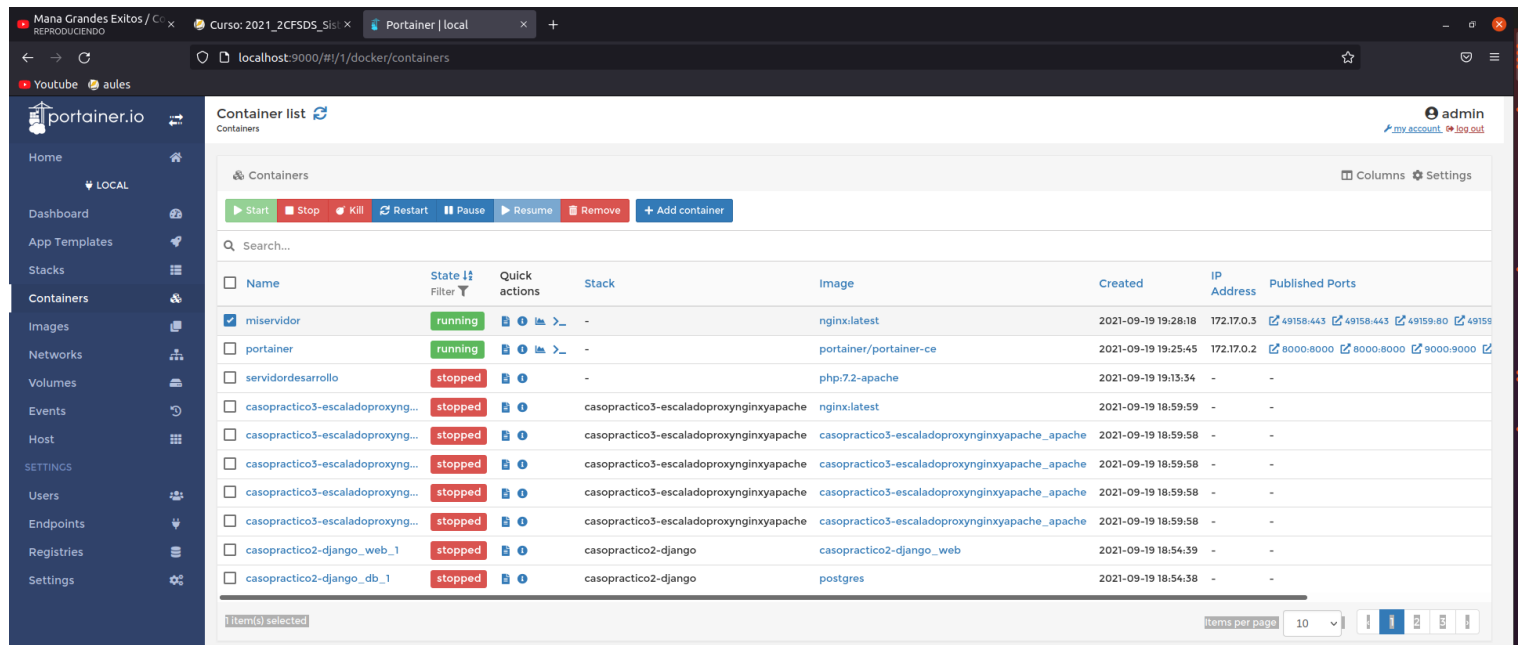
```
1 <?php
2     phpinfo();
3
```

The background of the VS Code window shows a table with system information for the container:

System	Linux 66b977993b82 5.11.0-34-generic #36~20.04.1-Ubuntu SMP Fri Aug 27 08:06:32 UTC 2021
Build Date	
Configure Command	
Server API	
Virtual Directory Support	
Configuration File (php.ini) Path	
Loaded Configuration File	
Scan this dir for additional .ini files	
Additional .ini files parsed	
PHP API	
PHP Extension	
Zend Extension	
Zend Extension Build	
PHP Extension Build	
Debug Build	
Thread Safety	
Zend Signal Handling	
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

At the bottom of the page, there is a copyright notice: "This program makes use of the Zend Scripting Language Engine: Zend Engine v3.2.0, Copyright (c) 1998-2018 Zend Technologies" and the Zend Engine logo.

7.02 - Adjunta una captura que se vea “Portainer CE” funcionando.



Actividad 02:

Kubernetes es un software libre creado por Google que nos permite crear un clúster de contenedores y “orquestrar” su despliegue, es decir, automatizar el despliegue de los diversos contenedores que componen nuestra APP. Mediante comandos y/o un fichero YAML le indicaremos como queremos que se despliegue y kubernetes se encargara de todo, facilitando enormemente el despliegue y escalado y, por ejemplos, olvidándonos de virtualizar en todas las maquinas posibles, puesto que kubernetes lo realiza automáticamente.

Actividad 03: Investiga como poner en marcha un servidor “Moodle” utilizando Docker Compose.

Accediendo al repositorio de GitHub (<https://github.com/bitnami/bitnami-docker-moodle>) nos explican cómo crear nuestra imagen de Docker con Moodle, nos dan directamente el archivo Docker-compose.yml, y nos indican los cambios que deberemos realizar para añadir persistencia a nuestro Docker.

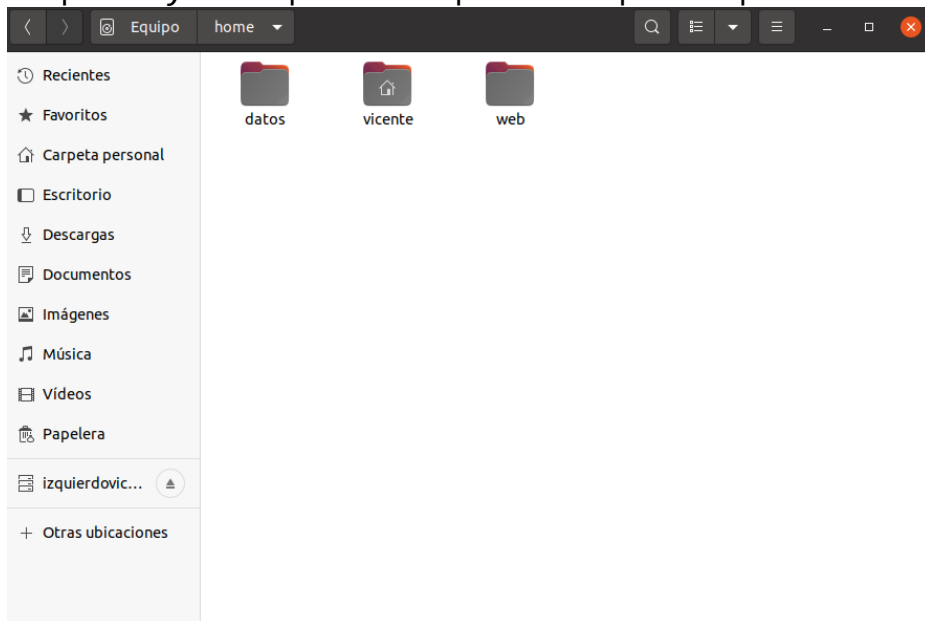
Únicamente deberemos cambiar los apartados “volumes” de la base de datos y del Moodle como se ve en la siguiente imagen.

```
1 version: '2'
2 services:
3   mariadb:
4     image: docker.io/bitnami/mariadb:10.5
5     environment:
6       # ALLOW_EMPTY_PASSWORD is recommended only for development.
7       - ALLOW_EMPTY_PASSWORD=yes
8       - MARIADB_USER=bn_moodle
9       - MARIADB_DATABASE=bitnami_moodle
10      - MARIADB_CHARACTER_SET=utf8mb4
11      - MARIADB_COLLATE=utf8mb4_unicode_ci
12   volumes:
13     - /home/datos:/bitnami/moodle
14   moodle:
15     image: docker.io/bitnami/moodle:3
16     ports:
17       - '80:8080'
18       - '443:8443'
19     environment:
20       - MOODLE_DATABASE_HOST=mariadb
21       - MOODLE_DATABASE_PORT_NUMBER=3306
22       - MOODLE_DATABASE_USER=bn_moodle
23       - MOODLE_DATABASE_NAME=bitnami_moodle
24       # ALLOW_EMPTY_PASSWORD is recommended only for development.
25       - ALLOW_EMPTY_PASSWORD=yes
26     volumes:
27       - /home/web:/bitnami/moodle
28     depends_on:
29       - mariadb
30   volumes:
31     mariadb_data:
32       driver: local
33     moodle_data:
34       driver: local
35     moodledata_data:
36       driver: local
```

En la siguiente imagen podemos observar que se cumple la condición de que la base de datos y el servidor de la página se encuentran en contenedores distintos

```
vicente@vicente-desktop:~/prueba1234$ docker-compose up -d
Creating network "prueba1234_default" with the default driver
Creating volume "prueba1234_mariadb_data" with local driver
Creating volume "prueba1234_moodle_data" with local driver
Creating volume "prueba1234_moodledata_data" with local driver
Creating prueba1234_mariadb_1 ... done
Creating prueba1234_moodle_1 ... done
vicente@vicente-desktop:~/prueba1234$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                               NAMES
3064812fd55a   bitnami/moodle:3   "/opt/bitnami/script..." 26 seconds ago Up 25 seconds 0.0.0.0:80->8080/tcp, :::80->8080/tcp, 0.0.0.0:443->8443/tcp, :::443->8443/tcp   prueba1234_moodle_1
887f4ce80d01   bitnami/mariadb:10.5 "/opt/bitnami/script..." 27 seconds ago Up 26 seconds 3306/tcp                          prueba1234_mariadb_1
```

Accediendo a la dirección que hemos indicado en el archivo Docker-composer.yml comprobamos que se cumple la persistencia demandada:



Y por último accedemos a la página de Moodle escribiendo en el buscador del navegador “Localhost”

