

PRACTICAS DOCKER COMPOSE

Escribir el fichero: **docker-compose.yml**

1) Servicio **Apache HTTP Server**

- a. Versión “3.9”
- b. La imagen httpd:2.4
- c. Puertos 80:80
- d. Configurar el bind mount
 - i. ./ El directorio actual del host de Docker
 - ii. En el contenedor: /usr/local/apache2/htdocs/
- e. Probar los comandos: up, log, down de Docker compose.

2) Servicio **MySQL**

- a. Versión “3.9”
- b. Servicio de mysql, imagen: mysql:8.0
- c. Puertos: 3306:3306
- d. Variables de entorno en una lista:
 - MYSQL_ROOT_PASSWORD=root
 - MYSQL_DATABASE=database
 - MYSQL_USER=user
 - MYSQL_PASSWORD=password
- e. Definir un volumen: mysql_data
- f. Se monta en el servicio en /var/lib/mysql
- g. Probar los comandos: up -d, ps, log, down, down -v de Docker compose
- h. Probar los comandos: docker volumen ls

3) Dos servicios: **MySQL** y **phpMyAdmin**, incluir las variables de entorno en un fichero: **.env**

- a. Versión “3.9”
- b. A los servicios asignar nombre con container_name
- c. Servicio de mysql, imagen: mysql:8.0
- d. Puertos: 3306:3306
- e. Variables de entorno en una lista,
- f. Definir un volumen: mysql_data
- g. Se monta en el servicio en /var/lib/mysql
- h. Servicio de phpmyadmin, imagen: phpmyadmin:5.1
- i. Puertos: 8080:80

- j. Variable de entorno sin fichero:
 - i. PMA_HOST=mysql,
 - ii. PMA_USER=myuser
 - iii. PMA_PASSWORD=mypassword
- k. depends_on=mysql
- l. Probar los comandos: up -d, ps, log, down, down -v de Docker compose
- m. Probar los comandos: docker volumen ls

4) Despliegue de una aplicación de WordPress, mysql y phpmyadmin

- a. Ampliar el anterior con wordpress
- b. Imagen: wordpress:latest
- c. Nombre del contenedor: wordpress_site
- d. depends_on=mysql
- e. puertos: "80:80"
- f. Variables de entorno:
 - i. WORDPRESS_DB_HOST: mysql
 - ii. WORDPRESS_DB_USER: wp_user (el mismo)
 - iii. WORDPRESS_DB_PASSWORD: wp_password
 - iv. WORDPRESS_DB_NAME: wordpress

5) Desplegar dos servicios en Python con la librería Flask, los dos servicios estarán en la misma red. Uno expone el puerto 5000 y el otro el 5001 y desde un servicio queremos hacer una petición get al otro servicio. Estructura de carpetas. El fichero de requisitos con las librerías: flask y request (cada una en una fila distinta).

```
project/
├── service1/
│   ├── app.py
│   ├── requirements.txt
│   └── Dockerfile
├── service2/
│   ├── app.py
│   ├── requirements.txt
│   └── Dockerfile
└── docker-compose.yml
```

6) s