

Contenedores con Docker

1. Se entregan los ficheros de una aplicación PHP (archivo app.zip que se adjunta) que permite realizar peticiones HTTP sobre el fichero api.php y se pide realizar el despliegue de la aplicación PHP sobre contenedores Docker.

Como resultado final **se han de entregar todos los ficheros y carpetas de la aplicación**, de tal modo que se pueda probar el despliegue.

Se deben tener en cuenta las siguientes consideraciones:

- a) Inicialmente se debe copiar el contenido de la carpeta ficheros dentro de una carpeta que llevará tu nombre de pila
- b) La aplicación se debe poder desplegar automáticamente tras ejecutar el comando **docker-compose up -d** dentro de la carpeta de la aplicación. Por tanto, es necesario crear un fichero dentro de la carpeta de la aplicación de nombre **docker-compose.yml** con los servicios necesarios que se describen en los siguientes puntos
- c) **Servicio bd.** Se debe crear un fichero **Dockerfile** dentro de la carpeta “bd” para la imagen de Mysql donde se debe precargar el script SQL que se entrega (este script crea la base de datos, las tablas e inserta algunos valores de ejemplo)

El contenedor generado llevará por nombre “tu_nombre_de_pila_bd”

Para este servicio NO se expondrá ningún puerto a la maquina Windows

- d) **Servicio backend.** Se deben crear un fichero **Dockerfile** en la carpeta “back” para la imagen de PHP (php:7.4-apache) donde se instalarán y habilitarán las extensiones necesarias para la conexión a la base de datos y también se copiarán los ficheros de la aplicación php que se entregan en la carpeta “back\src”.

El contenedor generado llevará por nombre “tu_nombre_de_pila_back”

Este servicio depende del servicio bd

Para este servicio se expondrá el puerto 8000 a la maquina Windows (si no tuviésemos ese puerto disponible en nuestro sistema, se podrá emplear otro distinto, pero en ese caso se deberá modificar también la petición fetch que se puede ver en el fichero “front\src\index.html”)

- e) **Servicio frontend.** Se debe crear un fichero Dockerfile en la carpeta “front” para crear una imagen personalizada de nginx que incluya el contenido de la carpeta “front\src”

El contenedor generado llevará por nombre “tu_nombre_de_pila_front”

Este servicio depende del servicio backend

Para este servicio se expondrá el puerto 80 a la maquina Windows.

- f) **Servicio pma.** Se debe crear también un contenedor para la aplicación phpmyadmin que facilitará el acceso a los datos que se van generando en la aplicación

Este servicio depende del servicio bd

Para este servicio se expondrá el puerto 9000 a la maquina Windows.

- g) Es importante observar que la conexión de la aplicación a la base de datos se realiza a través del fichero **db.php** donde se pueden ver los parámetros de la conexión que ya vienen preestablecidos.

- h) Se debe **crear un volumen Docker** de datos persistentes para la base de datos. Este volumen llevará por nombre “vol_bd”

- i) Se debe **verificar que la aplicación es funcional:**

- Accediendo directamente al backend mediante la aplicación postman y probando los siguientes métodos con la URL <http://localhost:8000/api.php>
 - **GET**, que deberá mostrar un JSON con los datos de todos los organizadores
 - **POST**, que deberás usar para insertar un nuevo organizador con datos a tu elección (enviando un JSON con la petición)
 - **PUT**, que deberás usar para actualizar alguno de los datos del organizador añadido anteriormente (enviando un JSON con la petición)
- Accediendo mediante <http://localhost> y verificando que se muestra en pantalla una tabla con los datos de todos los organizadores