

Manual de despliegue

React

Es necesario descargar e instalar node.js de la página oficial:

[Descarga | Node.js \(nodejs.org\)](https://nodejs.org)

Una vez descargado hay que seguir los siguientes pasos para desplegar el front-end:

- Abrir una terminal en la ruta de la carpeta del proyecto de front-end.
- Ejecutar los siguientes comandos:
 - npm install
 - npm start

Se abrirá el navegador web en la ruta localhost:3000 desplegando todo el front-end de la aplicación.

Una vez desplegado el entorno Docker, se podrá comenzar a utilizar.

Docker

Se ha adjuntado una carpeta denominada “setup Docker” que contiene las imágenes usadas, así como el fichero DockerCompose para el lanzamiento del entorno.

Se deberá descargar e instalar Docker. Está disponible para macOS, Linux y Windows:

[Docker: Accelerated, Containerized Application Development](#)

En el caso de Docker Desktop (versión Windows), es recomendable deseleccionar la opción de usar WSL 2 para evitar que surja algún problema que requiera la configuración de algunos aspectos de Windows. Véase:

Configuration

- ☐ Use WSL 2 instead of Hyper-V (recommended)
- ☒ Add shortcut to desktop

Ok

Todos los pasos del manual sirven para cualquiera de los sistemas operativos mencionados anteriormente.

Configuración entorno Docker

Antes del despliegue es necesario realizar unos pasos previos de configuración.

Primero hay que abrir la carpeta adjunta “setup Docker” y abrir una terminal en esta ruta.

Para cargar las imágenes en nuestro repositorio local, ejecutar los siguientes comandos:

- `docker load -i mariadb-image`
- `docker load -i adminer-image`
- `docker load -i servlet-image`

Podemos comprobar si se han añadido correctamente con el comando:

- `docker image ls`

Deberían aparecer las tres imágenes que hemos añadido.

Despliegue entorno Docker

Para desplegar la aplicación en el entorno Docker, es necesario abrir una consola de comandos y moverse a la carpeta adjuntada denominada “setup Docker”. Una vez allí ejecutar el siguiente comando:

- `docker-compose up -d`

Una vez finalizada la ejecución, todos los contenedores deben estar arrancados y funcionando correctamente.

Comprobar estado entorno Docker

Para comprobar el buen funcionamiento del entorno Docker, ejecutar el siguiente comando en una terminal:

- `docker ps`

El resultado debe ser algo parecido a lo siguiente:

docker ps							
CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES	
957c96c33bd7	web-app:latest	"catalina.sh run"	6 seconds ago	Up 2 seconds	0.0.0.0:8080->8080/tcp	web-app-cont	
94b0e8e42566	adminer:4.8.1	"entrypoint.sh php -_"	6 seconds ago	Up 2 seconds	0.0.0.0:8081->8080/tcp	adminer-cont	
2909b752df55	mariadb:10.9.4	"docker-entrypoint.s_"	6 seconds ago	Up 4 seconds	0.0.0.0:3306->3306/tcp	mariadb-cont	

Debemos fijarnos en que estén los tres contenedores levantados, para ello, mirar en la columna “STATUS” y comprobar que ponga “Up”.

Como observamos, el contenedor de la base de datos está expuesto en el puerto localhost:3306, el adminer en el localhost:8081 (que referencia al localhost:8080 de dentro del contenedor) y el de la aplicación web en el localhost:8080.

Apagado entorno Docker

Ejecutar en una terminal en la ruta “setup Docker” el siguiente comando:

- `docker-compose down`

Si se realiza una comprobación del estado del entorno, se podrá ver como los contenedores ya no aparecen.

Posibles problemas

Ports are not available

```
cker> docker-compose up -d
[+] Running 3/4
 - Network setupdocker_default Created      0.8s
 - Container mariadb-cont Started           3.4s
 - Container adminer-cont Started           4.7s
 - Container web-app-cont Starting          4.7s
Error response from daemon: Ports are not available: exposing port TCP 0.0.0.0:8080 -> 0.0.0.0:0: listen tcp 0.0.0.0:8080: bind: Only one usage of each sock
et address (protocol/network address/port) is normally permitted.
```

Este error (en este caso concreto) nos indica que no se ha podido levantar el contenedor web-app-cont debido a que ya hay un servicio utilizando el puerto localhost:8080.

Para solucionarlo, localizar el servicio que está usando el puerto con el comando:

- `netstat -ano`

Localizar el puerto `localhost:8080` y ver el PID del proceso que lo está utilizando:

```
cker> netstat -ano | findstr 8080
TCP    0.0.0.0:8080      0.0.0.0:0        LISTENING        3536
TCP    [::]:8080        [::]:0           LISTENING        3536
```

Finalizar el servicio con ese PID (los pasos varían en función del sistema operativo).

Una vez finalizado ese proceso, se puede volver a relanzar el comando “`docker-compose up -d`” sin necesidad de parar los otros contenedores que ya se habían levantado.

Docker Daemon is not running

```
C:\Users\Daniel>docker ps
error during connect: This error may indicate that the docker daemon is not running.: Get "http://%2F%2F.%2Fpipe%2Fdocker_engine/v1.24/containers/json": open //./pipe/docker_engine: The system cannot find the file specified.
```

Este error nos indica que el servicio de Docker no está corriendo. Para levantarlo:

- En Windows: ejecutar la aplicación Docker Desktop.
- En Linux: ejecutar el comando “`sudo systemctl start Docker`”