CURSO 23 - 24

Despliegue
BOOK 'N BOOK

REALIZADO POR:

MARÍA ESCRIBANO VERDE

2º DAW - IES ALIXAR

MANUAL DE DESPLIEGUE

En este apartado se describen todos los pasos necesarios para desplegar la aplicación *Book N' Book* utilizando Docker. La aplicación está compuesta por tres componentes principales: una base de datos MySQL, un backend desarrollado en Spring Boot y un frontend en Angular. El despliegue se realiza utilizando contenedores Docker y Docker Compose.

PRE-REQUISITOS

Antes de comenzar, asegúrate de tener instalados los siguientes componentes en tu máquina:

- Docker
- Docker Compose

ESTRUCTURA DE PROYECTO

La estructura del proyecto debe ser similar a la siguiente:

```
booknbook/
├── BooknBookBACK/
│   ├── Dockerfile
│   ├── target/
│   │   └── booknbook-0.0.1-SNAPSHOT.jar
│   └── ...
├── BookNBookFRONT/
│   ├── BooknBookFront/
│   │   ├── Dockerfile
│   │   ├── nginx.conf
│   │   └── ...
├── docker-compose.yml
└── init.sql
```

EXPLICACIÓN DE LOS ARCHIVOS DE DESPLIEGUE

Dockerfile del Frontend: se divide en dos etapas para optimizar la construcción y despliegue de la aplicación Angular. En la primera etapa, se utiliza una imagen ligera de Node.js para construir la aplicación. Esto incluye la instalación de dependencias y la construcción del proyecto en un directorio de trabajo específico. En la segunda etapa, se utiliza una imagen de Nginx para servir la aplicación construida. Los archivos estáticos generados se copian al directorio de Nginx y se configura el servidor web para servirlos. Esta separación en etapas permite mantener la imagen final ligera y eficiente, ya que solo incluye los archivos necesarios para servir la aplicación.

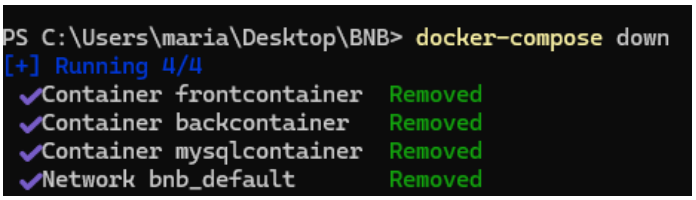
Dockerfile del Backend: define una configuración para ejecutar una aplicación Java utilizando OpenJDK 17. Se establecen variables de entorno para la configuración de la base de datos, incluyendo la URL, el nombre de usuario y la contraseña. El archivo JAR de la aplicación, generado previamente con Maven, se copia al contenedor. El comando de entrada especifica que la aplicación se ejecutará utilizando `java -jar` para iniciar el archivo JAR. Este enfoque simplifica el despliegue de la aplicación Java, asegurando que todas las dependencias y configuraciones necesarias estén contenidas dentro de la imagen del contenedor.

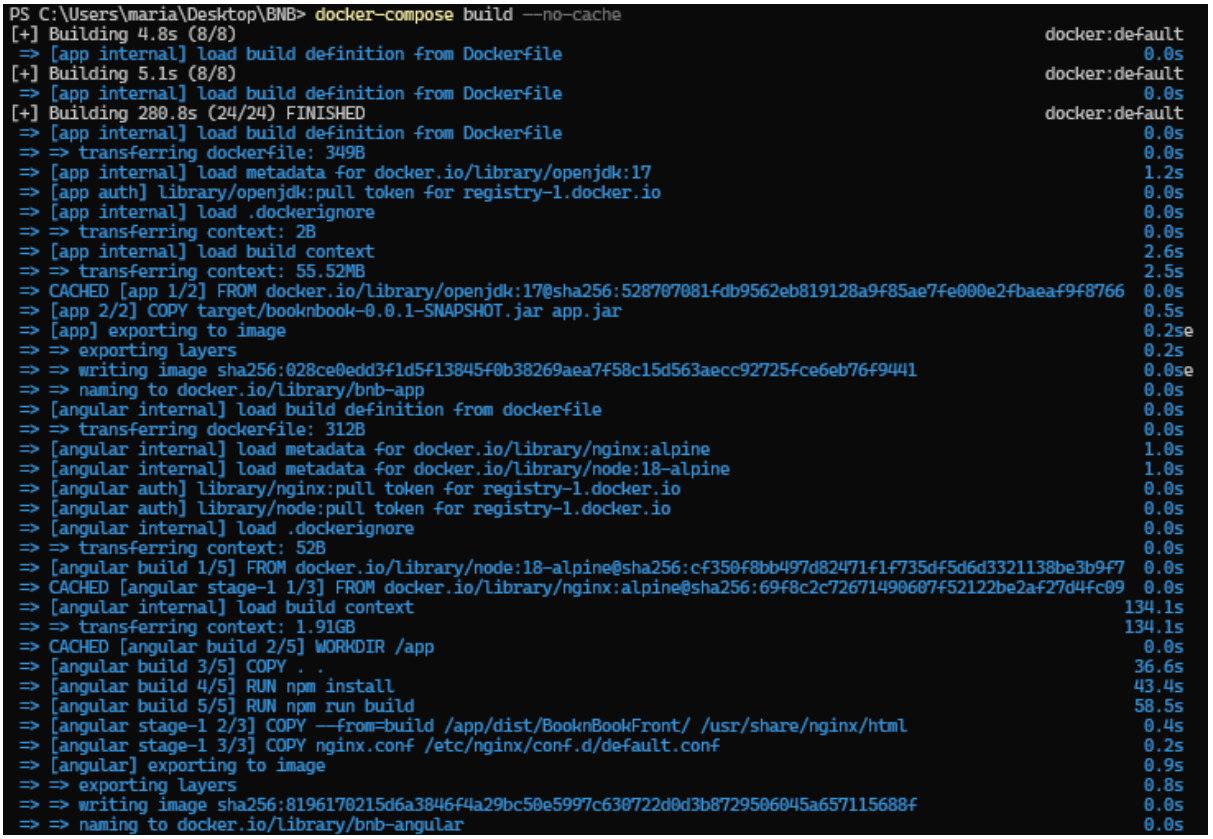
Archivo docker-compose.yml: agrupa el despliegue de múltiples servicios de contenedores necesarios para la aplicación. Define tres servicios principales: el frontend, el backend y la base de datos MySQL. Cada servicio tiene su propia configuración de construcción, puertos expuestos y dependencias. Por ejemplo, el servicio del frontend depende del backend para garantizar que el servidor de la aplicación esté listo antes de iniciar el servidor web. La configuración de MySQL incluye variables de entorno para la creación y autenticación de la base de datos, así como un health check para asegurar que el servicio de base de datos esté saludable antes de permitir que los demás servicios se inicien. Este enfoque asegura una coordinación adecuada entre los distintos componentes de la aplicación, facilitando un despliegue eficiente y ordenado.

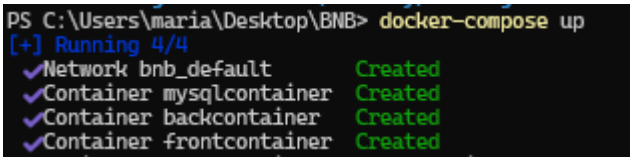
PASOS PARA EL DESPLIEGUE

Para desplegar la aplicación, desde una terminal en el directorio raíz del proyecto (booknbook) y ejecutar los siguientes comandos:

```
docker-compose down
docker-compose build --no-cache
docker-compose up
```



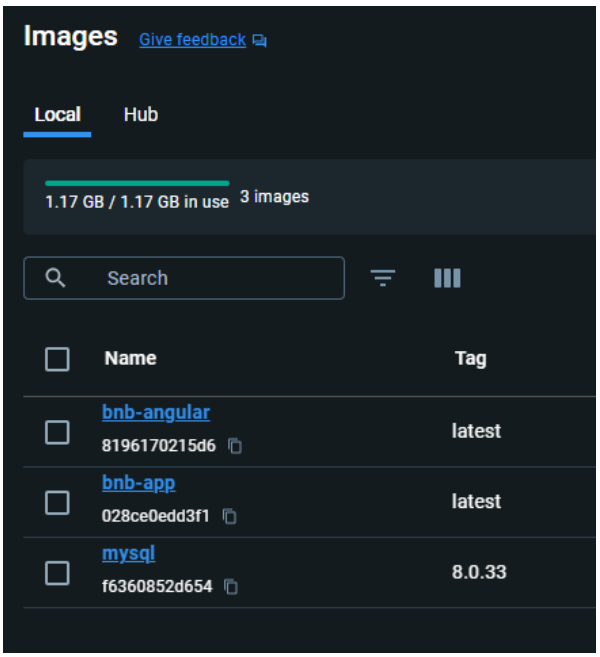


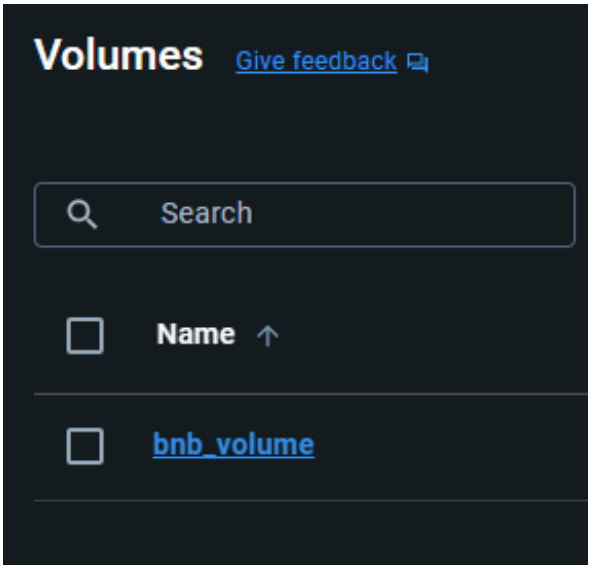


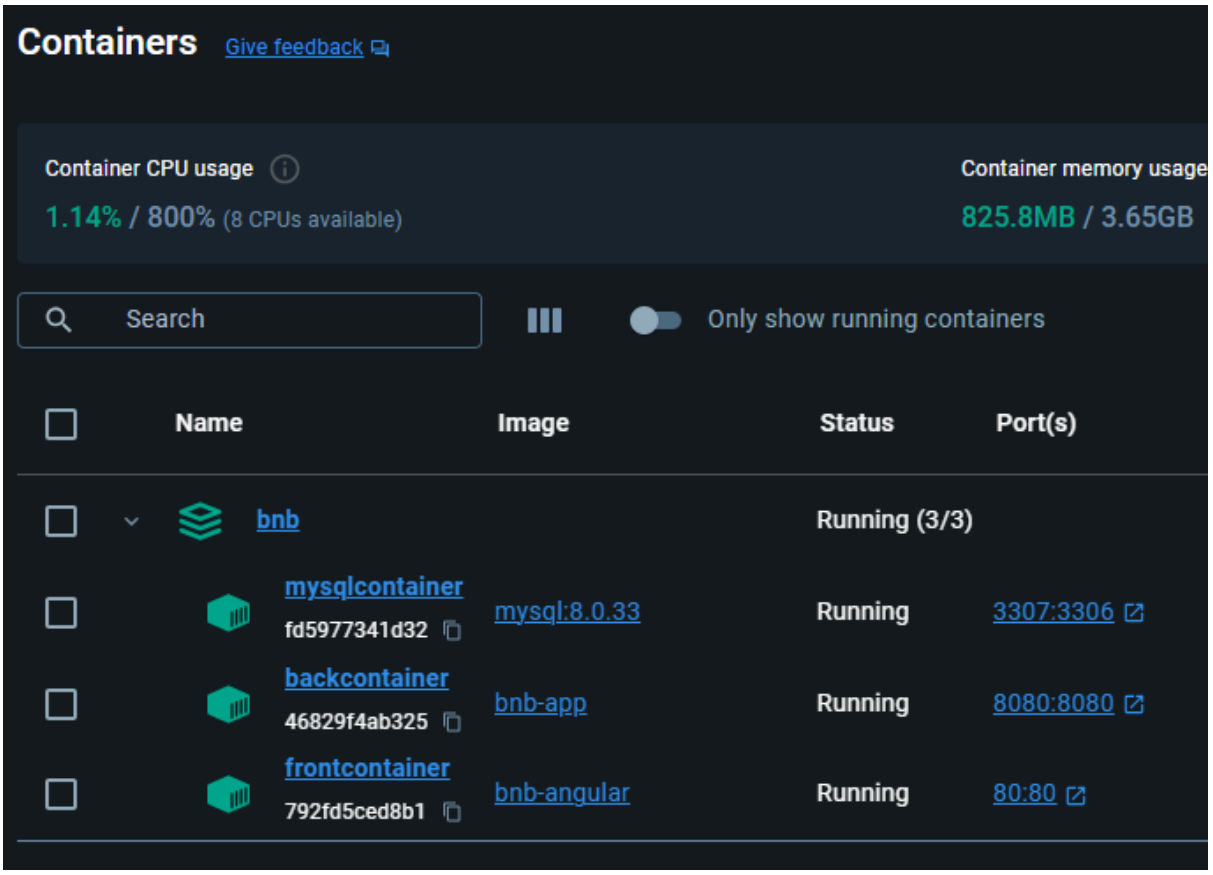
Estos comandos harán lo siguiente:

- docker-compose down: Detiene y elimina los contenedores existentes.

- docker-compose build --no-cache: Construye los contenedores sin utilizar la caché, asegurándose de que se utilicen las versiones más recientes de las imágenes.
- docker-compose up: Inicia los contenedores en modo adjunto, permitiendo ver los logs en tiempo real.



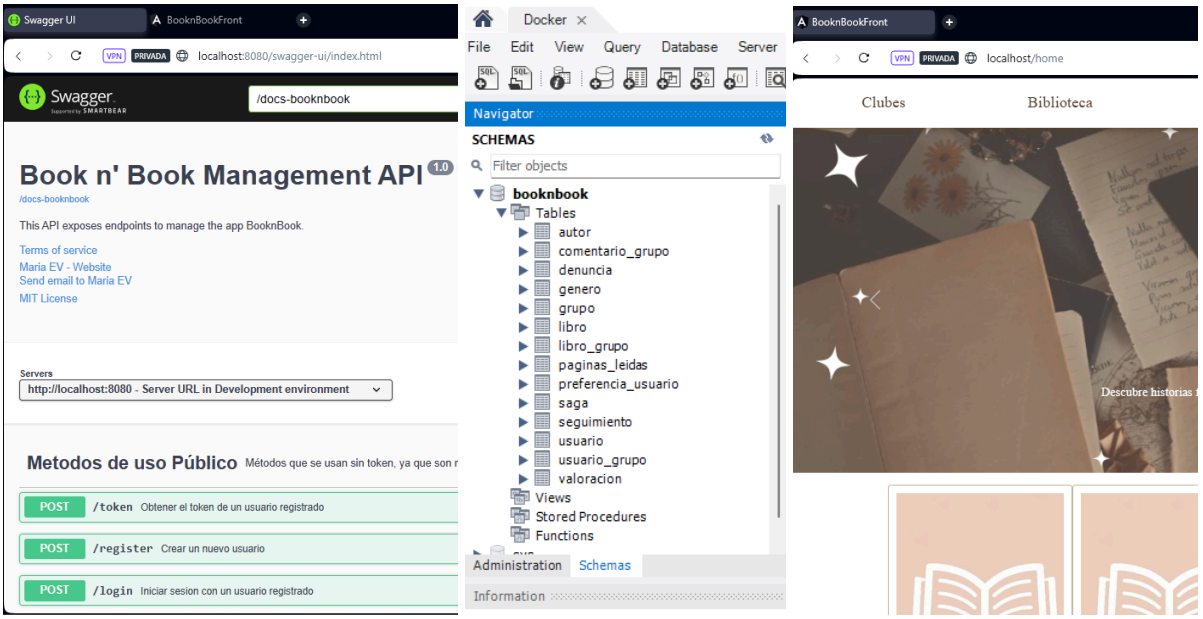




Tras ello, se puede verificar el despliegue de la siguiente forma:

- Backend: Acceder a http://localhost:8080/swagger para verificar que el backend de Spring Boot está corriendo correctamente.

- Frontend: Acceder a http://localhost para verificar que el frontend de Angular está corriendo correctamente.
- Base de Datos: Acceder a MySQL a través del puerto 3307 en tu máquina local con el usuario root y la contraseña 123456.



Por último a tener en cuenta, en este despliegue, se ha configurado un volumen para la base de datos MySQL, lo que permite la persistencia de datos incluso si los contenedores se detienen o reinician. Este volumen se define en el archivo docker-compose.yml y asegura que los datos de la aplicación no se pierdan, proporcionando una solución de almacenamiento confiable y persistente.

Además, el frontend de Angular se sirve utilizando Nginx, un servidor web ligero y de alto rendimiento. Nginx se utiliza para servir los archivos estáticos generados por Angular, asegurando una entrega rápida y eficiente de los recursos del frontend. La configuración de Nginx se especifica en el archivo nginx.conf, optimizando así la experiencia del usuario final al interactuar con la aplicación *Book N' Book*.