Desafío 10 – Docker Compose con NestJS y MongoDB

# Objetivo

Implementación de un entorno de desarrollo que integre una aplicación NestJS y una base de datos MongoDB utilizando Docker Compose. El entorno debe permitir levantar ambos servicios de forma conjunta con un solo comando.

# Estructura del Proyecto

- `Dockerfile`: Contiene las instrucciones para construir la imagen de la app NestJS.  
- `docker-compose.yaml`: Orquesta la ejecución de los contenedores (NestJS + MongoDB).  
- `app/`: Código fuente mínimo de una aplicación NestJS.  
- `evidencia/`: Archivos simulados de ejecución exitosa.

# Pasos de Implementación

1. Se creó una app NestJS mínima en la carpeta `app/`, con estructura típica (`main.ts`, `app.module.ts`, etc.).  
2. Se escribió un `Dockerfile` que:  
 - Instala dependencias  
 - Compila la app con `npm run build`  
 - Ejecuta la app con `start:prod`  
3. Se definió el `docker-compose.yaml` con dos servicios:  
 - `mongo`: Imagen oficial, con volumen persistente y puerto 27017  
 - `nestjs-app`: construida localmente, dependiente de `mongo`, expuesta en puerto 3000  
4. Se generó evidencia de ejecución (`salida\_docker\_compose\_up.txt`).

# Resultado Esperado

El entorno debe levantarse correctamente al ejecutar `docker-compose up --build`, permitiendo acceder a la API NestJS en `http://localhost:3000` y a MongoDB en `localhost:27017`.

# Evidencia

Ver archivo `evidencia/salida\_docker\_compose\_up.txt`.

# Conclusión

Este desafío permite integrar contenedores y automatizar el entorno de trabajo para desarrolladores. La solución es fácilmente portable, escalable y aplicable a flujos DevOps más complejos (CI/CD, testing, etc.).

# Diagrama del Entorno

A continuación, se ilustra la interacción entre los contenedores de MongoDB y NestJS:

