Examen de la parte del backend del Backend.

El ejercicio debe ser funcional y se deben de controlar los posibles errores que nos devuelva la conexión con la BBDD o cualquier otra operación. Se debe de trabajar con los procesos que nos ofrece js.

Utilizando Express, crear un backend que me permita gestionar libros de una biblioteca.

Nuestro backend tendrá el siguiente endpoint.

- "localhost:5173/búsqueda" Nos devolverá los libros que tengamos a través de un "título" que le pasemos.

Se pide

- Generar las diferentes rutas asignadas a cada uno de los endpoint´s.
- Generar la conexión con la base de datos. "Biblioteca2DAM"
- Desarrollar los controladores asociados a cada una de las rutas de entrada a nuestra API que hemos configurado en el punto anterior.
- Generar los schemas necesarios para la validación de datos.
    - Cuando insertamos datos, se debe de introducir:
        - Titulo – obligatorio
        - Autor - obligatorio
        - Editorial - obligatorio.
    - Buscar tareas:
        - Titulo: obligatorio.

Para conectarnos contra nuestra API, vamos a utilizar el Thunder Client. Hacemos las peticiones siguiendo los esquemas anteriores y recibiremos las respuestas en formato json.

Docker-Compose:

```
services:
  mongo:
    image: mongo
    container_name: mongo
    restart: always
    ports:
      - "27017:27017"
    volumes:
      - /home/usertar/proyectoDI/data:/data/db
    networks:
      - mongo-network

  mongo-express:
    image: mongo-express:1.0.2-20-alpine3.19
    container_name: mongo-express
    ports:
      - "8081:8081"
    depends_on:
      - mongo
    environment:
      - ME_CONFIG_BASICAUTH_USERNAME=root
      - ME_CONFIG_BASICAUTH_PASSWORD=root
      - ME_CONFIG_MONGODB_PORT=27017
      - ME_CONFIG_MONGODB_ADMINUSERNAME=root
      - ME_CONFIG_MONGODB_ADMINPASSWORD=root
    links:
      - mongo
    networks:
      - mongo-network

networks:
  mongo-network:
    driver: bridge
```