

1. Definição das Ferramentas, Bibliotecas, Frameworks e Dispositivos

Ferramentas

Java JDK 17 ou superior: Para desenvolvimento e execução do código em Java.

Spring Boot: Framework para desenvolvimento de aplicações web com Java.

Maven: Gerenciador de dependências e automação de builds.

MySQL: Banco de dados relacional utilizado no projeto.

MySQL Workbench: Interface para gerenciar o banco de dados.

Git: Controle de versão para gerenciar alterações no projeto.

Swagger (Springdoc OpenAPI): Documentação e testes de requisições HTTP para a API.

Bibliotecas e Frameworks

Spring Data JPA: Para integração com o banco de dados usando ORM.

Lombok: Para reduzir boilerplate no código (como getters, setters, etc.).

Spring Web: Para criação e gerenciamento de rotas no backend.

Validation API (Jakarta): Para validação de entradas e requisições.

Dispositivos

Computador com sistema operacional Windows, macOS, ou Linux.

Conexão à internet para baixar dependências e configurar o ambiente.

2. Comandos de Instalação e Configuração

Pré-requisitos

1. Java JDK 17 ou superior:

Faça o download e instalação pelo site oficial ou pelo gerenciador de pacotes do seu sistema operacional.

2. MySQL:

Faça o download e instalação pelo site oficial e configure um usuário e senha para o banco de dados.

3. Maven:

Caso esteja usando uma IDE como IntelliJ ou Eclipse, o Maven já estará integrado. Caso contrário, faça o download pelo site oficial.

Passos para Configuração

1. Clone o repositório do projeto:

```
git clone https://github.com/JsViniciusA/TasksApi.git  
cd TasksApi
```

2. Configure o banco de dados:

Abra o MySQL Workbench ou o terminal e crie o banco de dados:

```
CREATE DATABASE nome_do_banco;
```

3. Configure o arquivo application.properties:

Localize o arquivo em src/main/resources/application.properties e adicione as configurações do banco de dados:

```
spring.datasource.url=jdbc:mysql://127.0.0.1:3306/nome_do_banco  
spring.datasource.username=usuario  
spring.datasource.password=senha  
spring.jpa.hibernate.ddl-auto=update  
spring.jpa.show-sql=true
```

4. Instale as dependências do projeto:

Caso use Maven:

```
mvn clean install
```

5. Execute o projeto:

Inicie o servidor Spring Boot:

```
mvn spring-boot:run
```

6. Testando com o Swagger:

Acesse o Swagger UI no navegador:

<http://localhost:8080/swagger-ui.html>