

1 Instalar Dependências Necessárias

Antes de tudo, certifique-se de ter instalado:

✓ **Java JDK 17+** → Verifique com:

```
sh
CopiarEditar
java -version
```

Se não tiver instalado, baixe [aqui](#).

✓ **Maven** → Verifique com:

```
sh
CopiarEditar
mvn -version
```

Se não tiver instalado, baixe [aqui](#).

✓ **MySQL Server** → Verifique com:

```
sh
CopiarEditar
mysql --version
```

Se não tiver instalado, baixe [aqui](#).

✓ **Postman ou Front-end** para testar a API.

2 Configurar o Banco de Dados

Abra o MySQL e execute:

```
sql
CopiarEditar
CREATE DATABASE db_livros;
USE db_livros;

CREATE TABLE tbl_livros (
    id INT AUTO_INCREMENT PRIMARY KEY,
    titulo VARCHAR(100) NOT NULL,
    autor VARCHAR(100) NOT NULL,
    ano_publicacao INT,
    favorito BOOLEAN DEFAULT FALSE,
```

```
        foto VARCHAR(255)
    );

INSERT INTO tbl_livros (titulo, autor, ano_publicacao, favorito, foto)
VALUES
("O Livro Vermelho", "C.G. Jung", 2017, false,
"https://m.media-amazon.com/images/I/715a5Cfa3dL._SY466_.jpg");
```

3 Configurar o `application.properties`

No seu projeto Spring Boot (`src/main/resources/application.properties`), configure:

```
properties
CopiarEditar
spring.datasource.url=jdbc:mysql://localhost:3306/db_livros
spring.datasource.username=root
spring.datasource.password=SUASENHA
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver

spring.jpa.database-platform=org.hibernate.dialect.MySQL8Dialect
spring.jpa.hibernate.ddl-auto=update
spring.jpa.show-sql=true
```

Troque **SUASENHA** pela senha do MySQL!


Se não souber a senha, tente acessar o MySQL com:

```
sh
CopiarEditar
mysql -u root -p
```

4 Rodar a Aplicação

Agora, vá até a pasta do seu projeto e execute:

```
sh
CopiarEditar
mvn spring-boot:run
```

Ou, se estiver usando o **IntelliJ** ou **VS Code**, procure a classe **LivroApplication** (ou **Main**) e clique em **Run** .

Se tudo der certo, verá algo como:

```
sh
CopiarEditar
Tomcat started on port 8080
Started LivroApplication in X seconds
```

Isso significa que a API já está rodando! 🚀

5 Testar a API

Agora, você pode testar com **Postman** ou direto no navegador.

➤ Listar Livros (GET)

Abra no navegador:

```
bash
CopiarEditar
http://localhost:8080/api/livros/listarLivros
```

Ou no **Postman**:

- **Método:** GET
 - **URL:** <http://localhost:8080/api/livros/listarLivros>
-

➤ Cadastrar Livro (POST)

No **Postman**:

- **Método:** POST
- **URL:** <http://localhost:8080/api/livros/cadastrarLivro>
- **Headers:** Content-Type: application/json
- **Body (JSON):**

```
json
CopiarEditar
{
  "titulo": "Clean Code",
  "autor": "Robert C. Martin",
  "anoPublicacao": 2008,
  "favorito": true,
  "foto": "https://m.media-amazon.com/images/I/41SH-SvWPxL.jpg"
}
```

Se retornar **201 Created**, o livro foi cadastrado! ✅

6 Rodar o Front-end

Se já tiver um front-end feito, vá até a pasta dele e inicie com:

```
sh
CopiarEditar
npx live-server
```

ou

```
sh
CopiarEditar
python -m http.server 5500
```

Depois, acesse:

```
arduino
CopiarEditar
http://127.0.0.1:5500/index.html
```

E teste a interação com a API!