

# **Exercícios Trilha Teórica de Back-End:**

(SpringBoot)

## **Exercício 1: API Rest Básica:**

### **Requisitos:**

Crie um projeto Spring Boot usando Spring Initializr com as seguintes dependências:  
Implemente Swagger UI para teste e documentação dos endpoints.

**Spring Web**  
**Swagger UI**

### **Implemente uma classe Tarefa com os seguintes atributos:**

long id  
String descricao  
boolean completa

### **Crie um controlador REST (TarefaController) com os seguintes endpoints:**

GET /tarefas: Retorna a lista de todas as tarefas.  
POST /tarefas: Adiciona uma nova tarefa.  
PUT /tarefas/{id}: Atualiza uma tarefa existente.  
DELETE /tarefas/{id}: Remove uma tarefa.

## **Exercício 2: API Rest Básica com Banco de Dados:**

Crie uma aplicação Spring Boot que expõe uma API REST para gerenciar usuários, utilizando um banco de dados para persistência.  
Implemente Swagger UI para teste e documentação dos endpoints.

### **Requisitos:**

Crie um projeto Spring Boot usando Spring Initializr com as seguintes dependências:

**Spring Web**  
**Spring Data JPA**  
**Swagger UI**

**Implemente a entidade Usuario com os seguintes atributos:**

long id

String nome

String email

Crie um repositório (UsuarioRepository) estendendo JpaRepository.

**Crie um controlador REST (UserController) com os seguintes endpoints:**

GET /usuarios: Retorna a lista de todos os usuários.

POST /usuarios: Adiciona um novo usuário.

PUT /usuarios/{id}: Atualiza um usuário existente.

DELETE /usuarios/{id}: Remove um usuário.

## **Exercício 3: API REST com Segurança e Autenticação JWT**

**Descrição:**

Crie uma aplicação Spring Boot que expõe uma API REST para gerenciar produtos, utilizando autenticação JWT para proteger os endpoints.

**Requisitos:**

Crie um projeto Spring Boot usando Spring Initializr com as seguintes dependências:

Spring Web

Spring Data JPA

Spring Security

H2 Database

jjwt (JSON Web Token)

Implemente a entidade Produto com os seguintes atributos:

long id

String nome

double preco

Crie um repositório (ProdutoRepository) estendendo JpaRepository.

**Implemente o serviço de autenticação JWT, incluindo:**

Gerenciamento de usuários e roles

Geração de tokens JWT

Filtragem de requisições para autenticação JWT

Crie um controlador REST (ProdutoController) com os seguintes endpoints protegidos por autenticação JWT:

GET /produtos: Retorna a lista de todos os produtos.

POST /produtos: Adiciona um novo produto.

PUT /produtos/{id}: Atualiza um produto existente.

DELETE /produtos/{id}: Remove um produto.