
📝 Exercício – Construção de uma API REST com Spring Boot

Você deverá desenvolver uma **API REST aberta (sem autenticação)** utilizando **Spring Boot**, com suporte a **operações de cadastro (CRUD)** para duas entidades principais: **Disciplinas** e **Notas**.

🎯 Objetivo

Criar uma aplicação backend em Spring Boot que permita o gerenciamento de disciplinas e suas respectivas notas de alunos, por meio de uma API RESTful.

📦 Entidades

1. Disciplina

Representa uma disciplina de um curso.

Atributos obrigatórios:

- `id`: identificador único (gerado automaticamente)
- `nome`: nome da disciplina (ex: "Estrutura de Dados")
- `codigo`: código da disciplina (ex: "ED123")

2. Nota

Representa a nota de um aluno em uma determinada disciplina.

Atributos obrigatórios:

- `id`: identificador único (gerado automaticamente)
- `aluno`: nome do aluno (ex: "Carlos Alberto")
- `valor`: valor da nota (ex: 8.5)
- `disciplina`: disciplina associada à nota (relacionamento com a entidade Disciplina)

🔗 Relacionamentos

- Uma **disciplina** pode ter **várias notas**.
- Cada **nota** está associada a **uma única disciplina**.

📮 Requisitos da API REST

Implemente os seguintes endpoints:

📘 Disciplinas

- `GET /api/disciplinas` – Listar todas as disciplinas
- `GET /api/disciplinas/{id}` – Obter uma disciplina por ID
- `POST /api/disciplinas` – Criar uma nova disciplina
- `PUT /api/disciplinas/{id}` – Atualizar os dados de uma disciplina
- `DELETE /api/disciplinas/{id}` – Remover uma disciplina

📝 Notas

- `GET /api/disciplinas/{id}/notas` – Listar todas as notas de uma disciplina
- `POST /api/disciplinas/{id}/notas` – Adicionar uma nova nota à disciplina
- `PUT /api/notas/{id}` – Atualizar uma nota
- `DELETE /api/notas/{id}` – Remover uma nota

🛠️ Requisitos Técnicos

- Use **Spring Boot** com **Spring Web** e **Spring Data JPA**.
- Utilize um banco de dados relacional (H2, PostgreSQL ou MySQL).

- Use anotações JPA para mapeamento das entidades.
- Utilize o padrão **Controller-Service-Repository**.
- Retorne os status HTTP adequados para cada operação (`200 OK`, `201 Created`, `204 No Content`, `404 Not Found`, etc).

📑 Critérios de Avaliação

- Organização e clareza do código
- Uso adequado dos padrões REST
- Estruturação correta das entidades e relacionamentos
- Funcionamento completo dos endpoints
- Tratamento de erros adequado
