


Atividade: Sistema de Gerenciamento de Biblioteca	
SSM	
CRIADO: CRISTIANO MENESES	Data: 06/11/2024

Objetivo: Criar uma aplicação básica de gerenciamento de uma biblioteca utilizando Spring Boot com três tabelas no banco de dados: **Livro, Autor e Categoria**. Os alunos irão configurar um projeto que permita o cadastro de livros, autores e categorias, com as relações entre eles.

1. Estrutura do Projeto

Os alunos deverão criar um projeto Spring Boot com as seguintes camadas:

- **Model (Entidades):** Define as classes que representam as tabelas no banco de dados.
- **Repository:** Interfaces para interação com o banco de dados usando Spring Data JPA.
- **Service:** Camada de lógica de negócio.
- **Controller:** Camada para criação de endpoints REST.

2. Estrutura das Tabelas


- **Livro** (tabela tbl_livros):
 - id (Long, PK)
 - titulo (String)
 - isbn (String)
 - autor_id (Long, FK para a tabela autor)
 - categoria_id (Long, FK para a tabela categoria)
- **Autor** (tabela tbl_autores):
 - id (Long, PK)
 - nome (String)
 - nacionalidade (String)
- **Categoria** (tabela tbl_categorias):
 - id (Long, PK)
 - nome (String)

Relacionamentos:

- Livro -> Autor: Muitos para um (um autor pode ter vários livros, mas um livro possui apenas um autor).
- Livro -> Categoria: Muitos para um (uma categoria pode ter vários livros, mas um livro pertence a apenas uma categoria).

3. Passos para Implementação

1. Configuração Inicial:

Atividade: Sistema de Gerenciamento de Biblioteca	
SSM	
CRIADO: CRISTIANO MENESES	Data: 06/11/2024

- Baixar o projeto no meu git:
<https://github.com/Cristianomeneses2008/springboot>
 - **Spring Web** (para criar APIs REST)
 - **Spring Data JPA** (para persistência de dados)
 - **Mysql Database**

2. Criar Entidades:

- Definir as classes Livro, Autor e Categoria na camada de modelo com as anotações do JPA para mapear as tabelas e os relacionamentos.

3. Criar Interfaces de Repository:

- Criar as interfaces **LivroRepository**, **AutorRepository**, e **CategoriaRepository** que estendem JpaRepository para cada entidade.

4. Criar Services:

- Implementar uma camada de serviço com as classes **LivroService**, **AutorService**, e **CategoriaService**, contendo a lógica de negócio básica para manipular cada entidade.

5. Criar Controllers:

- Implementar controladores REST (**LivroController**, **AutorController**, e **CategoriaController**) para criar, listar, atualizar e excluir registros de livros, autores e categorias.

6. Banco de Dados e Testes:

- Configurar o banco de dados Mysql.
- Testar os endpoints usando o **Postman** ou outra ferramenta de API.

4. Entregáveis

- Código completo da aplicação, com os modelos, repositórios, serviços e controladores.
- Scripts SQL (se necessário) para popular o banco de dados.
- Testes realizados nos endpoints para confirmar que CRUD funciona para Livro, Autor e Categoria.