# FACULDADE UNYLEYA

**MURILO NUNES DOS SANTOS MATRÍCULA 1105126**

# ATIVIDADE SPRING RESTFUL

**DESENVOLVIMENTO BACK-END PROFESSOR HELDER ARAGÃO**

# JULHO DE 2022

**CURITIBA – PR**

**1 – Inicialização do projeto**

Para inicializar o projeto utilizei do Spring initializr com as depêndencias Spring web e Spring Data JPA, e fiz o download do projeto, essa ferramenta é excelente pois não temos muito trabalho para inciializar nosso projeto com o Spring Framework.

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

**2 – Abertura do Projeto**

Após o download do projeto abri o mesmo no netbeans. Decidi utilizar o netbeans por conta que é uma das IDES que estou mais familiarizado no desenvolvimento com a linguagem java.

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

**3 – Criação do Banco de dados.**

Para criar o banco de dados utilizei duas tecnologias:

**3.1 – XAMPP**

Utilizei o XAMPP para subir um servidor SQL local na porta 3306.

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

**3.2 – MYSQL Workbench**

Ele é uma interace gráfica para executar comandos SQL em um banco de dados que utiliza o MYSQL como serviço.

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Rodei os seguintes comandos para criar ae estrutura do banco de dados.

Criação do banco de dados:

*CREATE DATABASE cidadaos*

*default charset utf8*

*default collate utf8\_general\_ci*

Criação da tabela cidadãos:

*CREATE TABLE cidadaos(*

*id INT AUTO\_INCREMENT PRIMARY KEY,*

*first\_name varchar(200),*

*social\_security\_number varchar(14),*

*address varchar(300),*

*sex enum('M','F')*

*charset utf8);*

**Observação**:

* Criei as colunas em inglês para simular o projeto com um time internacional e nacional.
* A coluna social\_secutiry\_number (CPF) foi o nome que encontrei em inglês.
* As tabelas utilizam UTF8 uma vez que o sistema foi feito para ser utilizado no Brasil e temos caracteres especiais.
* Dentro do .ZIP deste trabalho vou deixar o arquivo SQL disponível.

**4 – Criação da Camada Model**

Uma vez que a tabela do banco de dados foi definida criei o pacote model dentro do projeto no Netbeans e criei a classe Cidadaos, com a entidade, e colunas definidas.

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente,

**5 – Configuração da conexão no application properties**

É necessário configurar porta, host, usuário.

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

**6 – Criação do repository**

Uma vez configurado podemos ter a integração com o banco de dados sem comandos SQL.

Interface gráfica do usuário, Texto

Descrição gerada automaticamente

**7 – Criação da camada Controller**

Agora é a etapa que começo a desenvolver os métodos GET, POST, PUT e DELETE.

Vou listar abaixo os métodos criados e prints de teste no banco de dados e Postman.

**7.1 – POST**

Uma imagem contendo Word

Descrição gerada automaticamente

URL: <http://localhost:8080/api/v1/cidadaos>

Body:

{

"nome":"Murilo Santos",

"cpf":"07604175040",

"endereco":"Rua Deputado Mário de Barros, 1700",

"sexo":"M"

}

POSTMAN

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

MySQL

Interface gráfica do usuário, Aplicativo, Word

Descrição gerada automaticamente

**7.2 – GET**

**7.2.1 – GET ALL**

Texto

Descrição gerada automaticamente com confiança média

URL: <http://localhost:8080/api/v1/cidadaos>

POSTMAN

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

MySQL

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

**7.2.2 – GET BY ID**

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

**Observação**: devido a não conseguir validar as classes exception utilizei os métodos isPresent para validar a existência de um Cidadão, o método isPresent estava disponibilizado nas referências da matéria no link abaixo:

<https://www.treinaweb.com.br/blog/criando-uma-api-rest-com-o-spring-boot>

URL: <http://localhost:8080/api/v1/cidadaos/2>

Tela de celular com aplicativo aberto

Descrição gerada automaticamente

MySQL

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

**7.3 – PUT**

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

URL: <http://localhost:8080/api/v1/cidadaos/1>

Banco antes da execução:

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

POSTMAN

Tela de celular com aplicativo aberto

Descrição gerada automaticamente

Banco após execução

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

**7.4 – Delete**

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

URL: <http://localhost:8080/api/v1/cidadaos/2>

Banco antes da execução:

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Tela de computador com fundo preto

Descrição gerada automaticamente

Banco após a execução:

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

**8. Considerações finais.**

Com a construção destes métodos pude criar um back-end Restful. Vejo o Spring com outros olhos agora e pretendo me aprofundar na tecnologia no futuro.

Os projetos da matéria estão disponíveis no meu GitHub.

<https://github.com/MuriloNSantos17/Spring-Boot>