



**IGE - INSTITUTO DE GEOCIENCIAS E ENGENHARIAS**  
**FACULDADE DE ENGENHARIA DA COMPUTAÇÃO**

**Relatório API – Web Services REST**

Gabriel Vilarino Gonçalves

Luis Guilherme Góis Matos

MARABÁ – PA

2022 – 2023

## **APRESENTAÇÃO**

O presente relatório tem por objetivo apresentar as atividades desenvolvidas utilizando os verbos GET, POST, PUT E DELETE. A aula sobre este tema foi ministrada pela docente Warley Muricy Valente Junior.

## SUMÁRIO

1. INTRODUÇÃO .....	4
2. OBJETIVO.....	5
3. DESENVOLVIMENTO.....	5
3.1 Seção 1 .....	5
3.2 Seção 2 .....	7
4. CONSIDERAÇÕES FINAIS .....	10

## **1. INTRODUÇÃO**

O Projeto envolve a utilização dos verbos GET, POST, PUT E DELETE, com o intuito de aprendizado para os discentes. Este projeto foi dividido em duas seções, onde na primeira seção foi utilizado o verbo GET HTTP, e na segunda seção foi utilizado os verbos GET, POST, PUT E DELETE.

## 2. OBJETIVO

Realizar uma aplicação prática de uma api web service, utilizando os seus verbos, para compreensão dos discentes.

## 3. DESENVOLVIMENTO

### 3.1 SEÇÃO 1

Foi utilizada duas classes para a realização da Seção 1:

-Saudacao.java

-SaudacaoController.java

A classe Saudacao.java tem um construtor que solicita o id e o nome, e há também o método get para id e nome.

A classe SaudacaoController.java tem uma constante privada MENSAGEM = "Olá %s"; que define a mensagem de saudação que será exibida, também há uma constante do tipo AtomicLong que é usada para gerar IDs únicos para cada saudação retornada pelo controlador. @GetMapping define uma rota que será usada para exibir uma saudação. @RequestParam indica que o valor do parâmetro nome será recebido a partir da query string na URL da requisição. O valor padrão para o parâmetro é "mundo" caso nenhum valor seja fornecido.

Depois é criado um objeto Saudacao com o ID único gerado pelo contador e a mensagem de saudação formatada com o valor do parâmetro nome.

The screenshot shows the Postman interface with a GET request to `http://localhost:8080/saudacao?nome=luis`. The request is configured with the following settings:

- Method: GET
- URL: `http://localhost:8080/saudacao?nome=luis`
- Params: (empty)
- Authorization: (empty)
- Headers: (7)
- Body: (empty)
- Pre-request Script: (empty)
- Tests: (empty)
- Settings: (empty)

The response body is displayed in the "Body" tab, showing the following JSON data:

```
1 {  
2   "id": 2,  
3   "nome": "Olá luis"  
4 }
```

### 3.2 SEÇÃO 2

Foi utilizada duas classes para a realização da Seção 2:

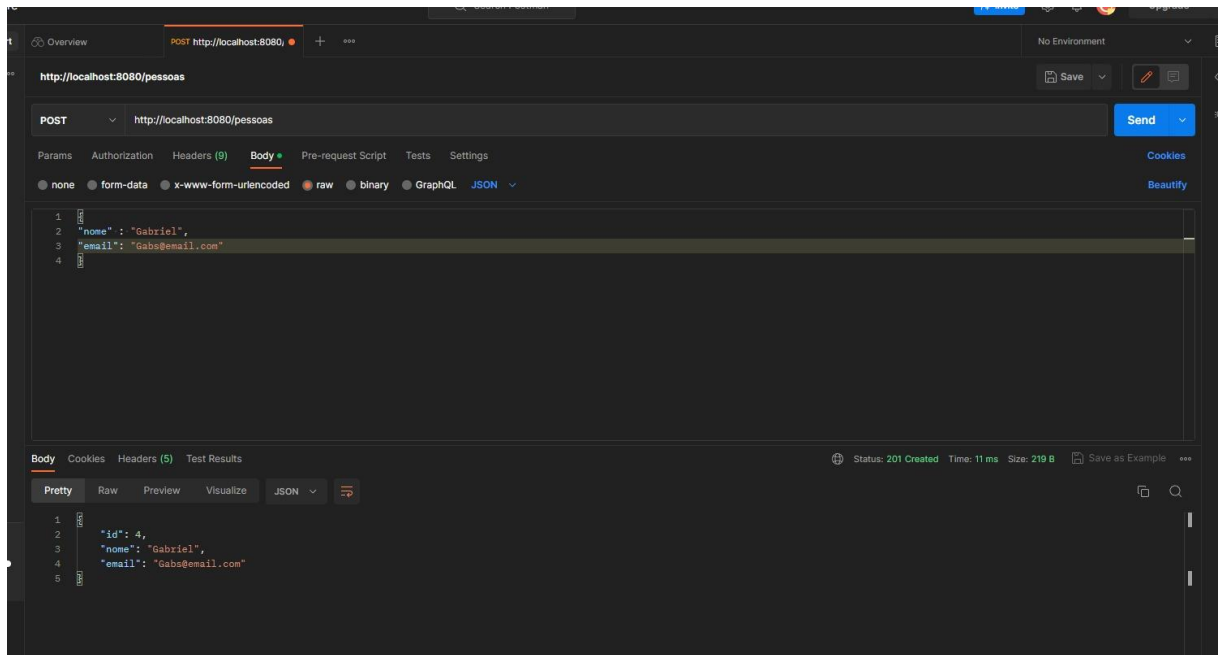
- Pessoa.java
- AgendaController.java
- PessoaNaoEncontrada.java

A Classe Pessoa.java é um Plain Old Java Object para representar uma pessoa na agenda de Contatos. Há dois construtores, um vazio e outro que solicita a entrada do id, nome e email e métodos getters e setters para essas variáveis.

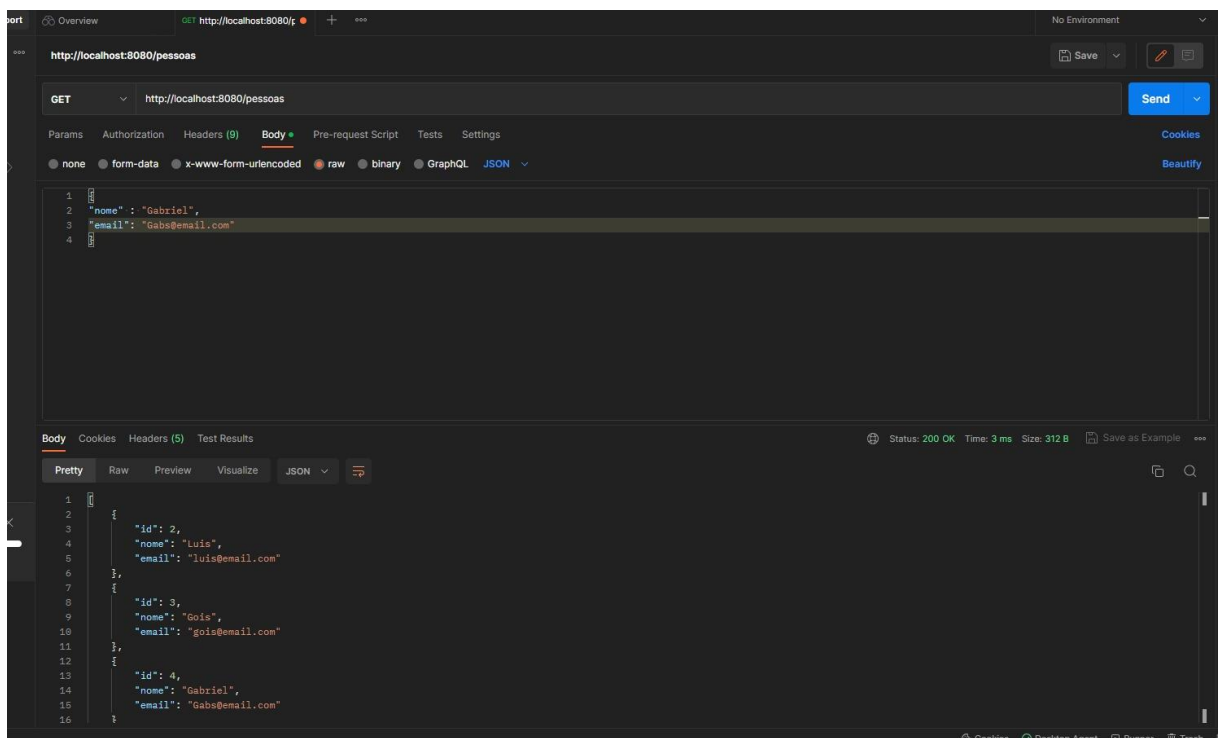
A Classe AgendaController.java é a classe responsável por processar os pedidos e instanciar objetos da classe Pessoa e fazer as alterações na agenda de contatos em memória.

A anotação @RestController indica que a classe é um controlador Spring REST e a anotação @RequestMapping define o prefixo de URL para todas as rotas que serão definidas nesse controlador. Há um ArrayList constante, onde armazena objetos do tipo Pessoa e também uma constante do tipo AtomicLog para gerar o ID Único.

A classe PessoaNaoEncontrada é um tratador de exceção para a classe PessoaNaoEncontradaException, que é lançada quando uma pessoa não é encontrada na agenda. Ele retorna uma resposta HTTP 404 (Não encontrado) com a mensagem da exceção.

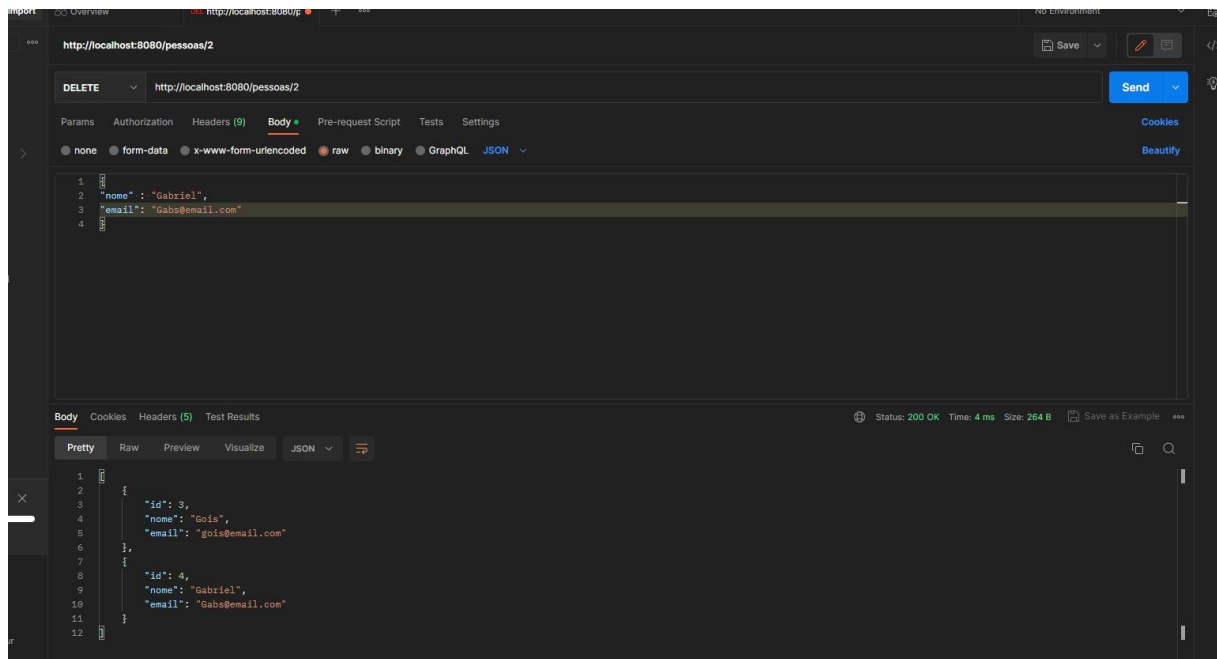


Nesta imagem foi utilizado o verbo POST para adicionar uma pessoa na lista de contatos.



Nesta imagem foi utilizado o verbo GET para informar todas as pessoas que estão adicionadas na lista de contatos.





Nesta imagem foi utilizado o verbo DELETE para deletar a pessoa que tem o ID = 2.



Por fim o terminal rodando o servidor.

#### **4. CONSIDERAÇÕES FINAIS**

Este projeto foi realizado com sucesso, utilizamos de todos os verbos propostos nas seções 1 e 2 para a realização da saudação e da adição de pessoas para uma lista de chamada utilizada uma api REST.