

# Documentação do Projeto de Cálculo de Frete

## Índice

1. [Introdução](#)
2. [Configuração de Projeto](#)
3. [Script SQL](#)
4. [Estrutura de Dados JSON](#)
5. [Endpoints Principais](#)
6. [Execução do Projeto](#)
7. [Testes](#)
8. [Conclusão](#)

## Introdução

Este documento fornece informações detalhadas sobre como configurar, rodar e utilizar o projeto Spring Boot para cálculo de frete. Inclui a descrição dos principais endpoints, funções, como executar o script MySQL, a estrutura de dados JSON esperada e outras informações pertinentes.

## Configurações de Projeto

1. **Clone o repositório**

```
git clone https://github.com/ardassejose/CD2_JavaTest.git
cd CD2_JavaTest
```

2. Configure as variáveis de ambiente para o banco de dados MySQL no arquivo *application.properties*:

```
spring.datasource.url=jdbc:mysql://localhost:3306/sigabem
```

```
spring.datasource.username=root  
spring.datasource.password=root  
spring.jpa.hibernate.ddl-auto=create-drop
```

## Script SQL

```
-- Deletar o schema se ele existir  
DROP DATABASE IF EXISTS sigabem;  
  
-- Criar o schema novamente  
CREATE DATABASE sigabem;  
  
-- Usar o novo schema  
USE sigabem;  
  
-- Criar a tabela com valores padrão definidos  
CREATE TABLE frete (  
    id BIGINT AUTO_INCREMENT PRIMARY KEY,  
    peso DOUBLE NOT NULL DEFAULT 0.0,  
    cep_origem VARCHAR(9) NOT NULL DEFAULT '00000-000',  
    cep_destino VARCHAR(9) NOT NULL DEFAULT '00000-000',  
    nome_destinatario VARCHAR(255) NOT NULL DEFAULT 'destinatario',  
    vl_total_frete DOUBLE NOT NULL DEFAULT 0.0,  
    data_prevista_entrega DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,  
    data_consulta DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP  
);
```

## Estrutura de Dados JSON

Requisição:

```
{  
  "cepOrigem": "29656-980",  
  "cepDestino": "29535-980",  
  "peso": 10.5,  
  "nomeDestinatario": "João da Silva"  
}
```

Resposta:

```
{  
  "id": 1,  
  "cepOrigem": "29656-980",  
  "cepDestino": "29535-980",  
  "dataConsulta": "2024-05-22T00:57:56.688+00:00",  
  "dataPrevistaEntrega": "2024-05-23T00:57:56.688+00:00",  
  "nomeDestinatario": "João da Silva",  
  "peso": 10.5,  
  "vITotalFrete": 25.0  
}
```

# Endpoints Principais

## 1. **Postar frete**

URL: `/api/frete/calcular`

Método HTTP: *POST*

Descrição: Calcula o valor do frete e a data prevista de entrega.

*Exemplo de Requisição:*

```
{  
  "cepOrigem": "01001-000",  
  "cepDestino": "02002-000",  
  "peso": 10.5,  
  "nomeDestinatario": "João da Silva"  
}
```

*Exemplo de Resposta:*

```
{  
  "id": 1,  
  "cepOrigem": "01001-000",  
  "cepDestino": "02002-000",  
  "dataConsulta": "2024-05-22T00:57:56.688+00:00",  
  "dataPrevistaEntrega": "2024-05-23T00:57:56.688+00:00",  
  "nomeDestinatario": "João da Silva",  
  "peso": 10.5,  
  "vITotalFrete": 25.0  
}
```

## 2. Listar Todos os Fretes

URL: /api/frete

Método HTTP: GET

Descrição: Retorna uma lista com todos os fretes calculados.

Exemplo de Resposta:

```
{
  "id": 1,
  "cepOrigem": "01001-000",
  "cepDestino": "02002-000",
  "dataConsulta": "2024-05-22T00:57:56.688+00:00",
  "dataPrevistaEntrega": "2024-05-23T00:57:56.688+00:00",
  "nomeDestinatario": "João da Silva",
  "peso": 10.5,
  "vITotalFrete": 25.0
}
```

## 3. Buscar Frete por ID

URL: /api/frete/{id}

Método HTTP: GET

Descrição: Retorna um frete específico com base no ID fornecido.

Exemplo de Resposta:

```
{
  "id": 1,
  "cepOrigem": "01001-000",
  "cepDestino": "02002-000",
  "dataConsulta": "2024-05-22T00:57:56.688+00:00",
  "dataPrevistaEntrega": "2024-05-23T00:57:56.688+00:00",
  "nomeDestinatario": "João da Silva",
  "peso": 10.5,
  "vITotalFrete": 25.0
}
```

#### 4. Deletar Frete por ID

URL: /api/frete/{id}

Método HTTP: DELETE

Descrição: Exclui um frete específico com base no ID fornecido.

## Execução do Projeto

Rodar o Script SQL:

1. Use um cliente MySQL (MySQL Workbench, DBeaver, etc.) para rodar o script SQL fornecido anteriormente.
2. Rodar a Aplicação:  
`./mvnw spring-boot:run`

## Testes

*Estrutura de Diretórios*

```
src
├── test
│   ├── java
│   │   ├── com
│   │   │   ├── sigabem
│   │   │   │   ├── demo
│   │   │   │   │   ├── controller
│   │   │   │   │   ├── service
│   │   │   │   │   └── repository
```

## **Testes Unitários**

*FreteServiceTest:*

*src/test/java/com/sigabem/demo/service/FreteServiceTest.java*

*FreteControllerTest:*

*src/test/java/com/sigabem/demo/controller/FreteControllerTest.java*

## **Testes de Integração**

*FreteServiceIntegrationTest:*

*src/test/java/com/sigabem/demo/service/FreteServiceIntegrationTest.java*

## **Executar Testes:**

Rodar Testes Unitários:

`./mvnw test`

Rodar Testes de Integração:

`./mvnw verify`

## **Conclusão**

Este documento detalha o processo de configuração, execução e testes do projeto Spring Boot para cálculo de frete, fornecendo todas as informações necessárias para utilizar e entender o sistema.