Documentação

Passo 1: Clone o Repositório

1. Abra o terminal ou a linha de comando em seu sistema operacional.

2. Navegue até o diretório onde você deseja armazenar o código-fonte da aplicação usando o comando cd.

3. Clone o repositório Git da aplicação. Você precisará da URL do repositório Git.

Passo 2: Instalar Dependências

Front-end

1. Navegue até o diretório da aplicação clonada usando o comando cd pelo terminal.

2. Você usara o codigo 'npm install' para instalar todas as dependencias usadas no projeto na sua maquina.

Back-end

1. Basta dar import na aplicação em sua IDE de preferência.

Passo 3: Configurar o banco de dados no Java (SpringBoot)

1. Dentro da package src/main/resources contem um arquivo chamado 'application.properties', é onde você vai usar para colocar as configurações do seu banco de dados (PostgreSQL).

2. Abrir o PostgreSQL e criar as tabelas com suas colunas. Vou deixar abaixo os comandos SQL para criar as tabelas com suas respectivas colunas.

2.1 NotaMoedaEstoque

```
CREATE TABLE nota_moeda_estoque (
    nota_moeda_estoque_id SERIAL PRIMARY KEY,
    valor NUMERIC(10, 2) NOT NULL,
    quantidade INT NOT NULL
);
```

2.2 Veiculos

```
CREATE TABLE veiculos (
    veiculo_id SERIAL PRIMARY KEY,
    placa VARCHAR(20) NOT NULL,
    tipo VARCHAR(20) NOT NULL,
    horario_entrada TIMESTAMP NOT NULL
);
```

2.3 MovimentaçãoDiaria

```
CREATE TABLE movimentacao_diaria (
    movimentacao_diaria_id SERIAL PRIMARY KEY,
    veiculo_id INT REFERENCES veiculos(veiculo_id),
    entrada TIMESTAMP NOT NULL,
    saida TIMESTAMP NOT NULL,
    valor_total NUMERIC(10, 2) NOT NULL,
```

```
    pago BOOLEAN
);
```

2.4 Usuario

```
CREATE TABLE usuario (
    id SERIAL PRIMARY KEY,
    username VARCHAR(255) NOT NULL,
    password VARCHAR(255) NOT NULL
);
```

2.5 Perfil

```
CREATE TABLE perfil (
    id SERIAL PRIMARY KEY,
    username VARCHAR(255) NOT NULL
);
```

2.6 Usuario_perfis

```
CREATE TABLE usuario_perfis (
    usuario_id BIGINT,
    perfis_id INTEGER,
    PRIMARY KEY (usuario_id, perfis_id),
    FOREIGN KEY (usuario_id) REFERENCES usuario (id),
    FOREIGN KEY (perfis_id) REFERENCES perfil (id)
);
```

Passo 4: Criar um usuario

1. Agora é necessario criar um usuario manualmente no banco de dados, para mais para frente gerar um token JWT para a autenticação na request.

2 Execute o comando SQL para criar o usuario:

INSERT INTO usuario (username, password) VALUES ('teste', '$2a$16$3UVmwywN3gpwowAzJeFwbu/lHiXjuvonOFT17CVaRxrcfr9T4amvm');

INSERT INTO perfil (username) VALUES 'teste';

INSERT INTO usuario_perfis (usuario_id, perfis_id)

VALUES (1, 1);

Passo 5: Executar a Aplicação back-end

1. Inicie a aplicação pela classe "EstacionamentoApplication".

Passo 6: Gerar o token JWT

1. Abra o Postman.

2. Coloque o caminho 'localhost:8080/auth' em uma requisição POST.

3. No Body do post, você deve colocar a informação abaixo e clicar em Send para enviar a requisição:

```
 {
   "username": "teste",
```

```
  "password": "12345678"
}
```

4. No Body de resposta você ira receber o token, salve ele.

Passo 7: Colocar o Token JWT para autenticação no front-end

1. Abra o front end na sua IDE de preferencia e vá até o arquivo 'token.interceptor.ts' que está no caminho de pasta src/app/auth.

2. Na linha 16 vai ter um campo escrito 'Bearer x', no lugar do x você deve colar o token que salvou no body do Postman.

Passo 8: Executar a Aplicação front-end

1. Após a instalação das dependências e configução do token JWT você pode iniciar a aplicação. Use o comando ng s no terminal.

Passo 9: Acessar a Aplicação

1. Com tudo já configurado, basta acessar a url 'http://localhost:4200/' e realizar os testes.