

Projeto FloodGuard

Sistema de Monitoramento Inteligente de Enchentes

Renato Silva Alexandre Bezerra | RM560928

Jhonatan Quispe Torrez | RM560601

Felipe Carlos Abreu | RM559476

2025

Sumário

Sistema de Monitoramento Inteligente de Enchentes	1
1. Introdução	3
2. Funcionalidades	3
3. Capturas de Tela	4
4. Tecnologias Utilizadas	6
5. Estrutura do Projeto	6
6. Como Executar	7
7. Contribuição	7

1. Introdução

Este projeto apresenta o FloodGuard, uma aplicação desenvolvida com Quarkus para monitoramento de níveis de água e geração de alertas de enchentes. Seu objetivo é fornecer uma API robusta que permita o cadastro de usuários, a configuração de alertas personalizados, o acompanhamento em tempo real dos níveis de água e a visualização de históricos de inundações. O sistema também possui uma interface web integrada para interação visual e controle das funcionalidades.

2. Funcionalidades

- Gerenciamento de Usuários

Permite o cadastro, login e manutenção de contas dos usuários do sistema.

- Configuração de Usuários

Usuários podem definir limites personalizados de risco para recebimento de alertas.

- Monitoramento de Nível de Água

Coleta e armazena leituras do nível da água em locais monitorados.

- Alertas de Enchente

Gera alertas automaticamente com base em níveis de água e preferências definidas.

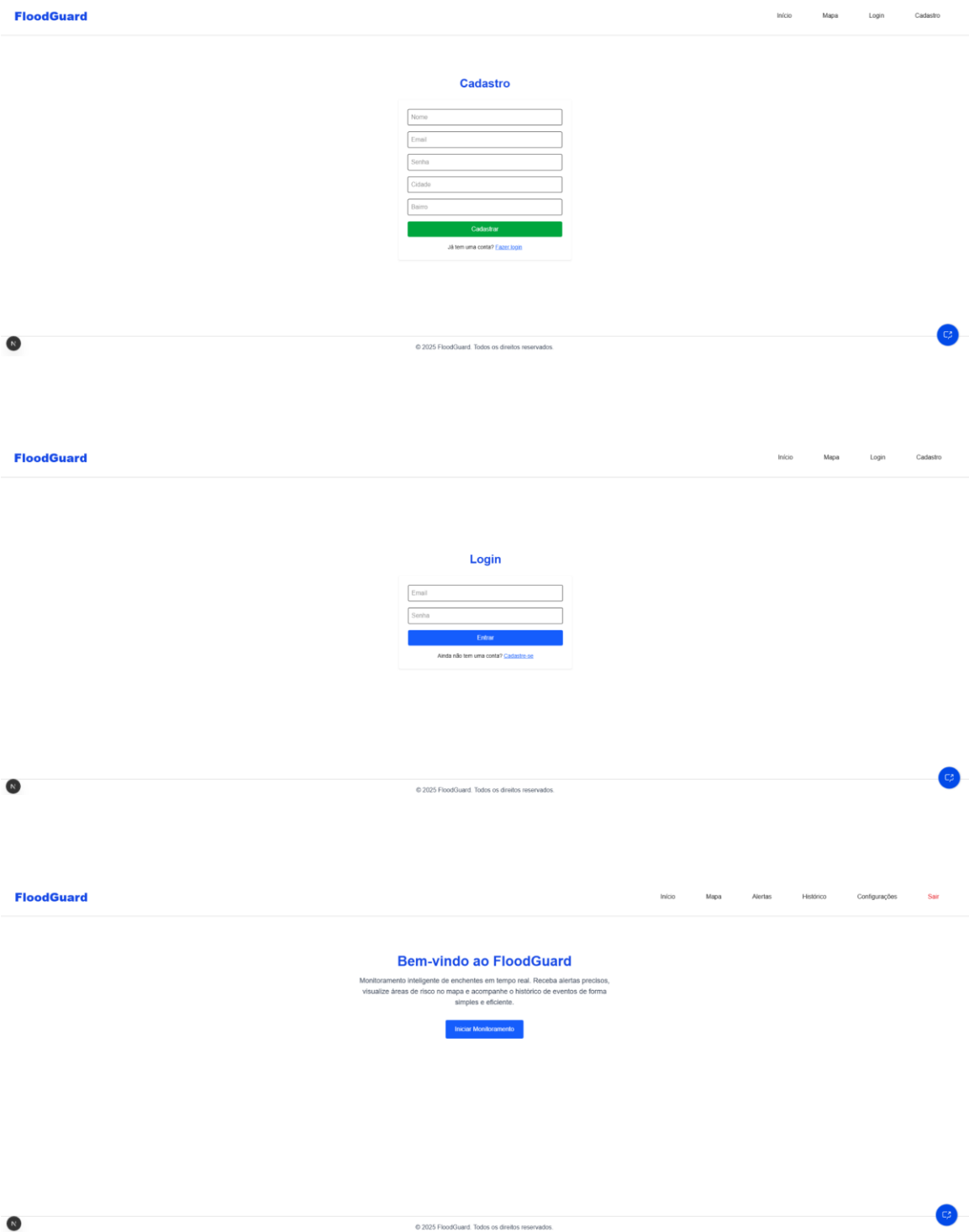
- Histórico de Enchentes

Apresenta um registro cronológico de eventos passados de enchentes com detalhes.

- Integração com OpenWeather

Possibilita obter dados climáticos externos para enriquecer o monitoramento.

3. Capturas de Tela



Mapa de Risco

Todos Ativos Encerrados Inativos



Alertas Ativos

Todos Ativos Encerrados Inativos

Histórico de Enchentes

Data Local Nivel

Configurações

Email
Nova Senha
Cidade
Bairro
Salvar Alterações

4. Tecnologias Utilizadas

- Quarkus: Framework Java para microserviços e aplicações reativas.
- Java: Linguagem principal do backend.
- Maven: Gerenciador de dependências e build da aplicação.
- RESTful API: Comunicação entre cliente e servidor.
- Docker: Suporte à containerização da aplicação.
- Banco de Dados Relacional: Persistência de dados da aplicação.

5. Estrutura do Projeto

A aplicação segue a estrutura padrão de projetos Quarkus. As principais pastas e arquivos incluem:

- `dto` : Objetos de transferência de dados.
- `entities` : Representações das tabelas do banco.
- `exceptions` : Manipulação de erros personalizados.
- `filters` : Filtros HTTP como CORS.
- `infrastructure` : Configuração da base de dados.
- `repositories` : Acesso e persistência dos dados.
- `resources` : Endpoints da API REST.
- `services` : Regras de negócio do sistema.
- `application.properties` : Configurações gerais.
- `pom.xml` : Dependências do Maven.
- `Dockerfile` : Containerização da aplicação.

6. Como Executar

1. Clone o repositório:

```
git clone https://github.com/felipecarlos1/gs-API-floodGuard.git  
cd gs-API-floodGuard
```

2. Configure o banco de dados em `application.properties`.

3. Execute a aplicação localmente:

```
./mvnw quarkus:dev
```

4. Para build de produção:

```
./mvnw package
```

5. Para executar via Docker:

```
docker build -f src/main/docker/Dockerfile.jvm -t gs-api-floodguard-jvm .  
docker run -i --rm -p 8080:8080 gs-api-floodguard-jvm
```

7. Contribuição

Contribuições são bem-vindas! Você pode colaborar abrindo issues com sugestões ou correções, ou criando pull requests com melhorias.