

Documentação do Projeto – API de Clima com MySQL e Docker

Autor: Gabriel Silva Lima de Sousa

1. Objetivo

Este projeto realiza a coleta de dados climáticos utilizando a API OpenWeather. Os dados obtidos são armazenados em um banco de dados MySQL e ficam disponíveis para consulta através de uma API REST desenvolvida em Flask. Todo o ambiente é executado através de containers Docker para garantir portabilidade e facilidade de implantação.

2. Arquitetura da Aplicação

Usuário → API Flask → Banco de Dados MySQL

↑

API Externa OpenWeather

3. Estrutura de Arquivos

```
clima-projeto/
- app.py
- requirements.txt
- Dockerfile
- docker-compose.yml
```

4. Tecnologias Utilizadas

- Python 3.12
- Flask
- Requests
- MySQL
- Docker e Docker Compose

5. API Externa – OpenWeather

Endpoint: <https://api.openweathermap.org/data/2.5/weather>

Parâmetros: q, appid, units, lang

6. Banco de Dados

```
CREATE TABLE clima (
    id INT AUTO_INCREMENT PRIMARY KEY,
    cidade VARCHAR(100),
    temperatura FLOAT,
    sensacao FLOAT,
    umidade INT,
    clima VARCHAR(100),
    data_hora TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

7. Endpoints da API REST

```
GET /
GET /clima
POST /coletar
```

8. Execução com Docker

```
docker-compose up --build
Executar em segundo plano:
docker-compose up -d --build
```

9. Testes da API

```
http://localhost:5000/
http://localhost:5000/clima
POST http://localhost:5000/coletar
```

10. Conclusão

A solução demonstra consumo de APIs externas, armazenamento em banco de dados, API REST e execução via Docker.