

CENTRO UNIVERSITÁRIO UNIFECAP

**ANÁLISE E DESENVOLVIMENTO
DE SISTEMAS**

Ingrid Xisto Santos

DOCUMENTAÇÃO DO PROJETO:

**Pipeline de Dados IoT com Docker e
PostgreSQL**

Taboão da Serra, SP

2025

INGRID XISTO SANTOS

DOCUMENTAÇÃO DO PROJETO:

Pipeline de Dados IoT com Docker e PostgreSQL

Trabalho apresentado como requisito parcial de avaliação da disciplina **Disruptive Architectures: IOT, Big Data e IA** do Curso de Graduação em **Análise e Desenvolvimento de Sistemas** do Centro Universitário UniFECAF.

Tutor(a): **Felipe Bonatto.**

Taboão da Serra, SP

2025

SUMÁRIO

Ingrid Xisto Santos	1
1 CONTEXTUALIZAÇÃO DO PROJETO	4
2 ESTRUTURA DO PROJETO	5
3 PASSOS REALIZADOS	6
3.1 Configuração do Ambiente	6
4 CONFIGURAÇÃO E EXECUÇÃO	6
4.1 Rodando o PostgreSQL com Docker.....	6
4.2 Criando a Tabela no PostgreSQL	8
4.3 Inserindo os Dados.....	9
4.4 Tabela no Dbeaver	9
4.5 Tabela no Streamlit	10
4.6 Criando as Views SQL.....	10
5 DASHBOARD STREAMLIT.....	14
5.1 Executando.....	14
5.2 Prints do Dashboard Streamlit e suas funcionalidades	14
6 INSIGHTS OBTIDOS.....	17
7 REFERÊNCIAS.....	18

1 CONTEXTUALIZAÇÃO DO PROJETO

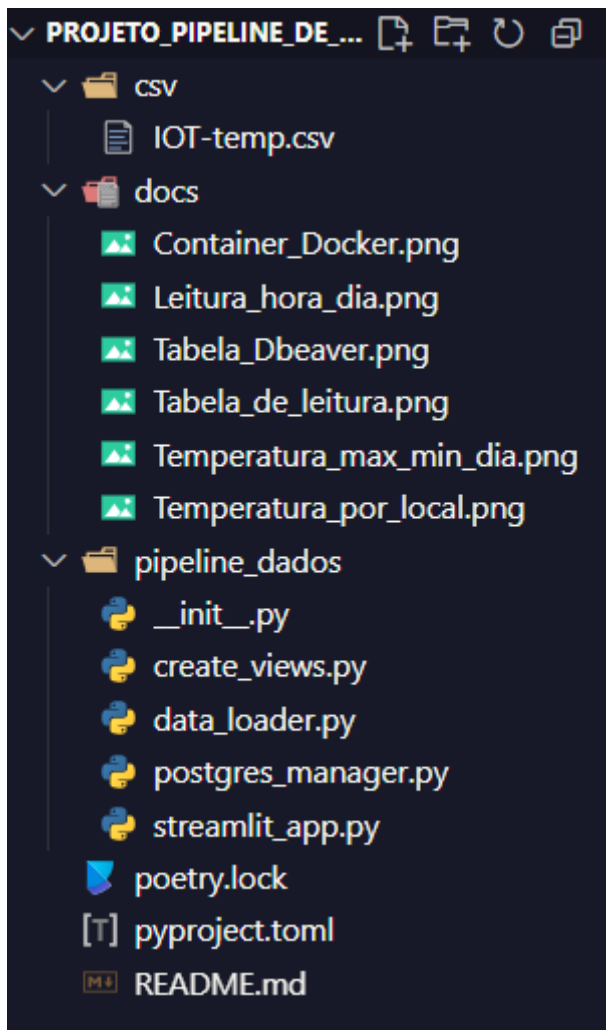
O projeto consiste em um **pipeline de dados** que processa leituras de temperatura de dispositivos IoT e armazena os dados em um **banco de dados PostgreSQL** usando Docker.

O objetivo é demonstrar o fluxo completo de dados:

1. **Extração:** leitura de um arquivo CSV com dados de temperatura de sensores IoT.
2. **Processamento:** conversão de timestamps, tratamento de valores numéricos e padronização dos dados.
3. **Armazenamento:** inserção dos dados no PostgreSQL dentro de uma tabela estruturada.
4. **Visualização:** criação de um dashboard em Streamlit para análise das informações.

Esse pipeline permite analisar padrões de temperatura em diferentes ambientes (internos e externos), identificar picos de atividade e monitorar o funcionamento dos sensores.

2 ESTRUTURA DO PROJETO



3 PASSOS REALIZADOS

3.1 Configuração do Ambiente

- Python 3.9 ou superior
- Docker
- PostgreSQL
- Criação de ambiente virtual Python
- Bibliotecas Python:

```
Pip install pandas psycopg2-binary sqlalchemy streamlit plotly
```

4 CONFIGURAÇÃO E EXECUÇÃO

4.1 Rodando o PostgreSQL com Docker

```
docker run --name postgres-iot -e POSTGRES_PASSWORD=12345 -e POSTGRES_DB=database -p 5432:5432 -d postgres
```

The screenshot shows the Docker Desktop application window. The top bar includes the Docker logo, a search bar, and a 'Sign in' button. The left sidebar contains navigation options: Containers, Images, Volumes, Builds, Dev Environments, Docker Scout, and Extensions. The main panel displays the 'postgres-iot' container, which is running. The 'Logs' tab is selected, showing a detailed log of the container's startup and operation. The logs include timestamps, container IDs, and various log messages from the PostgreSQL database system, such as 'received fast shutdown request', 'aborting any active transactions', 'terminating connection due to administrator command', 'background worker "logical replication launcher" (PID 32) exited with exit code 1', 'shutting down', 'checkpoint starting: shutdown immediate', 'checkpoint complete: wrote 0 buffers (0.0%); 0 WAL file(s) added, 0 removed, 0 recycled; wr', 'database system is shut down', 'starting PostgreSQL 17.5 (Debian 17.5-1.pgdg130+1) on x86_64-pc-linux-gnu, compiled by gcc', 'listening on IPv4 address "0.0.0.0", port 5432', 'listening on IPv6 address ":::", port 5432', 'listening on Unix socket "/var/run/postgresql/.s.PGSQL.5432"', 'database system was shut down at 2025-09-02 12:25:06 UTC', 'database system is ready to accept connections', 'checkpoint starting: time', 'checkpoint complete: wrote 6 buffers (0.0%); 0 WAL file(s) added, 0 removed, 0 recycled; wr', 'PostgreSQL Database directory appears to contain a database; Skipping initialization'. The bottom status bar shows 'Engine running', 'RAM 1.82 GB', 'CPU 10.51%', 'Not signed in', and a 'New version available' notification.

postgres-iot

STATUS
Running (13 minutes ago)

Logs Inspect Bind mounts Exec Files Stats

0/3F1EEA0

2025-09-02 09:25:05 2025-09-02 12:25:05.978 UTC [1] LOG: received fast shutdown request

2025-09-02 09:25:06 2025-09-02 12:25:06.056 UTC [1] LOG: aborting any active transactions

2025-09-02 09:25:06 2025-09-02 12:25:06.112 UTC [34] FATAL: terminating connection due to administrator command

2025-09-02 09:25:06 2025-09-02 12:25:06.116 UTC [33] FATAL: terminating connection due to administrator command

2025-09-02 09:25:06 2025-09-02 12:25:06.189 UTC [1] LOG: background worker "logical replication launcher" (PID 32) exited with exit code 1

2025-09-02 09:25:06 2025-09-02 12:25:06.256 UTC [27] LOG: shutting down

2025-09-02 09:25:06 2025-09-02 12:25:06.292 UTC [27] LOG: checkpoint starting: shutdown immediate

2025-09-02 09:25:06 2025-09-02 12:25:06.421 UTC [27] LOG: checkpoint complete: wrote 0 buffers (0.0%); 0 WAL file(s) added, 0 removed, 0 recycled; wr
ite=0.070 s, sync=0.001 s, total=0.154 s; sync files=0, longest=0.000 s, average=0.000 s; distance=0 kB, estimate=111 kB; lsn=0/3F1EFA8, redo lsn=0/3F
1EFA8

2025-09-02 09:25:06 2025-09-02 12:25:06.548 UTC [1] LOG: database system is shut down

2025-09-02 10:12:38 2025-09-02 13:12:38.784 UTC [1] LOG: starting PostgreSQL 17.5 (Debian 17.5-1.pgdg130+1) on x86_64-pc-linux-gnu, compiled by gcc (

Debian 14.2.0-19) 14.2.0, 64-bit

2025-09-02 10:12:38 2025-09-02 13:12:38.791 UTC [1] LOG: listening on IPv4 address "0.0.0.0", port 5432

2025-09-02 10:12:38 2025-09-02 13:12:38.791 UTC [1] LOG: listening on IPv6 address ":::", port 5432

2025-09-02 10:12:38 2025-09-02 13:12:38.798 UTC [1] LOG: listening on Unix socket "/var/run/postgresql/.s.PGSQL.5432"

2025-09-02 10:12:38 2025-09-02 13:12:38.850 UTC [29] LOG: database system was shut down at 2025-09-02 12:25:06 UTC

2025-09-02 10:12:38 2025-09-02 13:12:38.937 UTC [1] LOG: database system is ready to accept connections

2025-09-02 10:17:40 2025-09-02 13:17:40.355 UTC [27] LOG: checkpoint starting: time

2025-09-02 10:17:43 2025-09-02 13:17:43.344 UTC [27] LOG: checkpoint complete: wrote 6 buffers (0.0%); 0 WAL file(s) added, 0 removed, 0 recycled; wr
ite=0.020 s, sync=0.137 s, total=4.085 s; sync files=4, longest=0.093 s, average=0.035 s; distance=0 kB, estimate=0 kB; lsn=0/3F1F000, redo lsn=0/3F1F
058

2025-09-02 10:12:38

2025-09-02 10:12:38 PostgreSQL Database directory appears to contain a database; Skipping initialization

2025-09-02 10:12:38

Engine running RAM 1.82 GB CPU 10.51% Not signed in New version available 8

4.2 Criando a Tabela no PostgreSQL

Execute o script *postgres_manager.py* para criar a tabela *iot_temp_log*:

```
# Criação da tabela
def create_table(engine):
    query = """
        CREATE TABLE IF NOT EXISTS iot_temp_log (
            id VARCHAR(100) PRIMARY KEY,
            room_id VARCHAR(100),
            noted_date TIMESTAMP,
            temperature FLOAT,
            location VARCHAR(10)
        );
    """
    with engine.begin() as conn:
        conn.execute(text(query))
```

Executando:

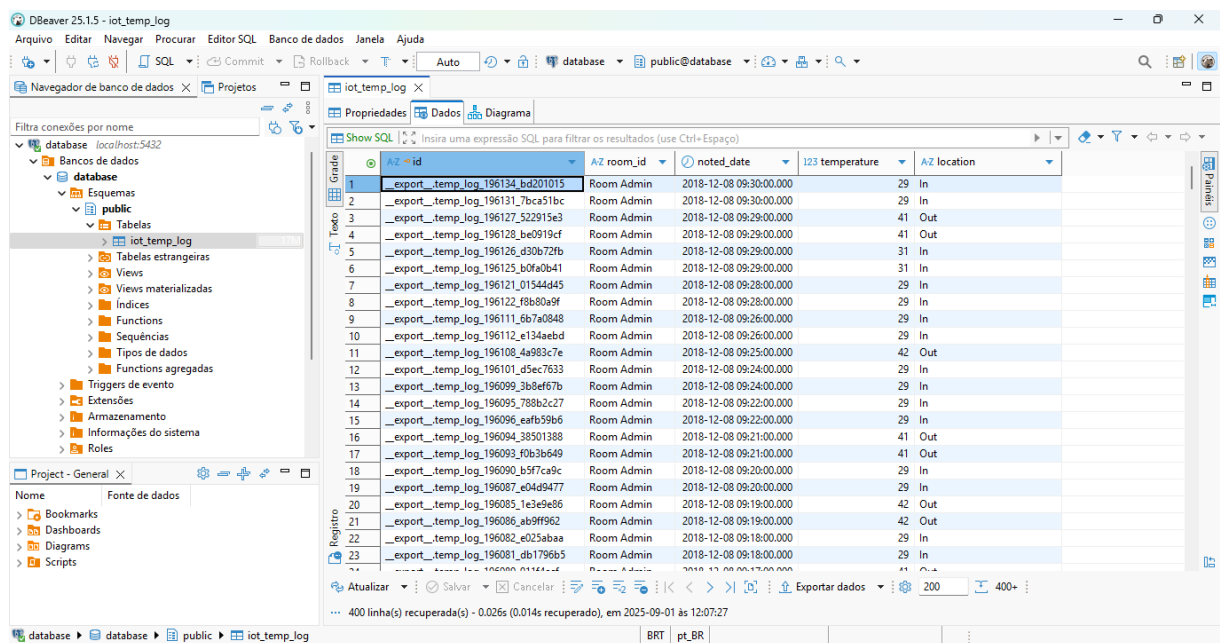
```
python pipeline_dados/postgres_manager.py
```


4.3 Inserindo os Dados

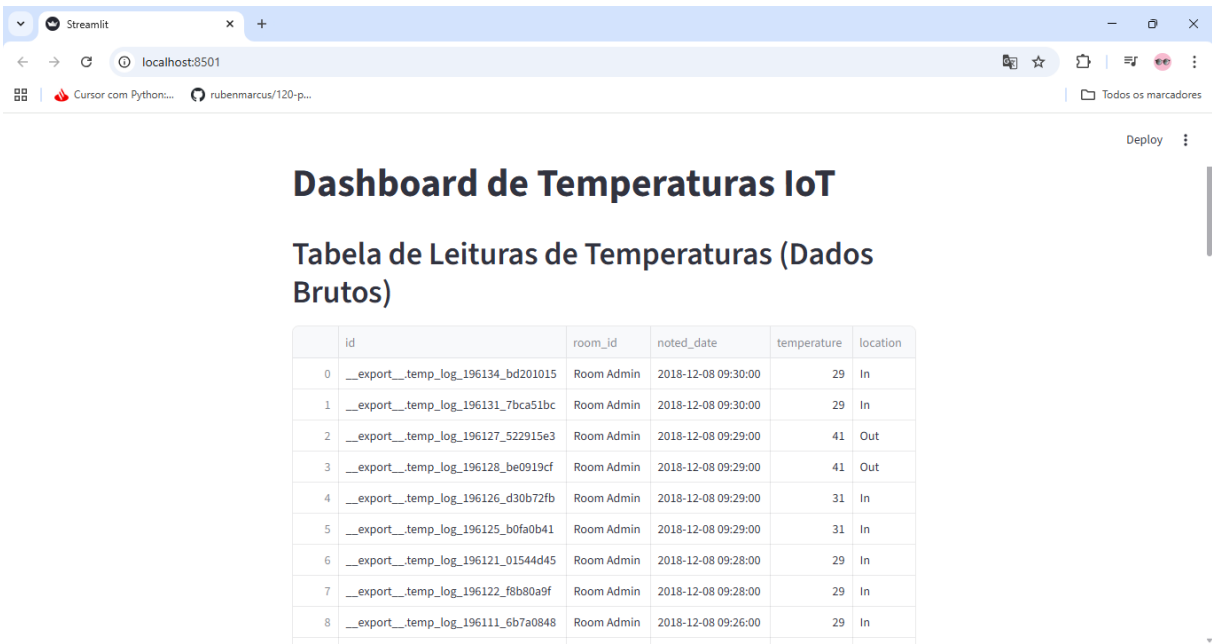
Use `data_loader.py` para ler o CSV e inserir os dados na tabela `iot_temp_log`.

- Converte o timestamp para o formato PostgreSQL.
- Converte a temperatura para float.
- Ignora conflitos de IDs duplicados.

4.4 Tabela no Dbeaver



4.5 Tabela no Streamlit



4.6 Criando as Views SQL

→ View 1: Média de temperatura por local (In / Out)

```
def create_views():  
    with engine.begin() as conn:  
        # Média de temperatura por local (In / Out)  
        conn.execute(text("""  
            CREATE OR REPLACE VIEW avg_temp_por_local AS  
            SELECT location, AVG(temperature) AS avg_temp  
            FROM iot_temp_log  
            GROUP BY location;  
        """))
```

Explicação:

- Esta view calcula a **média das temperaturas** registradas por cada tipo de sensor, identificado pela coluna `location` (“In” para sensores internos e “Out” para sensores externos).
- **Propósito:** Permite comparar a temperatura média entre sensores internos e externos, mostrando se há diferença significativa entre o ambiente interno e externo.

→ **View 2:** Leituras por hora do dia

```
# Contagem de leituras por hora
conn.execute(text("""
    CREATE OR REPLACE VIEW leituras_por_hora AS
    SELECT EXTRACT(HOUR FROM noted_date) AS hora, COUNT(*) AS contagem
    FROM iot_temp_log
    GROUP BY EXTRACT(HOUR FROM noted_date)
    ORDER BY hora;
"""))
```

Explicação:

- Esta view conta **quantas leituras foram registradas em cada hora do dia**.
- A função `EXTRACT(HOUR FROM noted_date)` extrai a hora do timestamp `noted_date`.
- **Propósito:** Identificar padrões de atividade ou horários de pico nos sensores. Por exemplo, se a maior parte das leituras ocorre em determinados períodos, podemos entender melhor o uso dos dispositivos ou o comportamento ambiental.

→ **View 3:** Temperatura máxima e mínima por dia

```
# Temperatura máxima e mínima por dia
conn.execute(text("""
    CREATE OR REPLACE VIEW temp_max_min_por_dia AS
    SELECT DATE(noted_date) AS data,
           MAX(temperature) AS temp_max,
           MIN(temperature) AS temp_min
    FROM iot_temp_log
    GROUP BY DATE(noted_date)
    ORDER BY data;
"""))
```

Explicação:

- Esta view calcula a **temperatura máxima e mínima registrada em cada dia**.
- A função `DATE(noted_date)` converte o timestamp para apenas a data, descartando hora, minutos e segundos.
- **Propósito:** Permitir análise de variação diária da temperatura, ajudando a identificar extremos de temperatura e padrões ao longo do tempo.

5 DASHBOARD STREAMLIT

5.1 Executando

```
streamlit run pipeline_dados/streamlit_app.py
```

5.2 Prints do Dashboard Streamlit e suas funcionalidades

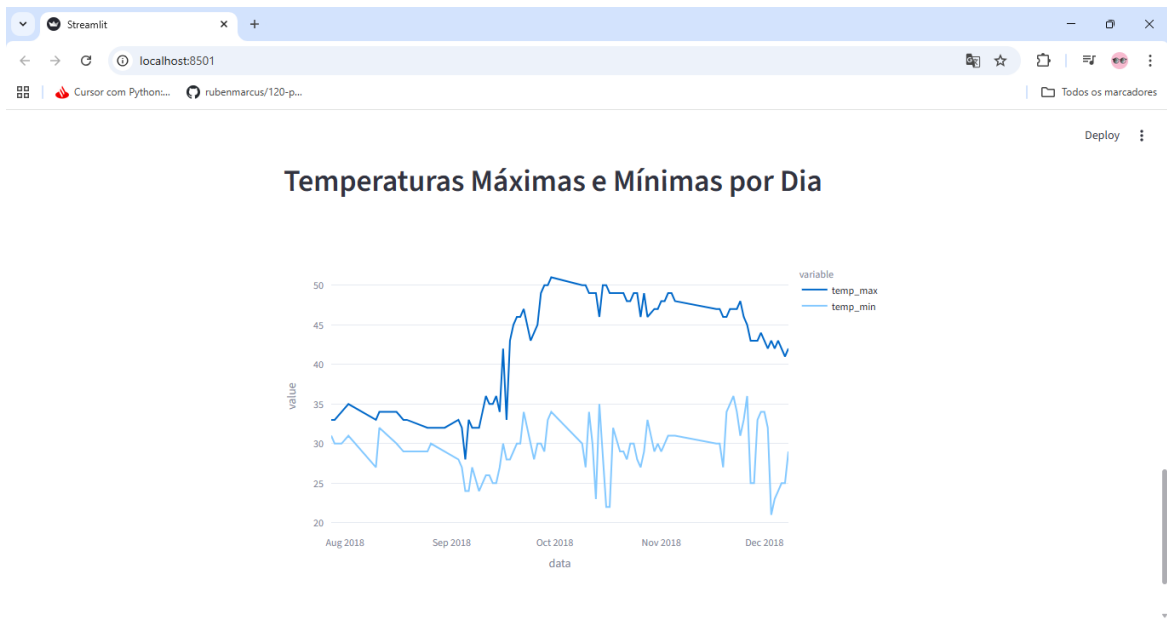
- **Gráfico de barras da média de temperatura por local (In / Out):** gráfico de barras mostrando a média de sensores internos e externos.



- **Leituras por hora do dia:** gráfico de linha mostrando a quantidade de leituras por horas.



- **Temperaturas máximas e mínimas por dia:** gráfico de linha mostrando a variação diária da temperatura.



6 INSIGHTS OBTIDOS

- Sensores internos (In) apresentam temperaturas mais estáveis, refletindo um ambiente mais controlado.
- Sensores externos (Out) apresentam maior variação, provavelmente influenciados pelo clima.
- Horários de maior registro de leituras indicam picos de atividade do sistema IoT.
- Dias com variação extrema de temperatura podem indicar eventos climáticos ou falhas nos sensores.

7 REFERÊNCIAS

ANANDJHA, Atul. *Temperature Readings: IoT Devices*. Kaggle, 2022. Disponível em: <https://www.kaggle.com/datasets/atulanandjha/temperature-readings-iot-devices>. Acesso em: 02 set. 2025.

STREAMLIT. *Streamlit Documentation*. Disponível em: <https://docs.streamlit.io>. Acesso em: 02 set. 2025.

SQLALCHEMY. *SQLAlchemy 2.0 Documentation*. Disponível em: <https://docs.sqlalchemy.org/en/20/>. Acesso em: 02 set. 2025.

PROFESSOR BRANDÃO, Afonso. *Unidade 4 – Disruptive Architectures: IoT, Big Data e IA*. Curso de Análise e Desenvolvimento de Sistemas. Faculdade UniFECAF – Ensino a Distância. 2025. Aula disponível na plataforma da instituição: <https://fecaf.brightspace.com/d21/le/enhancedSequenceViewer/26204?url=https%3A%2F%2Fcabd6a42-976a-4fa2-80a8-36b0ec721d14.sequences.api.brightspace.com%2F26204%2Factivity%2F482097%3FfilterOnDatesAndDepth%3D1>