



Relatório Teórico – Pipeline de Dados com IoT e Docker

1. Introdução

Este relatório foi desenvolvido como parte da disciplina Disruptive Architectures: IoT, Big Data e IA.

O objetivo é apresentar a construção de um pipeline de dados com dispositivos IoT, utilizando Python, PostgreSQL, Docker e Streamlit.

A proposta consiste em processar leituras de sensores de temperatura, armazená-las em um banco de dados relacional e criar um dashboard interativo para análise.

2. Contextualização do Projeto

- Atualmente, dispositivos IoT geram grandes volumes de dados continuamente. Para que esses dados sejam úteis, é necessário um pipeline capaz de **ingestar, armazenar, processar e visualizar informações** de forma escalável.
- Neste trabalho, simulamos um cenário de monitoramento de temperatura em diferentes ambientes. O pipeline recebe dados de sensores, organiza em um banco PostgreSQL e exibe gráficos interativos via Streamlit.

3. Etapas Realizadas

3.1 Configuração do Ambiente

- Instalação do **Python 3.11**.
- Instalação do **Docker** para criar o banco PostgreSQL em contêiner.
- Criação de ambiente virtual (**venv**).

Instalação de bibliotecas necessárias:

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

-

3.2 Criação do Contêiner PostgreSQL

O banco foi criado com o seguinte comando:

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

3.3 Inserção de Dados no Banco

Um script Python foi desenvolvido para ler o **CSV do Kaggle** e inserir os dados no PostgreSQL:

- Limpeza de dados antigos.
- Inserção dos novos registros na tabela `temperature_readings`.

3.4 Criação das Views SQL

Foram criadas 3 views no PostgreSQL para facilitar consultas analíticas:

1. **Média de temperatura por dispositivo.**
2. **Contagem de leituras por hora do dia.**
3. **Temperaturas máxima e mínima por dia.**

3.5 Construção do Dashboard

Um dashboard em **Streamlit** foi criado para visualizar os dados.

Os gráficos foram feitos com **Plotly Express**, permitindo interação e filtros dinâmicos.

4. Explicação das Views SQL

1. `avg_temp_por_dispositivo`

- `CREATE VIEW avg_temp_por_dispositivo AS`
- `SELECT device_id, AVG(temperature) AS avg_temp`

- FROM temperature_readings
- GROUP BY device_id;

Propósito: calcular a **média de temperatura** registrada por cada dispositivo.

Insight: permite identificar dispositivos que registram temperaturas mais altas ou mais baixas, ajudando a detectar anomalias ou ambientes específicos.

2. leituras_por_hora

- CREATE VIEW leituras_por_hora AS
- SELECT EXTRACT(HOUR FROM reading_time) AS hora, COUNT(*) AS contagem
- FROM temperature_readings
- GROUP BY hora
- ORDER BY hora;

Propósito: mostrar quantas leituras de temperatura foram registradas em cada hora do dia.

Insight: ajuda a verificar padrões de coleta — por exemplo, se os sensores coletam mais em determinados horários.

3. temp_max_min_por_dia

- CREATE VIEW temp_max_min_por_dia AS
- SELECT DATE(reading_time) AS data,
- MAX(temperature) AS temp_max,
- MIN(temperature) AS temp_min
- FROM temperature_readings
- GROUP BY DATE(reading_time)
- ORDER BY data;

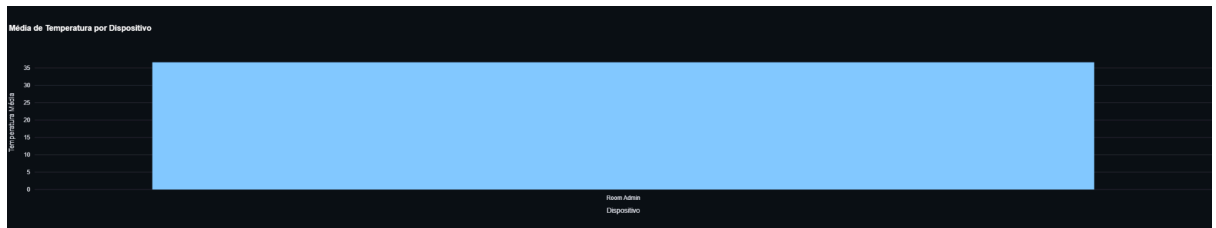
Propósito: mostrar a **temperatura máxima e mínima de cada dia**.

Insight: permite analisar variações climáticas ao longo do tempo e identificar dias atípicos.

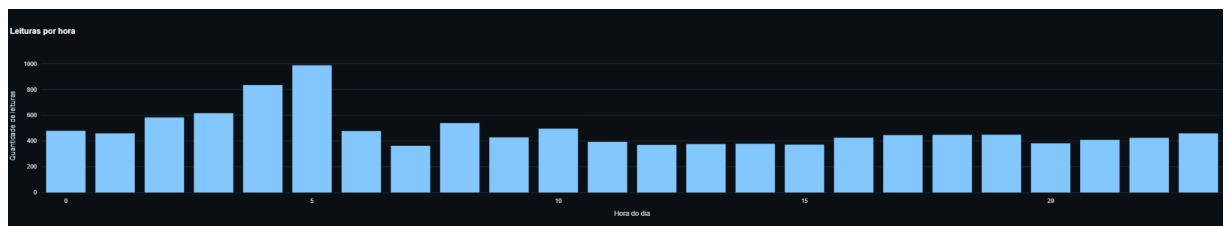
5. Prints do Dashboard

Os três gráficos principais gerados no Streamlit foram:

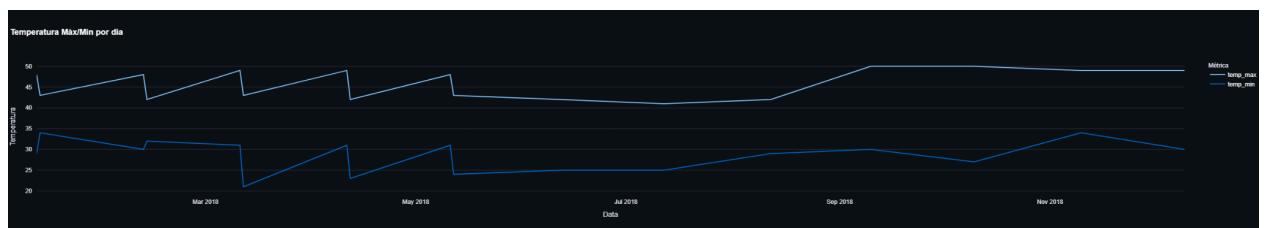
1. **Média de Temperatura por Dispositivo**
- 2.



3. **Contagem de Leituras por Hora do Dia**



4. **Temperatura Máxima e Mínima por Dia**



6. Principais Insights Obtidos

- O dispositivo **Room Admin** é responsável por todas as medições do dataset, sendo o ponto central de análise.
- A maior parte das leituras ocorre de forma distribuída ao longo do dia, mas há horários com picos de coleta.
- A análise de **temperatura máxima e mínima por dia** mostra variações importantes que poderiam ser usadas em sistemas de monitoramento climático ou automação predial.
-

7. Referências

- Kaggle – *Temperature Readings: IoT Devices*:
<https://www.kaggle.com/datasets/atulanandjha/temperature-readings-iot-devices>
- Docker Docs: <https://docs.docker.com/>
- Streamlit Docs: <https://docs.streamlit.io/>
- SQLAlchemy Docs: <https://docs.sqlalchemy.org/>