

CENTRO UNIVERSITÁRIO UNIFECAP
ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

Pâmela Xisto dos Santos

PIPELINE DE DADOS COM IOT E DOCKER
Análise de Dados

Taboão da Serra, SP

2025
PÂMELA XISTO DOS SANTOS

PIPELINE DE DADOS COM IOT E DOCKER Análise de Dados

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

1 INTRODUÇÃO.....	4
2 ETAPAS DO PROJETO	5
3 DASHBOARDS EM FUNCIONAMENTO (STREAMLIT)	7
4 INSIGHTS OBTIDOS E SUGESTÕES REAIS	10
REFERÊNCIAS	12

1 INTRODUÇÃO

O projeto tem como objetivo implementar um **pipeline de dados para dispositivos IoT**, especificamente sensores de temperatura. Dispositivos IoT enviam dados de temperatura periodicamente, e o pipeline tem como proposta:

- Capturar esses dados via arquivo CSV;
- Armazenar os dados em um banco de dados relacional PostgreSQL, facilitando consultas e análises;
- Fornecer uma **visualização interativa** por meio de um dashboard em Streamlit, permitindo análise de tendências, médias e comportamentos dos sensores;
- Utilizar Docker para facilitar a criação e execução do ambiente de banco de dados, garantindo portabilidade e reprodutibilidade.

2 ETAPAS DO PROJETO

2.1 Configuração do Ambiente Python

3.10+ instalado.

Criação de ambiente virtual Python:

```
python -m venv venv
source venv/bin/activate # Linux/Mac
venv\Scripts\activate.bat #
```

Instalação das dependências do projeto:

```
pip install -r requirements.txt
```

2.2 Criação do Container Docker com PostgreSQL

Baixou-se a imagem oficial do PostgreSQL:

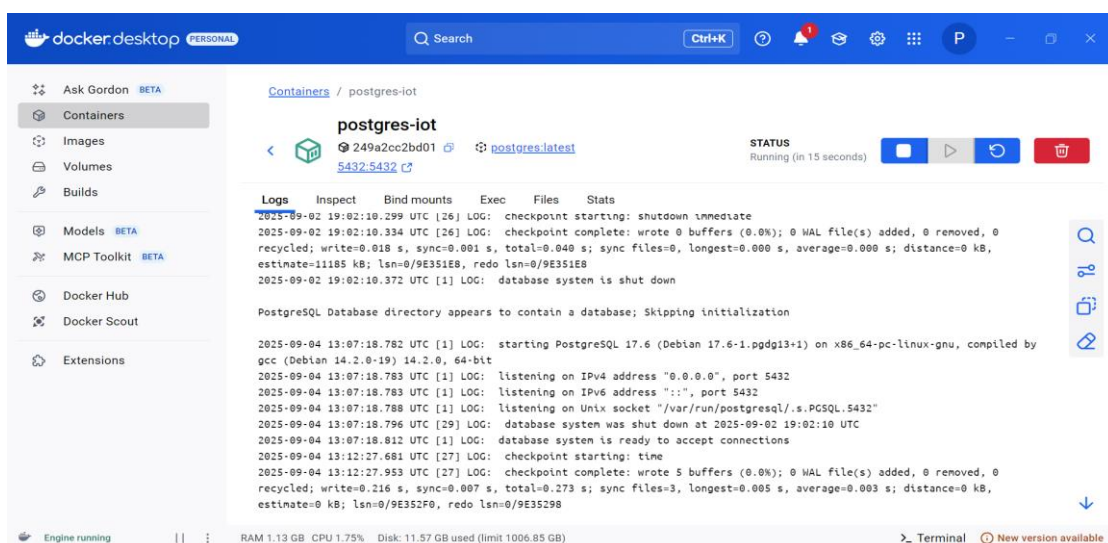
```
docker pull postgres:latest
```

-Criação e execução do container:

```
docker run --name postgres-iot -e POSTGRES_USER=postgres -e
POSTGRES_PASSWORD=123456 -e POSTGRES_DB=iot_db -p 5432:5432 -d postgres
```

-Verificação do container ativo:

```
docker ps
```



2.3 Inserção de Dados no Banco

Arquivo CSV com leituras de sensores: data/IOT-temp.csv

Script Python main.py conecta ao PostgreSQL e a tabela principal temperature_logs.

```
engine = get_db_connection() df.to_sql('temperature_logs', engine,
if_exists='replace', index=False)
```

id	room_id	noted_date	temp	out/in
1	Room Admin	08-12-2018 09:30	29	In
2	Room Admin	08-12-2018 09:30	29	In
3	Room Admin	08-12-2018 09:29	41	Out
4	Room Admin	08-12-2018 09:29	41	Out
5	Room Admin	08-12-2018 09:29	31	In
6	Room Admin	08-12-2018 09:29	31	In
7	Room Admin	08-12-2018 09:28	29	In
8	Room Admin	08-12-2018 09:28	29	In
9	Room Admin	08-12-2018 09:26	29	In
10	Room Admin	08-12-2018 09:26	29	In
11	Room Admin	08-12-2018 09:25	42	Out
12	Room Admin	08-12-2018 09:25	42	Out
13	Room Admin	08-12-2018 09:24	29	In
14	Room Admin	08-12-2018 09:24	29	In
15	Room Admin	08-12-2018 09:22	29	In
16	Room Admin	08-12-2018 09:22	29	In
17	Room Admin	08-12-2018 09:21	41	Out
18	Room Admin	08-12-2018 09:21	41	Out
19	Room Admin	08-12-2018 09:20	29	In

2.4 Criação das Views SQL

Foram desenvolvidas três views no banco de dados PostgreSQL para simplificar e otimizar as consultas necessárias ao dashboard.

- **avg_temp_por_dispositivo:** calculada para fornecer a temperatura média registrada por cada dispositivo IoT, facilitando a análise.

```
# Gráfico 1: Média de temperatura por dispositivo
st.header("Média de Temperatura por Dispositivo")
df_avg_temp = load_data("avg_temp_por_dispositivo")
fig1 = px.bar(df_avg_temp, x="device_id", y="avg_temp", Labels={"avg_temp": "Temperatura Média", "device_id": "Dispositivo"})
st.plotly_chart(fig1)
```

- **leituras_por_hora:** total de leituras feitas em cada hora do dia, evidenciando padrões temporais e frequência de coleta dos dados.

```
# Gráfico 2: Leituras por hora
st.header("Leituras por Hora do Dia")
df_leituras_hora = load_data("leituras_por_hora")
fig2 = px.line(df_leituras_hora, x="hora", y="contagem", Labels={"contagem": "Número de Leituras"})
st.plotly_chart(fig2)
```

- **temp_max_min_por_dia:** apresenta as temperaturas máximas e mínimas observadas diariamente, oferecendo uma visão clara das variações térmicas no ambiente monitorado.

```
# Gráfico 3: Temperaturas Máximas e Mínimas por dia
st.header("Temperaturas Máximas e Mínimas por Dia")
df_temp_max_min = load_data("temp_max_min_por_dia")
fig3 = px.line(df_temp_max_min, x="data", y=["temp_max", "temp_min"], labels={"value": "Temperatura"})
st.plotly_chart(fig3)
```

3 DASHBOARDS EM FUNCIONAMENTO (STREAMLIT)

3.1 Dashboard de Temperaturas IOT

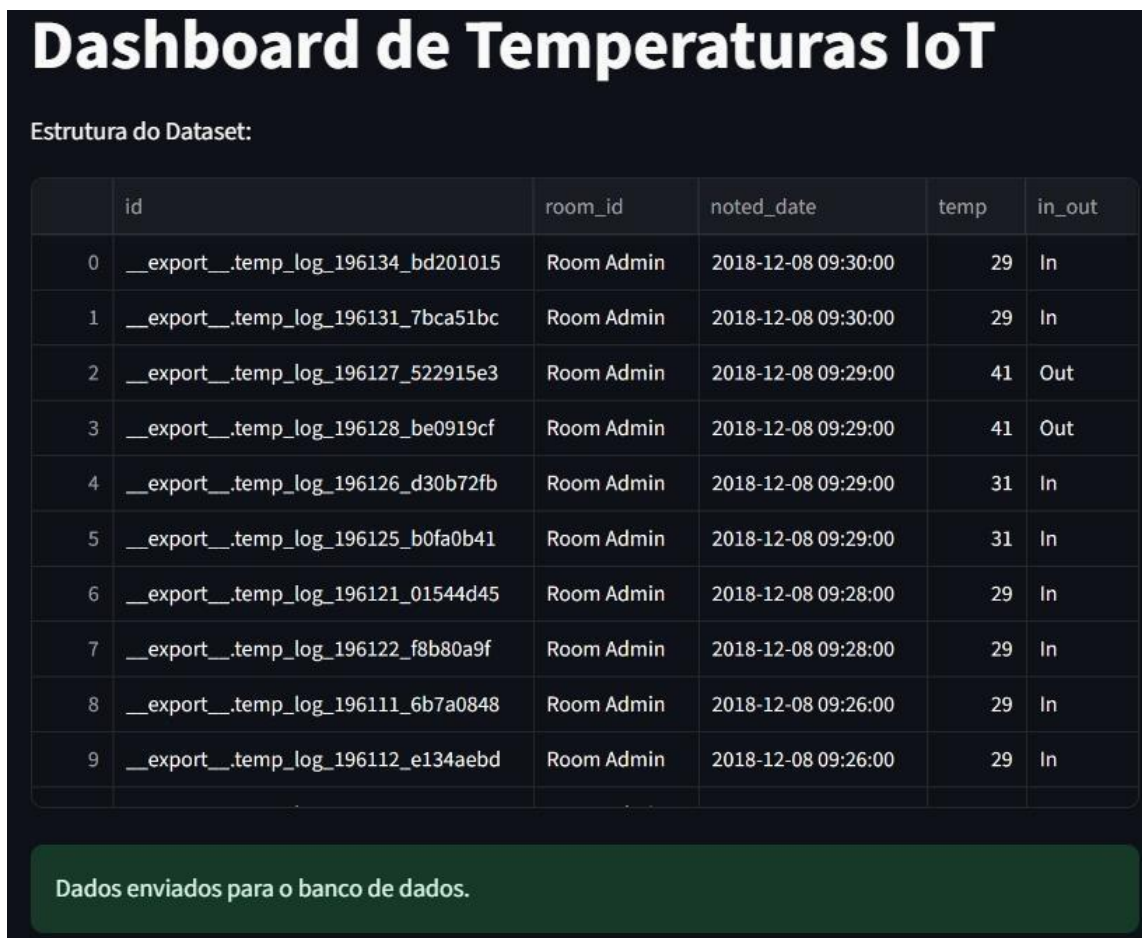


Figura 1.

A Figura 1 apresenta a visualização inicial dos dados carregados no Streamlit, exibindo as primeiras linhas do dataset. As colunas disponíveis incluem o identificador do registro, o dispositivo de origem da medição, a data e hora da coleta, a temperatura registrada e a indicação do ambiente (*interno* ou *externo*). Além disso, observa-se a confirmação de que os dados foram enviados com sucesso para o banco de dados PostgreSQL.

3.2 Média de Temperatura por Dispositivo



Figura 2.

O gráfico da Figura 2 apresenta a **temperatura média registrada por dispositivo IoT**. No eixo horizontal (X), encontram-se os dispositivos responsáveis pela coleta dos dados; no caso analisado, o dispositivo “Room Admin”. No eixo vertical (Y), visualiza-se a temperatura média de 35°C em graus Celsius.

3.3 Temperaturas Máximas e Mínimas por Dia

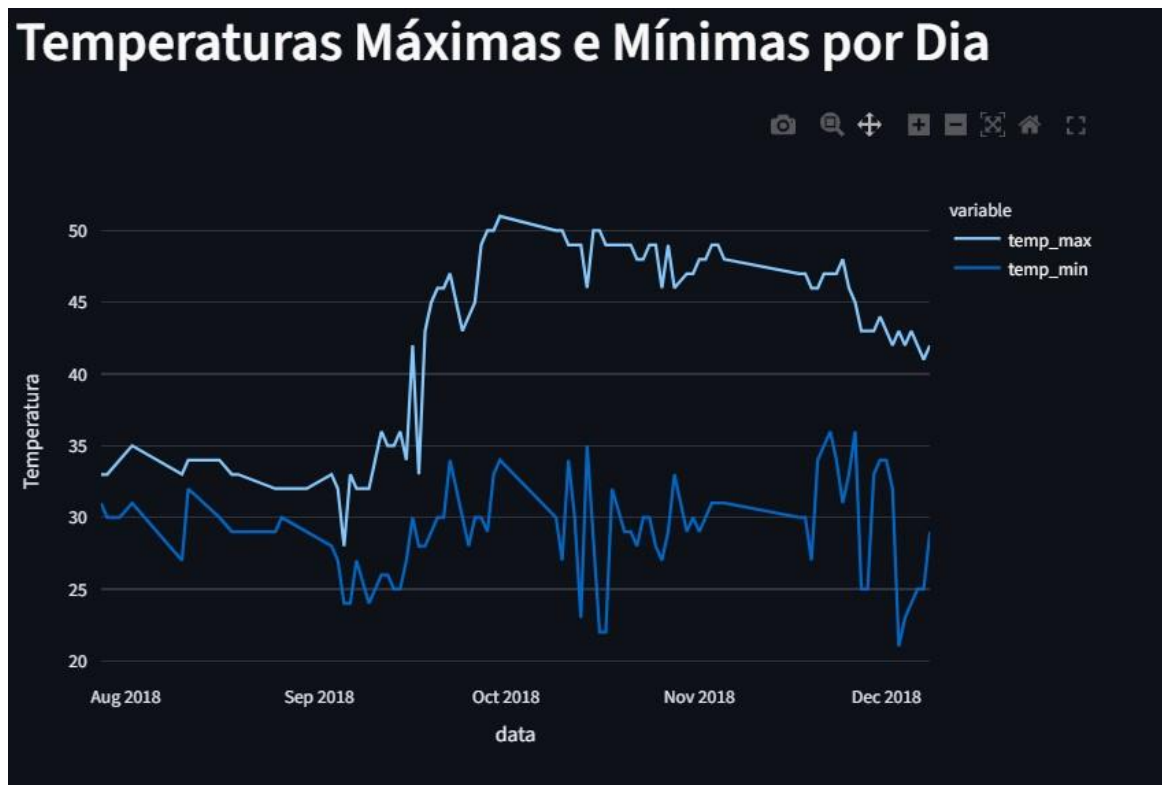


Figura 3.

A Figura 3 apresenta a variação das temperaturas máximas e mínimas registradas diariamente pelos dispositivos IoT. O eixo horizontal mostra o período analisado (agosto a dezembro de 2018), enquanto o eixo vertical representa as temperaturas em graus Celsius.

Observa-se que a temperatura máxima variou entre 35 °C e 50 °C, com destaque para o mês de outubro, em que ocorreram os maiores picos. Já a temperatura mínima oscilou entre 20 °C e 35 °C, com variações bruscas em diferentes dias.

3.4 Leituras por Hora do Dia

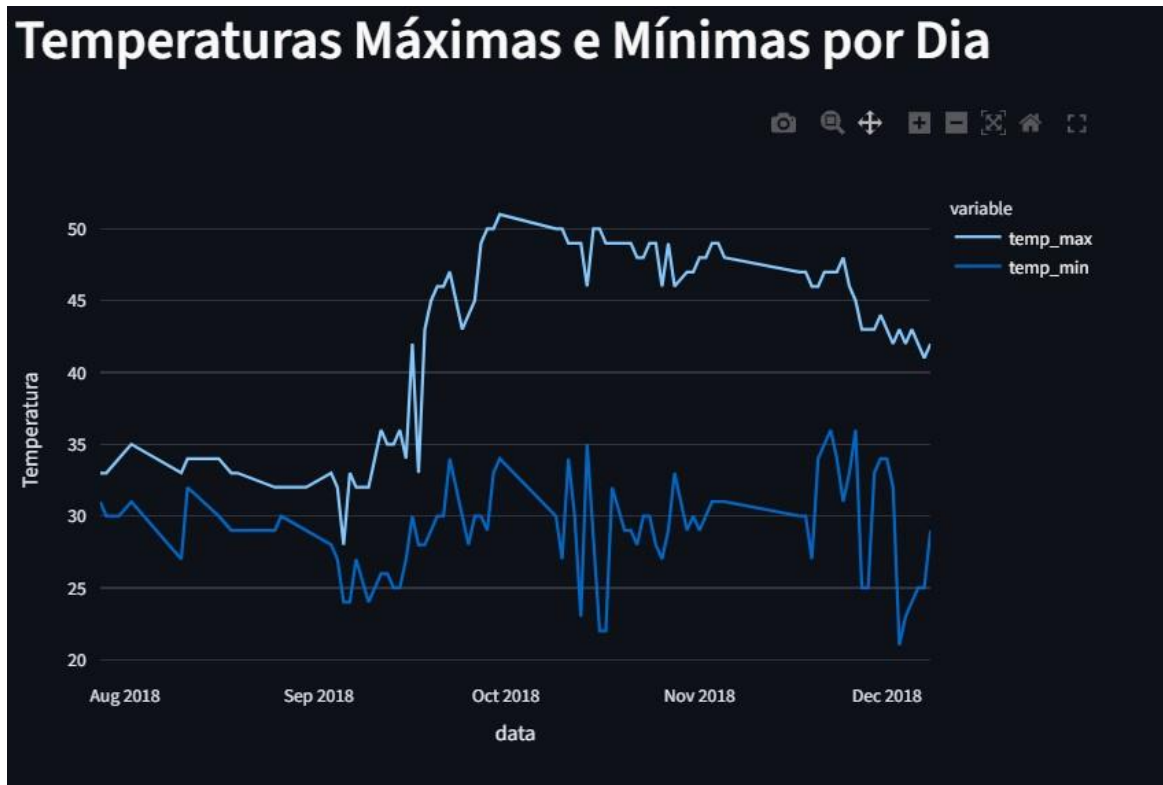


Figura 4.

A Figura 4 apresenta a quantidade de leituras de temperatura registradas ao longo das horas do dia. O eixo horizontal representa as horas (0h a 23h), enquanto o eixo vertical mostra o número total de registros realizados pelos dispositivos IoT.

Constata-se que há horários com maior concentração de leituras, indicando padrões de coleta de dados ao longo do dia. Esses picos podem estar relacionados a períodos de maior atividade ou a configurações específicas de monitoramento, permitindo identificar tendências de funcionamento dos sensores em tempo real.

4 INSIGHTS OBTIDOS E SUGESTÕES REAIS

A análise dos dados IoT permitiu identificar padrões de temperatura ao longo do tempo e entre aparelhos, detectando variações anormais que indicam falhas ou necessidade de manutenção. O monitoramento contínuo das leituras por hora revelou períodos de maior atividade ou instabilidades na coleta.

As temperaturas máximas e mínimas diárias mostraram a variação do ambiente monitorado. Em ambientes reais, como fábricas, hospital e laboratórios, a solução facilita decisões rápidas e precisas, com alertas em tempo real, contribuindo para a manutenção preditiva e redução de custos operacionais.

REFERÊNCIAS

Internet das Coisas: o que é, como funciona e exemplos de uso. CNN, 2023. Disponível em: <[Internet das Coisas: o que é, como funciona e exemplos de uso](#) />. Acesso em: 10 de agos. de 2025.

S., Bruno. O Que é Docker e Como Ele Funciona? – Docker Explicado. Hostinger, 2025. Disponível em: <[O Que é Docker e Como Ele Funciona? – Docker Explicado](#)/>. Acesso em: 15 de agos. de 2025.

mcDantas, Cintia. Como usar o Streamlit em um projeto de análise de dados. Medium, 2022. Disponível em: <[Como usar o Streamlit em um projeto de análise de dados | by Cintiamcdantas | Suzano DigitalTech | Medium](#)/>. Acesso em: 16 de agos. de 2025.

BRANDÃO, Afonso. Videoaulas da Unidade 4 da disciplina Disruptive Architectures: IOT, BIG DATA e IA. Universidade FECAF, 2025. Disponível em: [[Aulas 1 e 2](#)]. Acesso em: 25 ago. 2025.