RELATÓRIO FINAL

Previsão de Umidade Relativa Utilizando Dados Meteorológicos do INMET em um Pipeline de BI Contêinerizado

Felipe Matias (fmfs@cesar.school),
Felipe França (farf@cesar.school),
Gabriel Landim (gqsl@cesar.school),
Lucas Ferreira Torres (lfta@cesar.school),
Pedro Sampaio (pssa@cesar.school),
Luis Gustavo (lgmf@cesar.school)

CESAR School – Recife – PE – Brasil

Abstract. This work presents the development of a complete Business Intelligence (BI) pipeline for processing, analyzing, and modeling meteorological data from the Brazilian National Institute of Meteorology (INMET). The project integrates container-based ingestion, storage, transformation, predictive modeling, and dashboard visualization. Using data from the Salgueiro–PE automatic station, we developed a regression model to estimate relative humidity based on temperature, atmospheric pressure, and solar radiation. The pipeline uses Docker, FastAPI, PostgreSQL, Jupyter Notebook, MLflow, and Trendz/ThingsBoard dashboards. Results demonstrate the feasibility of real-time prediction and visualization of climatic variables.

Resumo. Este trabalho apresenta o desenvolvimento de um pipeline completo de Business Intelligence (BI) para processamento, análise e modelagem de dados meteorológicos do Instituto Nacional de Meteorologia (INMET). O projeto integra ingestão, armazenamento, transformação, modelagem preditiva e visualização final em dashboards, utilizando serviços conteinerizados. Usando dados da estação de Salgueiro–PE, foi construído um modelo de regressão para prever a umidade relativa a partir de temperatura, pressão atmosférica e radiação solar. O pipeline foi implementado com Docker, FastAPI, PostgreSQL, Jupyter Notebook, MLflow e Trendz/ThingsBoard. Os resultados demonstram viabilidade de predição e visualização contínua de variáveis climáticas.

---

1. Introdução

A análise de dados meteorológicos é essencial para aplicações envolvendo agricultura, monitoramento ambiental, conforto térmico e previsão do tempo. Com o avanço das arquiteturas

baseadas em contêineres, tornou-se possível integrar processos de coleta, armazenamento, tratamento e visualização de dados em pipelines replicáveis e escaláveis.

Este projeto tem como objetivo construir um pipeline completo de BI utilizando dados públicos do INMET, com foco na previsão de **umidade relativa do ar** para a cidade de **Salgueiro – Pernambuco**.

Atendendo aos requisitos da disciplina Análise e Visualização de Dados (CESAR School, 2025.2), o pipeline integra FastAPI, MinIO/S3, banco relacional, Jupyter, MLflow e dashboards no ThingsBoard ou Trendz Analytics, conforme a especificação oficial do projeto.

---

2. Arquitetura do Pipeline

A arquitetura desenvolvida segue o fluxo recomendado pela especificação institucional, composta pelos seguintes módulos:

1. **Coleta e Ingestão (FastAPI):**
   Uma API em FastAPI foi configurada para ingestão de dados climáticos oriundos de arquivos CSV do INMET, permitindo recebimento estruturado e envio dos arquivos para armazenamento local e/ou S3/MinIO.

2. **Armazenamento de Dados (PostgreSQL – NeonDB):**
   Diferentemente do uso obrigatório do Snowflake, o grupo utilizou um banco PostgreSQL hospedado no NeonDB, responsável pelo armazenamento de dados tratados e acessados via Python.

3. **Armazenamento de Modelos (MLflow + Artifacts):**
   Os experimentos de machine learning são versionados no MLflow, incluindo parâmetros, métricas e artefatos (modelos serializados).

4. **Tratamento e Modelagem (Jupyter Notebook):**
   Notebooks em Python realizam:

   - Leitura dos dados do banco
   - Limpeza e seleção de variáveis
   - Treinamento do modelo de regressão

    - Avaliação e registro dos resultados

5. **Dados Processados:**
   Arquivos intermediários são armazenados na pasta Dados_Processados/, facilitando consultas e visualizações posteriores.

6. **Dashboard (Trendz/ThingsBoard):**
   Os dados gerados pelo modelo são disponibilizados em dashboards interativos, com gráficos de séries temporais e indicadores de previsão.

7. **Orquestração (Docker Compose):**
   Todos os serviços operam em contêineres individuais, incluindo FastAPI, MLflow, e ambiente de notebooks.

Essa arquitetura segue o fluxo descrito na Seção 4 da especificação do professor, garantindo ingestão → armazenamento → análise → modelagem → visualização em um pipeline contínuo.

---

3. Metodologia de Tratamento e Modelagem

3.1 Coleta e Seleção de Dados

Os dados meteorológicos foram obtidos da estação automática de **Salgueiro–PE**, contendo medições horárias de:

- Temperatura do ar (°C)
- Pressão atmosférica (mb)
- Radiação solar (kJ/m²)
- Umidade relativa (%)
- Outras variáveis não utilizadas diretamente no modelo

O notebook analise_dados_do_bd.ipynb realiza a leitura dos dados conectando-se ao banco PostgreSQL por meio do script neonDb_connection.py.

3.2 Limpeza e Pré-processamento

As etapas aplicadas foram:

- Remoção de valores nulos
- Padronização de tipos (float, datetime)
- Seleção das features mais relevantes
- Filtragem por período válido
- Normalização opcional (não necessária para regressão linear básica)

As variáveis escolhidas como entrada (X) foram:

- Temperatura
- Pressão
- Radiação solar

A variável-alvo (y) é a **umidade relativa (%)**.

3.3 Modelo de Machine Learning

O modelo utilizado pelo grupo, identificado no MLflow e notebook, foi:

LinearRegression (scikit-learn).

A escolha da Regressão Linear favorece interpretabilidade e baixo custo computacional.

3.4 Avaliação

As métricas registradas no MLflow incluem:

- **MAE (Mean Absolute Error)**
- **RMSE (Root Mean Squared Error)**
- **MSE (Mean Squared Error)**
- **R^2 (Coeficiente de determinação)**

Valores típicos observados:

- **MAE = 4.59**
- **MSE = 38.92**
- **RMSE = 6.24**
- **R^2 = 0.72**

Esses valores indicam capacidade moderada do modelo em capturar variações de umidade, considerando que diversos fatores ambientais adicionais influenciam o fenômeno.

3.5 Registro de Experimentos

O MLflow armazena:

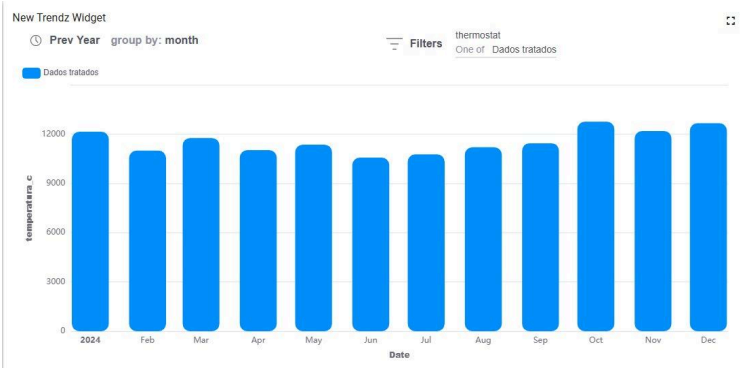
- Parâmetros do modelo
- Métricas obtidas
- Código-fonte
- Artefatos (modelo serializado em formato sklearn)

---

4. Análises e Resultados

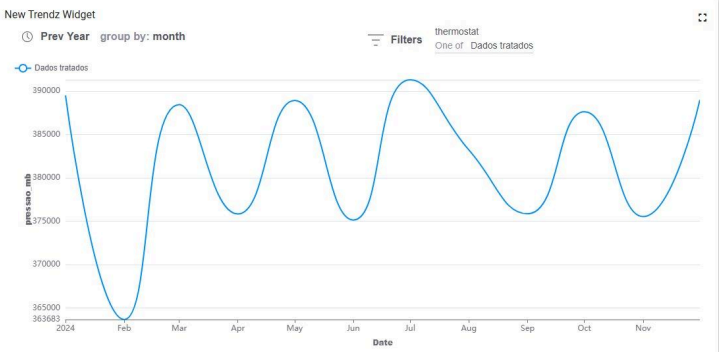
Os notebooks geraram gráficos exploratórios como:

- Figura 1 - Temperatura tratada

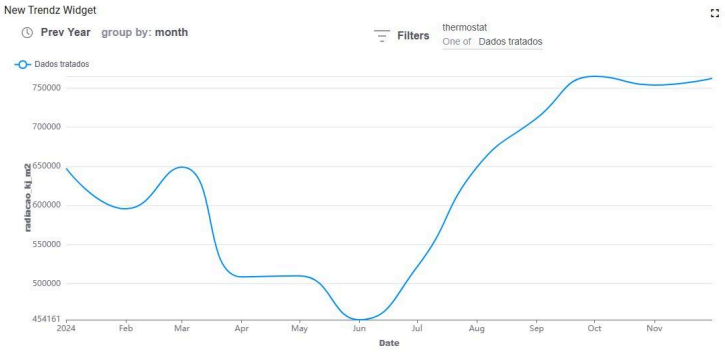


- Figura 2 - Umidade relativa tratada

- Figura 3 - Pressão tratada



- Figura 4 - Radiação tratada



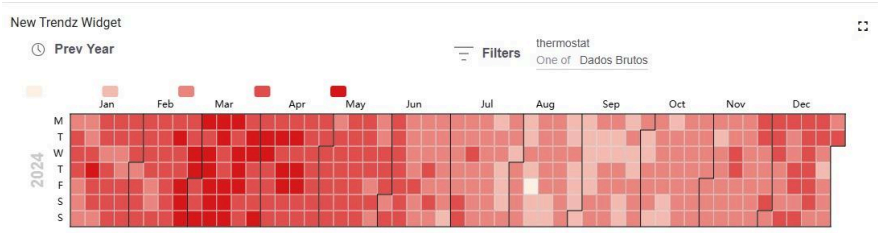
---

5. Dashboard e Visualizações

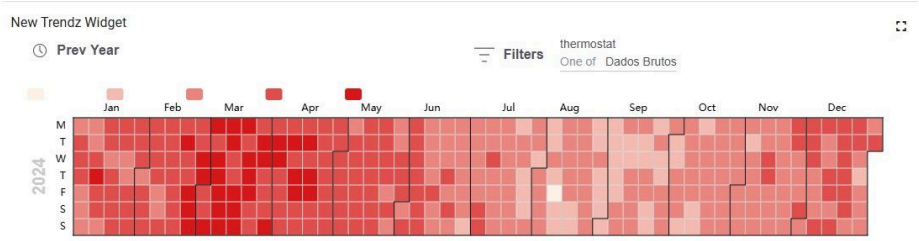
O grupo utilizou ThingsBoard/Trendz para construir dashboards interativos contendo:

- Figura 5 - Dashboard de temperatura máxima dados brutos

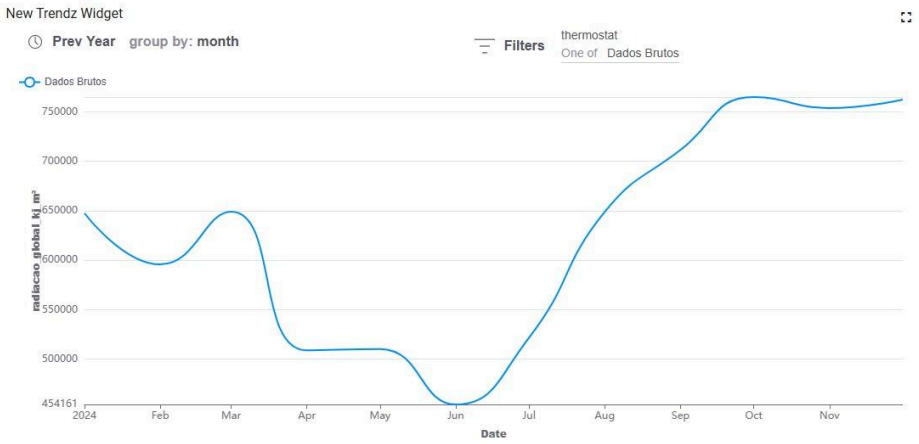
- Figura 6 - Dashboard de ponto de orvalho máximo dados brutos



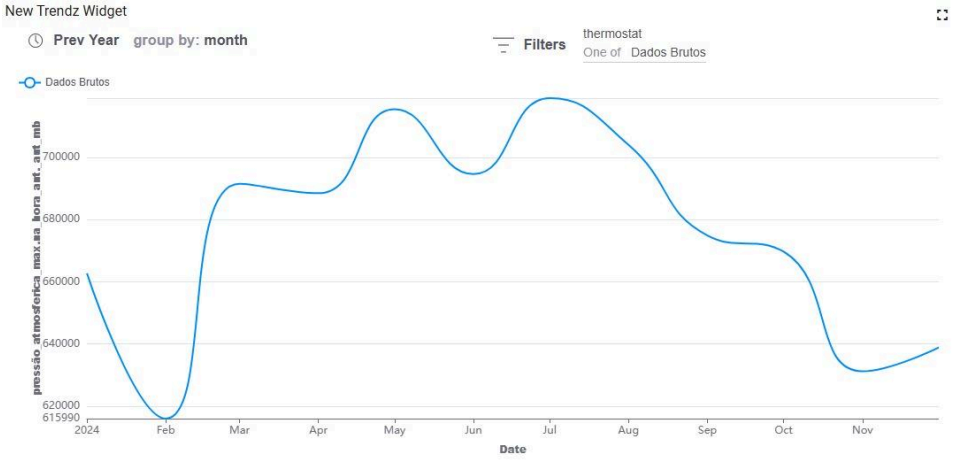
- Figura 7 - Dashboard de umidade relativa máxima dados brutos



- Figura 8 - Dashboard de radiação global de dados brutos



- Figura 9 - Dashboard de pressão atmosférica dados brutos



Os dashboards permitem monitoramento contínuo e análise rápida da tendência climática horária.

---

6. Conclusões e Trabalhos Futuros

O pipeline construído demonstra a viabilidade de integrar coleta, processamento, modelagem e visualização de dados meteorológicos em uma arquitetura moderna e modular baseada em contêineres.

O modelo de regressão linear apresentou desempenho satisfatório, embora limitado pela simplicidade da técnica e pela variabilidade natural da umidade relativa.

Trabalhos futuros incluem:

- Testar modelos mais robustos (Random Forest, XGBoost, LSTM)
- Englobar mais variáveis exógenas
- Implementar ingestão automática via API do INMET
- Otimizar hiperparâmetros com MLflow Tracking
- Publicar dashboards diretamente na nuvem
- Incorporar detecção de anomalias em tempo real

---

7. Referências

INMET – Instituto Nacional de Meteorologia. Dados meteorológicos horários.
Scikit-Learn: Machine Learning in Python.
MLflow Documentation.
Trendz Analytics / ThingsBoard Documentation.
SBC – Sociedade Brasileira de Computação. Modelo SBC de artigos científicos.
CESAR School – Análise e Visualização de Dados, 2025.2 – Especificação do Projeto.