



Predição Meteorológica: Meteo Abrantes

Um projeto integrado que combina recolha automática de dados, previsão estatística e visualização analítica para demonstrar técnicas de engenharia de dados e análise preditiva.

CTESP INFORMÁTICA

PROJETO INTEGRADO

Visão Geral do Projeto

O AbrantesMeteo é uma plataforma integrada para recolha, previsão e análise de dados meteorológicos na cidade de Abrantes. Combina dados históricos importados de bases externas, medições em tempo real e modelos preditivos com visualização centralizada em dashboards Power BI.

O objetivo principal é demonstrar, em contexto académico, a aplicação prática de tecnologias IoT, bases de dados NoSQL, modelação preditiva e Business Intelligence, assegurando atualização automática e consistência temporal dos dados.

01

Recolha Automática

Dados históricos e diários via API

02

Previsão Prophet

Modelos preditivos 365 dias

03

Visualização BI

Dashboards comparativos

Objetivos e Critérios de Sucesso

1

Recolha Histórica

Recolher histórico meteorológico via Open-Meteo Archive e armazenar em MongoDB local com atualização diária automática.

2

Previsões Diárias

Gerar previsões para 365 dias com Prophet, normalizando o output em formato tabular para análise comparativa.

3

Sincronização Cloud

Sincronizar automaticamente para MongoDB Atlas com autenticação segura e permissões mínimas.

4

Exposição Power BI

Expor dados via Atlas Data Federation/SQL (ODBC) para consumo analítico sem dependências de drivers nativos.

5

Modelo Analítico

Construir modelo de dados com calendário, medidas DAX e visualizações para comparação Real vs. Previsto.

6

Baseline Consistente

Garantir atualização diária e consistência do baseline de previsão para comparações temporais fiáveis.

01:20:52)

3.37% of 16 CPU(s)

0.02,0.09,0.11

⊖ IO delay

8 GiB of 29.16 GiB)

KSM sharing

2 GiB of 93.93 GiB)

↻ SWAP usage

16 x AMD Ryzen 7 8845HS

Lin

✓ Proxmox VE updates ! Non

ARQUITETURA

Ambiente de Execução e Tecnologias

Infraestrutura

- **Servidor:** MiniPC UM880 com Proxmox
- **Execução:** Contentor LXC dedicado (docker-ct) com Docker Engine e Docker Compose
- **Cliente:** PC Windows com Power BI Desktop e Power BI Service
- **Gateway:** On-premises Data Gateway (Personal Mode) para refresh automático

Stack Tecnológico

- **MongoDB 7:** Armazenamento local e testes
- **MongoDB Atlas:** Persistência cloud e consumo analítico
- **Python 3.11:** Ingestão, normalização e previsão
- **Prophet:** Forecasting com cmdstanpy
- **Power BI:** Modelação, DAX e dashboards
- **Atlas SQL/ODBC:** Exposição SQL para BI

Arquitetura Lógica e Fluxo de Dados

A solução foi projetada como um pipeline modular onde a ingestão e previsão correm no servidor junto ao armazenamento local, garantindo execução diária e independência do posto de trabalho. A cloud (Atlas) centraliza e disponibiliza os dados para consumo consistente via ODBC.



Organização do Projeto no Servidor

A implementação foi organizada numa pasta dedicada dentro do volume RAID do servidor, assegurando persistência dos dados e separação entre código, logs e segredos.

Localização: /mnt/storage/abrantes_meteo

Componentes Principais

- **docker-compose.yml:** Orquestração de serviços
- **secrets/:** Keyfile MongoDB
- **mongo-init/:** Scripts de bootstrap
- **app/:** Aplicação Python
- **mongo-data/:** Persistência BD
- **logs/:** Evidências operacionais

```
services:
  mongodb:
    image: mongo:7
    container_name: abrantes_meteo_mongodb
    command: ["--bind_ip_all", "--replSet", "rs0", "--keyFile", "/etc/mongo-keyfile", "--auth"]
    environment:
      MONGO_INITDB_ROOT_USERNAME: ${MONGO_ROOT_USERNAME}
      MONGO_INITDB_ROOT_PASSWORD: ${MONGO_ROOT_PASSWORD}
    ports:
      - "27018:27017" # 27018 no host para não colidir com outros projetos
    volumes:
      - ./mongo-data:/data/db
      - ./secrets/mongo-keyfile:/etc/mongo-keyfile:ro

  healthcheck:
    test: ["CMD-SHELL", "mongosh --quiet -u ${MONGO_INITDB_ROOT_USERNAME} -p ${MONGO_ROOT_PASSWORD} --eval 'use admin; db.runCommand({ping:1})'"]
    interval: 10s
    timeout: 5s
    retries: 30
    start_period: 20s
```



Implementação: Pipeline Aplicacional

O ficheiro app.py implementa o ciclo diário completo do AbrantesMeteo, desde a ingestão até à disponibilização para análise, garantindo execução automática e consistência temporal.

Ingestão de Dados

Chamadas à API Open-Meteo com normalização de campos e datas, persistência idempotente no MongoDB local sem duplicação.

Geração de Previsões

Treino de modelos Prophet para horizonte de 365 dias, identificação por run_id diário para rastreabilidade.

Sincronização Atlas

Envio de histórico recente (janela de segurança) e previsão do dia para consumo externo via Power BI.

Agendamento Automático

Execução inicial no arranque e ciclo diário à hora configurada, com tratamento de exceções para resiliência.

Artefactos de Configuração Docker

1

docker-compose.yml

Define serviços (MongoDB, bootstrap, aplicação Python) com dependências, volumes persistentes e políticas de reinício automático para resiliência.

2

Ficheiro .env

Centraliza variáveis de execução: fuso horário, coordenadas, datas, URIs de ligação, horizonte de previsão e janela de sincronização.

3

requirements.txt

Fixa dependências Python com versões específicas: requests, pymongo, pandas, prophet e cmdstanpy para reprodutibilidade.

4

01-setup.js

Script idempotente de bootstrap: inicializa ReplicaSet, cria utilizadores com permissões mínimas e define índices essenciais.

5

Dockerfile

Constrói imagem da aplicação com base Python 3.11-slim, instalação de dependências e definição do comando de arranque.

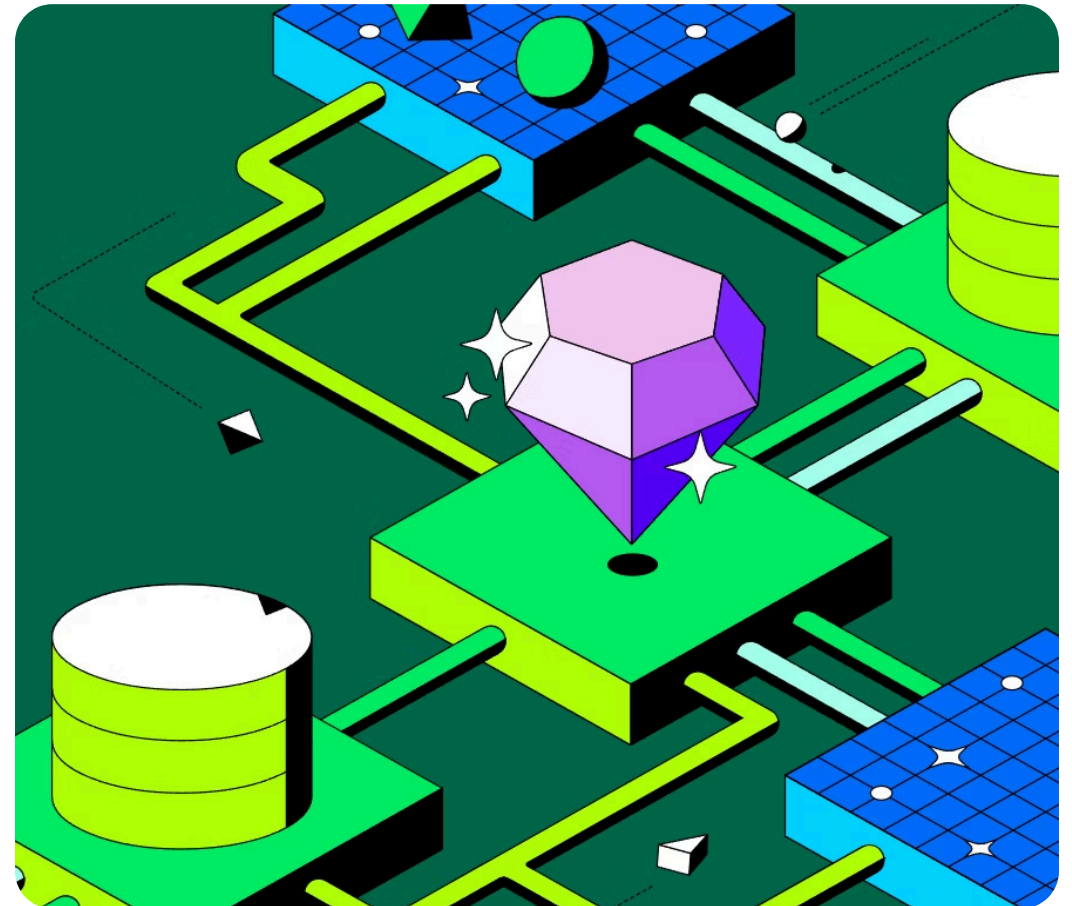
Sincronização para MongoDB Atlas

Após validação do armazenamento local, foi configurado um cluster MongoDB Atlas para alojar a base de dados em cloud e suportar o consumo analítico via Power BI.

Configuração de Segurança

Foram definidos perfis de acesso distintos seguindo o princípio do menor privilégio:

- **abrantes_writer:** Permissões de escrita para processos de ingestão e sincronização
- **abrantes_reader:** Permissões de leitura para consumo analítico (Atlas SQL/ODBC e Power BI)



Cluster0

Connect

View Monitoring

Browse Collections

...

Learn about MongoDB Monitoring

Take our new course on how to
monitor MongoDB deployments

R 0
W
1 month 4 hours ago

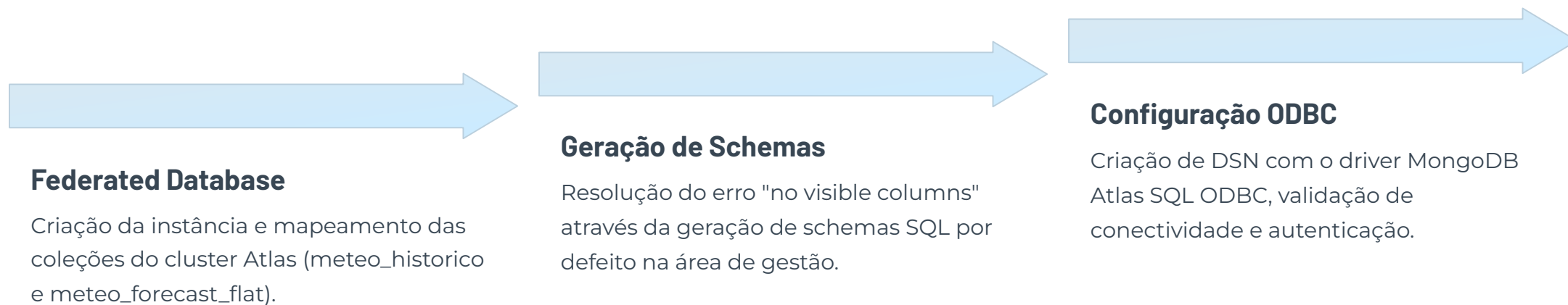
Connections 2.0
1 month 4 hours ago

In 25.01 B/s
Out 195.55 B/s
1 month 4 hours ago

ATLAS SQL

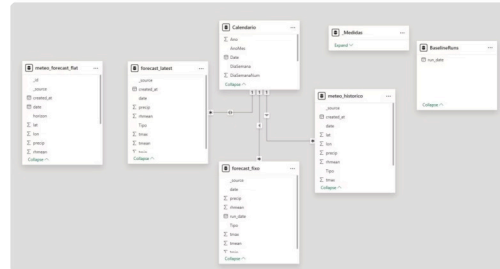
Exposição ao Power BI via Atlas SQL

O Power BI não consome MongoDB nativamente com a mesma maturidade de fontes relacionais. Foi adotada a abordagem Atlas Data Federation + Atlas SQL Interface para expor as coleções através de um endpoint SQL consumível via ODBC.



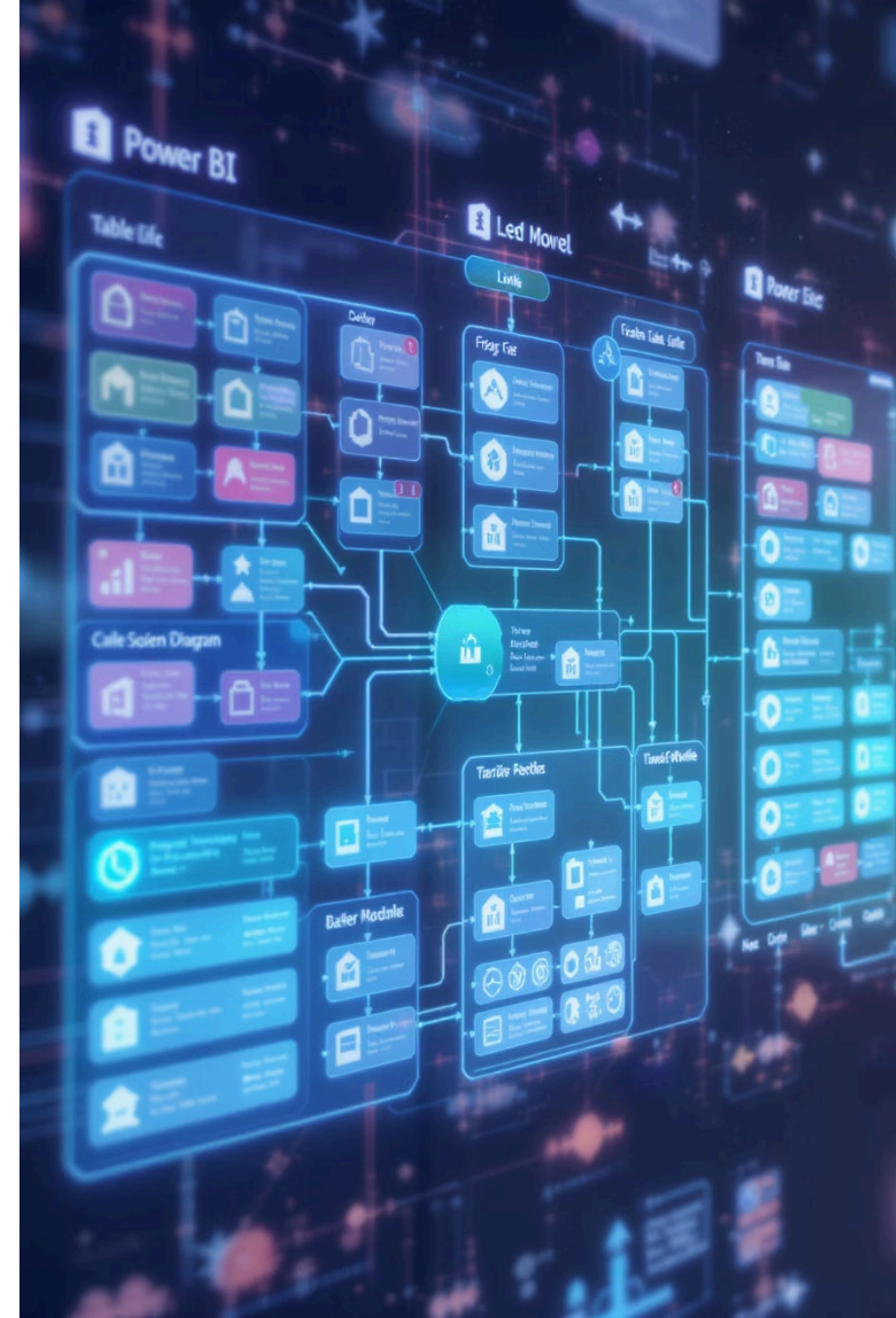
Modelação no Power BI

Com a ligação operacional, o foco passou para a modelação: definição de tipos de dados, tabela calendário, relacionamentos e medidas analíticas.



Componentes do Modelo

- **Tabela Calendário:** Criada com CALENDAR() para filtragem temporal consistente e suporte a comparações entre séries
- **Relacionamentos:** Tabela calendário relacionada 1:* com tabelas de factos através do campo de data
- **forecast_fixo:** Tabela derivada que fixa uma execução como baseline para comparações Real vs. Previsto
- **Medidas DAX:** Criadas para cada variável (Tmax, Tmin, Precip) com cálculos de Real, Previsto e Diferença



Integração Raspberry Pi 5 + Sensor BME280

Extensão do sistema com Raspberry Pi 5 equipado com sensor BME280 para recolher temperatura, humidade relativa e pressão atmosférica, disponibilizando-as no Power BI como tabela independente.

Hardware I2C

Sensor BME280 ligado via I2C (3.3V, GND, SDA, SCL), ativação do I2C no sistema operativo e validação do endereço.

1

2

Ambiente Python

Criação de venv isolado, instalação de bibliotecas Adafruit e PyMongo, parametrização via ficheiro .env local.

Script de Envio

Leitura do sensor e inserção no MongoDB Atlas com timestamp UTC, origem e identificação do equipamento.

3

4

Automatização Cron

Configuração de 5 execuções diárias (07:00, 11:00, 15:00, 19:00, 23:00) com logs para monitorização.

Exposição Power BI

Schema SQL gerado no Atlas, importação via ODBC e criação de visuais com última leitura e histórico temporal.

5

Dificuldades Encontradas e Resolução

Codificação UTF-8

Problema: Erros de execução quando ficheiros não estavam em UTF-8 no container Python.

Resolução: Garantir gravação em UTF-8 e evitar caracteres inválidos, definir locale correto.

Autenticação Atlas

Problema: Erro "bad auth" por password ou roles incorretos.

Resolução: Recriar utilizador no Database Access e validar com mongosh via mongodb+srv.

Colunas Invisíveis Power BI

Problema: Erro "no visible columns" - schema SQL não gerado no Atlas SQL.

Resolução: Generate all default schemas em Manage SQL Schemas.

Colunas Duplicadas

Problema: Colunas date e date.1 após expandir registos no Power Query.

Resolução: Remover date.1 e mudar tipo de dados de date para Date.

Ligação ODBC

Problema: Erros de ligação e driver incorreto.

Resolução: Seleção do driver Atlas SQL ODBC e URI correta do Power BI Connector.

Resultados Finais e Melhorias Futuras

Resultados Obtidos

- Dados históricos carregados com sucesso desde 01-01-2023
- Previsão diária gerada e persistida (365 dias por execução)
- Sincronização para Atlas e exposição SQL/ODBC operacional
- Dashboards no Power BI com comparação Real vs. Previsto
- Identificação de execuções por run_id para comparações temporais
- Integração de sensor BME280 com leituras em tempo real

Melhorias Futuras

- Automatizar criação de forecast_fixo no servidor
- Persistir todas execuções com métricas de desempenho (MAE/RMSE)
- Implementar observabilidade com Prometheus/Grafana
- Reforçar gestão de segredos (Docker Secrets/Vault)
- Adicionar camadas de qualidade de dados
- Escalar para múltiplas entidades com parametrização



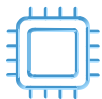
Conclusão e Aprendizagens

O AbrantesMeteo resultou numa implementação operacional e coerente de um pipeline de dados completo, desde a recolha automática até à análise em Power BI, com armazenamento em MongoDB e previsão em Python com Prophet.



Arquitetura Modular

A separação entre dados reais e previsões, com baseline de forecast, permitiu comparações fiáveis e repetíveis ao longo do tempo.



Componente IoT

A extensão com Raspberry Pi 5 e sensor BME280 adicionou complexidade relevante, demonstrando integração de medições físicas reais.



Integração Cloud

A exposição via MongoDB Atlas e Atlas SQL (ODBC) foi determinante para garantir compatibilidade e estabilidade no Power BI.



Escalabilidade

O projeto valida uma abordagem modular e escalável, facilmente adaptável a outros contextos além da meteorologia.

Este projeto evidencia competências práticas de integração de sistemas, automação e análise preditiva, aplicáveis a qualquer domínio que requeira ingestão, transformação e disponibilização de dados para apoio à decisão.