Hackathon de Engenharia de Dados

Plataforma de Dados Simplificada

Objetivo Geral

Desenvolver uma plataforma básica de dados que realize ingestão de arquivos ou dados externos, saneamento, transformação em camadas (Bronze → Silver → Gold), versionamento de código e automação mínima (CI/CD), finalizando com a apresentação de insights analíticos.

Dataset

Será disponibilizado um conjunto de arquivos semi-estruturados contendo dados climáticos do ano de 2024, com informações de estações meteorológicas de diversas cidades do Brasil. Esses datasets servirão como exemplos e ponto de partida para a prática, mas os participantes podem utilizar qualquer outro dataset público ou privado — incluindo dados estruturados, semi-estruturados, não estruturados ou até mesmo consumo de APIs públicas. O importante é que a solução demonstre uma pipeline completa (ingestão, transformação em camadas e geração de insights) independentemente da origem dos dados escolhidos.

Etapas do Pipeline de Dados

1. Ingestão (Raw → Bronze)

- Ler arquivos ou consumir dados de APIs.
- Separar metadados e dados principais.
- Padronizar separador decimal, encoding e delimitador.
- Salvar dados crus organizados em uma camada Bronze (ex.: parquet/CSV organizado por partições).

2. Limpeza & Padronização (Bronze → Silver)

- Normalizar nomes de colunas (snake_case, sem acentos/espaços).
- Converter tipos de dados (datas para timestamp, números com ponto decimal).
- Tratar valores faltantes e inválidos.
- Ajustar timezone (UTC → America/Fortaleza).
- Deduplicar registros.

3. Modelagem & Transformações (Silver → Gold)
- Criar tabelas agregadas por dia/estação/município (ex.: temperatura máxima/mínima, precipitação total).
- Construir métricas derivadas e tabelas analíticas organizadas para consumo.
- Manter boas práticas de particionamento e performance.

4. Testes & Qualidade de Dados
- Verificar ranges válidos (temperatura -20°C a 50°C, umidade 0–100%).
- Garantir campos obrigatórios não nulos e chaves únicas.
- Documentar regras de qualidade implementadas.

5. Orquestração / Pipeline
- Criar um script único que execute ingestão → silver → gold.
- Permitir execução simples localmente (ex.: `python run_all.py`).

6. Versionamento & CI/CD
- Organizar o código em repositório Git com pastas claras (ingestion, transforms, analytics).
- Configurar um workflow simples (ex.: GitHub Actions) para rodar testes e gerar as camadas automaticamente.

7. Analytics & Insights
- Responder perguntas obrigatórias:
  - Dia mais quente de 2024 em João Pessoa.
  - Dia mais chuvoso de 2024 em Patos.
- Gerar gráficos e análises criativas (ondas de calor, correlação radiação vs temperatura etc.).

8. Documentação & Arquitetura
- README.md explicando setup, dependências e execução.
- Diagrama simples (Draw.io, Miro ou PowerPoint) mostrando ingestão, camadas e CI/CD.

🗀 Estrutura de Diretórios Sugerida

```
/projeto_dados/
├─ data/
│   ├─ bronze/
│   ├─ silver/
│   └─ gold/
├─ ingestion/
├─ transforms/
├─ analytics/
├─ pipelines/
└─ .github/workflows/
```

Ferramentas Sugeridas

- Pandas / PySpark para processamento.
- DuckDB ou Parquet para armazenamento.
- GitHub Actions para CI/CD simples.
- Matplotlib/Plotly ou Power BI para visualização.

Critérios de Avaliação (100 pontos)

• Arquitetura & documentação: 15%
• Ingestão & Bronze: 15%
• Limpeza & Silver: 15%
• Modelagem & Gold: 15%
• Qualidade de dados: 10%
• CI/CD & versionamento: 10%
• Insights & visualização: 20%