

## Engenharia de Dados – Caso de Cervejarias

**Objetivo:** O objetivo deste teste é avaliar suas habilidades em consumir dados de uma API, transformá-los e persistir em um data lake seguindo a arquitetura de medalhão com três camadas: dados brutos, dados tratados particionados por localização e uma camada analítica agregada.

### Instruções:

1. **API:** Use a Open Brewery DB API para buscar dados. A API possui um endpoint para listar cervejarias: <<https://api.openbrewerydb.org/breweries>>
2. **Ferramenta de Orquestração:** Escolha a ferramenta de orquestração de sua preferência (Airflow, Databricks, Luigi, Mage etc.) para construir um pipeline de dados. Estamos interessados em ver sua habilidade em lidar com agendamento, tentativas de repetição e tratamento de erros no pipeline.
3. **Linguagem:** Use Python como linguagem base. Você pode usar quaisquer pacotes que achar necessário para as requisições e transformação de dados. Por favor, inclua casos de teste para seu código. PySpark e Spark são preferidos, mas não obrigatórios.
4. **Containerização:** Se você usar Docker ou Kubernetes para modularização, ganhará pontos extras.
5. **Arquitetura de Data Lake:** Seu data lake deve seguir a arquitetura de medalhão tendo uma camada bronze, prata e ouro:
  - **Camada Bronze:** Persistir os dados brutos da API em seu formato nativo ou qualquer formato que você considerar adequado.
  - **Camada Prata:** Transformar os dados para um formato de armazenamento colunar, como parquet ou delta, e particionar por localização da cervejaria. Por favor, explique quaisquer outras transformações que você realizar.
  - **Camada Ouro:** Criar uma visão agregada com a quantidade de cervejarias por tipo e localização.
6. **Monitoramento/Alertas:** Descreva como você implementaria um processo de monitoramento e alertas para este pipeline. Considere problemas de qualidade de dados, falhas no pipeline e outros problemas potenciais em sua resposta.
7. **Repositório:** Crie um repositório público no GitHub com sua solução. Documente suas escolhas de design, compensações e forneça instruções claras sobre como executar sua aplicação.
8. **Serviços em Nuvem:** Se sua solução requerer qualquer serviço em nuvem, forneça instruções sobre como configurá-los. Por favor, não poste essas informações em seu repositório público.

**Critérios de Avaliação:** Sua solução será avaliada com base nos seguintes critérios:

1. Qualidade do Código
2. Design da Solução
3. Eficiência

4. Integralidade
5. Documentação
6. Tratamento de Erros

**Prazo:** Por favor, complete o teste dentro de 1 semana e compartilhe o link para seu repositório GitHub conosco. Lembre-se, o objetivo deste teste é mostrar suas habilidades e abordagem para construir um pipeline de dados.

Boa sorte!