

# Case Técnico – Engenharia de Dados Maxinutri

## Visão Geral

**Objetivo:** Avaliar habilidades em engenharia de dados, desde ingestão de dados, modelagem e até automação de pipelines.

**Prazo de Entrega:** 7 dias a partir do recebimento deste documento.

**Envio:** Repositório GitHub para [dados@maxinutri.com.br](mailto:dados@maxinutri.com.br) com cópia para [dados1@maxinutri.com.br](mailto:dados1@maxinutri.com.br) ou whatsapp ( 18 ) 98156-3470 se identificando.

Aconselhamos a extrair os dados e salva-los em um arquivo seja ele .csv, .parquet, .json qualquer um de sua preferência. Api para por inatividade a cada 50 segundos, pedimos desculpa por isso 🙏. Será levado em consideração na avaliação.

---

## Detalhes Técnicos

### 1. API de Dados

- **Base URL:** <https://teste-engenheiro.onrender.com>
- **Endpoint:** </data>
- **Parâmetros Obrigatórios:**
  - **token:** [63daac8dac3d57e6c6f9f71d3a9fa4ed](#)
  - **page:** Número da página (ex: [page=1](#)).
- **Exemplo de Requisição:**

 Text

```
GET /data?token=63daac8dac3d57e6c6f9f71d3a9fa4ed&page=1
```

- **Resposta Paginada:**

```
{ } JSON
{
  "pagina": 1,
  "total_paginas": 5,
  "total_linhas": 1000,
  "dados": [{ "coluna1": "valor1", "coluna2": "valor2" }]
}
```

---

### 2. Tarefas do Desafio

## 1. Ingestão de Dados

- Pagar e armazenar todos os dados da API.

## 2. Exploração e Tratamento

- Identificar estrutura, relacionamentos e chaves.
- Realizar tratamento mínimo necessário.

## 3. Modelagem do DW

- Criar schema dimensional (estrela ou floco de neve).
- Banco de dados preferencial: **PostgreSQL**.

## 4. Processo ETL

- Extrair, transformar (conforme modelagem) e carregar no DW que você construiu.

## 5. Automatização

- Usar **Airflow** para orquestração e agendamento **(será um diferencial)**
- *Novos dados serão adicionados 1 dia antes da entrevista para validação.*

## 6. Documentação no README do GitHub

- Diagrama de arquitetura.
- Justificativas de tecnologias e escolhas.

## 7. Apresentação do case

- Será utilizado o **README do GitHub**:
  - Por que esta arquitetura?
  - Benefícios, escalabilidade e melhorias para on-premises.

---

## Critérios de Avaliação

- ✓ Clareza e organização do código.
- ✓ Boas práticas de engenharia de dados.
- ✓ Modelagem eficiente do DW.
- ✓ Automação reutilizável **(Airflow como diferencial)**.
- ✓ Documentação objetiva e justificativas técnicas.

---

## Entregáveis

- **Repositório GitHub com:**
  - Código-fonte (ETL, scripts SQL, DDL).
  - Schema do banco de dados.
  - Diagrama de arquitetura.
  - Apresentação será no **README do GitHub**.

---

## Ferramentas Sugeridas

- **Linguagem:** Python.
- **Banco de Dados:** PostgreSQL.
- **Orquestração:** Apache Airflow (opcional).
- **Visualização arquitetura:** Draw.io, Excalidraw.

---

## Dicas Bônus para o Desafio!

Aqui estão algumas dicas extras para deixar seu projeto ainda mais incrível:

## Dicas Técnicas ( Não obrigatórios ):

### 1. Tratamento de Erros:

- Implemente retries com backoff para lidar com falhas temporárias na API.
- Valide os dados antes de carregá-los no DW (ex: checar nulos, formatos, duplicados).

### 2. Airflow (se usar):

- Adicione logs claros e alertas em caso de falha ( email / pode usar o dados@maxinutri.com.br também para informar os alertas).

### 3. Modelagem do DW:

- Documente as decisões de modelagem (ex: "Por que escolhi modelo estrela?").
  - Adicione índices nas chaves de dimensão para melhor performance.
-

## 🎯 Dicas de Organização:

- **README do GitHub:**
  - Inclua um passo a passo visual (ex: diagrama do fluxo de dados).
  - Deixe sua documentação estruturada conforme os passos da arquitetura desenvolvida
- **Apresentação:**
  - Compare sua solução com alternativas (ex: "Por que não usamos Kafka? ou porque não usei Python ").
  - Destaque um **diferencial criativo** (ex: monitoramento em tempo real com Grafana).

---

## 💡 Dica Final:

- **Teste com Dados Novos:**
  - 💡 Simule a adição de dados (como mencionado no desafio) para garantir que a automação funciona!
  - 💡 Foca nos fundamentos isso é ouro, não precisa inventar a roda.

Qualquer dúvida ou erros da API pode entrar em contato.

**Boa sorte! Seu projeto já vai sair na frente com essas dicas! 😊**