Descrição do Processo: Desafio Técnico de Análise de Dados (AltoQi)

Candidato: Ramiro Manoel

**Tarefa 1 - Dashboard:**

**Objetivo:** Desenvolver um dashboard de acompanhamento trimestral para uma empresa com software SaaS e com liberdade total na escolha de métricas, layout e storytelling.

**Etapas Executadas:**

* Criação e enriquecimento dos dados: Gerei um conjunto de dados fictício com auxílio de IA, simulando informações de vendas SaaS.
  + Link do prompt: <https://chatgpt.com/share/68fda84e-4634-800a-9a68-762831dc48ff>
* Realizei o enriquecimento manual adicionando colunas de Vendedor e Cidade, e criei dois arquivos auxiliares: Metas de Vendas por Vendedor e Leads por Vendedor. Essa estrutura permitiu construir análises mais completas e realistas.
* Armazenamento e integração: Criei um script Python para ingestão dos arquivos excel e gravação em um database PostgreSQL hospedado no Render. Essa abordagem simulou um ambiente de produção e garantiu escalabilidade para futuras atualizações.
* Transformação e modelagem de dados: Conectei o Power BI diretamente ao banco de dados. Como as tabelas não estavam normalizadas, realizei o processo de ETL no Power Query, criando tabelas fato e dimensão. Também normalizei nomes de clientes (remoção de pronomes de tratamento) e ajustei relacionamentos.
* Desenvolvimento das métricas DAX: Foram criadas medidas para faturamento, ticket médio, curva ABC, meta vs realizado, top vendedores, cidades prioritárias e variações temporais (MOM, YOY e YTD). A estrutura das métricas foi validada conforme o documento de levantamento de requisitos elaborado para o projeto “Levantamento\_Requisitos\_BI\_AltoQi.docx”.
* Design e experiência do usuário: O layout foi planejado no Figma, respeitando a paleta de cores da marca da empresa. Cada página apresenta indicadores, filtros hierárquicos e ícones de informação contextual sobre as métricas. O documento foi publicado no Power BI Service e salvo em formato .pbix, na pasta “report”.
  + Links:
    - Figma: <https://www.figma.com/design/0hMKKxj7JXceH9Ath0SlR9/AltoQI?node-id=0-1&t=Wz8jnLCVn8a4aU29-1>
    - Power Bi serviço: <https://app.powerbi.com/view?r=eyJrIjoiMWU3MzlkZTUtNGY2OC00NGI1LTgyNzEtOTI2NDE4YzFiMWM4IiwidCI6ImU2MmRlZmMwLWVmZGEtNDJhNy1hMDNkLWZjNTVhNjRhYzQyNCJ9>
* Gestão e versionamento: Todo o projeto foi versionado via Git, garantindo rastreabilidade e organização dos artefatos (scripts, dados e relatórios).
  + Link para o repositório: <https://github.com/ramiromanoel/data_analyst_altoqi>
* Também elaborei um documento formal de levantamento de requisitos, incluindo objetivos, KPIs, escopo e cronograma de execução.
* Resultado: Um painel consolidado e funcional, com storytelling visual orientado a indicadores comerciais e estrutura técnica aderente às boas práticas de BI corporativo.

**Tarefa 2 - Query:**

**Objetivo:** Listar clientes com mais de 5 compras no último ano e nenhuma compra no último mês.

**Etapas Executadas:**

* Utilizei as tabelas Clientes, Pedidos e DetalhesPedidos. As tabelas Clientes e Pedidos foram as principais fontes para análise de frequência de compra, enquanto DetalhesPedidos estava disponível para possíveis extensões analíticas.
* Realizei a junção entre Clientes e Pedidos com base na chave ClienteID. A contagem de compras foi obtida com COUNT(PedidoID), agrupando por cliente.
* Implementei o filtro para considerar apenas pedidos realizados no último ano, utilizando DATEADD(YEAR, -1, GETDATE()).
* Na sequência, criei uma subquery que identifica clientes com pedidos no último mês, usando DATEADD(MONTH, -1, GETDATE()), e os excluí com NOT IN.
* A cláusula HAVING foi utilizada para exibir apenas clientes com mais de 5 compras no período filtrado e o resultado é ordenado pelo total de compras, exibindo primeiro os clientes mais ativos historicamente.
* Resultado esperado: Uma lista de clientes considerados inativos recentemente, mas com histórico de compras.

Query Final:

SELECT   
 c.ClienteID,  
 c.Nome,  
 c.Email,  
 COUNT(p.PedidoID) AS TotalComprasUltimoAno  
FROM Clientes AS c  
INNER JOIN Pedidos AS p ON c.ClienteID = p.ClienteID  
WHERE   
 p.DataPedido >= DATEADD(YEAR, -1, GETDATE())  
 AND c.ClienteID NOT IN (  
 SELECT DISTINCT ClienteID  
 FROM Pedidos  
 WHERE DataPedido >= DATEADD(MONTH, -1, GETDATE())  
 )  
GROUP BY   
 c.ClienteID, c.Nome, c.Email  
HAVING   
 COUNT(p.PedidoID) > 5  
ORDER BY   
 TotalComprasUltimoAno DESC;

Obs.: Query também disponível no arquivo “query\_altoqi.sql”, contido na pasta “sql”.

**Tarefa 3: Limpeza de Arquivo Usando Python**

**Objetivo:** Criar um script para limpar e padronizar números de telefone de uma coluna CSV, removendo caracteres não numéricos e normalizando o formato.

**Etapas Executadas:**

* Importação das bibliotecas: pandas para manipulação de dados e re para filtragem de caracteres usando expressões regulares.
* Leitura e inspeção inicial: O arquivo foi lido via pd.read\_csv() e verificado estrutura com df.head() e df.tail().
* Função de limpeza: Na função “limpar\_e\_padronizar\_telefone”, comecei removendo todas as letras, símbolos e espaços, deixando só os números. Depois, removo o zero inicial, caso exista, e conto quantos dígitos restaram para entender se o número tem código do país. Com base nessa contagem, mantenho apenas os 10 ou 11 dígitos finais, que correspondem ao formato nacional. Se o número for muito curto ou não fizer sentido, retorno “None” para indicar que é inválido.
* Para facilitar o rastreio e mitigar dúvidas/erros, criei uma nova coluna chamada “telefone\_limpo” para manter o histórico e facilitar auditorias.
* Após validação, exportei o resultado em CSV e Excel com os dados limpos.

Obs.: Script disponível no arquivo “limpa\_csv.ipynb”, contido na pasta “python”.