

## Case: ANALYTICS ENGINEER SR

Este case engloba 3 partes que acreditamos serem importantes para um Analytics Engineer: pipeline de dados (aqui com Airflow), Analytics e Data Science (básico).

Não é obrigatório você fazer todas as 3 partes que compõe este case. Se você foi selecionado para fazer este case é porque achamos que você tem os requisitos, seja pela experiência ou pelo potencial.

Mas encorajamos que tentem responder tudo.

### Parte 1 – Analytics:

Utilize a base de dados fornecida e usando Python, faça os cruzamentos e tratativas necessárias para ajudar o time comercial a entender a cesta de compra dos nossos sócios e a possibilidade de criar combos promocionais.

O time comercial precisa saber quais foram os 10 itens de maior relevância no período e os top 15 itens que acompanham cada um deles.

Salve o resultado como você achar melhor e crie um relatório *One Page* (preferencialmente no Looker Studio ou Tableau) que tenha a possibilidade de cada comprador visualizar seu departamento e o gerente consiga ter uma visão macro do período.

O que deverá ser enviado:

- Breve resumo do raciocínio utilizado
- Arquivo com código Python utilizado
- Prints ou link do relatório

### Parte 2 – Data Science:

O problema anterior limita-se a achar combos com apenas 2 itens. A partir da sua solução da Parte 1, desenvolva uma nova solução com alguma técnica de Machine Learning que permita achar combos maiores (p.ex. com 3 itens).

O que deverá ser enviado:

- Explicação sucinta da técnica utilizada
- Arquivo com código Python utilizado
- Print dos resultados

### Parte 3 – Pipeline de dados/Airflow:

Você agora precisa automatizar esse processo com Airflow.

Considerando que:

- Há uma query “Q1” que roda no BigQuery e gera a base que você recebeu neste case, onde é necessário passar como valor do parâmetro **@date\_max** a data atual e **@date\_min** a data de 3 dias atrás;
- Assuma que a query Q1 pode estar armazenada tanto em um arquivo “Q1.sql” quanto em uma variável no Python, “Q1\_query”;
- Você vai usar seu programa Python da Parte 2 para gerar as informações necessárias ao seu dashboard. Se você não tiver feito a Parte 2, considere um código Python “caixa-preta”;
- Esse processo vai ser executado toda quinta-feira às 7h da manhã;
- Queremos tentar executar as tasks 3 vezes, depois disso o job vai resultar em “Failed”.

O que queremos:

- a) Construa uma DAG do Airflow que execute a query “Q1” e seu programa Python da Parte 2. Estamos focando em observar a estrutura utilizada para construir a Dag e erros de sintaxe não serão considerados aqui, logo, não é preciso testar a execução da DAG num ambiente real.
- b) Se quisermos gravar a saída do seu programa Python da Parte 2 numa tabela do BigQuery, descreva quais alterações/inclusões você faria na sua DAG do Airflow.

O que deverá ser enviado:

- Arquivo com o código da DAG do item (a)
- Descrição para solução do item (b)

*OBS. Quem quiser compartilhar os resultados pelo Git, é só enviar o link do seu repositório com as respostas/códigos deste case.*