Desafio Data Engineer II

Observações:

- O conteúdo deste desafio não pode ser copiado/divulgado;
- Você enviará os arquivos-resposta com o título: **Desafio: Bemol Digital Data Engineer II - Seu Nome** para talent@bemol.com.br

O que esperamos de você:

Você será responsável por projetar e implementar uma solução de engenharia de dados utilizando PySpark ou Scala para gerenciar e disponibilizar um conjunto de dados de e-commerce brasileiro para a área de negócio. Seu trabalho será avaliado pela construção de uma casa de dados robusta e utilização de técnicas que possam melhorar ou aperfeiçoar o processo de engenharia de dados e que ofereçam insights estratégicos para o negócio.

1. **Arquitetura de Dados:**

- **Desenho da Arquitetura:** Crie um diagrama detalhado de uma arquitetura de dados escalável utilizando Apache Spark. Inclua componentes como: ingestão de dados, processamento em batch e/ou em tempo real, armazenamento de dados (Data Lake e/ou Data Warehouse), e mecanismos de consulta e visualização.
- **Tecnologias a Serem Utilizadas:** Utilize PySpark ou Scala para a implementação. Sinta-se livre para integrar outras tecnologias de Big Data como Apache Kafka para processamento de streams, Apache Hadoop para armazenamento, e Apache Airflow para orquestração de workflows.

2. **Implementação da Casa de Dados:**

- **Ingestão de Dados:** Automatize a ingestão de dados usando PySpark ou Scala, assegurando a captura eficiente de dados em formatos variados.
- **Organização e Otimização:** Estruture os dados em um Data Lake utilizando práticas como particionamento e bucketing para otimizar consultas.
- **Limpeza e Transformação:** Implemente transformações para normalizar, limpar e enriquecer os dados. Crie um dicionário de dados que descreva os campos processados.

3. **Documentação e Código:**

- **Documentação Completa:** Elabore documentação detalhada para cada etapa do processo, incluindo arquitetura, código e análises.
- **Qualidade do Código:** Escreva código limpo e eficiente em PySpark ou Scala, seguindo as melhores práticas de desenvolvimento e engenharia de software.

4. **Entrega Final:**

Repositório GitHub: Todo o código, documentação e recursos devem ser entregues via um repositório GitHub dedicado, com commits organizados e descrições claras para cada etapa implementada.

Diferenciais:

- Documentação organizada referente as análises e os tópicos
- Código limpo, legível e organizado
- Utilização de Cloud na arquitetura
- Mini Data Warehouse para armazenamento dos dados
- Sugestão de gerir a governança de dados em seu desafio
- Tratamento de Dados Sensíveis

Apoio para entendimento das bases disponibilizadas: