Projeto de Ciência de Dados e Big Data

1. Visão Geral

Este projeto tem como objetivo construir uma solução completa de Ciência de Dados/Big Data, envolvendo coleta, processamento, armazenamento e apresentação de insights. A entrega será composta por documentação no Confluence e código-fonte em repositório GitHub/Bitbucket.

2. Documentação Geral

Descrição do problema: Analisar dados de vendas de uma empresa de e-commerce para identificar padrões de consumo e prever demanda.

Objetivos: Criar um pipeline robusto para ingestão, processamento e análise dos dados, com visualizações de KPIs.

Escopo: Inclui ingestão de dados de logs de vendas, processamento em Spark, armazenamento em Data Lake e visualização em Metabase. Não inclui desenvolvimento de APIs externas.

Arquitetura: Pipeline completo com ingestão, processamento, armazenamento e análise.

Ferramentas: Kafka, Airflow, Spark, MinIO, Metabase.

Decisões técnicas: Uso de Data Lake para flexibilidade, Spark para escalabilidade e Kafka para ingestão em tempo real.

Guia de execução: Clone o repositório, configure Docker Compose e execute os serviços.

Dependências: Python 3.9, Spark 3.2, Kafka 2.8, MinIO, Metabase.

Dados: Logs de vendas em formato JSON, armazenados em camadas Raw/Bronze, Silver e Gold.

Limitações: Dependência de infraestrutura de cluster e custos de armazenamento.

Trabalho individual: Cada integrante será responsável por uma parte do pipeline (ingestão,

processamento, armazenamento, visualização, documentação).

3. Arquitetura do Projeto

Diagrama de componentes: UML/Data Flow Diagram representando ingestão → processamento → armazenamento → análise.

Fluxo do pipeline: Kafka → Spark → MinIO (Raw/Bronze/Silver/Gold) → Metabase.

Camadas: Raw (dados brutos), Bronze (dados tratados), Silver (dados agregados), Gold (dados prontos para análise).

Infraestrutura: Containers Docker, serviços externos em cluster Kubernetes.

Formato dos dados: JSON e Parquet.

Governança: Catálogo de dados, validação de esquemas, versionamento com Delta Lake.

4. Componentes Técnicos

4.1 Ingestão

Kafka para streaming de dados.

Airflow para ingestão batch.

Pré-processamento com limpeza de dados.

4.2 Processamento

Spark para transformações e agregações.

Lógica de negócio: cálculo de KPIs de vendas (ticket médio, churn, lifetime value).

4.3 Armazenamento

MinIO como Data Lake.

Camadas Raw/Bronze/Silver/Gold.

Formato Parquet para eficiência.

4.4 Análise e Visualização

Metabase para dashboards.

KPIs: vendas por região, ticket médio, previsão de demanda.

4.5 API (Opcional)

Endpoint Flask para servir dados processados.

5. Critérios de Avaliação

Entendimento da solução.

Capacidade de explicar componentes técnicos.

Clareza sobre papel individual.

Noções de arquitetura de dados.

Domínio das ferramentas.

6. Requisitos de Apresentação

Resumo do problema.

Demonstração do pipeline.

Explicação da arquitetura.

Melhorias futuras: uso de ML para previsão mais precisa.

Perguntas individuais.

7. Organização do Repositório

/docs - documentação.

/src - código-fonte.

/infra - configs Docker/Terraform.

/notebooks - análise exploratória.

/datasets - dados de teste.

README.md - guia de execução.

8. Considerações Finais

Este trabalho reflete práticas reais de Engenharia e Ciência de Dados. A clareza, organização e justificativas técnicas são diferenciais para avaliação. O próximo passo é distribuir tarefas entre os membros do grupo.