Documento de Requisitos

Prova Prática de Ciência de Dados e Big Data

1. Visão Geral do Trabalho

A entrega é dividida em **duas partes obrigatórias**:

- **Documentação no Confluence** (estrutura organizada em páginas, links e anexos quando necessário).
- **Código-fonte no Bitbucket ou GitHub** (repositório versionado, com branches organizadas e README).

O projeto consiste na construção de uma solução completa de Ciência de Dados/Big Data, envolvendo coleta, processamento, armazenamento e apresentação de insights. O trabalho deve ser realizado em **grupo de até 5 participantes**, porém a **avaliação é individual**, incluindo perguntas específicas feitas a cada integrante.

O objetivo é demonstrar domínio técnico, clareza conceitual e capacidade de estruturar um fluxo de dados robusto e bem documentado.

---

2. Entregáveis Obrigatórios

2.1 Documentação Geral

A documentação deve incluir:

- **Descrição do problema** abordado pelo projeto.
- **Objetivos do sistema** e justificativa técnica.
- **Escopo da solução** (o que está incluído e o que não está).
- **Arquitetura completa** do pipeline.
- **Detalhes das ferramentas e tecnologias** utilizadas.
- **Decisões técnicas** (trade-offs, alternativas consideradas e por quê).
- **Guia de execução**: como rodar o projeto do zero.
- **Guia de dependências** (versões de libs, serviços, containers e configurações).
- **Descrição dos dados**: origem, formato, esquema, dicionário de dados.
- **Pontos de falha e limitações**.
- **Trabalho individual**: descrição breve das responsabilidades de cada integrante.

---

3. Arquitetura do Projeto

A arquitetura deve ser apresentada de forma clara, incluindo:

- **Diagrama de componentes** (pode ser UML, Data Flow Diagram, Mermaid, etc.).
- **Fluxo do pipeline de dados** (ingestão → processamento → armazenamento → análise).
- **Camadas da arquitetura** (Raw, Bronze, Silver, Gold, caso use Data Lake/Lakehouse).
- **Descrição da infraestrutura**: containers, serviços externos, cluster ou nuvem.
- **Explicação do formato dos dados** (Parquet, JSON, CSV, etc.).
- **Estratégias de governança e qualidade de dados** (catalogação, validação, versionamento).

---

4. Componentes Técnicos Esperados

4.1 Ingestão de Dados

- Pode ser streaming (Kafka), batch (Airflow), ou ambos.
- Explicar como a coleta é agendada ou consumida.
- Incluir tratamentos básicos de pré-processamento.

4.2 Processamento

- Pode usar Spark, Pandas, SQL, ou similar.
- Explicar transformações, agregações e cálculos.
- Detalhar lógica de negócio aplicada aos dados.

4.3 Armazenamento

- Preferencialmente Data Lake (MinIO, S3, HDFS).
- Explicar camadas (Raw/Bronze, Silver, Gold).
- Justificar o formato usado.

4.4 Análise e Visualização

- Pode usar Metabase, Grafana, Power BI, Superset ou dashboards customizados.
- Deve apresentar KPIs, métricas e visualizações com base nos dados tratados.

4.5 API ou Interface (Opcional)

- Endpoint para servir dados processados.

---

5. Critérios de Avaliação

A avaliação será focada de forma **individual**, considerando:

- Entendimento real da solução entregue.

- Capacidade de explicar componentes técnicos.
- Clareza sobre seu papel no grupo.
- Noções de arquitetura de dados.
- Domínio das ferramentas utilizadas.

---

6. Requisitos de Apresentação

A apresentação deve conter:

- Resumo do problema resolvido.
- Demonstração do pipeline.
- Explicação da arquitetura.
- Melhorias que poderiam ser implementadas.
- Perguntas individuais do avaliador.

---

7. Organização do Repositório

O repositório deve conter:

- `/docs` — documentação completa.
- `/src` — scripts e código-fonte.
- `/infra` — Docker Compose, Terraform ou configs.
- `/notebooks` — notebooks de análise exploratória.
- `/datasets` — dados de teste.
- `README.md` — guia de execução + visão geral.

---

8. Considerações Finais

O trabalho deve refletir práticas reais de Engenharia e Ciência de Dados. Clareza, organização, justificativas técnicas e entendimento profundo do pipeline serão diferenciais marcantes na avaliação.

Este documento de requisitos serve como referência para garantir que o projeto final seja completo, bem estruturado e tecnicamente sólido. O próximo passo natural é transformar esses requisitos em tarefas distribuídas entre os membros do grupo.