Proposta de Projeto em Fundamentos de Big Data

1. Introdução

Este documento detalha a proposta do projeto final para a disciplina de Fundamentos de Big Data. O objetivo principal é proporcionar uma experiência prática e imersiva no desenvolvimento de uma solução completa para um problema do mundo real, utilizando um pipeline de Big Data. Os alunos terão a liberdade de escolher, propor e desenvolver um projeto, desde a concepção da ideia até a apresentação de uma solução funcional, simulando um ambiente de trabalho e as demandas do mercado.

2. Objetivo do Projeto

Desenvolver uma solução baseada em dados que resolva um problema real, aplicando os conceitos e ferramentas de Big Data aprendidos em sala de aula. O projeto deve obrigatoriamente abranger todas as etapas de um pipeline de dados, desde a coleta em fontes diversas até a disponibilização dos insights gerados para análise.

3. Requisitos Gerais

- **Equipes:** O projeto deverá ser realizado em equipes de exatamente **3 (três) participantes**.
- **Pipeline Completo:** A solução deve implementar e documentar um pipeline de Big Data, especificando cada uma de suas etapas.
- **Problema Real:** A escolha do tema é livre, contanto que o problema a ser resolvido seja relevante e a proposta seja viável de ser executada no tempo determinado.
- **Código Aberto:** Todo o código e documentação do projeto devem ser versionados e publicados em um repositório no GitHub.

4. O Pipeline de Big Data: Etapas Obrigatórias

A metodologia do projeto deve ser centrada na construção de um pipeline de dados. A equipe deverá detalhar as ferramentas e os processos utilizados em cada uma das seguintes etapas:

1. **Fontes de Dados (Data Sources):**
   - **Descrição:** O ponto de origem dos dados. Podem ser estruturados, semiestruturados ou não estruturados.
   - **Exemplos:** APIs de redes sociais (Twitter (X), Reddit), dados de sensores de IoT, logs de servidores web, bancos de dados transacionais (SQL), bancos de dados NoSQL, arquivos CSV/JSON, dados governamentais abertos, etc.
2. **Ingestão (Ingestion):**
   - **Descrição:** O processo de coletar os dados brutos das fontes e trazê-los para dentro do ecossistema de dados. Pode ser em batch (lotes) ou streaming (tempo real).

- **Exemplos de Ferramentas:** Batch (mais comum): pandas.read_csv, requests para baixar arquivos, ou upload manual no Colab.
- Streaming (simulação): gerador Python que lê arquivo linha a linha em loop e grava em "micro-lotes".

3. **Transformação (Transformation):**
- **Descrição:** A etapa mais crítica, onde os dados são limpos, normalizados, enriquecidos, agregados e preparados para análise. É aqui que os dados brutos ganham valor.
    - **Exemplos de Ferramentas e Processos:** Pandas (principal). NumPy (operações vetorizadas). PyArrow + Parquet para salvar com mais eficiência. Validação simples: drop de nulos, normalização, criação de colunas derivadas. Entrega: CSV ou Parquet transformado em /silver.
4. **Carregamento (Loading):**
- **Descrição:** O processo de carregar os dados já transformados e refinados para um sistema de armazenamento final.
    - **Exemplos de Ferramentas:** Salvar dados limpos em CSV/Parquet no Colab (/gold). O o próprio Parquet/CSV já funciona como destino final.
5. **Destino (Destination):**
- **Descrição:** O local final onde os dados processados são armazenados e ficam disponíveis para consumo, seja por analistas, cientistas de dados ou ferramentas de visualização.
    - **Exemplos de Ferramentas:** CSV/Parquet no /gold. Jupyter/Colab com gráficos (matplotlib, seaborn, plotly).

6. Entregáveis

6.1 AV1

Primeira Entrega – AV1 (13/10)

Objetivo: Garantir que cada grupo tenha o pipeline de dados iniciado e documentado, demonstrando o progresso nas etapas de ingestão, armazenamento e transformação, usando a base escolhida.

Minientregas Obrigatórias

1. **Documento de Arquitetura (PDF ou DOCX)**
    - Diagrama do pipeline de dados atual (ingestão, armazenamento e transformação).
    - Tecnologias já utilizadas e quais poderiam ser usadas para refinamento (tecnologias pagas) e justificativa da escolha.
    - Arquitetura parcial implementada (mesmo que em ambiente simulado).
    - Equipe responsável e divisão de tarefas.
2. **Repositório GitHub do grupo**
    - Estrutura organizada com:
        - /dados (amostras de dados, se aplicável)
        - /codigo ou /src (scripts e notebooks)

        - /documentacao (diagramas, PDFs, etc.)
    - README inicial com:
        - Nome e descrição do projeto
        - Fonte dos dados
        - Ferramentas já aplicadas
    - Commits visíveis e com mensagens claras (cada membro deve ter contribuição registrada).
3. **Demonstração Técnica (em aula)**
    - Mostra do funcionamento da ingestão e/ou transformação com prints, outputs ou notebook.
    - Pode ser simulação parcial caso o pipeline ainda não esteja completo.
    - Cada grupo terá 8 min para apresentar a evolução do seu projeto
4. **Checklist Preenchido**
    - Formulário com o estado atual de cada parte do pipeline:
        - Ingestão: ( ) Em progresso / ( ) Finalizado / ( ) Pendente
        - Armazenamento: ( ) Em progresso / ( ) Finalizado / ( ) Pendente
        - Transformação: ( ) Em progresso / ( ) Finalizado / ( ) Pendente

6.2 AV2

A. Repositório no GitHub (Entrega Principal)

Esta é a forma de entrega preferencial. O repositório deve ser público e ter os 3 membros da equipe como colaboradores. A estrutura do repositório deve funcionar como o relatório do projeto, contendo:

- **README.md:** Um arquivo de apresentação robusto, seguindo a estrutura de um relatório ABNT, com:
    - **Introdução:** Apresentação do tema e do problema.
    - **Motivação:** Relevância e justificativa da escolha do projeto.
    - **Objetivo do Projeto:** O que se pretende alcançar com a solução.
    - **Metodologia (Pipeline de Dados):** Descrição detalhada de cada etapa do pipeline (Fontes, Ingestão, Transformação, Carregamento, Destino), incluindo as tecnologias e a arquitetura da solução.
    - **Resultados e Visualizações:** Apresentação dos dashboards, gráficos e insights gerados.
    - **Conclusões:** Análise crítica dos resultados, dificuldades encontradas e trabalhos futuros.
- **Pasta /codigo ou /src:** Contendo todos os scripts, notebooks e códigos desenvolvidos.
- **Pasta /notebooks:** Jupyter Notebooks utilizados para exploração e análise.
- **Pasta /dados (opcional):** Amostras pequenas dos dados. Arquivos grandes não devem ser "commitados".
- **Pasta /documentacao:** Arquivos adicionais, como diagramas de arquitetura.

B. Apresentação Final

- **Duração:** Até **20 minutos** por equipe.
- **Conteúdo:** A apresentação deve focar nos resultados e na demonstração do projeto funcionando. Mostrem a pipeline em ação ou, no mínimo, os resultados consolidados em um dashboard interativo.

6. Cronograma

Data	Atividade
24/11 (seg)	Simulação de Pitch técnico com feedback cruzado (simular a apresentação com os possíveis ajustes finais)
26/11 (qua)	Aula de dúvidas e orientações finais
01/12 (seg)	**Apresentações Finais – Grupos 1 a 5**
03/12 (qua)	**Apresentações Finais – Grupos 6 a 10**
10/12 (qua)	**Apresentações Finais – Grupos 11 a 15**

7. Critérios de Avaliação

7.1 AV1

A avaliação AV1 do projeto será composta pelos seguintes critérios:

Mini entregas: Documento de Arquitetura (20%) + Repositório Git (20%) + Checklist preenchido (20%) + Demonstração Técnica (40%)

7.2 AV2

A avaliação AV2 do projeto será composta pelos seguintes critérios:

Qualidade técnica do pipeline (30%) + Profundidade da análise (25%) + Ética e documentação (15%) + Visualizações e storytelling (15%) + Apresentação final (15%)

8. Sugestões de Temas, Problemas e Datasets

Mobilidade Urbana e Cidades Inteligentes

1. **Problema:** Otimizar o sistema de transporte público em uma cidade, prevendo a demanda de passageiros por rota e horário para evitar superlotação.
   - **Dataset Sugerido:** Dados de bilhetagem eletrônica (se disponíveis em portais de dados abertos de prefeituras) ou dados de GPS de ônibus de cidades como o Rio de Janeiro (Data.Rio - GPS de Ônibus).

2. **Problema:** Analisar padrões de acidentes de trânsito para identificar os pontos mais perigosos de uma cidade e propor intervenções.
   - **Dataset Sugerido:** Dados da Polícia Rodoviária Federal (Dados Abertos PRF).
3. **Problema:** Mapear e prever a disponibilidade de bicicletas ou patinetes compartilhados em diferentes regiões da cidade ao longo do dia.
   - **Dataset Sugerido:** Muitas empresas de compartilhamento possuem APIs ou dados históricos. Um dataset genérico pode ser o Bike Share Systems (EUA).

Saúde e Bem-Estar

4. **Problema:** Prever surtos de doenças (como Dengue ou COVID-19) em nível municipal, correlacionando dados de saúde com dados climáticos e de redes sociais.
   - **Dataset Sugerido:** InfoDengue, dados do OpenDataSUS e dados climáticos do INMET.
5. **Problema:** Analisar sentimentos em redes sociais sobre campanhas de vacinação para identificar focos de desinformação e hesitação vacinal.
   - **Dataset Sugerido:** Coleta de dados via API do Twitter/X ou Reddit, focando em palavras-chave relacionadas.
6. **Problema:** Identificar fatores de risco para internações hospitalares a partir de dados anonimizados de pacientes.
   - **Dataset Sugerido:** Dados de internações hospitalares (SIH/SUS) do OpenDataSUS.

E-commerce e Varejo

7. **Problema:** Criar um sistema de recomendação de produtos em tempo real baseado na navegação do usuário e no histórico de compras.
   - **Dataset Sugerido:** Brazilian E-Commerce Public Dataset by Olist. É um dos datasets brasileiros mais completos para este fim.
8. **Problema:** Analisar avaliações de clientes (reviews) para identificar automaticamente os principais pontos fortes e fracos de produtos e serviços.
   - **Dataset Sugerido:** O mesmo dataset da Olist (contém reviews) ou datasets de reviews da Amazon.
9. **Problema:** Prever o risco de *churn* (cancelamento de serviço) de clientes, identificando os principais motivos que levam um cliente a abandonar a plataforma.
   - **Dataset Sugerido:** Telco Customer Churn (dataset clássico e ótimo para o problema).

Meio Ambiente e Agronegócio

10. **Problema:** Monitorar e prever focos de desmatamento ou queimadas na Amazônia utilizando imagens de satélite e outros dados.
    - **Dataset Sugerido:** Dados do INPE (PRODES/DETER) (TerraBrasilis) e imagens de satélite (Landsat/Sentinel).
11. **Problema:** Otimizar o uso de água e fertilizantes na agricultura, prevendo a necessidade da lavoura com base em dados de sensores IoT, clima e imagens de drone/satélite.
    - **Dataset Sugerido:** Syntetic Data for Farming ou dados climáticos do INMET combinados com dados de safras da CONAB.

Finanças e Economia

12. **Problema:** Detectar transações fraudulentas com cartão de crédito em tempo real.

- **Dataset Sugerido:** Credit Card Fraud Detection. É um dataset clássico, excelente para testar modelos de detecção de anomalias.

13. **Problema:** Analisar o sentimento do mercado financeiro a partir de notícias e posts em redes sociais para correlacionar com a volatilidade de ações da B3.
    - **Dataset Sugerido:** Coleta de dados de portais de notícias (via web scraping) e API do Twitter/X, cruzando com dados históricos de ações do Yahoo Finance.

Mídia e Entretenimento

14. **Problema:** Prever o sucesso de um filme (nota ou bilheteria) antes de seu lançamento, com base em dados como elenco, diretor, gênero, orçamento e buzz nas redes sociais.
    - **Dataset Sugerido:** The Movies Dataset (contém metadados de 45 mil filmes).
15. **Problema:** Analisar milhões de partidas de jogos online (ex: League of Legends, CS:GO) para identificar padrões de vitória e estratégias emergentes.
    - **Dataset Sugerido:** League of Legends Diamond Ranked Games.