

Relatório - Movies Dataset

1. Diagrama do Pipeline de Dados

Fluxo Geral do Pipeline

(Representação visual do pipeline com as etapas de ingestão, bronze, silver e gold — conforme o documento original.)

Fluxo Detalhado por Etapa

Ingestão

- Modo: Batch (lotes)
- Formato: CSV
- Volume: ~45K registros (movies_metadata.csv)
- Frequência: Uma vez (dataset estático do Kaggle)

Armazenamento

- Bronze: Parquet (dados brutos padronizados)
- Silver: Parquet (dados limpos e enriquecidos)
- Gold: Parquet + CSV (dados analíticos)

Transformação

- Bronze → Silver: Limpeza, tipagem, parsing de JSON, feature engineering
- Silver → Gold: Seleção, agregação, preparação para consumo

2. Tecnologias Utilizadas e Propostas de Refinamento

2.1 Tecnologias Atualmente Implementadas (Open Source)

Componente | Tecnologia | Justificativa

Processamento | Python + Pandas | Simplicidade, biblioteca madura para manipulação de dados tabulares

Armazenamento | Parquet (Apache Arrow) | Formato colunar eficiente, compressão nativa, compatível com Big Data

Análise Exploratória | Matplotlib | Visualizações simples e diretas

Modelagem | Scikit-learn | Biblioteca padrão para ML em Python, ideal para baseline

Ambiente | Jupyter Notebook / Google Colab | Prototipagem rápida, documentação viva

2.2 Limitações da Arquitetura Atual

- Escalabilidade limitada (Pandas em memória)
- Sem processamento distribuído
- Execução manual, sem orquestração
- Falta de monitoramento e métricas estruturadas
- Ausência de versionamento de datasets

2.3 Tecnologias Propostas para Refinamento (Pegas / Enterprise)

Opção 1: Stack AWS (Cloud-Native)

- Ingestão: AWS S3 + AWS Glue
- Processamento: AWS EMR (Spark)
- Orquestração: Step Functions ou Airflow (MWAA)
- Armazenamento: S3 (Data Lake)
- Catálogo de Dados: AWS Glue Data Catalog
- Query Engine: Athena

- Dashboards: Amazon QuickSight

Custo estimado: entre 500 e 1400 dólares por mês.

Justificativas: escalabilidade elástica, integração nativa, segurança e maturidade da plataforma.

Opção 2: Stack Databricks (Lakehouse Architecture)

- Plataforma unificada: Databricks Lakehouse
- Armazenamento: Delta Lake
- Orquestração: Databricks Workflows
- Feature Store: Databricks Feature Store
- ML: MLflow
- Dashboards: Databricks SQL Dashboards

Custo estimado: entre 900 e 2200 dólares por mês.

Benefícios: performance otimizada, governança com Unity Catalog e notebooks colaborativos.

Opção 3: Stack Google Cloud (Serverless-First)

- Ingestão: Cloud Storage + Dataflow
- Processamento: BigQuery
- Orquestração: Cloud Composer
- Feature Engineering: Vertex AI Feature Store
- ML: Vertex AI
- Dashboards: Looker Studio

Custo estimado: entre 500 e 1500 dólares por mês.

Benefícios: simplicidade, zero manutenção de cluster e integração nativa com ferramentas Google.

2.4 Recomendação Final

Para este projeto acadêmico, a Stack AWS é a mais recomendada devido ao custo-benefício, documentação extensa, flexibilidade e relevância de mercado.

Arquitetura recomendada:

```
S3 (Raw) → AWS Glue ETL → S3 (Bronze/Silver/Gold) → Athena (Query) → QuickSight (Viz)
↓
Step Functions (Orquestração)
```

3. Arquitetura Parcial Implementada

3.1 Ambiente Atual (Simulado Localmente)

Estrutura de diretórios:

```
/dados
  /raw → CSV original
  /bronze → Parquet padronizado
  /silver → Parquet limpo e enriquecido
  /gold → Datasets finais (Parquet + CSV)
```

3.2 Componentes Implementados

Ingestão

- Leitura de CSV com tratamento de erros
- Suporte a múltiplas fontes (Upload, Drive, Kaggle API)
- Validação básica de integridade

Armazenamento

- Estrutura Bronze/Silver/Gold
- Conversão para Parquet com compressão snappy
- Preservação de schema

Transformação

- Bronze: normalização de nomes de colunas

- Silver: coerção de tipos, parsing JSON, criação de ROI e vote_density
- Gold: seleção de colunas core e exportação final

Análise

- EDA básica com histogramas e scatter plots
- Análise de gêneros
- Baseline de modelagem com regressão linear

3.3 Componentes Não Implementados

Orquestração

- Agendamento automático
- Retry em falhas
- Notificações

Monitoramento

- Logs estruturados
- Métricas de performance
- Alertas de anomalias

Qualidade de Dados

- Testes automatizados de schema
- Validação de regras de negócio
- Data profiling

Streaming

- Ingestão em tempo real
 - Processamento incremental
-

4. Equipe Responsável e Divisão de Tarefas

4.1 Estrutura da Equipe

Engenheiro de Dados 1 – Anthony Kevin: camada Bronze, ingestão e padronização.

Engenheiro de Dados 2 – Beatriz Vilarim: camada Silver, limpeza e feature engineering.

Engenheiro de Dados 3 – João Pedro: camada Gold, análise e visualizações.

4.2 Divisão de Tarefas por Sprint

Sprint 1: Fundação (Semanas 1-2)

- Configurar estrutura de diretórios
- Implementar ingestão e padronização
- Preparar notebooks base e documentação inicial

Sprint 2: Transformação (Semanas 3-4)

- Bronze: compressão e validações
- Silver: limpeza, parsing e criação de features
- Gold: deduplicação e consistência de dados

Sprint 3: Analytics e Consolidação (Semanas 5-6)

- Gold: exportação final
- Silver: otimização de queries
- Gold: EDA e visualizações

Sprint 4: Documentação e Entrega (Semana 7)

- Revisão de código
- Consolidação de notebooks
- Relatório técnico e apresentação final

4.3 Cerimônias Ágeis

- Daily Standup: 15 min por dia
- Sprint Planning: 1h no início da sprint
- Sprint Review: 45 min no fim da sprint
- Retrospectiva: 30 min para ajustes de processo

4.4 Ferramentas de Colaboração

- GitHub (branches main, dev, feature/*)
 - GitHub Projects ou Trello
 - Google Colab / Jupyter Notebook
 - Markdown versionado no repositório
 - Discord / WhatsApp
-

5. Próximos Passos e Roadmap

Curto Prazo (1-2 meses)

- Testes automatizados de schema
- Logging estruturado
- Dashboard com Plotly/Streamlit

- Versionamento com DVC

Médio Prazo (3-6 meses)

- Migração para AWS S3 + Glue
- Orquestração com Airflow
- Streaming simulado com Kafka
- Feature store

Longo Prazo (6-12 meses)

- Deploy com CI/CD
- Monitoramento com Prometheus + Grafana
- Data quality checks automáticos
- Escalar para novos datasets

6. Referências Técnicas

- [Databricks Medallion Architecture](#)
- [AWS Data Lake Best Practices](#)
- [Google Cloud Data Engineering](#)
- [The Movies Dataset \(Kaggle\)](#)