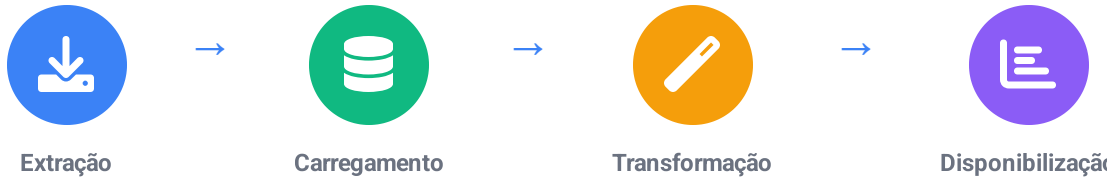
Construção de Pipeline de Dados

ETL vs ELT: Movendo e Transformando Dados em Escala

> ⚠ **Um Data Lake sem um Pipeline de Dados é apenas um pântano de dados**



Abertura: O que é um Pipeline de Dados?

Um Data Lake sem um Pipeline de Dados é apenas um pântano de dados

DEFINIÇÃO DE PIPELINE

Um **pipeline de dados** é o processo automatizado que **move e transforma dados brutos** da fonte até o destino final (análise, relatórios, machine learning). O pipeline garante que os dados cheguem **limpos, organizados e prontos para uso**.

Ciclo de Vida do Pipeline: 4 Fases

1

Extração (E)

Coleta de dados das fontes: APIs, bancos OLTP, arquivos CSV, logs, streaming

2

Carregamento (L)

Ingestão no Data Lake (Camada Bronze). Dados brutos armazenados em S3, ADLS, GCS

3

Transformação (T)

Limpeza, padronização, agregação e enriquecimento. Camadas Prata e Ouro

4

Disponibilização

Carregamento final para consumo: Dashboards, APIs, Modelos de ML

Por que o Pipeline é Crítico?

O pipeline é o **processo mais crítico da Engenharia de Dados**. Projetar um pipeline eficiente garante que a Ciência de Dados e o BI tenham a matéria-prima certa, no tempo certo e com a qualidade necessária. Um pipeline ruim = dados ruins = decisões ruins.

ETL vs ELT: Qual é a Diferença?

Aspecto	ETL (Extract, Transform, Load)	ELT (Extract, Load, Transform)
Ordem	**Extract → Transform → Load**	**Extract → Load → Transform**
Transformação	Feita **antes** do carregamento (servidor local)	Feita **depois** do carregamento (na nuvem)
Escalabilidade	**Limitada** (servidor local)	**Ilimitada** (poder da nuvem)
Latência	**Alta** (transformação lenta)	**Baixa** (transformação paralela)
Custo	**Alto** (servidor dedicado)	**Baixo** (paga-se por uso)
Caso de Uso	Data Warehouses tradicionais	Data Lakes modernos

ETL (Padrão Antigo)

```
MySQL → Transform (Pentaho) → DW
```

Dados são transformados em um servidor local antes de serem carregados no Data Warehouse. Lento e caro.

Características:

- → Transformação centralizada
- → Requer servidor dedicado
- → Não escala bem
- → Exemplo: Pentaho, Informatica

ELT (Padrão Moderno)

```
MySQL → S3 (Bronze) → Spark Transform → Prata/Ouro
```

Dados são carregados brutos na nuvem, depois transformados em paralelo. Rápido e barato.

Características:

- → Transformação distribuída
- → Aproveita poder da nuvem
- → Escala horizontalmente
- → Exemplo: Databricks, AWS Glue

Ciclo de Vida do Pipeline: As 4 Fases

1. Extração

O QUE FAZ

Coleta dados das fontes originais

FONTES

- APIs REST
- Bancos OLTP
- Arquivos CSV
- Logs
- Streaming (Kafka)

Exemplo:
Conector JDBC lê MySQL a cada 1 hora

2. Carregamento

O QUE FAZ

Ingestão no Data Lake (Bronze)

DESTINO

- AWS S3
- Azure ADLS
- Google GCS
- Formato Parquet

Exemplo:
s3://datalake/bronze/vendas/ano=2025/mes=01/

3. Transformação

O QUE FAZ

Limpeza, validação e agregação

FERRAMENTAS

- Apache Spark
- AWS Glue
- SQL
- Python/Pandas

Exemplo:
Remove nulos, padroniza datas, remove duplicatas

4. Disponibilização

O QUE FAZ

Carregamento final para consumo

DESTINOS

- Dashboards BI
- APIs REST
- Modelos ML
- Relatórios

Exemplo:
Power BI, Tableau, Grafana

Ferramentas e Tecnologias por Fase

Extração
FASE 1

APIS & CONECTORES

- **APIs REST**
  Coleta de dados via HTTP. Exemplo: Twitter API, Shopify
- **JDBC/ODBC**
  Conectores de bancos: MySQL, PostgreSQL, Oracle
- **Kafka/Kinesis**
  Streaming em tempo real. Dados contínuos
- **Web Scraping**
  Coleta de dados de websites. Selenium, BeautifulSoup

Storage (L)
FASE 2

CLOUD STORAGE

- **AWS S3**
  Armazenamento de objetos escalável. Padrão de mercado
- **Azure ADLS**
  Data Lake Storage Gen2. Integração Microsoft
- **GCS**
  Google Cloud Storage. Integração BigQuery
- **Formato Parquet**
  Compressão 10x vs CSV. Padrão moderno

Transformação
FASE 3

PROCESSAMENTO

- **Apache Spark**
  Processamento distribuído. Padrão ouro
- **AWS Glue**
  ETL serverless. Spark gerenciado
- **SQL/Python**
  Linguagens: SQL para queries, Python para lógica
- **Airflow**
  Orquestração de workflows. Agendamento

Disponibilização
FASE 4

CONSUMO DE DADOS

- **BigQuery/Redshift**
  Data Warehouses. Queries SQL rápidas
- **Power BI/Tableau**
  Dashboards e visualizações. BI
- **APIs REST**
  Exposição de dados para aplicações
- **ML Models**
  Dados para modelos de Machine Learning

Nota: A escolha de ferramentas depende do volume de dados, latência aceitável e orçamento. Spark é o padrão para transformações em escala. Airflow é essencial para orquestração. A maioria das empresas usa ELT moderno (dados brutos na nuvem, transformação distribuída).

Atividade 1: Planejamento do Pipeline

Instruções: Divida a turma em grupos de 3-4 alunos. Cada grupo escolhe um cenário de negócio e deve detalhar o pipeline em um fluxograma, justificando as escolhas de tecnologia e formato de dados.

Varejo

Contexto: Monitoramento de estoque e vendas em lojas físicas. Ingestão de dados de POS (Ponto de Venda) e atualização de estoque em tempo real.

Tipos de Dados:

- Transações de POS (tempo real)
- Dados de clientes (estruturados)
- Dados de produtos (estruturados)
- Dados de estoque (arquivos CSV)

Sua Tarefa:

- → Defina as fontes de dados (POS, MySQL, CSV)
- → Escolha método de extração (API, JDBC, arquivo)
- → Defina frequência de ingestão
- → Planeje transformações (Bronze → Prata → Ouro)
- → Defina destino final (BI, ML, API)

Saúde

Contexto: Coleta de dados de dispositivos vestíveis (wearables) e prontuários eletrônicos. Dados de streaming e estruturados.

Tipos de Dados:

- Wearables (streaming - alta frequência)
- Prontuários eletrônicos (batch)
- Sensores de cama (streaming)
- Dados de consultas (estruturados)

Sua Tarefa:

- → Defina fontes (Kafka, APIs, sensores)
- → Escolha método para streaming vs batch
- → Planeje anonimização (HIPAA/LGPD)
- → Defina detecção de anomalias
- → Planeje alertas para médicos

Evento

Contexto: Ingestão de dados de venda de ingressos e tráfego em redes sociais durante o evento. Análise em tempo real.

Tipos de Dados:

- Vendas de ingressos (APIs, tempo real)
- Posts de redes sociais (Web scraping)
- Dados de eventos (banco de dados)
- Dados de hashtags e menções

Sua Tarefa:

- → Defina fontes (APIs, Web scraping)
- → Planeje ingestão em tempo real
- → Defina análise de sentimento (NLP)
- → Planeje cálculo de buzz score
- → Defina recomendações de preço dinâmico

Atividade 2: Desenho Detalhado do Fluxograma

Instruções: Os grupos devem usar uma ferramenta de desenho (Draw.io, Miro, Lucidchart) ou papel para criar o fluxograma do ELT. Incluir todos os componentes obrigatórios e justificar as escolhas de tecnologia.

Componentes Obrigatórios

1 Fonte de Dados
Especificar: Banco MySQL, API REST, Arquivo CSV, Streaming Kafka, etc.

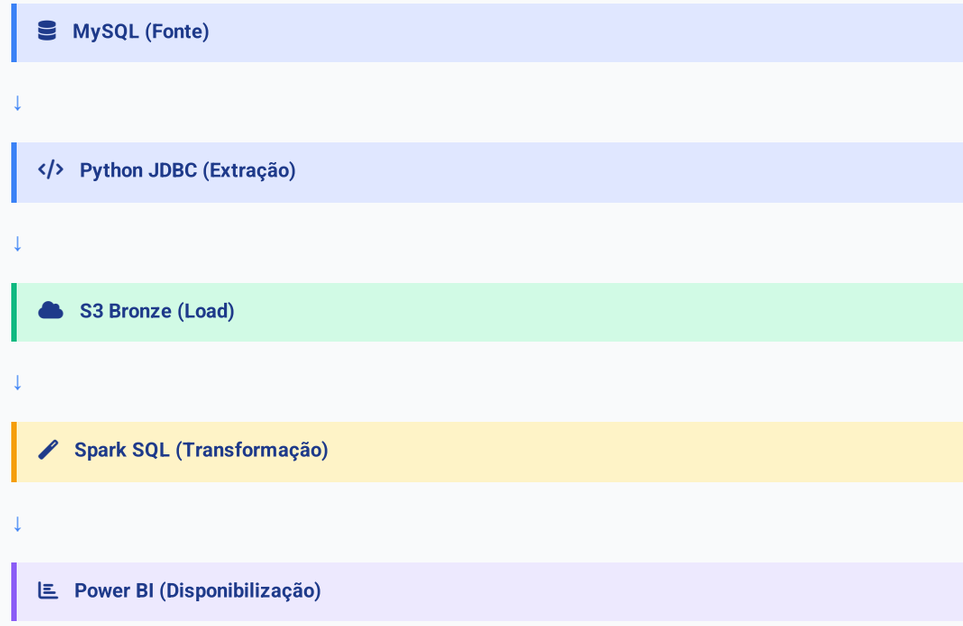
2 Extração (E)
Método de coleta: Conector JDBC, Script Python, Web Scraping, etc.

3 Load - Camada Bronze
Onde dados brutos são depositados: S3, ADLS, GCS com particionamento

4 Transformação (T)
Etapas de limpeza e agregação: Prata (limpo) → Ouro (agregado)

5 Destino (Disponibilização)
Onde dado final vai: Dashboard BI, API, Modelo de ML, Relatório

Exemplo de Fluxograma



Justificativas Necessárias (Escrever um parágrafo para cada)

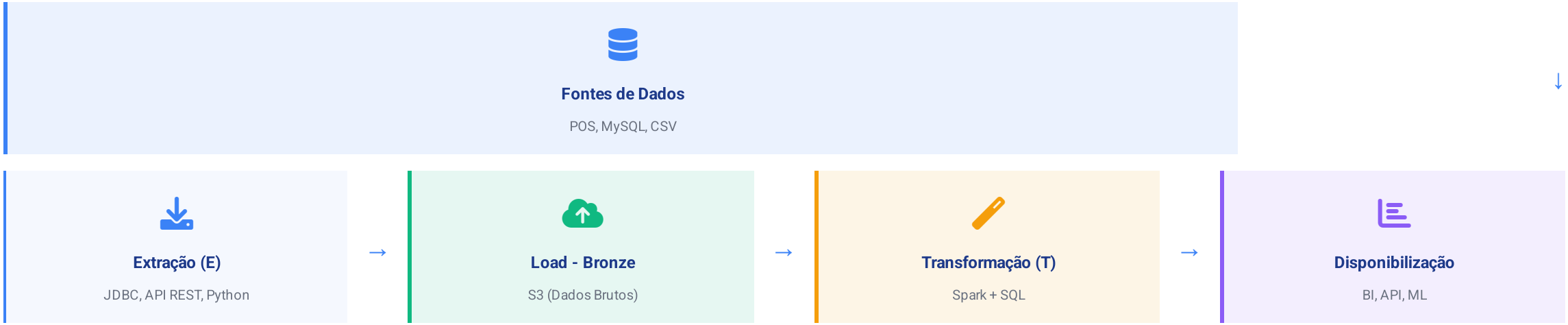
→ **Formato de Dados:** Por que escolheram CSV/JSON/Parquet? Qual é o volume? Qual é a frequência?

→ **Tecnologia de Transformação:** Por que escolheram Python/Pandas vs. SQL vs. Spark? Qual é a complexidade?

→ **Camadas:** Como o dado é limpo ao passar da Bronze para Prata (ex: removemos nulos, padronizamos datas)?

→ **Frequência de Ingestão:** Tempo real, horária, diária? Por quê? Qual é a latência aceitável?

Exemplo Prático: Pipeline de Varejo



Fontes

- POS em tempo real
- MySQL (1h)
- CSV estoque (diário)

Transformações

- Prata: Remove nulos
- Padroniza datas
- Ouro: Agrega vendas

Destinos

- Dashboard Power BI
- API REST (mobile)
- Modelo de ML

EXTRAÇÃO	STORAGE	TRANSFORMAÇÃO	ORQUESTRAÇÃO
JDBC, Python	AWS S3	Spark + SQL	Apache Airflow

Exemplo Prático: Pipeline de Saúde

Fontes de Dados

Wearables, Prontuários, Sensores

Extração (E)

Kafka, APIs, MQTT

→

Load - Bronze

S3 (Dados Brutos)

→

Transformação (T)

Spark Streaming + SQL

→

Disponibilização

Dashboard, Alertas, ML

Fontes

- Wearables (streaming)
- Prontuários (batch 6h)
- Sensores de cama (streaming)

Transformações

- Prata: Anonimiza dados
- Remove outliers vitais
- Ouro: Detecta anomalias

Disponibilização

- Dashboard médico (tempo real)
- Alertas para médicos
- Modelo de ML (previsão)

ANONIMIZAÇÃO (PRATA)

ID paciente → Hash criptográfico. Remove PII. Conformidade HIPAA/LGPD

DETECÇÃO DE ANOMALIAS (OURO)

FC > 200 bpm? Oxigenação < 90%? Gera alerta automático para médico

EXTRAÇÃO	STORAGE	TRANSFORMAÇÃO	ORQUESTRAÇÃO
Kafka, MQTT	AWS S3	Spark Streaming	Apache Airflow

Debate e Apresentação de Grupos

Cada grupo apresenta seu fluxograma do pipeline (5-7 minutos). Foco na lógica de transformação, justificativa de tecnologias e como os dados fluem de Bronze → Prata → Ouro. Após cada apresentação, a turma faz perguntas críticas.

1

Se o volume de dados fosse 10 vezes maior, qual parte do seu pipeline falharia primeiro?

Resposta esperada: O processo de transformação ou a ferramenta de ingestão. Discussão sobre escalabilidade horizontal vs vertical.

2

Como vocês garantem a qualidade dos dados em cada camada (Bronze → Prata → Ouro)?

Resposta esperada: Validações, testes de dados, monitoramento de qualidade, alertas de anomalias.

3

Qual é o custo de manter este pipeline em produção? Como vocês otimizariam para reduzir custos?

Resposta esperada: Armazenamento, processamento, transferência de dados. Compressão, particionamento, retenção de dados.

4

Como vocês lidariam com falhas no pipeline? Qual é o plano de recuperação?

Resposta esperada: Retry automático, dead letter queues, alertas, rollback, testes de recuperação.

Alinhamento Coletivo & Consolidação

→ **Padrões Observados:** Quais foram as escolhas de tecnologia mais comuns? Por quê? Spark vs SQL? Batch vs Streaming?

→ **Diferenças Importantes:** Quais foram as maiores diferenças entre os pipelines? Como cada cenário (Varejo, Saúde, Evento) exigiu abordagens diferentes?

→ **Lições Aprendidas:** Qual foi a maior dificuldade ao projetar o pipeline? O que vocês fariam diferente?

→ **Próximos Passos:** Como este pipeline evoluiria em 6 meses? Quais seriam as melhorias prioritárias?

Conclusão: O Pipeline é a Base da Engenharia de Dados

O pipeline é o processo mais crítico da Engenharia de Dados

Checklist de Design

- [x] **Fontes Identificadas**
  Todas as fontes de dados foram mapeadas?
- [x] **Frequência Definida**
  Tempo real, horária, diária? Latência aceitável?
- [x] **Formato Escolhido**
  CSV, JSON, Parquet? Volume de dados?
- [x] **Transformações Planejadas**
  Bronze → Prata → Ouro? Lógica de negócio?
- [x] **Ferramentas Selecionadas**
  Spark, SQL, Python? Orquestração com Airflow?
- [x] **Destino Final Definido**
  Dashboard, API, ML? Quem consome os dados?
- [x] **Qualidade Validada**
  Testes de dados? Monitoramento? Alertas?

Impacto do Pipeline

Um pipeline bem projetado garante que a Ciência de Dados e o BI tenham a **matéria-prima certa, no tempo certo e com a qualidade necessária**. Um pipeline ruim = dados ruins = decisões ruins.

Conceitos-Chave Aprendidos:

- → **ETL vs ELT:** ELT é o padrão moderno para Data Lakes em nuvem
- → **Ciclo de Vida:** Extração → Load → Transformação → Disponibilização
- → **Arquitetura em Camadas:** Bronze (bruto) → Prata (limpo) → Ouro (agregado)
- → **Ferramentas Essenciais:** Spark, SQL, Airflow, Cloud Storage

Projetar um pipeline eficiente é garantir o sucesso da sua estratégia de dados