

# Projeto Teste Técnico Data Services

## Analista de Dados Pleno - Pmweb

Este projeto implementa uma solução completa de ETL (Extract, Transform, Load) com 9 análises estratégicas e visualização interativa para um sistema de vendas/marketing. Desenvolvido como resposta ao teste técnico para Analista de Dados Pleno, utilizando:

- Processamento de dados com Python e Pandas
- Engenharia de dados com SQL e modelagem de banco
- Análise exploratória com múltiplas perspectivas de negócio
- Visualização de dados interativa com Pygwalker
- Automação de pipelines de dados

### Objetivos do Projeto

- 1.ETL Automatizado: Processar arquivos CSV de clientes e pedidos
- 2.Análises Estratégicas: Gerar 9 relatórios CSV com insights de negócio
- 3.Visualização Interativa: Dashboard exploratório com Pygwalker
- 4.Documentação Completa: Diagrama DER e relatório consolidado

### Tecnologias Utilizadas

Tecnologia	Versão	Finalidade
Python	3.8+	Linguagem principal
Pandas	2.0.3	Processamento de dados
SQLite	3.35+	Banco de dados local
Pygwalker	0.3.9	Visualização interativa
SQLAlchemy	2.0.23	ORM para SQLite
Matplotlib	3.7.2	Visualizações estáticas
Plotly	5.18.0	Gráficos interativos

### Análises Implementadas

#### 1. Pedidos por Semestre (Parcelados)

- **Objetivo:** Identificar padrões de compra parcelada por período
- **Métrica:** Quantidade de pedidos por cliente semestralmente
- **Insight:** Sazonalidade de vendas parceladas

## **2. Ticket Médio por Cliente**

- **Objetivo:** Analisar valor médio gasto por cliente
- **Métrica:** Ticket médio mensal e anual
- **Insight:** Comportamento de gasto ao longo do tempo

## **3. Intervalo entre Compras**

- **Objetivo:** Medir fidelidade e frequência de compra
- **Métrica:** Dias médios entre compras por cliente
- **Insight:** Identificação de clientes ativos vs. ocasionais

## **4. Classificação em Tiers**

- **Objetivo:** Segmentar clientes por valor gasto
- **Categorias:** Básico (<1k), Prata (1-2k), Ouro (2-5k), Super (>5k)
- **Insight:** Foco em clientes de alto valor

## **5. Comparativo Segmentos 2019 vs 2020**

- **Objetivo:** Analisar evolução de segmentos estratégicos
- **Segmentos:** Som e Papelaria
- **Insight:** Performance por categoria de produto

## **6. Análise por Faixa Etária**

- **Objetivo:** Compreender comportamento por idade
- **Faixas:** 18-25, 26-35, 36-45, 46-55, 56-65, 66+
- **Insight:** Segmentação para marketing etário

## **7. Análise por Localização Geográfica**

- **Objetivo:** Identificar padrões regionais de vendas
- **Dimensões:** UF e cidade
- **Insight:** Foco geográfico para expansão

## **8. Sazonalidade por Dia da Semana**

- **Objetivo:** Identificar padrões temporais de compra
- **Métrica:** Vendas por dia da semana
- **Insight:** Otimização de campanhas por período

## **9. Clientes com Pagamentos Não Confirmados**

- **Objetivo:** Identificar oportunidades de recuperação
- **Segmentação:** Com/sem permissão de email
- **Insight:** Campanhas de remarketing segmentadas

# Como Executar

## Instalação

### 1. Clone o repositório

```
git clone https://github.com/GustavoNeri/teste_data_services_pmweb.git  
Ou download do arquivo "teste_data_services_pmweb_gustavo_neri.zip" e extrair
```

```
cd teste_data_services_pmweb
```

### 2. Instale as dependências

```
pip install -r requirements.txt
```

Ou execute o projeto que instalará mediante aceitação

```
python3 run.py
```

## Execução Completa

```
# Coloque os arquivos CSV na pasta data/ caso não existam  
# data/CADASTROS.csv  
# data/PEDIDOS.csv  
  
# Execute o projeto completo  
python3 run.py
```

O script executará automaticamente:

1. Instalação de dependências (se necessário)
2. Processamento ETL dos dados
3. Geração das 9 análises
4. Criação do diagrama DER
5. Abertura do dashboard interativo

## Execuções Individuais

### Apenas ETL

```
python3 -c "from src.etl import executar_etl; executar_etl()"
```

### Apenas análises

```
python3 -c "from src.analises import gerar_analises; gerar_analises()"
```

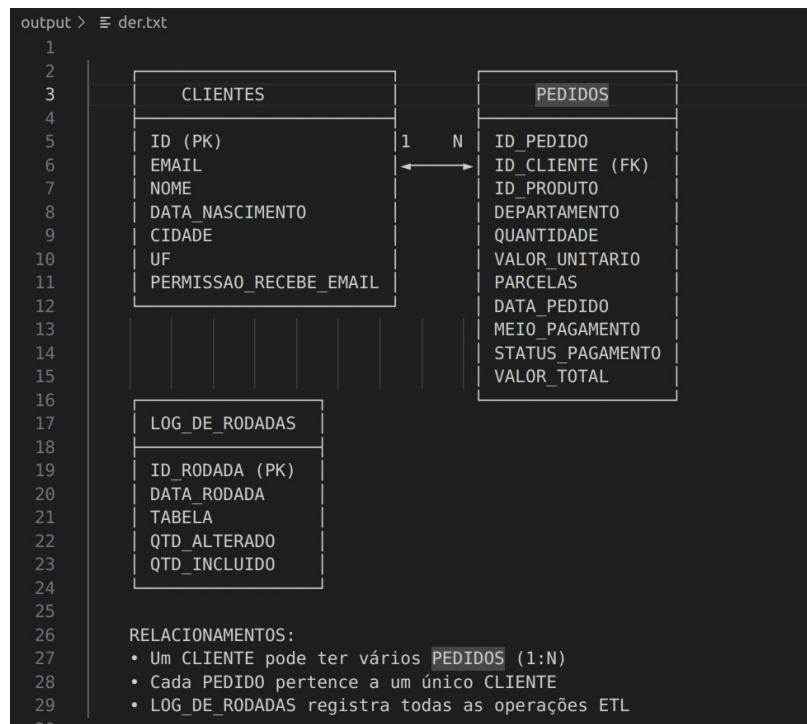
## Apenas dashboard

`python3 src/visualizacao.py`

## Verificar estrutura do banco

`python3 src/verifica_tabelas.py`

## Modelo de Dados – Diagrama de Entidade de Relacionamento (DER)



## Detalhes Técnicos

### Pipeline ETL

1. **Extração**: Leitura de CSVs com tratamento de encoding (latin1)

2. **Transformação**:

- Mapeamento de colunas para estrutura padronizada
- Cálculo de VALOR\_TOTAL (quantidade × valor\_unitario)
- Conversão de datas
- Normalização de tipos de dados

3. **Carga**: Inserção em SQLite com logging completo

### Sistema de Logging

- Tabela LOG\_DE\_RODADAS**: Registra todas as operações ETL
- Métricas**: Quantidade de registros incluídos/alterados
- Timestamp**: Data/hora exata de execução

## Tratamento de Erros

- **CSV Problemáticos:** `on_bad_lines='skip'`
- **Encoding:** Suporte a latin1 e UTF-8
- **Dados Faltantes:** Preenchimento com valores padrão
- **Tipos de Dados:** Conversão segura com `pd.to_numeric()`

## Visualização Interativa

### Dashboard Pygwalker

- **Interface Drag-and-Drop:** Criação de gráficos sem código
- **Múltiplas Visualizações:** Barras, linhas, pizza, dispersão
- **Filtros Dinâmicos:** Exploração interativa dos dados
- **Exportação:** Salve gráficos como PNG/HTML

## Como Usar o Dashboard

**Execute apenas o dashboard ou todo projeto**

`python3 src/visualizacao.py`

Ou `python3 run.py`

**Escolha entre 11 opções:**

- 1-2: Dados brutos (Pedidos/Clientes)
- 3-11: Análises processadas (1-9)

## Principais Diferenciais

1. **Modularidade:** Queries SQL separadas do código Python
2. **Robustez:** Tratamento completo de erros e logs
3. **Interatividade:** Dashboard exploratório com Pygwalker
4. **Documentação:** Diagrama DER e relatório consolidado
5. **Automação:** Pipeline completo em script único

Desenvolvido por Gustavo dos Santos Neri de Souza como resposta ao Teste Técnico para Analista de Dados Pleno da Empresa *Pmweb*.