Fundação Escola de Comércio Álvares Penteado

Ciência da Computação 4º Semestre

Projeto Interdisciplinar: Ciência de Dados

Gabriel Henrique Coelho Marussi - 24026609

Lucas Kenichi Soares – 24026179

Felipe Oluwaseun Santos Ojo – 24026245

Arthur Rodrigues Ferreira – 24026567

Pedro Dimitry Zyrianoff – 24026165

São Paulo – 2025

0 - Sumário

1 - Introdução Geral

A preparação de dados é uma etapa essencial para garantir a qualidade e integridade das análises executadas nos dashboards financeiros (CFO) e operacionais (CEO).
Ambos os dashboards utilizam a base *PicMoneyDados.xlsx*, porém cada um acessa:

- Aba "PicMoney-Unificada" → Análises financeiras do CFO
- Aba "PicMoney-Base-Simulada" → Análises operacionais do CEO

Antes de gerar os dashboards, foi aplicada uma série de transformações fundamentais para:

- Limpar
- Selecionar
- Derivar
- Integrar
- Uniformizar
- Formatar

Essas etapas garantem consistência e desempenho adequado no Streamlit, Plotly e cálculos internos.

2 - Seleção dos Dados

Nesta fase foram selecionadas apenas as colunas relevantes para cada análise.

Exemplos:

CFO — colunas relevantes

- categoria_estabelecimento
- valor_cupom
- valor_compra
- tipo_cupom
- data_captura

CEO — colunas relevantes

- local / região
- gênero
- idade
- user_id
- data_cadastro

Código de Seleção (Colab)

```
import pandas as pd

df = pd.read_excel("PicMoneyDados.xlsx", sheet_name="PicMoney-
Unificada")

colunas_interesse = [
    "categoria_estabelecimento", "valor_cupom", "valor_compra",
    "tipo_cupom", "data_captura"
]

df = df[colunas_interesse]
```

3 - Limpeza e Uniformização dos Dados

A limpeza foi essencial porque as bases possuíam:

- dados fora de padrão
- formatação inconsistente
- nomes de colunas diferentes entre abas
- datas com erros
- possíveis duplicatas
- valores nulos

Tratamento de Nulos

```
df = df.fillna({
    "categoria_estabelecimento": "Não Informado",
    "valor_cupom": 0,
    "valor_compra": 0
})
```

Tratamento de Datas

```
df["data_captura"] = pd.to_datetime(df["data_captura"],
errors="coerce")
```

Padronização de Textos

```
df["categoria_estabelecimento"] = (
    df["categoria_estabelecimento"]
    .str.strip()
    .str.title()
)
```

Remoção de Duplicatas

```
df = df.drop_duplicates()
```

4 - Derivação de Dados

Nesta etapa foram geradas novas informações a partir das existentes.

Exemplos aplicados:

- Ano da captura
- Mês
- Dia da semana
- Faixa etária
- Classificação de valor (baixo / médio / alto)

Código — Derivação Temporal

```
df["ano"] = df["data_captura"].dt.year
df["mes"] = df["data_captura"].dt.month
df["dia_semana"] = df["data_captura"].dt.day_name()
```

Código — Faixas Etárias (CEO)

```
df["faixa_etaria"] = pd.cut(
    df["idade"],
    bins=[0, 17, 25, 35, 45, 60, 80, 120],
    labels=["<18", "18–25", "26–35", "36–45", "46–60", "61–80", "80+"]
)
```

Código — Categorias Numéricas

```
df["classe_valor"] = pd.cut(
    df["valor_cupom"],
    bins=[0, 50, 200, 1000],
    labels=["Baixo", "Médio", "Alto"]
)
```

5 - Integração dos Dados

A integração consistiu em duas estratégias:

1. Normalização de nomes de colunas

Como mostrado no arquivo do CEO, foi necessário criar uma função robusta para "descobrir" colunas mesmo quando o nome estava diferente:

```
import unicodedata

def normalize(text):
    text = str(text).strip().lower()
    text = ''.join(ch for ch in unicodedata.normalize('NFD', text)
                   if unicodedata.category(ch) != 'Mn')
    return text.replace(" ", "").replace("_", "").replace("-", "")
```

Essa função detectava colunas como:

- gênero / genero / sexo
- local / região / estado / uf
- data / data_cadastro / registro

2. Integração entre tabelas (quando possível)

```
df_total = pd.merge(df_operacional, df_financeiro, on="id_usuario",
how="outer")
```

6 - Formatação dos Dados

Antes da exportação e plotagem, várias formatações foram aplicadas.

Ordenação

```
df = df.sort_values(by="data_captura")
```

Formatação de moeda

```
df["valor_formatado"] = df["valor_cupom"].apply(lambda x: f"R$
{x:,.2f}")
```

Reorganização de colunas

```
df = df[[
    "data_captura", "categoria_estabelecimento", "valor_cupom",
    "valor_compra", "tipo_cupom", "ano", "mes", "dia_semana"
]]
```

Exportação final (simulação de entrega)

```
df.to_excel("dados_preparados_final.xlsx", index=False)
```

7 - Como as etapas foram aplicadas nos dashboards

Esta seção conecta o processo de preparação com o uso real nos dashboards CFO e CEO:

7.1 Dashboard CFO (Análise Financeira)

Com base no arquivo **analise_cfo.py**

Análise do CFO

A preparação permitiu:

- somas por categoria
- médias e totais por data
- análise temporal
- comparações por valor
- distribuição de proporções
- agrupamentos para funções do Streamlit

Códigos como:

```
df_cat =
df.groupby("categoria_estabelecimento")[["valor_cupom","valor_compra"]
].sum()
```

e

```
df_data = df.groupby("data_captura")["valor_cupom"].sum()
```

dependem de dados já limpos e uniformizados.

7.2 Dashboard CEO (Análise Operacional)

Com base no arquivo **analise_ceo.py**

Análise do CEO

A preparação foi essencial para:

- detectar colunas independente do nome real
- gerar faixas etárias
- agrupar por local
- contar perfis por gênero
- normalizar dados inconsistentes

Trechos como:

```
df_local = df[col_local].value_counts()
```

e

```
df["faixa_etaria"] = pd.cut(...)
```

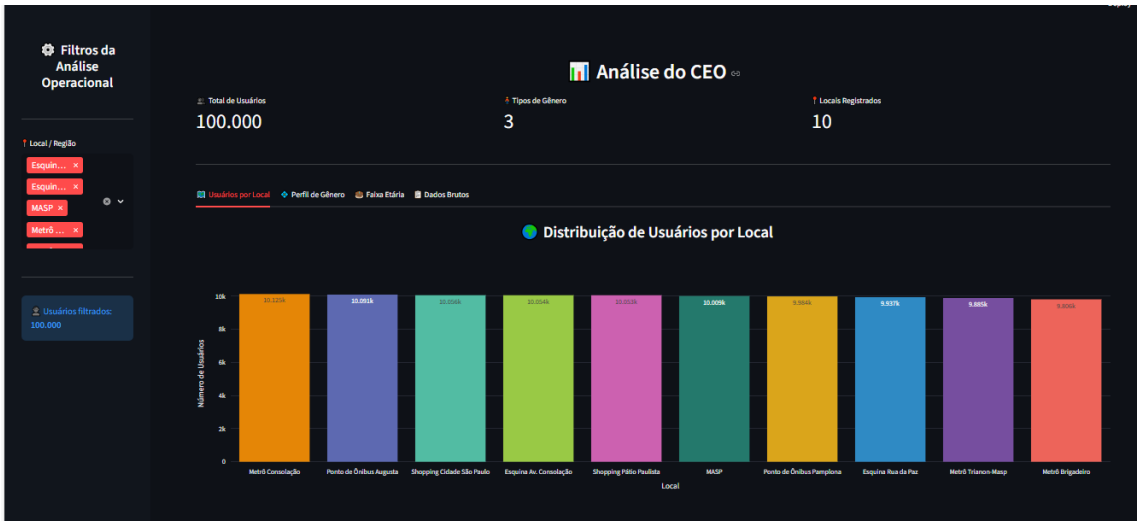
dependem de dados previamente uniformizados.

8 - Conclusão

A preparação de dados foi um processo fundamental que deu suporte direto:

✅ à análise financeira (CFO)
✅ à análise operacional (CEO)
✅ aos gráficos dinâmicos
✅ aos filtros
✅ às métricas
✅ à consistência das abas
✅ à performance no Streamlit

Sem essa preparação, seria impossível gerar análises consistentes, previsões e indicadores confiáveis.


Filtros da Análise Operacional
Local / Região
Esquin...
Esquin...
MASP
Metrô ...
Usuários filtrados: 100.000
Análise do CEO
Total de Usuários
100.000
Tipos de Gênero
3
Locais Registrados
10
Usuários por Local
Perfil de Gênero
Faixa Etária
Dados Brutos
Distribuição de Usuários por Local
Número de Usuários
Metrô Consolação
Ponto de Ônibus Augusta
Shopping Cidade São Paulo
Esquina Av. Consolação
Shopping Pátio Paulista
MASP
Ponto de Ônibus Pamplona
Esquina Rua da Paz
Metrô Trianon-Masp
Metrô Brigadeiro
Local


Filtros do Dashboard
Período de Captura
2025/01/07 – 2025/12/07
Categoria do Estabelecimento
Cafeteria
Clube /...
Clínica...
Eletro ...
Registros filtrados: 43.121
Análise do CFO
Total de Registros
43.121
Soma dos Valores
R$ 23.777.195,76
Média por Registro
R$ 551,41
Categoria
Evolução
Proporção
Tipo de Cupom
Dados
Comparação de Valor do Cupom x Valor da Compra por Categoria
Valor Total (R$)
Tipo de Valor
valor_cupom
valor_compra
Cafeteria
Clube / Cultura & Esporte
Clínicas Médicas e Laboratórios
Eletro & Móveis
Esporte & Fitness
Farmácia
Livraria
Moda
Papelaria
Restaurante
Saúde
Supermercado & Conveniência
Categoria do Estabelecimento