CONIC SEMESP
24º Congresso Nacional de Iniciação Científica

TÍTULO DO TRABALHO: Transformando dados públicos em informação: Um Data Mart para Preços de Combustíveis por Localidade .

AUTOR(ES): Gustavo Araujo Meira Dias, Luan Matheus da Silva Pereira de Sousa, Flávio Resende Ribeiro.
ORIENTADOR(ES): Patricia Bellin Ribeiro

1. RESUMO

Este trabalho apresenta a construção de um Data Mart temático voltado à análise dos preços de combustíveis por litro em diferentes localidades do Brasil, com base em dados públicos disponibilizados pela Agência Nacional do Petróleo (ANP). Para transformar as informações brutas em um ambiente analítico estruturado, foi aplicado o processo de ETL (Extração, Transformação e Carga), utilizando Python para o tratamento dos dados e MariaDB para modelagem e armazenamento. A modelagem adotada segue o padrão dimensional do tipo estrela, com foco na dimensão localidade. Após a implementação, foram realizadas consultas SQL e visualizações gráficas que permitiram identificar padrões regionais de preços, variações entre estados e possíveis distorções associadas à logística e distribuição. Os resultados evidenciam o potencial da solução para apoiar análises exploratórias, subsidiar decisões e servir como base para futuras expansões do modelo analítico.

2. INTRODUÇÃO

A dinâmica de preços dos combustíveis no Brasil é influenciada por uma combinação de fatores estruturais, como a logística de distribuição, a carga tributária estadual, políticas de precificação das distribuidoras e variações na demanda regional. Essa complexidade torna essencial o desenvolvimento de soluções analíticas capazes de proporcionar uma visão mais clara sobre os valores médios praticados em diferentes localidades. Analisar o valor de venda de combustível (R$/L) por localidade permite identificar padrões regionais, variações sazonais e possíveis distorções de mercado. Bases públicas, como as disponibilizadas pela Agência Nacional do Petróleo, Gás Natural e Biocombustíveis (ANP), fornecem dados detalhados sobre os preços praticados por revendedores em todo o território nacional. No entanto, tais dados são frequentemente apresentados em formatos brutos, fragmentados ou com informações pouco padronizadas, o que dificulta sua análise direta por profissionais de negócios ou formuladores de políticas públicas.

Nesse contexto, o presente trabalho propõe a construção de um Data Mart temático, projetado especificamente para a análise dos valores de venda de combustíveis por litro em nível local, por meio da aplicação de um processo chamado ETL (Extração, Transformação e Carga). A solução possibilita a organização e consolidação dos dados em um modelo dimensional adequado à realização de análises exploratórias, consultas específicas e visualizações interativas, alinhando-se aos princípios do Business Intelligence (BI). O objetivo principal é permitir que usuários obtenham respostas precisas e ágeis a partir de dados confiáveis e bem estruturados.

3. OBJETIVO

O objetivo deste trabalho é desenvolver um Data Mart temático para organização e análise de dados públicos sobre preços de combustíveis por localidade no Brasil. A proposta visa possibilitar, por meio da integração entre linguagens e ferramentas como Python e MariaDB, a construção de um ambiente analítico que permita gerar consultas específicas e visualizações que revelem padrões regionais, variações e possíveis distorções nos preços praticados.

4. METODOLOGIA

Este trabalho adota uma abordagem quantitativa e aplicada, com foco na construção de um Data Mart temático para análise dos preços de combustíveis por localidade. O projeto seguiu as etapas clássicas de um processo de Business Intelligence: coleta, modelagem, transformação e carga de dados (INMON, 2005; KIMBALL; ROSS, 2013).

Os dados foram obtidos em formato CSV, a partir da base pública disponibilizada pela Agência Nacional do Petróleo (ANP), contendo informações como município, estado, tipo de combustível, bandeira e valor de venda. Após análise inicial, decidiu-se por um recorte temático voltado à localidade.

A modelagem escolhida foi do tipo estrela (star schema), amplamente utilizada por sua simplicidade e desempenho em consultas analíticas (GOLFARELLI; RIZZI, 2009). Embora o modelo inicial considerasse múltiplas dimensões, o Data Mart implementado foi simplificado, com uma única tabela de dimensão (localidade) e uma tabela fato (preço por litro e chaves de ligação).

O tratamento dos dados foi realizado em Python, na versão 3.13.3, com apoio das bibliotecas pandas e SQLAlchemy, realizando limpeza, renomeação de campos e conversão de tipos. Como aponta Monteiro (2017), a transformação adequada dos dados é fundamental para garantir integridade e utilidade analítica.

Após o tratamento, os dados foram carregados no banco MariaDB, na versão 10.4.32. As tabelas foram criadas previamente com suas chaves e relacionamentos. A carga seguiu a ordem lógica de inserção na tabela de dimensão e, depois, na tabela fato, seguida de validações básicas de integridade.

Todo o processo foi executado localmente com auxílio do XAMPP e PhpMyAdmin. As análises e visualizações finais foram realizadas com as bibliotecas matplotlib e seaborn, em ambiente Python.

5. DESENVOLVIMENTO

O desenvolvimento deste projeto seguiu uma abordagem prática, com foco na construção de um Data Mart temático voltado à análise dos preços de combustíveis por localidade. Inicialmente, foi realizada a obtenção dos dados públicos da Agência Nacional do Petróleo, disponibilizados em arquivos CSV, contendo informações como município, estado, tipo de combustível, bandeira e valor por litro.

Com base nas necessidades de análise, foi elaborado um modelo dimensional do tipo estrela mais amplo, que considera múltiplas perspectivas analíticas. Esse modelo conceitual inclui uma tabela fato central e diversas dimensões, como localidade, combustível, bandeira, posto, período e coleta, permitindo análises futuras mais completas. A Figura 1 apresenta esse modelo mais abrangente.

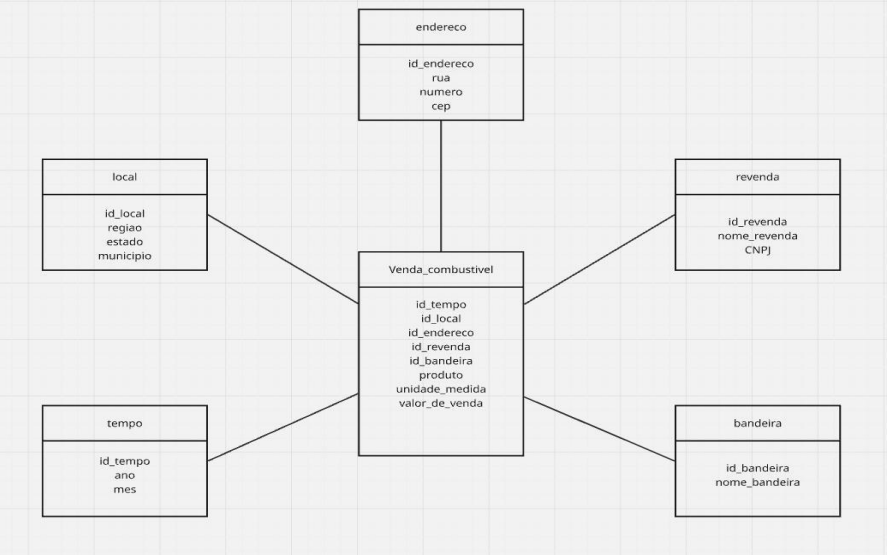

Figura 1 – Modelo estrela proposto com múltiplas dimensões para o Data Warehouse.

Entretanto, para a implementação prática neste projeto, optou-se por um recorte temático mais enxuto, focado exclusivamente na análise por localidade. Assim, foi implementado um Data Mart com apenas uma dimensão (local) e uma tabela fato com os valores de venda. Essa estrutura está ilustrada na Figura 2.

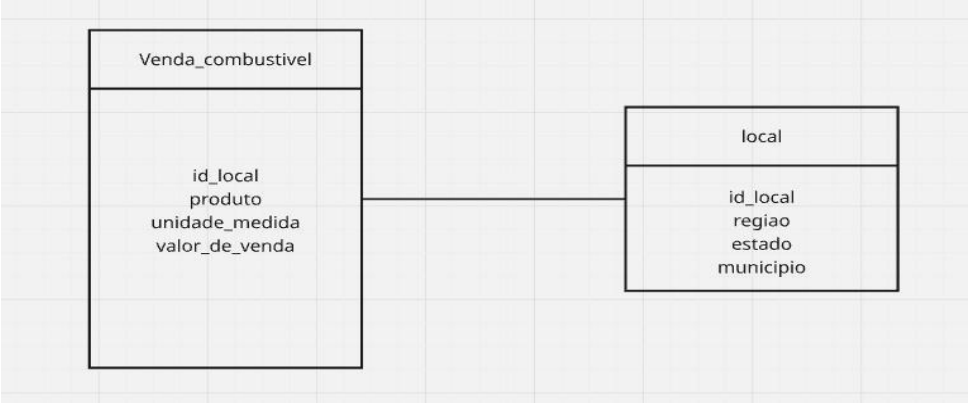

Figura 2 – Modelo estrela implementado no Data Mart temático (foco em localidade).

A seguir, os dados foram preparados para carga. O tratamento foi realizado com Python, utilizando bibliotecas como pandas. As principais ações incluíram a remoção de colunas desnecessárias, renomeação de campos e normalização de tipos. As Figuras 3 e 4 apresentam os códigos responsáveis por essa etapa.

```python
import pandas as pd

# Caminho do CSV bruto
caminho_csv_original = 'dados_combustiveis.csv'

# Caminho do CSV tratado de saída
caminho_csv_tratado = 'dados_tratados_combustiveis.csv'

# Leitura do CSV original
df = pd.read_csv("Preços semestrais - AUTOMOTIVOS_2024.02.csv", sep=";", encoding="utf-8")

# Seleção das colunas desejadas
colunas_desejadas = ['Produto', 'Unidade de Medida', 'Valor de Venda']
df = df[colunas_desejadas]

# Renomeação das colunas para coincidir com os nomes do banco
df.columns = ['produto', 'unidade_medida', 'valor_venda']

# Adiciona o ID sequencial
df.insert(0, 'id_local', range(1, len(df) + 1))

# Salva o CSV tratado
df.to_csv("preco_tratado.csv", index=False, sep=";", encoding="utf-8")
```

Figura 3 – Códigos Python utilizados para criação de CSV para tratamento de dados para tabela fato combustiveiscomercializados.

```python
import pandas as pd

# Caminho do CSV original
caminho_csv_original = 'dados_combustiveis.csv'

# Caminho do novo CSV tratado
caminho_csv_tratado = 'dados_tratados_local.csv'

# Leitura do CSV original
df = pd.read_csv("Preços semestrais - AUTOMOTIVOS_2024.02.csv", sep=";", encoding="utf-8")

# Seleção das colunas desejadas
colunas_desejadas = ['Regiao - Sigla', 'Estado - Sigla', 'Municipio']
df = df[colunas_desejadas]

# Renomeação das colunas
df.columns = ['regiao', 'estado', 'municipio']

# Adiciona o id_local sequencial (simulando AUTO_INCREMENT)
df.insert(0, 'id_local', range(1, len(df) + 1))

# Salva o novo CSV tratado com o id_local
df.to_csv("preco_tratado.csv", index=False, sep=";", encoding="utf-8")
```

Figura 4 – Códigos Python utilizados para criação de CSV para tratamento de dados para dimensão local.

Após o tratamento, as tabelas foram criadas no MariaDB com base na modelagem definida. Foram estabelecidas as chaves primárias e estrangeiras necessárias para garantir integridade referencial. A Figura 5 ilustra a criação da estrutura no banco de dados.

```sql
CREATE DATABASE dados_combustiveis;

CREATE TABLE combustiveiscomercializados (
  id_local int(11) DEFAULT NULL,
  produto varchar(100) NOT NULL,
  unidade_medida varchar(50) NOT NULL,
  valor_venda decimal(10,2) NOT NULL
);

CREATE TABLE local (
  id_local int(11) NOT NULL,
  regiao varchar(5) NOT NULL,
  estado varchar(5) NOT NULL,
  municipio varchar(100) NOT NULL
);

ALTER TABLE combustiveiscomercializados
  ADD KEY id_local (id_local);

ALTER TABLE local
  ADD PRIMARY KEY (id_local);

ALTER TABLE combustiveiscomercializados
  ADD CONSTRAINT combustiveiscomercializados_ibfk_1 FOREIGN KEY (id_local) REFERENCES local (id_local) ON DELETE CASCADE ON UPDATE CASCADE;
COMMIT;
```

Figura 5 – Criação das tabelas no MariaDB com chaves e relacionamentos.

A carga dos dados foi realizada por meio de comandos SQL, respeitando a ordem entre a inserção nas tabelas dimensionais e fato. A execução da carga está representada nas Figuras 6 e 7.

```python
import pandas as pd
from sqlalchemy import create_engine

# Caminho do CSV tratado
caminho_csv_tratado = 'preco_tratado.csv'

# Leitura do CSV com id_local já incluído
df = pd.read_csv("local_tratado.csv", sep=";", encoding="utf-8")

# Dados da conexão com o MySQL (substitua conforme o seu caso)
usuario = "root"
senha = ""
host = "localhost"
porta = 3306
banco = "dados_combustiveis"

# Criação da conexão
engine = create_engine(f'mysql+pymysql://{usuario}:{senha}@{host}:{porta}/{banco}')

# Insere os dados na tabela 'Local' (não altera id_local pois ele já está presente)
df.to_sql(name='Local', con=engine, if_exists='append', index=False)
```

Figura 6 – Código para inserção do CSV tratado para tabela local.

Figura 7 – Código para inserção do CSV tratado para tabela combustiveiscomercializados.

Após a carga, foram realizadas consultas SQL com o objetivo de validar a estrutura implementada e verificar a integridade da base. Foram feitas consultas do tipo valor médio de venda por produto e região, contagem de registros por estado e média de valores de combustíveis por estado. As Figuras de 8 a 12 apresentam a conexão com o banco, a função para consultas e as consultas realizadas diretamente sobre o Data Mart.

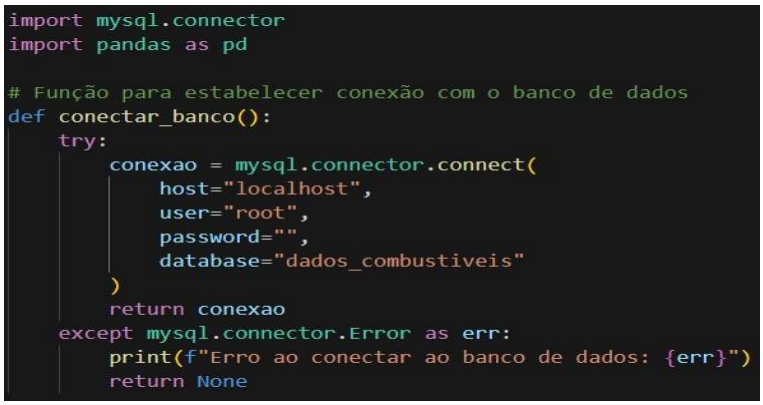

Figura 8 – Conexão com o banco.

```python
def executar_consulta(query):
    conexao = conectar_banco()
    if conexao is not None:
        try:
            cursor = conexao.cursor()
            cursor.execute(query)
            resultados = cursor.fetchall()
            colunas = [desc[0] for desc in cursor.description]
            df = pd.DataFrame(resultados, columns=colunas)
            return df
        except Exception as e:
            print(f"Erro ao executar consulta: {e}")
        finally:
            cursor.close()
            conexao.close()
    else:
        print("Não foi possível conectar ao banco.")
```

Figura 9 – Criando função para a consulta.

```python
#consulta sem tipo_localidade
# Análise 1: consulta para gerar o valor de venda médio por produto e região
consulta_dados = '''
SELECT
    cc.produto,
    l.regiao,
    ROUND(AVG(cc.valor_venda), 3) AS valor_venda_medio
FROM
    combustiveiscomercializados cc
JOIN
    local l ON cc.id_local = l.id_local
GROUP BY
    cc.produto, l.regiao
ORDER BY
    cc.produto, l.regiao;
'''
df_dados = executar_consulta(consulta_dados)
df_dados
```

Figura 10 – Consulta para gerar o valor de venda médio por produto e região.

```python
# Consulta SQL para contar registros por estado
query = """
SELECT
  l.estado AS estado,
  COUNT(*) AS quantidade_registros
FROM
  combustiveiscomercializados c
INNER JOIN
  local l ON c.id_local = l.id_local
WHERE
  c.produto = 'GASOLINA'
GROUP BY
  l.estado
ORDER BY
  quantidade_registros DESC;
"""
```

Figura 11 – Consulta para contagem de registros por estado

```python
#media valores de combustiveis por estado
query2 = """
SELECT
    l.estado,
    c.produto,
    ROUND(AVG(c.valor_venda), 3) AS media_preco
FROM
    combustiveiscomercializados c
JOIN
    local l ON c.id_local = l.id_local
WHERE
    c.produto IN ('GASOLINA', 'ETANOL', 'GASOLINA ADITIVADA', 'DIESEL S10', 'DIESEL', 'GNV')
GROUP BY
    l.estado, c.produto
ORDER BY
    c.produto, l.estado;
"""
```

Figura 12 – Consulta para gerar valor médio de combustíveis por estado

6. RESULTADOS

Após a estruturação do Data Mart e a realização das consultas SQL, os dados foram exportados para análise gráfica em Python. Os resultados revelaram padrões claros de variação nos preços de combustíveis por estado e por região.

O primeiro gráfico mostra a relação entre o volume de dados registrados e o preço médio da gasolina comum nos estados brasileiros. Observa-se que, embora estados como São Paulo e Minas Gerais concentrem o maior volume de registros, não apresentam os maiores preços médios. Já estados com menor volume de dados, como Acre e Roraima, tendem a registrar preços mais elevados, indicando possível influência de logística e distância dos centros de distribuição.

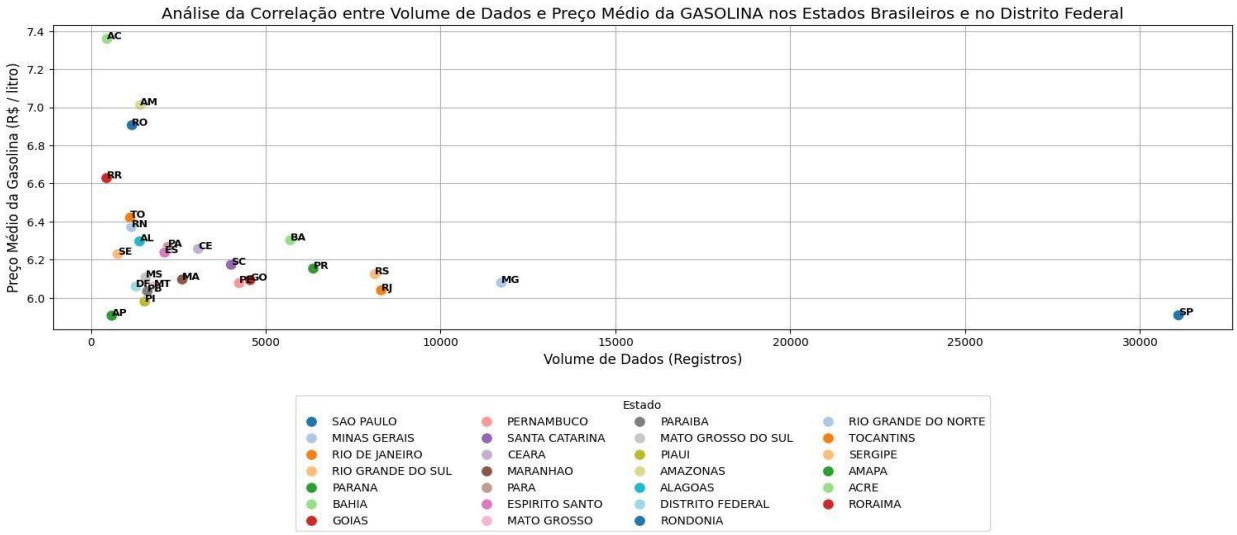

Figura 12 – Relação entre volume de dados e preço médio da gasolina por estado.

No segundo gráfico, um mapa de calor mostra os preços médios de diferentes tipos de combustíveis por região. A região Norte apresenta os maiores preços para quase todos os produtos, enquanto a região Centro-Oeste possui, em geral, os menores valores. O etanol, por exemplo, tem seu menor preço médio na região Centro-Oeste (R$ 4,02) e o maior na região Norte (R$ 4,80). Já a gasolina aditivada atinge os valores mais altos na região Norte e Sudeste, superando R$ 6,70.

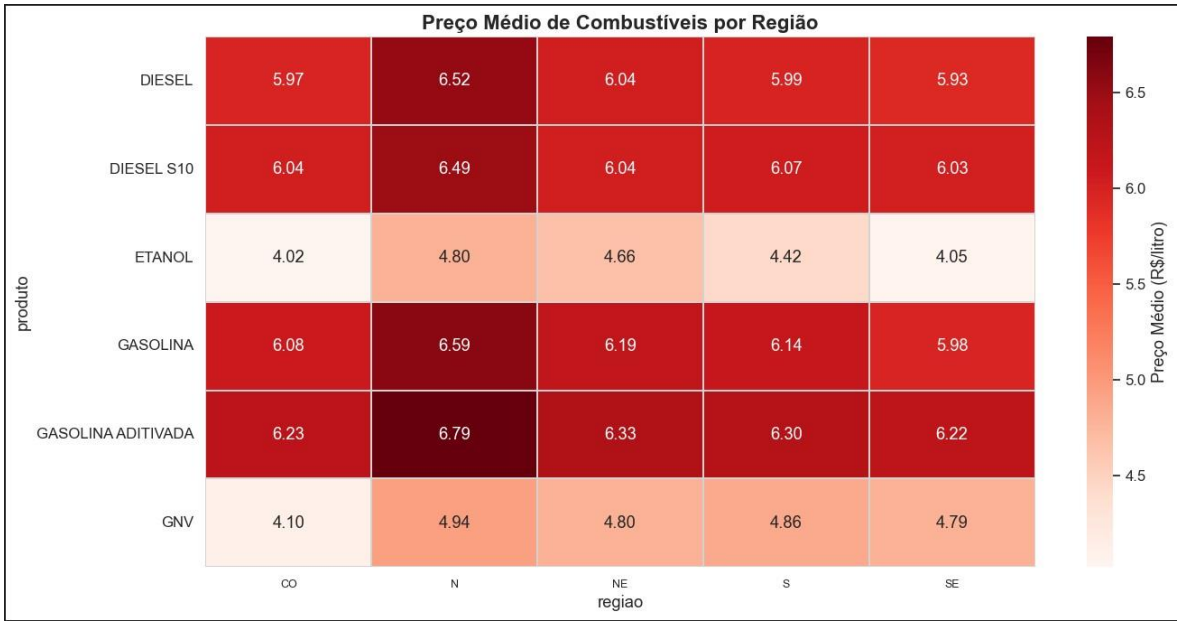

Figura 8 – Preço médio de combustíveis por região (mapa de calor).

Esses resultados demonstram a utilidade do ambiente analítico implementado, permitindo comparações claras e detalhadas entre regiões e tipos de combustível, a partir de dados públicos e atualizados.

7. CONSIDERAÇÕES FINAIS

O desenvolvimento deste projeto demonstrou como dados públicos podem ser organizados e analisados de forma eficiente com técnicas de *Business Intelligence*. A construção de um Data Mart temático voltado à análise de preços de combustíveis por localidade transformou dados brutos em um ambiente estruturado e acessível para consultas analíticas.

A metodologia adotada, com modelagem estrela, ETL e uso de Python, atendeu aos objetivos ao permitir comparações entre estados e municípios e gerar insights com dados reais. Como continuidade, recomenda-se incluir novas dimensões, como tempo e bandeira, e integrar ferramentas como o Power BI para análises interativas.

8. FONTES CONSULTADAS

AGÊNCIA NACIONAL DO PETRÓLEO, GÁS NATURAL E BIOCOMBUSTÍVEIS (ANP). Levantamento de preços de combustíveis. Disponível em: https://www.gov.br/anp/pt-br. Acesso em: 14 jun. 2025.

BRASIL. Ministério da Economia. Portal Brasileiro de Dados Abertos. Disponível em: https://dados.gov.br. Acesso em: 14 jun. 2025.

GOLFARELLI, Matteo; RIZZI, Stefano. *Data warehouse: teoria e prática de projeto.* Rio de Janeiro: Elsevier, 2009.

INMON, William H. *Building the data warehouse.* 4. ed. Indianapolis: Wiley, 2005.

KIMBALL, Ralph; ROSS, Margy. *The data warehouse toolkit: the definitive guide to dimensional modeling.* 3. ed. Indianapolis: Wiley, 2013.

MONTEIRO, Marcelo Sales. *Business Intelligence: modelos de análise e a construção de um Data Warehouse.* São Paulo: Érica, 2017.