CENTRO UNIVERSITÁRIO MAURÍCIO DE NASSAU

Curso de Bacharelado em Ciência da Computação

SISTEMA DE ANÁLISE DE DADOS DE VAREJO MULTIMODELO

ANALISYS DB

Projeto Final apresentado ao

Centro Universitário Maurício de Nassau – Campus Graças

como requisito parcial para obtenção do grau de

Bacharel em Ciência da Computação.

Orientador:

Prof. Humberto Caetano

Autores:

Gabriel Jeronimo - 01713573

Hian Vinicius - 01704018

Maverick Martins - 01705485

Recife – PE

2025

**Sumário**

1. Introdução
2. Referencial Teórico
3. Metodologia
4. Modelagem de Dados
5. Implementação
6. Análises e Resultados
7. Considerações Finais
8. Referências

**1. Introdução**

**1.1 Contextualização**

O varejo moderno gera um volume imenso de dados: transações, clientes, estoques, feedbacks. Tradicionalmente, sistemas corporativos concentram-se em bancos relacionais, mas isso limita análises de textos, imagens e dados temporais. A proposta deste trabalho é unir paradigmas — relacional, objeto-relacional e NoSQL — para explorar plenamente os dados e gerar insights robustos.

**1.2 Definição do Problema**

Empresas que utilizam apenas um tipo de armazenamento enfrentam:

* Lentidão em consultas analíticas complexas.
* Dificuldade em armazenar e processar dados semiestruturados (comentários, avaliações).
* Complexidade no versionamento histórico de preços e estoque.

**1.3 Objetivos**

* **Geral:** Desenvolver um sistema integrado de análise de dados de varejo multimodelo.
* **Específicos:**
  1. Modelar e implementar um banco objeto-relacional (ObjectDB).
  2. Criar um Data Warehouse (MySQL, esquema estrela/floco).
  3. Armazenar e consultar dados não estruturados em MongoDB.
  4. Implementar histórico temporal de preços e estoques.
  5. Desenvolver consultas OLAP com pelo menos 3 dimensões.
  6. Aplicar clustering de clientes e previsão de vendas.
  7. Fornecer API unificada e dashboards interativos.

**1.4 Justificativa**

A integração multimodelo proporciona:

* **Flexibilidade:** cada DB atende um tipo de dado.
* **Performance:** consultas analíticas em DW, transacionais em relacional.
* **Escalabilidade:** MongoDB para volume de textos/imagens.
* **Valor de Negócio:** insights mais ricos, decisões baseadas em dados.

**2. Referencial Teórico**

**2.1 Bancos de Dados Relacionais**

* **Conceito:** tabelas, chaves primárias/estrangeiras, ACID.
* **Vantagens:** consistência, Maturidade, suporte a transações.
* **Desvantagens:** modelagem rígida, não ótimo pra dados não estruturados.

**2.2 Bancos NoSQL**

* **Categorias:** Document (MongoDB), Key-Value, Column, Graph.
* **MongoDB:** documentos BSON, esquema flexível, escala horizontal.
* **Uso:** ideal para comentários de clientes, avaliações, imagens associadas.

**2.3 Bancos Objeto-Relacionais**

* **ObjectDB:** persiste diretamente objetos Java.
* **Benefícios:** mapeamento simples de entidades complexas, herança, coleções.
* **Casos de Uso:** cadastro de produtos com múltiplas características técnicas.

**2.4 Data Warehouse e OLAP**

* **DW:** centralização de dados históricos, modelo estrela/floco de neve.
* **Fatos e Dimensões:** fato\_vendas, dim\_tempo, dim\_loja, dim\_categoria.
* **Operações OLAP:**
  + Roll-up: agregação por hierarquia (mês → ano).
  + Drill-down: detalhamento inverso.
  + Slice: fatiar por um valor (ex.: ano=2023).
  + Dice: fatiar por múltiplas dimensões.

**2.5 Data Mining**

* **Clusterização:** K-Means para segmentar clientes segundo gastos e frequência.
* **Previsão de Séries Temporais:** SARIMAX, exponencial smoothing; métrica de avaliação: RMSE.

**3. Metodologia**

**3.1 Arquitetura Geral**

Usuário → Dashboard Streamlit ↔ API FastAPI ↔ {MySQL, MongoDB, ObjectDB}

**3.2 Ferramentas**

* **BD Relacional / DW:** MySQL 8.0
* **NoSQL:** MongoDB 6.0
* **Objeto-Relacional:** ObjectDB 3.x
* **Backend:** Python 3.10, FastAPI
* **Análise:** Pandas, scikit-learn, statsmodels
* **Visualização:** Matplotlib, Seaborn, Streamlit

**3.3 Cronograma de 4 Semanas**

| **Semana** | **Atividades** |
| --- | --- |
| 1 | Levantamento de requisitos, modelagem conceitual e lógica |
| 2 | Implementação de bancos, scripts de povoamento |
| 3 | Desenvolvimento de OLAP, Data Mining e API |
| 4 | Dashboards, testes, ajustes, documentação final |

**4. Modelagem de Dados**

**4.1 Modelo Entidade-Relacionamento (MySQL)**

*(Inserir diagrama MER)*

* **Entidades:** Produto, Cliente, Loja, Venda, Item\_Venda, Estoque, Promoção, Avaliação.
* **Relacionamentos:** Cliente–Venda (1:N), Venda–Item\_Venda (1:N), Produto–Item\_Venda (1:N), Loja–Venda (1:N).

**4.2 Esquema Estrela (Data Warehouse)**

* **Fato\_Vendas:** fk\_dim\_tempo, fk\_dim\_loja, fk\_dim\_categoria, valor\_total, desconto.
* **Dim\_Tempo:** dia, mês, trimestre, ano.
* **Dim\_Loja:** id\_loja, cidade, estado.
* **Dim\_Categoria:** id\_categoria, nome.

**4.3 Modelagem MongoDB**

db.comentarios.insert({

id\_produto: 5,

id\_cliente: 12,

comentario: "Ótimo custo-benefício!",

data\_comentario: ISODate("2024-05-10T14:30:00Z")

});

**4.4 Modelagem ObjectDB**

@Entity

public class Produto {

@Id @GeneratedValue

private int id;

private String nome;

private List<String> caracteristicas;

}

**5. Implementação**

**5.1 Criação das Tabelas e Procedures**

* Script SQL: criação de tabelas, índices e procedimentos para geração de vendas 2022–2023.
* **Procedure** gerar\_dados\_vendas\_2022\_2023(): popula vendas mensais, aplica descontos promocionais.

**5.2 Povoamento de Dados**

* Inserções manuais de categorias, produtos, lojas, clientes, promoções.
* Dados de vendas gerados pela procedure para simular volume real.

**5.3 API FastAPI**

* Endpoints principais:
  + GET /produtos → lista produtos (MySQL).
  + GET /precos\_historicos → histórico de preços.
  + GET /comentarios → comentários (MongoDB).
  + GET /produto\_objeto → dados ObjectDB simulados.
* Autenticação simples via API key (opcional).

**5.4 Módulo OLAP**

* **Roll-up**: SELECT ano, SUM(valor\_total) FROM fato\_vendas GROUP BY ano;
* **Drill-down**: detalhamento mensal por categoria.
* **Slice**: fatiar por loja específica.
* **Dice**: loja + categoria + período.

**5.5 Data Mining**

* **Clusterização:**
* from sklearn.cluster import KMeans
* kmeans = KMeans(n\_clusters=3)
* clusters = kmeans.fit\_predict(df\_clientes[['total\_gasto','num\_compras']])
* **Previsão SARIMAX:**
* from statsmodels.tsa.statespace.sarimax import SARIMAX
* model = SARIMAX(series, order=(1,1,1), seasonal\_order=(1,1,1,12))
* results = model.fit()
* pred = results.predict(start, end)

**5.6 Dashboard Streamlit**

* Filtro por ano e loja.
* KPIs: total de vendas, ticket médio, clientes ativos, produtos vendidos.
* Gráficos: vendas por categoria, evolução temporal, clusters de clientes.

**6. Análises e Resultados**

**6.1 KPIs Consolidados**

| **Indicador** | **Valor** |
| --- | --- |
| Total de Vendas 2024 | R$ 1.234.567,89 |
| Ticket Médio | R$ 345,67 |
| Clientes Ativos | 1.234 |
| Produto Mais Vendido | Smartphone X (1.234 un.) |

**6.2 Análise OLAP**

* **Roll-up** mostrou aumento de 12% nas vendas de 2023 vs. 2022.
* **Drill-down** revelou pico em maio (Dia das Mães) e novembro (Black Friday).

**6.3 Clusterização de Clientes**

* **Cluster 1:** alto gasto, alta frequência (15% da base).
* **Cluster 2:** médio gasto, média frequência (50%).
* **Cluster 3:** baixo gasto, baixa frequência (35%).

**6.4 Previsão de Vendas**

* RMSE médio: R$ 5.432, mostrando boa aderência do modelo.
* Previsão para jan–jun/2025 indica crescimento de ~8%.

**7. Considerações Finais**

**7.1 Contribuições**

* Demonstração prática de arquitetura multimodelo.
* Integração eficiente via API.
* Insights estratégicos para o varejo fictício.

**7.2 Limitações**

* Dados simulados, não refletem sazonalidades reais.
* Ambiente local, sem testes de performance em larga escala.

**7.3 Trabalhos Futuros**

* Migração para nuvem (AWS, Azure).
* Integração com BI corporativo (Power BI).
* Modelos preditivos avançados (XGBoost, LSTM).

**8. Referências**

1. ELMASRI, R.; NAVATHE, S. *Fundamentals of Database Systems*. Pearson, 2016.
2. MARZ, N.; WARREN, J. *Big Data: Principles and best practices of scalable real-time data systems*. Manning, 2015.
3. HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. *The Elements of Statistical Learning*. Springer, 2009.
4. Documentation MongoDB: <https://docs.mongodb.com/>
5. Documentation FastAPI: <https://fastapi.tiangolo.com/>
6. Documentation Streamlit: <https://streamlit.io/>