Documento de Arquitetura - DataDriven Store

Este documento descreve a arquitetura de persistência poliglota para a aplicação de e-commerce DataDriven Store. A arquitetura visa otimizar o desempenho, a escalabilidade e a flexibilidade, utilizando diferentes tecnologias de banco de dados para atender a requisitos específicos de cada domínio de dados. O documento detalha o diagrama conceitual da arquitetura, o modelo de dados para cada tecnologia, a responsabilidade de cada banco de dados e o fluxo de dados entre os sistemas.

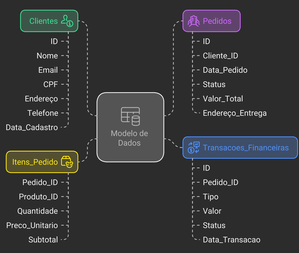
Diagrama Conceitual da Arquitetura

Modelo de Dados por Tecnologia

1. PostgreSQL - Dados Transacionais (OLTP)

Responsabilidade: Garantir consistência ACID para operações críticas do negócio.

Entidades Modeladas:

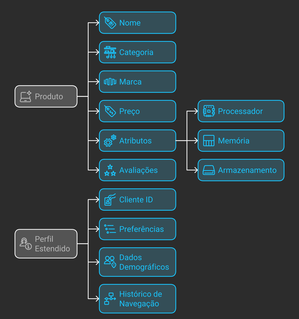


Justificativa: O modelo relacional com propriedades ACID é fundamental para transações financeiras e pedidos, onde a integridade dos dados é crítica. A estrutura rígida previne inconsistências e garante atomicidade nas operações de venda.

2. MongoDB - Dados Semi-estruturados

Responsabilidade: Flexibilidade para dados com estruturas variáveis.

Documentos Modelados:

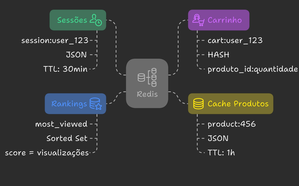


Justificativa: A flexibilidade do esquema permite evolução do catálogo sem migrações complexas. Produtos têm atributos únicos por categoria, e perfis de usuário podem crescer organicamente.

3. Redis - Cache e Dados Voláteis

Responsabilidade: Acesso de baixa latência para dados temporários.

Estruturas de Dados em Redis



Justificativa: Armazenamento em memória oferece latência sub-milissegundo essencial para experiência do usuário em tempo real. Dados voláteis não precisam de persistência permanente.

4. ClickHouse - Analytics (OLAP)

Responsabilidade: Processamento analítico de grandes volumes de eventos.

Tabelas Colunares:

```
-- Eventos de Interação
CREATE TABLE eventos (
    timestamp DateTime,
    user_id UInt64,
    evento_tipo String,
    produto_id String,
    sessao_id String,
    utm_source String,
    metadata String
) ENGINE = MergeTree()
ORDER BY (timestamp, user_id);
```

Justificativa: Arquitetura colunar otimizada para agregações e varreduras em massa. Ideal para dashboards, relatórios e análise de comportamento do usuário com alta performance em consultas analíticas.

5. Neo4j - Relacionamentos e Recomendações

Responsabilidade: Modelar e consultar relacionamentos complexos.

Modelo de Grafo:

```
// Nós
(c:Cliente {id, nome})
(p:Produto {id, nome, categoria})
(m:Marca {nome})
(cat:Categoria {nome})

// Relacionamentos
(c)-[:COMPROU {data, avaliacao}]->(p)
(c)-[:VISUALIZOU {timestamp}]->(p)
(p)-[:PERTENCE_A]->(cat)
(p)-[:DA_MARCA]->(m)
(c)-[:SIMILAR_A {score}]->(c2)
```

Justificativa: Estrutura de grafo permite consultas eficientes de relacionamentos complexos, essencial para algoritmos de filtragem colaborativa e sistemas de recomendação baseados em padrões de comportamento.

Fluxo de Dados e Integração

A arquitetura segue o padrão de responsabilidade única, onde cada banco atende necessidades específicas:

Tecnologias de Banco de Dados

Os dados fluem entre os sistemas através de eventos e sincronização eventual, mantendo cada tecnologia otimizada para seu domínio específico. Por exemplo, um novo pedido criado no PostgreSQL pode gerar um evento que atualiza o histórico de compras do cliente no MongoDB e dispara a atualização de recomendações no Neo4j. Os logs de acesso e eventos de interação são continuamente enviados para o ClickHouse para análise.