

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE  
SÃO PAULO**

**CRISTIANO DE ALMEIDA TOMAZ**

**DISCIPLINA: BANCO DE DADOS**

**PROFESSOR: PAULO GIOVANI DE FARIA ZEFERINO**

**CURSO: ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

**Projeto de Banco de Dados NoSQL**

**CAMPOS DO JORDÃO**

**2024**

## RESUMO

Este relatório apresenta um estudo detalhado sobre bancos de dados não relacionais (NoSQL), com ênfase na aplicação prática do MongoDB, uma das ferramentas mais populares desta categoria. Inicialmente, são analisadas as principais características dos bancos NoSQL, incluindo suas vantagens, limitações e os contextos em que são amplamente utilizados, como sistemas de grande escala e alta disponibilidade.

Para ilustrar a aplicação prática, escolheu-se o MongoDB como ferramenta central para o desenvolvimento de um sistema de gerenciamento de dados para uma loja virtual. A escolha baseou-se em sua capacidade de lidar com estruturas de dados flexíveis, suporte a consultas avançadas e escalabilidade horizontal.

O estudo abrange as tecnologias utilizadas, a metodologia adotada e as etapas do desenvolvimento do sistema, destacando as vantagens obtidas com o uso do MongoDB em comparação aos bancos de dados relacionais tradicionais (SQL). Entre os principais benefícios observados estão o aumento da flexibilidade no armazenamento e modelagem de dados, a escalabilidade para grandes volumes de informação e a melhoria no desempenho das consultas.

## **1 INTRODUÇÃO**

Com o crescimento exponencial do volume de dados gerados diariamente e a necessidade de processamento em tempo real, os bancos de dados tradicionais começaram a enfrentar desafios significativos, especialmente no que diz respeito à escalabilidade, flexibilidade e capacidade de lidar com dados heterogêneos. Nesse contexto, os bancos de dados não relacionais (NoSQL) surgiram como uma alternativa inovadora, oferecendo vantagens como alta escalabilidade horizontal, maior tolerância a falhas e estruturas de dados adaptáveis para atender às demandas de sistemas modernos.

Este trabalho tem como objetivo explorar as potencialidades do MongoDB, um dos bancos de dados NoSQL mais amplamente adotados no mercado, aplicando-o no desenvolvimento de um sistema prático de gerenciamento de dados. O projeto proposto consiste na criação de uma solução para uma loja virtual, simulando cenários reais que evidenciem as vantagens do MongoDB em relação aos bancos de dados relacionais tradicionais. Entre os aspectos analisados estão a flexibilidade no armazenamento de dados, o desempenho em consultas e a capacidade de escalabilidade em ambientes dinâmicos.

## **2 Metodologia**

Os bancos de dados NoSQL, também conhecidos como "não somente SQL", foram desenvolvidos para atender às demandas crescentes de aplicações modernas, que exigem maior escalabilidade horizontal e flexibilidade na estruturação de dados (STONEBRAKER, 2010). Diferentemente dos bancos relacionais, que organizam informações em tabelas rígidas e seguem o modelo ACID (Atomicidade, Consistência, Isolamento e Durabilidade), os bancos NoSQL oferecem maior adaptabilidade e priorizam características como alta disponibilidade e tolerância a falhas. Para isso, frequentemente adotam o modelo BASE (Basicamente Disponível, Estado Flexível e Consistência Eventual), que proporciona desempenho otimizado em sistemas distribuídos, mesmo que a consistência imediata seja sacrificada em prol da escalabilidade (CATTELL, 2011).

### **2.1 Modelos de Dados NoSQL**

Os bancos NoSQL se dividem em diferentes modelos, cada um projetado para atender a casos de uso específicos:

- **Chave-Valor:** Dados são armazenados como pares chave-valor, ideais para aplicações que demandam alta velocidade de leitura e escrita, como cache de dados ou configurações de sistemas.
- **Documentos:** Utilizam estruturas semelhantes ao formato JSON ou BSON, oferecendo flexibilidade para armazenar dados sem a necessidade de um esquema fixo. Este modelo é amplamente adotado em aplicações web e mobile devido à sua capacidade de representar objetos complexos de forma simples.
- **Colunas Largas:** Projetado para manipular grandes volumes de dados organizados em colunas e linhas, com foco em desempenho em análises massivas, sendo comum em aplicações de Big Data.
- **Grafos:** Especializado em mapear e gerenciar relações complexas, como conexões entre usuários em redes sociais ou rotas de transporte.

## 2.2 Exemplos de Uso

Os bancos de dados NoSQL têm ampla aplicação em diversos setores, destacando-se:

- **Redes Sociais:** Plataformas como LinkedIn e Twitter utilizam bancos de grafos para mapear conexões entre usuários, proporcionando alta eficiência na gestão de relacionamentos e interações.
- **E-commerce:** Empresas como Amazon adotam bancos orientados a documentos para gerenciar catálogos de produtos, dados de clientes e histórico de compras, garantindo flexibilidade e personalização.
- **Streaming:** Serviços como Netflix utilizam bancos NoSQL para armazenar preferências e histórico de usuários, oferecendo experiências altamente personalizadas com desempenho consistente, mesmo em grande escala (VAUGHAN-NICHOLS, 2020).

## 2.3 MongoDB e JSON

Para esta pesquisa foi escolhido o Sistema de Gerenciamento de Banco de Dados MongoDB como base para o estudo de caso. A escolha fundamenta-se em suas características notáveis, como flexibilidade estrutural, escalabilidade horizontal e suporte avançado a consultas complexas. O MongoDB utiliza um modelo orientado

a documentos, permitindo representar dados de maneira intuitiva, refletindo objetos do mundo real e suas relações de forma direta (CHODOROW, 2013).

O JSON (JavaScript Object Notation) foi adotado neste projeto devido à sua compatibilidade com o modelo orientado a documentos do MongoDB, que utiliza o BSON (Binary JSON) para armazenamento em suas coleções. Essa integração direta entre o JSON e o MongoDB simplifica a manipulação dos dados e reduz a complexidade na comunicação entre a aplicação e o banco de dados (CHODOROW, 2013). Além disso, o JSON oferece uma estrutura flexível, permitindo o armazenamento de informações hierárquicas e aninhadas. Essa característica é ideal para modelar objetos complexos do mundo real, como clientes com múltiplos endereços ou pedidos compostos por diversos itens, eliminando a necessidade de relações rígidas e esquemas predefinidos, típicos de bancos relacionais.

Outro benefício significativo do JSON é sua ampla adoção em aplicações web modernas, especialmente em APIs RESTful, o que facilita a comunicação entre servidores e clientes. Sua leitura e escrita são intuitivas, mesmo para desenvolvedores iniciantes, graças à sua semântica clara e utilização em diversas linguagens de programação. Em combinação com o MongoDB, o JSON proporciona alta performance em operações de leitura e escrita, permitindo maior flexibilidade na estruturação e manipulação dos dados. Isso o torna uma solução eficiente para sistemas que lidam com grandes volumes de dados e demandam adaptabilidade estrutural.

## **2.4 Sistema Desenvolvido**

O projeto consistiu na criação de um sistema de gerenciamento para uma loja virtual, com foco na organização e manipulação de dados em três coleções principais:

- Clientes: Dados pessoais e histórico de compras são armazenados. Campos incluem:
  - Nome, e-mail e CPF.
  - Endereço detalhado (rua, número, cidade, estado e CEP).
  - Histórico de compras, contendo detalhes dos pedidos realizados.
- Produtos: Informações sobre o estoque da loja, com os seguintes campos:
  - Produtos: Informações sobre o estoque da loja, com os seguintes campos:

- Nome, categoria, preço e descrição do produto.
  - Quantidade em estoque.
  - Fornecedor responsável pelo fornecimento do item.
- Pedidos: Dados que relacionam clientes aos produtos adquiridos. Os campos incluem:
  - Identificação do cliente.
  - Lista de produtos comprados, com quantidade e preço unitário.
  - Data do pedido, valor total e status (entregue, pendente, etc.).

## 2.5 Exemplo de Estruturas Criadas

Exemplo de Documento: Clientes

```
{
  "nome": "João Silva", // Nome do cliente
  "email": "joao@email.com", // Contato principal
  "cpf": "123.456.789-00", // Identificação fiscal
  "endereco": {
    "rua": "Av. Central",
    "numero": 123,
    "cidade": "São Paulo",
    "estado": "SP"
  },
  "historico_compras": [
    {"pedido_id": "001", "data": "2024-12-01", "valor_total": 150.00}
  ]
}
```

Exemplo de Documento: Produtos

```
{
  "nome": "Cadeira Gamer",
```

```

"categoria": "Móveis",
"preco": 850.00,
"estoque": 50,
"descricao": "Cadeira ergonômica com ajustes personalizáveis",
"fornecedor": "Móveis XYZ"
}

```

Exemplo de Documento: Pedidos

```

{
  "cliente_id": "12345678900", // CPF do cliente
  "produtos": [
    {"nome": "Cadeira Gamer", "quantidade": 2, "preco_unitario": 850.00}
  ],
  "data_pedido": "2024-12-01",
  "valor_total": 1700.00,
  "status": "Entregue"
}

```

### 3 Resultado Obtidos

O MongoDB apresentou vantagens significativas em comparação com bancos relacionais tradicionais:

1. Flexibilidade: A possibilidade de alterar o esquema sem interrupções no sistema facilitou ajustes dinâmicos, como a inclusão de novos campos.
2. Desempenho: Consultas complexas, como a agregação de dados por período, foram executadas em menos tempo devido à indexação otimizada.
3. Escalabilidade: A capacidade de adicionar nós para aumentar o armazenamento garantiu suporte a grandes volumes de dados sem impacto no desempenho.

Ao utilizar SQL, a modelagem do sistema teria exigido várias tabelas com relações complexas, aumentando a dificuldade de manutenção. No MongoDB, as relações foram embutidas diretamente nos documentos, reduzindo a necessidade de junções complexas e otimizando a performance (CATTELL, 2011).

## CONCLUSÃO

O MongoDB mostrou-se uma solução altamente robusta e eficaz para o gerenciamento de dados não estruturados, atendendo de forma exemplar às demandas de aplicações modernas. Durante o desenvolvimento do sistema de gerenciamento de dados para a loja virtual, destacou-se a capacidade do MongoDB de oferecer flexibilidade na modelagem de dados, eliminando a necessidade de esquemas rígidos e facilitando ajustes estruturais ao longo do processo. Além disso, o desempenho em consultas complexas e a escalabilidade horizontal foram evidentes, especialmente em cenários que simulavam grandes volumes de dados e requisições simultâneas.

Os resultados obtidos confirmaram que o uso do MongoDB agrega vantagens significativas em relação aos bancos de dados relacionais tradicionais, particularmente em sistemas que exigem alta disponibilidade e estruturação dinâmica. No entanto, algumas melhorias podem ser implementadas para otimizar ainda mais o projeto. A integração de ferramentas de análise de dados em tempo real, como o MongoDB Atlas ou frameworks de Big Data, pode ampliar a capacidade de extração de insights a partir dos dados. Adicionalmente, a implementação de réplicas para maior tolerância a falhas e a utilização de mecanismos avançados de indexação podem aprimorar a disponibilidade e o desempenho do sistema.

Por fim, a integração com serviços de computação em nuvem permitiria maior escalabilidade e flexibilidade operacional, tornando o sistema mais resiliente e preparado para demandas futuras. O MongoDB, com suas características modernas e adaptáveis, demonstra ser uma escolha estratégica para aplicações de grande escala, mas o sucesso contínuo dependerá de atualizações e ajustes constantes alinhados às novas necessidades tecnológicas.



## REFERÊNCIAS BIBLIOGRÁFICAS

CATTELL, R. Scalable SQL and NoSQL data stores. ACM SIGMOD Record, 2011. Disponível em: <https://doi.org/10.1145/1978915.1978919>. Acesso em: 04 dez. 2024.

CHODOROW, Kristina. MongoDB: The Definitive Guide. O'Reilly Media, 2013.  
MONGODB INC. MongoDB Documentation. Disponível em: <https://www.mongodb.com/docs/>. Acesso em: 04 dez. 2024.

STONEBRAKER, Michael. SQL databases v. NoSQL databases. Communications of the ACM, 2010. Disponível em: <https://doi.org/10.1145/1866739.1866757>. Acesso em: 04 dez. 2024.

VAUGHAN-NICHOLS, Steven J. Understanding NoSQL databases. ZDNet, 2020. Disponível em: <https://www.zdnet.com/article/understanding-nosql-databases/>. Acesso em: 04 dez. 2024.