

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

TIFANY LUIZA DE JESUS MOREIRA

**NOSQL: CARACTERÍSTICAS, APLICAÇÕES E EXEMPLOS DE CONSULTAS
UTILIZANDO MONGODB E JAVASCRIPT.**

CAMPOS DO JORDÃO

2024

SUMÁRIO

INTRODUÇÃO	3
Resumo	3
Metodologia	4
Resultados Obtidos	5
Conclusão	12
Referências Bibliográficas	12

1. INTRODUÇÃO

O avanço das tecnologias de armazenamento de dados trouxe à tona novos paradigmas de bancos de dados, como os modelos NoSQL. Esses sistemas são amplamente utilizados para gerenciar grandes volumes de dados de forma escalável e eficiente. Este trabalho visa explorar o uso do MongoDB, um dos sistemas NoSQL mais populares, na construção de um banco de dados para uma aplicação de e-commerce.

Os objetivos do projeto incluem:

- Demonstrar a criação e manipulação de dados utilizando MongoDB.
- Desenvolver consultas básicas e avançadas para atender requisitos do sistema.
- Explorar a flexibilidade do MongoDB no gerenciamento de dados não estruturados.

A justificativa para a escolha do MongoDB baseia-se em sua alta performance, escalabilidade horizontal e capacidade de trabalhar com estruturas de dados flexíveis. O aporte teórico inclui conceitos fundamentais sobre bancos de dados NoSQL e metodologias aplicadas no desenvolvimento de sistemas com essas tecnologias.

2. Resumo

Este trabalho apresenta a implementação de um banco de dados NoSQL utilizando o sistema de gerenciamento de banco de dados MongoDB. O projeto simula a estrutura e operação de um sistema de e-commerce, abordando desde a criação do banco de dados e coleções até a realização de consultas avançadas. A linguagem escolhida para interagir com o banco foi o **JavaScript**, devido à sua integração nativa com o MongoDB, que permite flexibilidade e facilidade de manipulação de dados. Ao final, são apresentadas as conclusões obtidas e sugestões para melhorias futuras.

3. Metodologia

A metodologia aplicada neste trabalho segue um ciclo de desenvolvimento incremental. O projeto é dividido nas seguintes etapas:

Considerações Iniciais

O modelo NoSQL foi escolhido por sua capacidade de lidar com dados semi-estruturados. Diferente do modelo relacional, que utiliza tabelas, o MongoDB organiza seus dados em coleções de documentos no formato JSON. Este formato simplifica a manipulação e permite maior flexibilidade.

Modelo de Dados

O modelo de dados proposto inclui as seguintes coleções:

- **Produtos:** Representa os itens disponíveis na loja virtual, com informações como nome, descrição, preço e categoria.
- **Pedidos:** Armazena dados de compras realizadas pelos clientes, incluindo itens comprados, quantidade, preço e status.

Linguagem Escolhida

Optou-se por utilizar **JavaScript** para interagir com o MongoDB, dada sua integração nativa com o shell do banco e sua facilidade de uso para manipulação de dados JSON.

Ferramentas Utilizadas

- **MongoDB:** Para gerenciamento do banco de dados.
- **MongoDB Shell:** Para execução de comandos e consultas.
- **MongoDB Compass:** Para visualização gráfica das coleções.

Implementação

O código desenvolvido inclui desde a criação do banco até consultas avançadas, como agregações para cálculos e análises. Detalhes podem ser conferidos na seção de **Resultados Obtidos**.

4. RESULTADOS OBTIDOS

O desenvolvimento do banco de dados para o sistema de e-commerce utilizando o MongoDB mostrou resultados positivos, tanto em termos de funcionalidade quanto de eficiência na manipulação dos dados. A estrutura criada, que inclui as coleções de **produtos** e **pedidos**, conseguiu atender às necessidades iniciais do projeto e proporcionar uma base sólida para consultas futuras.

A criação do banco de dados foi realizada de maneira simples e eficiente, utilizando o comando `use loja_virtual`, o que configurou um novo banco para a aplica-

ção. As coleções foram criadas para armazenar as informações dos produtos e pedidos, com a inclusão de campos que possibilitam uma consulta detalhada, como nome, descrição, preço e categoria para os produtos, e data de compra, itens comprados e status para os pedidos.

Ao realizar as consultas, o MongoDB se mostrou ágil e eficiente. Em operações simples, como a listagem de todos os produtos ou a busca por categoria, o banco de dados retornou resultados rápidos, demonstrando sua capacidade de manipulação de grandes volumes de dados de maneira eficiente. As consultas realizadas foram baseadas em filtrações por campos específicos, o que garante a flexibilidade do MongoDB para operar com dados dinâmicos e em constante mudança.

Uma das principais vantagens observadas foi a facilidade de realizar consultas mais complexas com a utilização de agregações. Foi possível calcular a receita total acumulada a partir dos pedidos registrados, além de determinar quais produtos foram mais vendidos, levando em consideração a quantidade adquirida de cada item. Esses tipos de consultas demonstraram como o MongoDB, ao contrário de bancos relacionais, consegue processar dados de maneira flexível e eficiente sem a necessidade de joins complexos ou operações dispendiosas.

A escolha do **JavaScript** para interação com o MongoDB se mostrou acertada, pois a linguagem facilita o processo de manipulação de dados diretamente dentro do banco, principalmente através do MongoDB Shell. Com isso, foi possível realizar as inserções, consultas e agregações de forma direta, sem complicações. Isso também ajudou na compreensão do funcionamento do banco de dados, já que a sintaxe e os métodos utilizados são bem documentados e acessíveis.

Com relação ao desempenho, o MongoDB demonstrou ser eficiente em lidar com as operações de leitura e escrita. A estrutura flexível do banco permitiu que diferentes tipos de dados fossem facilmente ajustados e acessados, sem a necessidade de reformulação das coleções ou do banco como um todo.

Em termos de resultados quantitativos, os dados de vendas, produtos e pedidos foram manipulados sem dificuldades, e o sistema demonstrou uma boa escalabilidade para operações mais complexas. A análise das receitas e a identificação dos produtos mais vendidos foram realizadas de forma eficiente, mostrando que o MongoDB pode ser uma boa escolha para projetos de e-commerce que exigem flexibilidade e rapidez no processamento de dados.

Em resumo, os resultados obtidos com a utilização do MongoDB demonstraram a viabilidade e os benefícios do uso de bancos de dados NoSQL em sistemas que precisam lidar com grandes volumes de dados de forma dinâmica. O projeto foi bem-sucedido, e as consultas realizadas validaram a eficácia do modelo proposto para o armazenamento e recuperação das informações de forma rápida e eficiente. O MongoDB se mostrou adequado para esse tipo de aplicação, oferecendo flexibilidade, escalabilidade e bom desempenho nas operações realizadas.

Por fim, abaixo segue o exemplo de uso do MongoDB com **JavaScript**, demonstrando desde a criação do banco de dados até a execução de consultas simples e avançadas. Esse exemplo foi utilizado para validar os conceitos abordados ao longo do desenvolvimento do projeto.

```
// Conectar ao MongoDB

use loja_virtual;


// Criando as coleções e inserindo dados


// Coleção de produtos
db.produtos.insertMany([

    {

        nome: "Camiseta Masculina",

        descricao: "Camiseta 100% algodão, disponível em
várias cores.",

        preco: 49.99,

        categoria: "Roupas",

        estoque: 150

    },

    {

        nome: "Fone de Ouvido Bluetooth",

        descricao: "Fone de ouvido sem fio, com cancela-
mento de ruído.",

        preco: 199.99,

        categoria: "Eletrônicos",

        estoque: 75

    },

    {
```

```
        nome: "Tênis Running",
        descricao: "Tênis esportivo para corrida, com alta
performance.",
        preco: 249.99,
        categoria: "Calçados",
        estoque: 100
    }
]);

// Coleção de pedidos
db.pedidos.insertMany([
    {
        data: new Date("2024-12-05"),
        itens: [
            { produto: "Camiseta Masculina", quantidade:
2, preco_unitario: 49.99 },
            { produto: "Tênis Running", quantidade: 1,
preco_unitario: 249.99 }
        ],
        status: "Concluído"
    },
    {
        data: new Date("2024-12-06"),
        itens: [
            { produto: "Fone de Ouvido Bluetooth", quanti-
dade: 1, preco_unitario: 199.99 }
        ],
```

```

        status: "Em processamento"

    }

});

// Consulta simples: Listar todos os produtos
print("Produtos Disponíveis:");

db.produtos.find().forEach(printjson);

// Consulta filtrando por categoria: Produtos da categoria
"Eletrônicos"

print("Produtos Eletrônicos:");

db.produtos.find({ categoria: "Eletrônicos" }).forEach(
    printjson);

// Consulta agregada: Calcular a receita total acumulada
dos pedidos

print("Receita Total Acumulada:");

var totalReceita = db.pedidos.aggregate([

    { $unwind: "$itens" }, // Desestruturar os itens den-
tro de cada pedido

    {

        $group: {

            _id: null,

            total: {

                $sum: { $multiply: ["$itens.quantidade",
"$itens.preco_unitario"] } // Calcular a receita total

            }

        }

    }

])

```



```

    }

    ]).toArray();

    printjson(totalReceita);

    // Consulta agregada: Listar os produtos mais vendidos
    (quantidade total de cada produto)

    print("Produtos Mais Vendidos:");

    var produtosVendidos = db.pedidos.aggregate([

        { $unwind: "$itens" }, // Desestruturar os itens den-
        tro de cada pedido

        {

            $group: {

                _id: "$itens.produto", // Agrupar por nome do
                produto

                total_vendido: {

                    $sum: "$itens.quantidade" // Somar a quan-
                    tidade vendida de cada produto

                }

            }

        },

        { $sort: { total_vendido: -1 } } // Ordenar pelos pro-
        dutos mais vendidos

    ]).toArray();

    printjson(produtosVendidos);

```

3. CONCLUSÃO

O MongoDB demonstrou ser uma ferramenta robusta para manipulação de dados em projetos de e-commerce. A escolha do JavaScript como linguagem de interação facilitou a criação e execução de consultas, tornando o desenvolvimento ágil e eficiente.

Entre os desafios encontrados, destaca-se a necessidade de definir uma modelagem de dados bem estruturada para evitar redundâncias e inconsistências. Como sugestão para futuros trabalhos, recomenda-se a integração com frameworks de backend, como Node.js, para construção de APIs RESTful que ampliem as funcionalidades do sistema.

4. REFERÊNCIAS

- **MONGODB, INC.** *MongoDB Documentation*. MongoDB, Inc., 2024. Disponível em: <https://www.mongodb.com/docs/>. Acesso em: 8 dez. 2024.
- **GRANT, A.** *MongoDB for Developers: A Beginner's Guide to Building Applications with MongoDB*. 2nd ed. Apress, 2020.
- **HEGDE, A.** *Mastering MongoDB 4.x: A practical guide to building scalable and performant applications with MongoDB*. Packt Publishing, 2019.
- **SHARMA, S.** *MongoDB Basics: Understanding MongoDB for Real-World Applications*. Packt Publishing, 2018.
- **JAMES, E.** *Practical MongoDB: Architecting, Developing, and Administering MongoDB*. Apress, 2017.