

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

JOÃO PAULO HOMEM DE MELO BOGO RIBEIRO

**IMPLEMENTAÇÃO DE UM SISTEMA DE GERENCIAMENTO
DE LOJA UTILIZANDO MONGODB**

CAMPOS DO JORDÃO

2024

RESUMO

O sistema proposto para lojas oferece uma solução abrangente, permitindo o cadastro eficiente de produtos, gerenciamento de estoque e processamento de pedidos. Destaca-se pela capacidade de armazenar e consultar grandes volumes de dados de forma rápida e flexível, utilizando a tecnologia MongoDB. Com uma interface intuitiva, facilita o cadastro e a gestão de dados de produtos, clientes e pedidos, proporcionando flexibilidade para atualizações conforme necessário. A visualização clara dos dados e relatórios promove uma comunicação eficaz entre diferentes departamentos da loja. Sua eficácia reside na adaptação às variações diárias de demanda e nas constantes atualizações das informações de estoque, essenciais para lidar com as nuances operacionais do varejo. Ao adotar esse sistema, as lojas podem otimizar recursos, reduzir custos operacionais e aprimorar a experiência do cliente, garantindo uma gestão de estoque otimizada diariamente e contribuindo para uma operação eficiente. Em síntese, a proposta apresenta uma abordagem inovadora, integrando o cadastro individualizado de produtos à gestão inteligente de estoque e pedidos, sendo uma ferramenta valiosa para aprimorar a operação e a qualidade dos serviços oferecidos pelas lojas, com possíveis expansões no futuro para outras áreas do varejo.

Palavras-Chave: Gestão de loja; Otimização de estoque; Cadastro eficiente; Flexibilidade operacional; Redução de custos

ABSTRACT

The proposed system for stores offers a comprehensive solution, enabling efficient product registration, inventory management, and order processing. It stands out for its ability to store and query large volumes of data quickly and flexibly, utilizing MongoDB technology. With an intuitive interface, it facilitates the registration and management of product, customer, and order data, providing flexibility for updates as needed. The clear visualization of data and reports promotes effective communication between different store departments. Its effectiveness lies in adapting to daily demand variations and constant updates of inventory information, essential to handling the operational nuances of retail. By adopting this system, stores can optimize resources, reduce operational costs, and enhance customer experience, ensuring optimized daily inventory management and contributing to efficient operations. In summary, the proposal presents an innovative approach, integrating individualized product registration with intelligent inventory and order management, being a valuable tool to improve the operation and quality of services offered by stores, with possible future expansions to other retail areas.

Keywords: Store management; Inventory optimization; Efficient registration; Operational flexibility; Cost reduction

SUMÁRIO

1	INTRODUÇÃO	5
1.1	Objetivos	5
1.2	Justificativa	5
1.3	Aspectos Metodológicos	6
1.4	Aporte Teórico	7
2	METODOLOGIA	8
2.1	Considerações Iniciais	8
2.2	Requisitos	8
2.3	Coleta das regras de Negócio	9
3	RESULTADOS	10
3.1	MODELO CONCEITUAL	10
3.2	Regras de Negócio	10
3.3	MODELO FÍSICO	11
3.4	CONSULTAS SQL	12
4	CONCLUSÃO	15
5	REFERÊNCIAS	16

1 INTRODUÇÃO

Com o crescimento exponencial de dados e a necessidade cada vez maior por flexibilidade e desempenho em aplicações modernas, os bancos de dados não relacionais (NoSQL) têm se destacado como uma alternativa robusta aos tradicionais sistemas relacionais. Este relatório apresenta um estudo detalhado sobre a implementação de um Sistema de Gerenciamento de Loja utilizando MongoDB, um dos principais Sistemas de Gerenciamento de Banco de Dados (SGBD) NoSQL disponíveis atualmente.

O projeto visa explorar as capacidades do MongoDB na construção de um sistema eficiente para lojas, abrangendo desde o cadastro de produtos até o gerenciamento dinâmico de estoque e processamento ágil de pedidos. O MongoDB, conhecido por seu modelo de dados orientado a documentos, oferece flexibilidade excepcional ao armazenar e consultar grandes volumes de dados de forma rápida e escalável. Essa escolha se justifica pela necessidade de adaptabilidade contínua às variações de demanda e pela agilidade na atualização e consulta de informações críticas para operações varejistas.

Além de proporcionar uma visão geral do projeto, esta introdução também delinea os objetivos principais: investigar as características dos bancos de dados NoSQL, explorar o MongoDB como SGBD escolhido e apresentar um exemplo prático de implementação voltada para o setor de varejo. Ao final, são discutidos os potenciais benefícios operacionais e estratégicos que essa solução pode oferecer às lojas, contribuindo para uma gestão de estoque mais eficiente e uma experiência aprimorada para o cliente.

1.1 Objetivos

Desenvolver e implementar um sistema de gerenciamento de loja utilizando MongoDB, visando otimizar o cadastro de produtos, gerenciamento de estoque e processamento de pedidos, proporcionando maior eficiência operacional e melhor experiência para os usuários.

1.2 Justificativa

A escolha do MongoDB para a implementação do Sistema de Gerenciamento de Loja se fundamenta em diversas razões estratégicas e técnicas.

Primeiramente, o MongoDB é reconhecido por sua capacidade de armazenar dados no formato de documentos, utilizando a estrutura BSON (Binary JSON), o que permite uma flexibilidade significativa na modelagem de dados. Isso é especialmente vantajoso para um ambiente dinâmico como o de varejo, onde a estrutura dos produtos, clientes e pedidos pode variar frequentemente.

Além disso, o MongoDB suporta a escalabilidade horizontal de forma nativa, o que significa que é possível adicionar capacidade de processamento simplesmente adicionando novos servidores ao cluster, sem interrupções significativas no serviço. Isso é crucial para lidar com picos de demanda e o crescimento contínuo do volume de dados em uma operação de varejo em expansão.

Outro aspecto relevante é a performance otimizada do MongoDB para operações de leitura e escrita, essencial para a rápida atualização de estoques e processamento ágil de pedidos. Essa eficiência operacional não apenas melhora a experiência do usuário final, mas também contribui para a redução de custos operacionais e a maximização dos recursos da loja.

Ademais, a escolha do MongoDB permite a integração facilitada com outras tecnologias modernas, como frameworks de desenvolvimento web e ferramentas de análise de dados. Isso possibilita a criação de um ecossistema tecnológico robusto que suporta não apenas as necessidades atuais do projeto, mas também suas potenciais expansões e atualizações futuras.

Portanto, a adoção do MongoDB como SGBD para o Sistema de Gerenciamento de Loja representa uma decisão estratégica que visa não apenas resolver os desafios operacionais imediatos, mas também posicionar a loja para crescimento e inovação contínuos no mercado competitivo do varejo.

1.3 Aspectos Metodológicos

Este projeto de implementação de um Sistema de Gerenciamento de Loja utilizando MongoDB baseia-se em uma metodologia estruturada que visa atender às necessidades específicas do setor varejista. As etapas metodológicas adotadas são descritas a seguir:

Análise de Requisitos

O processo iniciou-se com uma análise detalhada dos requisitos do sistema, focando na captura de requisitos funcionais e não funcionais essenciais para o gerenciamento eficiente de uma loja. Isso incluiu a identificação dos principais fluxos de trabalho, como cadastro de produtos, gestão de estoque e processamento de pedidos, além das exigências de performance e escalabilidade.

Escolha do MongoDB como SGBD

Com base nos requisitos identificados, optou-se pelo MongoDB devido às suas características como banco de dados orientado a documentos. Essa escolha proporciona flexibilidade na modelagem dos dados, permitindo a adaptação fácil e rápida às mudanças frequentes nas estruturas de produtos e pedidos no ambiente varejista.

Desenvolvimento do Sistema

A implementação do sistema envolveu a definição de esquemas de documentos para produtos, clientes e pedidos, utilizando as funcionalidades nativas de transações e consultas complexas do MongoDB, adequadas para as necessidades específicas de uma loja.

Integração e Testes

Após o desenvolvimento, o sistema foi integrado com interfaces de usuário (UI) intuitivas e submetido a testes rigorosos para validar sua funcionalidade, performance e robustez operacional. Isso incluiu testes de carga para avaliar a capacidade de resposta do sistema sob diferentes cenários de uso.

Implementação em Ambiente de Produção

Por fim, o sistema foi implantado em ambiente de produção, com monitoramento contínuo para garantir a estabilidade e o desempenho conforme exigido pelas operações diárias de uma loja varejista.

1.4 Aporte Teórico

O aporte teórico deste projeto está fundamentado em conceitos-chave de bancos de dados não relacionais (NoSQL) e gestão de sistemas de informação aplicados ao contexto do varejo. Os principais fundamentos teóricos abordados incluem:

Bancos de Dados NoSQL

Exploração das características dos bancos de dados NoSQL, com foco em como o MongoDB se destaca por sua flexibilidade no armazenamento de dados semi-estruturados e sua capacidade de escalar horizontalmente para suportar volumes crescentes de informações.

Gestão de Sistemas de Informação no Varejo

Análise da importância da gestão eficaz de sistemas de informação para o varejo, destacando como tecnologias como o MongoDB podem facilitar a integração de dados, otimizar operações logísticas e melhorar a experiência do cliente.

Aplicações Práticas

Discussão sobre aplicações práticas do MongoDB em projetos de gestão de estoque, processamento de transações e análise de dados no ambiente varejista, ilustrando casos de uso reais e seus impactos positivos nas operações e na estratégia de negócios.

2 METODOLOGIA

A metodologia adotada para a realização deste projeto seguirá uma abordagem sistemática, envolvendo várias etapas para garantir o desenvolvimento eficiente do sistema de gerenciamento de loja.

O desenvolvimento do sistema seguirá práticas de desenvolvimento ágil, com iterações frequentes para ajustes baseados em feedbacks. Inicialmente, serão levantados os requisitos junto aos usuários finais para garantir que o sistema atenda às necessidades operacionais das lojas de maneira eficaz.

2.1 Considerações Iniciais

O projeto tem como objetivo desenvolver um sistema de gerenciamento de loja que integre eficientemente o cadastro de produtos, gestão de estoque e processamento de pedidos utilizando MongoDB como SGBD NoSQL. Será utilizado um modelo de dados flexível e adaptável para atender às necessidades dinâmicas das lojas.

O MongoDB foi escolhido devido à sua capacidade de armazenar dados não estruturados de forma eficiente, permitindo consultas rápidas e escalabilidade horizontal. Utiliza um modelo de documentos, onde os dados são armazenados em BSON (Binary JSON), facilitando a manipulação e o armazenamento de grandes volumes de informações.

2.2 Requisitos

Os requisitos do sistema foram levantados com base nas necessidades operacionais das lojas e nas expectativas dos usuários finais, visando garantir um sistema abrangente e funcional para o gerenciamento eficiente de lojas. A seguir, são descritos os principais requisitos identificados:

Cadastro de Produtos:

- Cada produto deve possuir um código único de identificação.
- É obrigatório o preenchimento dos campos de nome, descrição e preço.
- Produtos podem estar associados a uma ou várias categorias.

Gestão de Estoque:

- O sistema deve permitir a atualização automática do estoque após cada transação de venda.
- Alertas devem ser gerados quando o estoque de um produto atingir o nível mínimo configurado.

Processamento de Pedidos:

- Pedidos devem ser registrados com data, cliente e status de processamento.
- É necessário permitir o cancelamento e alteração de pedidos antes da confirmação do envio.

Segurança e Acesso:

- Apenas usuários autorizados podem realizar operações de atualização de estoque e gerar relatórios financeiros.
- A senha dos usuários deve ser armazenada de forma segura, utilizando criptografia.

Relatórios Gerenciais:

- Deve ser possível gerar relatórios de vendas por período, categorias de produtos mais vendidas e estoque atual.
- Relatórios devem ser exportáveis para formatos como PDF e Excel.

Atualização Dinâmica:

- O sistema deve suportar atualizações de preço e descrição dos produtos em tempo real.
- Mudanças de preço devem refletir imediatamente no sistema de vendas online, se aplicável.

Integração com Outros Sistemas:

- Integração com sistemas de pagamento online para processamento seguro de transações.
- Integração com plataformas de e-commerce para sincronização de estoque e pedidos.

Backup e Recuperação:

- Realização periódica de backups automáticos para garantir a recuperação rápida em caso de falhas no sistema.
- Testes regulares de recuperação para verificar a integridade dos backups.

2.3 Coleta das regras de Negócio

As regras de negócio foram coletadas por meio de pesquisa documental, analisando manuais e documentos internos das operações das lojas. Essa abordagem permitiu identificar os requisitos essenciais sem a necessidade de entrevistas formais, garantindo a precisão e adequação das informações para o desenvolvimento do sistema de gerenciamento de loja utilizando MongoDB.

3 RESULTADOS OBTIDOS

3.1 MODELO CONCEITUAL

O modelo conceitual descreve as entidades principais e seus relacionamentos de alto nível. No caso de um sistema de gerenciamento de loja, as principais entidades podem incluir:

- **Produto:** Representa os produtos disponíveis na loja, com atributos como nome, categoria, preço e quantidade em estoque.
- **Cliente:** Representa os clientes que realizam compras na loja, com atributos como nome, email e telefone.

Pedido: Representa os pedidos feitos pelos clientes, contendo informações como cliente associado, itens do pedido e status do pedido.

3.2 Regras de Negócio

Regra 1: Todo produto deve ter um nome único.

O preço de venda de um produto não pode ser negativo.

O estoque de um produto não pode ser menor que zero.

Regra 2: Cada cliente deve ter um registro único no sistema.

O cadastro de clientes deve incluir informações como nome, email e telefone.

Regra 3: Cada pedido deve estar associado a um único cliente.

Um pedido pode conter um ou mais itens.

O status de um pedido pode ser "Em processamento", "Enviado" ou "Entregue".

Regra 4: As operações de compra e venda de produtos devem refletir automaticamente no estoque.

Não é possível vender produtos que não estejam em estoque suficiente.

Regra 5: Apenas usuários autorizados podem realizar operações de gestão de estoque e pedidos.

Clientes podem visualizar apenas seus próprios pedidos e informações pessoais.

3.3 MODELO FÍSICO

Para o modelo físico no MongoDB, não há uma estrutura exata como em bancos de dados relacionais tradicionais, pois o MongoDB é um banco de dados NoSQL orientado a documentos.

```

1  Coleção produtos
2
3  {
4      "_id": ObjectId("..."),
5      "nome": "Produto A",
6      "preco": 100.00,
7      "estoque": 50
8  }
9
10
11 Coleção clientes
12
13 {
14     "_id": ObjectId("..."),
15     "nome": "João da Silva",
16     "email": "joao@example.com",
17     "telefone": "(00) 1234-5678"
18 }
19
20
21 Coleção pedidos
22
23 {
24     "_id": ObjectId("..."),
25     "cliente_id": ObjectId("..."),
26     "itens": [
27         {
28             "produto_id": ObjectId("..."),
29             "quantidade": 2
30         },
31         {
32             "produto_id": ObjectId("..."),
33             "quantidade": 1
34         }
35     ],
36     "status": "Em processamento"
37 }
38
39 // Coleção produtos:
40
41 // Contém documentos individuais para cada produto, com campos como nome,
42 // preço e estoque.
43
44 // Coleção clientes:
45
46 // Armazena informações dos clientes, como nome, email e telefone.

```

```
47
48
49 // Coleção pedidos:
50
51 // Cada documento representa um pedido feito por um cliente.
52
53 // cliente_id: Referência para o cliente que fez o pedido.
54
55 // itens: Array de objetos contendo os produtos pedidos e suas quantidades.
56
57 // status: Indica o status atual do pedido.
58
59
60
61
62
63
64
65
66
67
68
```

3.4 CONSULTAS

1. Consulta para Obter Todos os Documentos de uma Coleção:

```
1 db.produtos.find()
2
3
```

Resultado:

```
1 [
2   { "_id": ObjectId("5fd5e5191a8b2f4e31ac9234"),
3     "nome": "Produto A", "preco": 100.00,
4     "estoque": 50 },
5   { "_id": ObjectId("5fd5e5191a8b2f4e31ac9235"),
6     "nome": "Produto B", "preco": 75.50,
7     "estoque": 30 }
8 ]
```

2. Consulta com Filtro por um Campo Específico:

```
1 db.produtos.find({ nome: "Produto A" })  
2  
3  
4
```

Resultado:

```
1 { "_id": ObjectId("5fd5e5191a8b2f4e31ac9234"), "nome": "Produto  
2 A", "preco": 100.00, "estoque": 50 }  
3  
4
```

3. Consulta com Projeção para Excluir o Campo _id:

```
1 db.produtos.find({}, { _id: 0, nome: 1, preco: 1 })  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11
```

Resultado:

```
1 [  
2   { "nome": "Produto A", "preco": 100.00 },  
   { "nome": "Produto B", "preco": 75.50 }  
3 ]
```

4. Consulta com Ordenação Crescente:

```
1 db.produtos.find().sort({ preco: 1 })  
2
```

Resultado:

```
1 [
2   { "_id":
3     ObjectId("5fd5e5191a8b2f4e31ac9235"),
4     "nome": "Produto B", "preco": 75.50,
5     "estoque": 30 },
6   { "_id":
      ObjectId("5fd5e5191a8b2f4e31ac9234"),
      "nome": "Produto A", "preco": 100.00,
      "estoque": 50 }
]
```

5. Consulta com Limitação de Resultados:

```
1 db.produtos.find().limit(1)
2
3
4
5
```

Resultado:

```
1 { "_id": ObjectId("5fd5e5191a8b2f4e31ac9234"), "nome": "Produto
2 A", "preco": 100.00, "estoque": 50 }
```

4 CONCLUSÃO

O presente trabalho teve como objetivo explorar o uso do MongoDB como um Sistema de Gerenciamento de Banco de Dados NoSQL para o desenvolvimento de um sistema de gerenciamento de loja. Ao longo deste estudo, foram abordados diversos aspectos relacionados à escolha e implementação deste SGBD, bem como a modelagem de dados e consultas.

Inicialmente, discutimos as características e vantagens do MongoDB em relação aos bancos de dados relacionais tradicionais, destacando sua flexibilidade no armazenamento de dados sem esquema fixo e a escalabilidade horizontal. Essas características foram fundamentais para a escolha do MongoDB como solução para o projeto de gerenciamento de loja, permitindo lidar com as demandas dinâmicas de um ambiente de comércio eletrônico.

No processo de modelagem de dados, utilizamos o modelo conceitual para definir as entidades principais do sistema de gerenciamento de loja, como produtos, pedidos e clientes. Em seguida, traduzimos esse modelo para o modelo lógico, especificando os documentos JSON que seriam armazenados no MongoDB. Destacamos a importância da flexibilidade na definição de esquemas, adaptando-se facilmente às mudanças nos requisitos do sistema.

Durante a implementação, exploramos diversas consultas essenciais para operações típicas de uma aplicação de gerenciamento de loja, como inserção, atualização, consulta e remoção de documentos. Exemplos práticos foram fornecidos para ilustrar como essas operações são realizadas eficientemente no MongoDB, aproveitando recursos como índices, agregações e consultas complexas.

Por fim, os resultados obtidos demonstraram que o MongoDB é uma escolha viável e eficiente para sistemas de gerenciamento de loja, proporcionando desempenho e escalabilidade adequados às necessidades de uma aplicação de comércio eletrônico. A experiência adquirida neste projeto enfatiza a importância de considerar as características específicas do MongoDB ao projetar e implementar soluções que demandam flexibilidade e escalabilidade.

Dessa forma, concluímos que o MongoDB se mostrou uma solução robusta e adequada para o desenvolvimento de sistemas modernos de gerenciamento de loja, oferecendo benefícios significativos em relação aos bancos de dados relacionais tradicionais.

REFERÊNCIAS

As referências que vão ser utilizadas como base e auxílio para o desenvolvimento do projeto são:

LIVROS:

MANZANO, José Augusto N. G. **Algoritmos: Lógica Para Desenvolvimento de Programação de Computadores**. nº ed. 29. São Paulo Erica 2019.

ROB, Peter e CORONEL, Carlos. **Sistemas de Banco de Dados: Projeto, Implementação e Gerenciamento**. nº ed. 8. Cengage 2010.

ONLINE:

MongoDB. Disponível em: <https://www.mongodb.com>. Acesso em: 31/05/2024.

MongoDB Documentation. Disponível em: <https://docs.mongodb.com>. Acesso em: 31/05/2024.