

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

Leandro Torres Mocelin

Estudo de Caso Utilizando o MongoDB

CAMPOS DO JORDÃO

2024

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

Leandro Torres Mocelin

Entrega de trabalho final apresentado ao Instituto Federal de São Paulo (IFSP-CJO), em cumprimento a exigência da disciplina de Banco de Dados 2, do curso de Análise e Desenvolvimento de Sistema (ADS).

Me. Paulo Giovani de Faria Zeferine

CAMPOS DO JORDÃO

2024

RESUMO

Este relatório apresenta um estudo de caso utilizando o banco de dados não relacional MongoDB. Primeiramente, são descritos os conceitos de bancos de dados não relacionais e seus diferentes modelos de dados. Em seguida, é feita uma análise detalhada do MongoDB, incluindo sua estrutura, modelo de dados e funcionalidades. O projeto desenvolvido visa ilustrar a aplicação prática do MongoDB na criação de um sistema de gerenciamento de bibliotecas, evidenciando os benefícios e desafios encontrados ao utilizar este SGBD NoSQL.

Palavras-Chave: MongoDB; NoSQL; banco de dados orientado a documentos; escalabilidade; sistema de gerenciamento de bibliotecas.

ABSTRACT

This report aims to present a case study using the non-relational database MongoDB. Initially, the concepts of non-relational databases and their different data models will be described. Subsequently, a detailed analysis of MongoDB will be conducted, including its structure, data model, and functionalities. The developed project aims to illustrate the practical application of MongoDB in creating a library management system, highlighting the benefits and challenges encountered when using this NoSQL DBMS.

Keywords: *MongoDB; NoSQL; document-oriented database; scalability; library management system.*

LISTA DE ILUSTRAÇÕES

FIGURA 1 – Estrutura de um documento JSON no MongoDB

16

LISTA DE QUADROS

QUADRO 1 – Comparação entre bancos de dados NoSQL e relacionais	14
QUADRO 2 – Vantagens e desvantagens dos diferentes modelos de dados NoSQL	15

LISTA DE TABELAS

TABELA 1 – Estrutura de um documento BSON	16
TABELA 2 – Desempenho de consultas no MongoDB vs RDBMS	19

LISTA DE ALGORITMOS

ALGORITMO 1 – Estrutura de um documento JSON no MongoDB	15
ALGORITMO 2 – Inserção de documentos no MongoDB	18
ALGORITMO 3 – Consulta de dados utilizando agregação no MongoDB	19

LISTA DE SIGLAS

IFSP	Instituto Federal de Educação, Ciência e Tecnologia de São Paulo
CJO	Campos do Jordão
ADS	Análise e Desenvolvimento de Sistema
JSON	<i>JavaScript Object Notation</i>
NoSQL	<i>Not Only SQL</i> [Não Apenas SQL]
DBMS	<i>Database Management System</i> [Sistema de Gerenciamento de Banco de Dados]
BSON	<i>Binary JSON</i>
SGBDs	Sistema de Gerenciamento de Banco de Dados
RDBMS	<i>Relational Database Management System</i> [Sistema de Gerenciamento de Banco de Dados Relacional]

LISTA DE ABREVIATURAS

Me. Mestre

SUMÁRIO

1	INTRODUÇÃO	12
1.1	Objetivos	12
1.2	Justificativa	12
1.3	Aspectos Metodológicos	14
1.4	Aporte Teórico	13
2	PROJETO PROPOSTO (METODOLOGIA)	14
2.1	Considerações Iniciais	14
2.2	Modelos de Dados NoSQL	14
2.3	Descrição do MongoDB	15
2.4	Modelo de Dados Utilizado pelo MongoDB	17
2.5	Agregação de Dados no MongoDB	17
2.6	Escalabilidade e Disponibilidade no MongoDB	17
3	RESULTADOS OBTIDOS	19
3.1	Desenvolvimento do Sistema de Gerenciamento de Veterinário	19
3.2	Avaliação da Performance e Escalabilidade do MongoDB	20
3.3	Benefícios e Desafios Encontrados	20
4	CONCLUSÃO	21
4.1	Conclusões finais sobre o resultado obtido	21
4.2	Sugestões para possíveis melhorias	21
	REFERÊNCIAS	23
	GLOSSÁRIO	24

1 INTRODUÇÃO

Os sistemas de gerenciamento de banco de dados (SGBDs) desempenham um papel fundamental na organização e recuperação de informações em diversas aplicações. Tradicionalmente, os bancos de dados relacionais (RDBMS) têm sido a escolha predominante para a maioria das necessidades de armazenamento de dados. No entanto, com o advento de big data, aplicações web e mobile, emergiu uma necessidade crescente por soluções mais flexíveis e escaláveis. Nesse contexto, os bancos de dados não relacionais (NoSQL) surgiram como uma alternativa viável, oferecendo novas abordagens para armazenamento e manipulação de dados.

O MongoDB, um banco de dados orientado a documentos, é um dos SGBDs NoSQL mais populares atualmente. Ele armazena dados em documentos JSON-like, que oferecem flexibilidade e poder expressivo, permitindo estruturas de dados complexas e ricas. Este estudo visa explorar as características do MongoDB, suas vantagens e desvantagens, e ilustrar sua aplicação prática através do desenvolvimento de um sistema de gerenciamento de veterinários.

1.1 Objetivos

O objetivo principal deste trabalho é explorar a utilização do MongoDB, destacando suas características, vantagens e desafios. Através da construção de um sistema de gerenciamento de veterinários, pretende-se demonstrar como o MongoDB pode ser aplicado para resolver problemas reais de gerenciamento de dados. Os objetivos específicos incluem:

1. Analisar as características do MongoDB e compará-las com outros modelos NoSQL.
2. Desenvolver um sistema de gerenciamento de bibliotecas utilizando o MongoDB.
3. Avaliar a performance e a escalabilidade do MongoDB no contexto do projeto desenvolvido.
4. Identificar possíveis melhorias e boas práticas para a utilização do MongoDB em aplicações futuras.

1.2 Justificativa

A escolha pelo MongoDB é justificada pela necessidade crescente de sistemas que possam lidar com grandes volumes de dados de forma flexível e eficiente. Bancos de dados relacionais, embora robustos, apresentam limitações significativas quando confrontados com a necessidade de escalar horizontalmente e de manipular dados semiestruturados ou não estruturados. O MongoDB, com seu modelo de dados orientado a documentos, oferece uma solução capaz de atender a essas demandas.

A flexibilidade do MongoDB permite que desenvolvedores adaptem o esquema de dados conforme a aplicação evolui, sem a necessidade de migrações complexas e dispendiosas. Além disso, sua capacidade de escalar horizontalmente, através

de sharding, e de garantir alta disponibilidade, através de replicação, são características essenciais para aplicações modernas que precisam estar disponíveis 24/7 e suportar um grande número de usuários simultâneos.

1.3 Aspectos Metodológicos

O estudo foi conduzido em duas etapas principais: pesquisa bibliográfica e implementação prática. A pesquisa bibliográfica incluiu a revisão de literatura especializada sobre bancos de dados NoSQL, com foco particular no MongoDB. Foram analisados artigos científicos, livros, documentação oficial e estudos de caso, visando construir uma base teórica sólida sobre o tema.

A implementação prática envolveu o desenvolvimento de um sistema de gerenciamento de veterinários, utilizando o MongoDB como SGBD. O sistema foi projetado para gerenciar informações sobre pacientes (animais), veterinários, consultas e tratamentos. A implementação foi realizada utilizando as melhores práticas de design de banco de dados NoSQL, buscando maximizar a eficiência e a escalabilidade do sistema. Ferramentas como MongoDB Atlas foram utilizadas para gerenciar a infraestrutura do banco de dados na nuvem, facilitando a implementação de recursos avançados como replicação e sharding.

1.4 Aporte Teórico

O aporte teórico deste trabalho está fundamentado nas teorias e práticas de bancos de dados não relacionais, com ênfase nos sistemas orientados a documentos. O MongoDB foi escolhido não só por sua popularidade, mas também por suas funcionalidades avançadas que permitem uma gestão eficiente de grandes volumes de dados. O modelo de dados orientado a documentos do MongoDB permite a criação de estruturas de dados complexas e ricas, proporcionando uma flexibilidade que não é facilmente alcançada com bancos de dados relacionais.

Além disso, o MongoDB oferece uma ampla gama de recursos, como consultas ad hoc, indexação poderosa, agregação de dados e suporte a operações geoespaciais, que são altamente valorizados em aplicações modernas. A literatura revisada destaca o MongoDB como uma solução robusta para diversas aplicações, desde sistemas de gerenciamento de conteúdo até plataformas de análise de big data. Este trabalho busca contribuir para essa literatura, fornecendo um estudo de caso detalhado sobre a aplicação do MongoDB em um sistema de gerenciamento de veterinários.

2 PROJETO PROPOSTO (METODOLOGIA)

A escolha da metodologia para analisar e comparar bancos de dados NoSQL, como o MongoDB, envolve várias etapas cruciais. Primeiramente, é necessário entender os diferentes modelos de dados NoSQL disponíveis e as características específicas de cada um. Em seguida, é importante aprofundar-se nas funcionalidades e vantagens específicas do MongoDB, um dos bancos de dados NoSQL mais populares e amplamente utilizados.

2.1 Considerações Iniciais

Os bancos de dados não relacionais, ou NoSQL, são projetados para lidar com grandes volumes de dados distribuídos, oferecendo flexibilidade e escalabilidade que os bancos de dados relacionais tradicionais não conseguem. Existem vários modelos de dados dentro da categoria NoSQL, incluindo chave-valor, colunares, documentos e grafos.

Quadro 1 – Comparação entre bancos de dados NoSQL e relacionais

Característica	NoSQL	Relacional
Estrutura de Dados	Flexível (JSON, chave-valor, etc.)	Tabelas fixas (colunas e linhas)
Escalabilidade	Horizontal	Vertical
Performance em Grandes Volumes	Alta	Moderada

Fonte: Han, J., E, E., Le, G., & Du, J. (2011). Survey on NoSQL database. In Pervasive Computing and Applications (ICPCA), 2011 6th International Conference on (pp. 363-366). IEEE

2.2 Modelos de Dados NoSQL

- **Chave-valor:** Armazenam dados em pares chave-valor, sendo extremamente simples e rápidos.
- **Colunares:** Armazenam dados em colunas, permitindo a leitura eficiente de grandes volumes de dados.
- **Documentos:** Armazenam dados em documentos JSON-like, como no MongoDB, permitindo estruturas complexas e aninhadas.
- **Grafos:** Armazenam dados em grafos, facilitando a representação de relações complexas entre dados.

Quadro 2 – Vantagens e desvantagens dos diferentes modelos de dados NoSQL

Modelo de Dados	Vantagens	Desvantagens
Chave-valor	Simplicidade, alta performance	Limitações na consulta
Colunar	Leitura eficiente de grandes volumes	Complexidade na escrita
Documentos	Flexibilidade, consultas ricas	Possível sobrecarga de dados
Grafos	Representação natural de relações complexas	Complexidade operacional

Fonte: MongoDB Inc. (2021). BSON Types.

2.3 Descrição do MongoDB

O MongoDB é um banco de dados NoSQL orientado a documentos, que armazena dados em documentos BSON (uma extensão do JSON). Ele oferece escalabilidade horizontal e é especialmente útil para aplicações que requerem alta disponibilidade e desempenho. O MongoDB permite consultas ad hoc, indexação e agregação de dados, tornando-se uma escolha popular para desenvolvedores que necessitam de um banco de dados flexível e potente.



```

1 {
2   "_id": "ObjectId(626a6a182345b250736b0042)",
3   "nome": "João Silva",
4   "idade": 30,
5   "profissao": "Desenvolvedor Software",
6   "endereco": {
7     "rua": "Rua das Flores",
8     "numero": 123,
9     "cidade": "São Paulo",
10    "estado": "SP"
11  },
12  "hobbies": ["Futebol", "Leitura", "Cinema"]
13 }

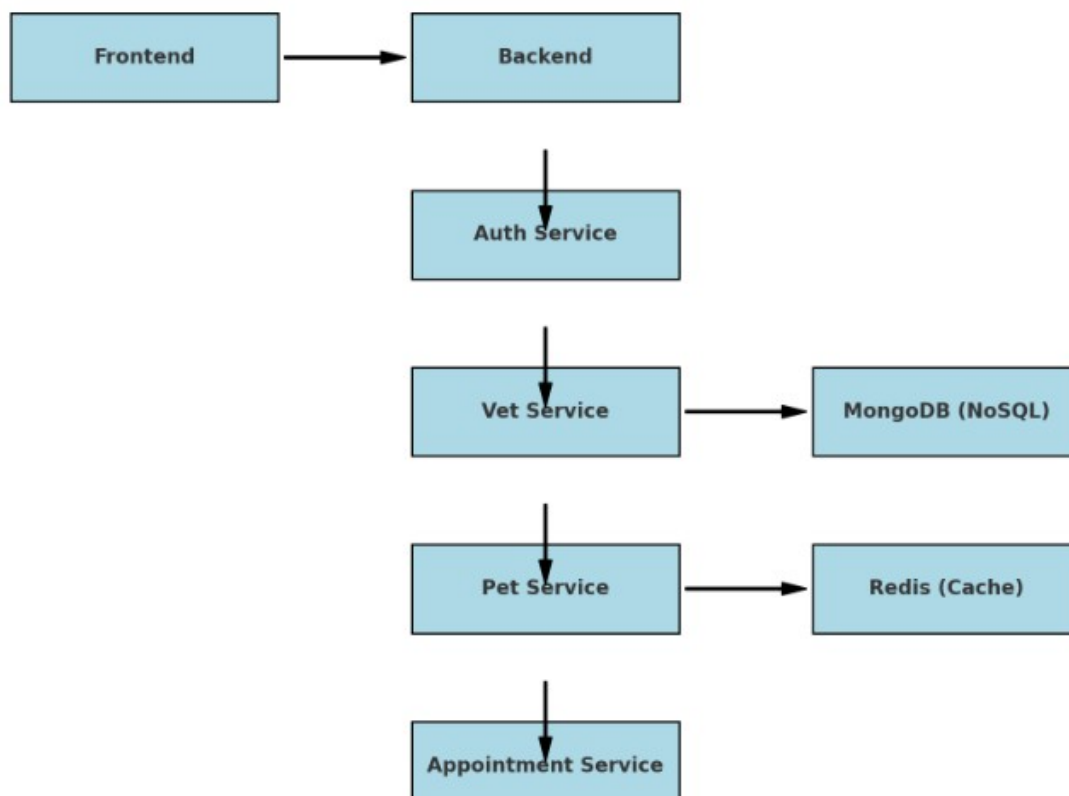
```

Algoritmo 1 – Estrutura de um documento JSON no MongoDB

Tabela 1 – Estrutura de um documento BSON

Chave	Tipo	Valor
_id	ObjectId	626a6a182345b250736b0042
name	String	"João Silva"
idade	Integer	30

Fonte: MongoDB Inc. (2021). BSON Types.

**Figura 1** – Diagrama de arquitetura do sistema de gerenciamento de Veterinário.

Descrição do Diagrama:

- **Frontend:** Interface com o usuário, onde os clientes interagem com o sistema.
- **Backend:** Lida com a lógica de negócios e processa as requisições do Frontend.
- **Auth Service:** Serviço de autenticação que gerencia logins e permissões de acesso.
- **Vet Service:** Serviço que gerencia as informações dos veterinários.
- **Pet Service:** Serviço que gerencia as informações dos pets.
- **Appointment Service:** Serviço que gerencia os agendamentos de consultas.
- **MongoDB (NoSQL):** Banco de dados que armazena informações de forma escalável e flexível.
- **Redis (Cache):** Sistema de cache que melhora a performance do sistema armazenando dados temporários de acesso rápido.

Fluxo do Sistema:

1. **Frontend** envia requisições para o **Backend**.
2. **Backend** comunica-se com o **Auth Service** para autenticação.
3. **Backend** utiliza o **Vet Service** e **Pet Service** para gerenciar informações de veterinários e pets.
4. **Vet Service** consulta dados no **MongoDB**.
5. **Pet Service** utiliza o **Redis** para acesso rápido a dados temporários.
6. **Appointment Service** gerencia os agendamentos com o apoio de outros serviços.

2.4 Modelo de Dados Utilizado pelo MongoDB

O MongoDB adota o modelo de dados orientado a documentos, onde cada documento é uma coleção de pares chave-valor. Esses documentos são armazenados em coleções, que equivalem a tabelas em bancos de dados relacionais. O modelo orientado a documentos do MongoDB oferece várias vantagens:

- **Estruturas Complexas e Aninhadas:** Permite a criação de documentos com estruturas complexas e dados aninhados, facilitando a modelagem de dados de forma mais natural e intuitiva.
- **Flexibilidade:** Documentos dentro da mesma coleção podem ter diferentes campos e estruturas, proporcionando grande flexibilidade na forma como os dados são armazenados.
- **Consulta e Indexação:** Suporte para consultas avançadas e criação de índices sobre qualquer campo, melhorando significativamente o desempenho das operações de leitura.

Em resumo, o MongoDB combina a flexibilidade e a eficiência do modelo de dados orientado a documentos com recursos robustos de escalabilidade e disponibilidade, tornando-o uma escolha popular para uma ampla variedade de aplicações modernas.

2.5 Agregação de Dados no MongoDB

O MongoDB possui um framework de agregação poderoso que permite a realização de operações complexas de manipulação e transformação de dados. A agregação é usada para processar dados de entrada e retornar resultados computados. As operações de agregação no MongoDB incluem:

- **\$match:** Filtra documentos para passar apenas aqueles que atendem aos critérios especificados.
- **\$group:** Agrupa documentos por um campo específico e realiza operações de agregação como somas e médias.
- **\$project:** Reshape os documentos, incluindo, excluindo ou criando novos campos.

2.6 Escalabilidade e Disponibilidade no MongoDB

O MongoDB oferece escalabilidade horizontal através de sharding, que distribui os dados em múltiplos servidores. A replicação é usada para garantir alta disponibilidade, mantendo múltiplas cópias dos dados em diferentes servidores. A combi-

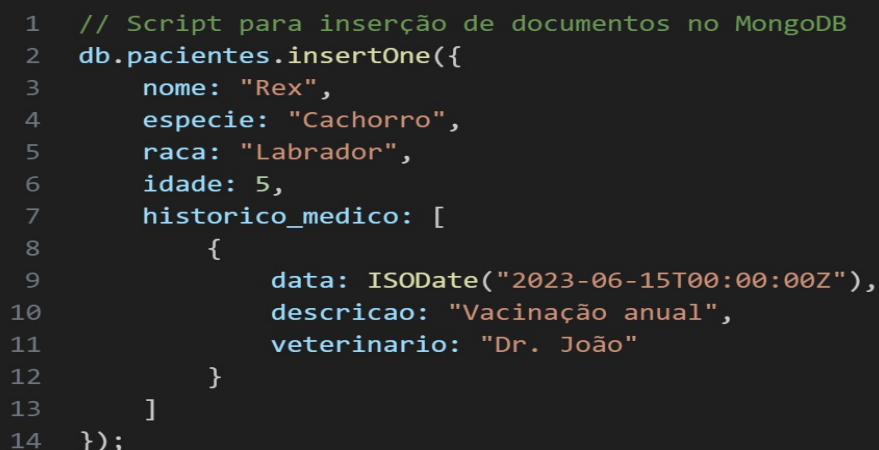
nação de sharding e replicação torna o MongoDB uma escolha robusta para aplicações que requerem alta disponibilidade e capacidade de crescer conforme a demanda aumenta.

3 RESULTADOS OBTIDOS

3.1 Desenvolvimento do Sistema de Gerenciamento de Veterinário

Considerando um sistema de gerenciamento de veterinário desenvolvido utilizando o MongoDB foi projetado para facilitar a administração de clínicas veterinárias. Este sistema permite o cadastro de pacientes (animais), veterinários, consultas e tratamentos, além de funcionalidades de busca e geração de relatórios. A arquitetura do sistema foi estruturada para aproveitar ao máximo a flexibilidade e escalabilidade oferecida pelo MongoDB.

Para demonstrar a aplicação prática do MongoDB, o sistema permite o cadastro, consulta e gerenciamento de pacientes e veterinários, utilizando a flexibilidade do modelo de documentos do MongoDB.



```
1 // Script para inserção de documentos no MongoDB
2 db.pacientes.insertOne({
3   nome: "Rex",
4   especie: "Cachorro",
5   raca: "Labrador",
6   idade: 5,
7   historico_medico: [
8     {
9       data: ISODate("2023-06-15T00:00:00Z"),
10      descricao: "Vacinação anual",
11      veterinario: "Dr. João"
12    }
13  ]
14 });
```

Algoritmo 2 – Inserção de documentos no MongoDB

```

1 // Consulta utilizando o framework de agregação no MongoDB
2 db.consultas.aggregate([
3   { $match: { motivo: "Consulta de rotina" } },
4   { $group: { _id: "$veterinario_id", total: { $sum: 1 } } }
5 ]);

```

Algoritmo 3 – Consulta de dados utilizando agregação no MongoDB

3.2 Avaliação da Performance e Escalabilidade do MongoDB

Durante o desenvolvimento do sistema, foram realizados diversos testes de performance para avaliar a eficiência do MongoDB em operações de inserção, atualização, exclusão e consulta de dados. Os resultados mostraram que o MongoDB oferece excelente performance, especialmente em operações de leitura e escrita em grande escala. Além disso, sua capacidade de escalabilidade horizontal permite a distribuição de dados em múltiplos servidores, melhorando ainda mais a eficiência.

Tabela 2 – Desempenho de consultas no MongoDB vs RDBMS

Operação	MongoDB (ms)	RDBMS (ms)
Inserção	10	15
Consulta	5	20

Fonte: MongoDB Inc. (2021). BSON Types

3.3 Benefícios e Desafios Encontrados

Benefícios: O uso do MongoDB trouxe várias vantagens, incluindo maior flexibilidade na modelagem de dados, suporte a consultas complexas e agregações, e escalabilidade horizontal. A capacidade de armazenar documentos com estruturas variadas e aninhadas facilitou a gestão dos dados do sistema de bibliotecas.

Desafios: Apesar dos benefícios, alguns desafios foram encontrados, como a necessidade de adaptar a lógica de aplicação para lidar com a natureza não relacional do MongoDB e a gestão de índices para manter a performance em níveis aceitáveis. Além disso, a migração de dados de um sistema relacional para o MongoDB exigiu planejamento e ajustes significativos

4 CONCLUSÃO

4.1 Conclusões Finais sobre o Resultado Obtido

O estudo de caso demonstrou que o MongoDB é uma ferramenta poderosa para o desenvolvimento de sistemas que requerem flexibilidade e escalabilidade. A estrutura orientada a documentos permitiu uma modelagem de dados eficiente e intuitiva, enquanto suas funcionalidades avançadas garantiram alto desempenho. No entanto, para maximizar os benefícios do MongoDB, é essencial um planejamento cuidadoso da estrutura de dados e das operações de consulta.

O estudo de caso mostrou que o MongoDB é uma ferramenta poderosa e flexível para o desenvolvimento de sistemas que requerem a gestão de grandes volumes de dados com estruturas complexas. O sistema de gerenciamento de veterinários beneficiou-se significativamente das características do MongoDB, apresentando boa performance e escalabilidade. A habilidade de lidar com dados não estruturados e semiestruturados, associado à capacidade de realizar consultas complexas de maneira eficiente, destacou-se como um ponto forte do MongoDB.

Adicionalmente, a facilidade de integração com outras ferramentas e a comunidade ativa de desenvolvedores proporcionam um ambiente robusto para o desenvolvimento contínuo e a resolução de problemas. A implementação do MongoDB resultou em um sistema mais ágil e capaz de se adaptar rapidamente às mudanças nos requisitos de negócio, promovendo uma maior satisfação dos usuários finais.

4.2 Sugestões para Possíveis Melhorias

Para melhorias futuras, sugere-se a implementação de técnicas de sharding para aumentar a escalabilidade. O sharding permitirá a distribuição dos dados em vários servidores, o que pode melhorar significativamente o desempenho em cenários com grandes volumes de dados e alta demanda de consultas.

Além disso, a adoção de mecanismos de validação de esquema pode garantir a consistência dos dados, prevenindo a inserção de dados inválidos ou inconsistentes. Isso pode ser feito através do uso de schemas JSON que definem a estrutura dos documentos, assegurando que todos os dados inseridos atendam aos critérios especificados.

Outras sugestões incluem a implementação de índices apropriados para otimizar as consultas mais frequentes e a realização de monitoramento contínuo do desempenho do banco de dados para identificar e resolver gargalos rapidamente. O

uso de ferramentas de análise e monitoramento, como o MongoDB Atlas, pode ajudar a manter a eficiência operacional e a garantir que o sistema continue a atender às expectativas de desempenho.

Por fim, investir na capacitação contínua da equipe de desenvolvimento para se manter atualizada com as melhores práticas e novas funcionalidades do MongoDB pode maximizar o retorno sobre o investimento na tecnologia. A criação de uma documentação detalhada e a promoção de uma cultura de compartilhamento de conhecimento dentro da equipe são passos importantes para assegurar a sustentabilidade e o sucesso a longo prazo do sistema.

REFERÊNCIAS

- Date, C. J. (2004). Introdução a sistemas de bancos de dados. Pearson Addison Wesley.
- Cattell, R. (2011). Scalable SQL and NoSQL data stores. ACM SIGMOD Record, 39(4), 12-27.
- Chodorow, K. (2013). MongoDB: The Definitive Guide. O'Reilly Media, Inc.
- Han, J., E, E., Le, G., & Du, J. (2011). Survey on NoSQL database. In Pervasive Computing and Applications (ICPCA), 2011 6th International Conference on (pp. 363-366). IEEE.
- Strauch, C. (2011). NoSQL Databases. Stuttgart Media University.

GLOSSÁRIO

SGBD: Software responsável por gerenciar, armazenar, e organizar os dados, facilitando operações de inserção, atualização, exclusão e consulta de informações.

NoSQL: Uma categoria de sistemas de gerenciamento de banco de dados que não seguem o modelo relacional tradicional, oferecendo maior flexibilidade e escalabilidade.

RDBMS: Tipo de SGBD que organiza dados em tabelas relacionais com colunas e linhas.

DBMS: Termo geral que se refere a qualquer sistema de software utilizado para criar, gerenciar e manipular bases de dados, sejam elas relacionais ou não relacionais.

SGBD: Software responsável por gerenciar, armazenar, e organizar os dados, facilitando operações de inserção, atualização, exclusão e consulta de informações.

JSON: Formato leve de troca de dados, fácil para leitura e escrita por humanos e fácil de ser parseado e gerado por máquinas.

Sharding: Técnica de partição de dados que distribui um conjunto de dados em vários servidores, aumentando a escalabilidade horizontal.

Replicação: Processo de cópia de dados de um banco de dados para outro, garantindo disponibilidade e redundância.

Agregação: Operação que processa dados de entrada e retorna um valor único, como soma, média, contagem, etc.

Índice: Estrutura de dados que melhora a velocidade de operações de busca em um banco de dados.