

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

MARCOS BENNER DINIZ MOREIRA

ESTUDO DE USO DO NOSQL USANDO O MONGODB

CAMPOS DO JORDÃO

2024

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

MARCOS BENNER DINIZ MOREIRA

Entrega de um projeto de pesquisa sobre o NoSQL apresentado ao Instituto Federal de São Paulo (IFSP), em cumprimento à exigência da disciplina de Banco de Dados 2 (BDD2), do curso de Análise e Desenvolvimento de Sistemas.

PROFESSOR: Paulo Giovani de Faria Zeferine.

RESUMO

O crescimento do mundo digital sobretudo a partir do final do século XXI com o surgimento da internet, gerou um grande volume de dados oriundos das aplicações web, deste modo observou-se que essa geração desenfreada de dados necessita de alguns requisitos específicos os quais essas aplicações exigem, como a escalabilidade sob demanda junto ao elevado grau de disponibilidade e estes fatores têm contribuído para o surgimento de novos paradigmas e tecnologias. Neste contexto, uma nova categoria de Banco de Dados, chamada NoSQL (Not Only SQL), foi proposta com o objetivo de atender aos requisitos de gerenciamento de grandes volumes de dados, semi-estruturados ou não estruturados. Este trabalho apresenta os fundamentos da tecnologia de banco de dados NoSQL, enfatizando as suas principais características e diferenciais, bem como suas áreas de aplicação. No projeto no qual foi feita a criação de um banco para biblioteca foi mostrado o modo como esses conceitos podem ser aplicados, para tanto usou-se o mongoDB um NoSQL orientado a documentos.

Palavras-Chave: NoSQL; Biblioteca; Dados; Digital.

ABSTRACT

The growth of the digital world especially from the end of the twenty-first century with the emergence of the Internet, has generated a large volume of data from web applications, thus it has been noted that this unrestricted generation of data requires some specific requirements which these applications require, such as scalability on demand along with the high degree of availability and these factors have contributed to the appearance of new paradigms and technologies. In this context, a new category of Database, called NoSQL (Not Only SQL), has been proposed with the aim of meeting the requirements of managing large volumes of data, semi-structured or unstructured. This paper presents the fundamentals of NoSQL database technology, emphasizing its main characteristics and differentials, as well as its areas of application. In the project in which was made the creation of a library bank was shown how these concepts can be applied, so used mongoDB a NoSQL document oriented.

Keywords:NoSQL; Library; Data; Digital.

LISTA DE FIGURAS

ALGORITMO 01 –	Criação do Banco e Collations
ALGORITMO 02 –	Insert 01
ALGORITMO 03 –	Insert 02
ALGORITMO 04 –	Insert 03
ALGORITMO 05 –	Consulta e Resultado 01
ALGORITMO 06 –	Consulta e Resultado 02
ALGORITMO 07 –	Consulta e Resultado 03
ALGORITMO 08 –	Consulta e Resultado 04
ALGORITMO 09 –	Consulta e Resultado 05
ALGORITMO 10 –	Consulta e Resultado 06
ALGORITMO 11 –	Consulta e Resultado 07
ALGORITMO 12 –	Consulta e Resultado 08
ALGORITMO 13 –	Consulta e Resultado 09
ALGORITMO 14 –	Consulta e Resultado 10
ALGORITMO 15 –	Consulta e Resultado 04

LISTA DE FIGURAS

SUMÁRIO

1	Introdução _____	7
2	Objetivos _____	9
2.1	Modelo Orientado a Documentos _____	9
3	Resultados Obtidos _____	10
3.1	Criação do Banco _____	11
3.2	Conclusão _____	21
2.1	Referências Bibliográficas _____	22

1 INTRODUÇÃO

O crescimento exponencial da geração de dados exigiram tratamentos diferenciados para seu manuseio, tais como a escalabilidade de acordo com a demanda juntamente a um elevado grau de disponibilidade, têm contribuído para o surgimento de novos paradigmas e tecnologias, aplicações como em redes sociais e streamings de multimídias, requerem o gerenciamento de grandes quantidades de dados e por muitas vezes não estruturados, sendo estes gerados diariamente por milhões de usuários em busca do compartilhamento de informações, conhecimentos e interesse.

A partir dessa necessidade, surgiu uma nova maneira de trabalhar com dados, Esta chamada NoSQL (Not Only SQL), que foi proposta com o objetivo de atender aos requisitos de gerenciamento de grandes volumes de dados, semi-estruturados ou não estruturados, que necessitam de alta disponibilidade e escalabilidade. A necessidade de uma nova tecnologia de BD surgiu como consequência da ineficiência dos bancos de dados relacionais em lidar com esta tarefa. Vale ressaltar que a tecnologia relacional foi proposta em meados do século XX, quando as aplicações de banco de dados caracterizavam-se por lidar com dados estruturados, ou seja, dados que possuem uma estrutura fixa e bem definida. Além disso, por maior que fossem os volumes de dados gerados por tais aplicações, ainda assim não se comparam ao volume de dados gerado pelas redes sociais, por exemplo, que também possuem um universo de usuários que cresce de maneira contínua e rápida, o que não acontece nas aplicações convencionais de banco de dados. A missão de solucionar esses problemas levou ao surgimento de uma nova classe de bancos de dados, que consiste em *column-oriented data stores*, *key/value pair databases*, e *document databases*, todos esses são identificados como NoSQL.

Os bancos de dados NoSQL têm sido bem vistos e por consequência adotados em empresas como Facebook, Amazon e Google com o intuito de atender às suas demandas de escalabilidade, alta disponibilidade e dados não estruturados. Além disso, atualmente, diversos bancos de dados NoSQL de código livre estão disponíveis, como: Cassandra¹, Hypertable², MongoDB³, Redis⁴, CouchDB⁵ e Dynamo⁶. O sistema de gerenciamento de banco de dados escolhido para estudo foi o MongoDB devido a sua forte presença no mercado.

2 METODOLOGIA

A metodologia adotada para o desenvolvimento desta pesquisa é de caráter qualitativo orientada a buscas bibliográficas fundamentada na análise de artigos publicados em grandes centros acadêmicos da área da tecnologia. O Sistema de Gerenciamento de Banco de Dados Escolhido para detalhar-se foi o MongoDB devido a sua importância para o setor. O MongoDB é um banco que não utiliza o modelo relacional, que até hoje ainda concentra grande parte dos Data Center, O fato é que este pertence à classe dos bancos NoSQL. Ferramentas, bases de dados e estruturas NoSQL, como foi apresentado no capítulo anterior modelos de armazenagem de dados que se diferem em partes do tradicional modelo entidade-relacionamento, o NoSQL tem ganhado importante notoriedade pois estão sendo utilizadas em grande escala hoje em dia. Deste modo, a pesquisa objetiva demonstrar de maneira simples e Objetiva a utilização destas ferramentas dando ênfase no Sistema de Gerenciamento de Banco de Dados MongoDB a fim de mostrar que esses novos conceitos têm seu surgimento inevitável devido às constantes mudanças nas Tecnologia, sobretudo após a viabilização da internet para usuários comuns o que tem ocasionado em uma geração de dados exponencial. neste trabalho será abordado o modelo orientado a documentos o qual será apresentado no tópico a seguir.

2.1 MODELO ORIENTADO A DOCUMENTOS

O modelo baseado em Documentos faz uso de associações entre pares de chaves e valores; porém, nesse modelo, os dados não são dispostos em uma única estrutura de dados, mas sim agrupados em documentos, que podem seguir a codificação XML ou JSON, sendo esta última a mais usada. Um documento é uma coleção de chaves e valores que estão relacionados a uma instância de dados. Os vários documentos pertencentes a um mesmo conceito do domínio são armazenados em uma coleção de documentos, da mesma maneira como as linhas são armazenadas em tabelas, no Modelo Relacional. alinhadas com as melhores práticas da indústria, atendendo de maneira eficaz às necessidades dos estudantes que se preparam para concursos públicos.

Porém, assim como em outros sistemas NoSQL, o Modelo em Documentos é flexível, permitindo que documentos de uma mesma coleção de dados possam apresentar campos distintos uns dos outros. As implementações dos sistemas NoSQL podem apresentar características diferentes umas das outras, ainda que pertençam a um mesmo modelo de dados. Para este trabalho, foi tomado como exemplo de Modelo em Documentos o do sistema MongoDB . Para este sistema, a chave de uma instância – nesse caso, o documento – é chamada de ObjectId ou simplesmente oid, um tipo de dados BSON de 12 bytes. A chave pode ser definida no momento da persistência do documento, ou pode ter seu valor gerado aleatoriamente pelo próprio banco. É importante mencionar que, no MongoDB, não é possível definir atributos com propriedades de unicidade, como ocorre com as chaves primárias do Modelo Relacional.

O Modelo orientado a Documentos permite que consultas mais complexas, envolvendo documentos de coleções distintas, ocorram. Para essas consultas, é necessário que um documento possua uma DBRef (Database Reference) do documento relacionado. Uma DBRef é uma estrutura que armazena as informações da coleção de documentos e o ObjectId do documento referenciado. Apesar de permitirem referências, as DBRef não garantem integridade referencial, pois não é possível assegurar que a referência terá o mesmo valor do ObjectId. Além disso, esse tipo de consulta não está disponível na linguagem de consulta do sistema, não podendo ser feita a requisição diretamente sobre a base de dados. Ela apenas é possível por meio de aplicações com acesso ao banco de dados, fazendo uso da sua API. Essas consultas, no entanto, são mais custosas, em termos de desempenho, em comparação com consultas realizadas considerando o aninhamento de mais dados em um determinado documento (dados relacionados são armazenados em um mesmo documento).

3 RESULTADOS OBTIDOS

Para estudo de caso foi desenvolvido um banco de dados NoSQL para uma biblioteca com as *collation* Livros, Autores e Editoras em seguida o próprio MongoDB gerou os endereços dos *documents*. Após a criação da base de dados foi possível fazer consultas, a fim de demonstrar o uso de um banco de dados que não segue o estilo entidade-relacionamento.

3.1 CRIAÇÃO DO BANCO E DAS COLLATIONS

A ferramenta utilizada para criar o banco e todo o restante dos comandos foi o MongoShell. Para esse projeto foram criadas 3 (três) collections: Livros, Autores e Editoras. No mongoDB não existe um comando em específico para a criação do Data Base, para isso basta dizer o nome do banco que deseja usar se este não existir automaticamente um espaço na memória é reservado, então basta fazer um primeiro insert ou a criação de uma collation para que este banco seja criado.

```
test> use biblioteca
switched to db biblioteca
biblioteca> db.createCollection("livros", { collation: { locale: "pt", strength: 2 }});
{ ok: 1 }
biblioteca> db.createCollection("autores", {collation: { locale: "pt", strength: 2 }});
{ ok: 1 }
biblioteca> db.createCollection("editoras", {collation: { locale: "pt", strength: 2 }});
{ ok: 1 }
biblioteca> _
```

Algoritmo 1 - Criação do banco e collation

Este exemplo cria três coleções (livros, autores e editoras) com um collation configurado para o idioma português (locale: "pt") e força (strength: 2), que faz com que a comparação de strings seja insensível a maiúsculas/minúsculas e insensível a acentos.

3.2 INSERT DE DADOS NAS COLLECTIONS

Para a inserção de dados foi usado o comando insertMany, que serve para a inserção de vários dados de uma única vez, é válido ressaltar que no mongoDB também existe a opção de inserimos dados usando arquivos CSV, no entanto por fins didáticos optou-se pelo processo em linhas de códigos.

3.2.1 INSERT DE DADOS NA COLLECTION LIVROS.

```
biblioteca> db.livros.insertMany([
...   { titulo: "Dom Casmurro", autor: "Machado de Assis", ano: 1899 },
...   { titulo: "Memórias Póstumas de Brás Cubas", autor: "Machado de Assis", ano: 1881 },
...   { titulo: "O Guarani", autor: "José de Alencar", ano: 1857 },
...   { titulo: "Iracema", autor: "José de Alencar", ano: 1865 },
...   { titulo: "Grande Sertão: Veredas", autor: "João Guimarães Rosa", ano: 1956 },
...   { titulo: "Vidas Secas", autor: "Graciliano Ramos", ano: 1938 },
...   { titulo: "A Moreninha", autor: "Joaquim Manuel de Macedo", ano: 1844 },
...   { titulo: "A Hora da Estrela", autor: "Clarice Lispector", ano: 1977 },
...   { titulo: "Capitães da Areia", autor: "Jorge Amado", ano: 1937 },
...   { titulo: "Os Sertões", autor: "Euclides da Cunha", ano: 1902 }
... ]);
```

Após o insert ser realizado o MongoDB automaticamente gera documents que representam cada elemento dentro de uma collection.

```
acknowledged: true,
insertedIds: {
  '0': ObjectId('667c1872caf9b68cd9cc8988'),
  '1': ObjectId('667c1872caf9b68cd9cc8989'),
  '2': ObjectId('667c1872caf9b68cd9cc898a'),
  '3': ObjectId('667c1872caf9b68cd9cc898b'),
  '4': ObjectId('667c1872caf9b68cd9cc898c'),
  '5': ObjectId('667c1872caf9b68cd9cc898d'),
  '6': ObjectId('667c1872caf9b68cd9cc898e'),
  '7': ObjectId('667c1872caf9b68cd9cc898f'),
  '8': ObjectId('667c1872caf9b68cd9cc8990'),
  '9': ObjectId('667c1872caf9b68cd9cc8991')
}
```

Algoritmo 2 - Insert 1

3.2.2 INSERT DE DADOS NA COLLECTION AUTORES.

```
biblioteca> db.autores.insertMany([
...   { nome: "Machado de Assis", nascimento: 1839, falecimento: 1908 },
...   { nome: "José de Alencar", nascimento: 1829, falecimento: 1877 },
...   { nome: "João Guimarães Rosa", nascimento: 1908, falecimento: 1967 },
...   { nome: "Graciliano Ramos", nascimento: 1892, falecimento: 1953 },
...   { nome: "Joaquim Manuel de Macedo", nascimento: 1820, falecimento: 1882 },
...   { nome: "Clarice Lispector", nascimento: 1920, falecimento: 1977 },
...   { nome: "Jorge Amado", nascimento: 1912, falecimento: 2001 },
...   { nome: "Euclides da Cunha", nascimento: 1866, falecimento: 1909 },
...   { nome: "Manuel Bandeira", nascimento: 1886, falecimento: 1968 },
...   { nome: "Carlos Drummond de Andrade", nascimento: 1902, falecimento: 1987 }
... ]);
```

```
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('667c1cdecaf9b68cd9cc8992'),
    '1': ObjectId('667c1cdecaf9b68cd9cc8993'),
    '2': ObjectId('667c1cdecaf9b68cd9cc8994'),
    '3': ObjectId('667c1cdecaf9b68cd9cc8995'),
    '4': ObjectId('667c1cdecaf9b68cd9cc8996'),
    '5': ObjectId('667c1cdecaf9b68cd9cc8997'),
    '6': ObjectId('667c1cdecaf9b68cd9cc8998'),
    '7': ObjectId('667c1cdecaf9b68cd9cc8999'),
    '8': ObjectId('667c1cdecaf9b68cd9cc899a'),
    '9': ObjectId('667c1cdecaf9b68cd9cc899b'),
  } '9': ObjectId('667c1872caf9b68cd9cc8991')
}
```

Algoritmo 3 - Insert 2

3.2.3 INSERT DE DADOS NA COLLECTION EDITORAS

```
biblioteca> db.editoras.insertMany([
...   { nome: "Companhia das Letras", fundacao: 1986 },
...   { nome: "Editora Record", fundacao: 1940 },
...   { nome: "Editora Globo", fundacao: 1929 },
...   { nome: "Editora Rocco", fundacao: 1975 },
...   { nome: "Editora Nova Fronteira", fundacao: 1965 },
...   { nome: "Editora Objetiva", fundacao: 1991 },
...   { nome: "Editora Intrínseca", fundacao: 2003 },
...   { nome: "Martins Fontes", fundacao: 1960 },
...   { nome: "Editora Paz e Terra", fundacao: 1967 },
...   { nome: "Editora Saraiva", fundacao: 1914 }
... ]);
```

```
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('667c4705b2232a5bebcc8992'),
    '1': ObjectId('667c4705b2232a5bebcc8993'),
    '2': ObjectId('667c4705b2232a5bebcc8994'),
    '3': ObjectId('667c4705b2232a5bebcc8995'),
    '4': ObjectId('667c4705b2232a5bebcc8996'),
    '5': ObjectId('667c4705b2232a5bebcc8997'),
    '6': ObjectId('667c4705b2232a5bebcc8998'),
    '7': ObjectId('667c4705b2232a5bebcc8999'),
    '8': ObjectId('667c4705b2232a5bebcc899a'),
    '9': ObjectId('667c4705b2232a5bebcc899b')
  }
}
biblioteca>
```

Algoritmo 4 - Insert 3

3.3 EXEMPLOS DE CONSULTAS

Para demonstração de uso do MongoDB, foram feitas algumas consultas dentro do banco de dados. consultas essas que exemplificam as diferenças entre um banco de dados relacional para outro no modelo NoSQL.

- 1) Selecionar livros de um autor específico (por exemplo, "Machado de Assis"):

```
biblioteca> db.livros.find({ autor: "Machado de Assis" })
[
  {
    _id: ObjectId('667c1872caf9b68cd9cc8988'),
    titulo: 'Dom Casmurro',
    autor: 'Machado de Assis',
    ano: 1899
  },
  {
    _id: ObjectId('667c1872caf9b68cd9cc8989'),
    titulo: 'O Alienista',
    autor: 'Machado de Assis',
    ano: 1881
  }
]
```

Algoritmo 5 - consulta e resultado 1

- 2) Selecionar autores nascidos antes de 1830:

```
biblioteca> db.autores.find({ nascimento: { $lt: 1830 } })
[
  {
    _id: ObjectId('667c1cdecaf9b68cd9cc8993'),
    nome: 'José de Alencar',
    nascimento: 1829,
    falecimento: 1877
  },
  {
    _id: ObjectId('667c1cdecaf9b68cd9cc8996'),
    nome: 'Joaquim Manuel de Macedo',
    nascimento: 1820,
    falecimento: 1882
  },
  {
    _id: ObjectId('667c1e50b2232a5bebcc8989'),
    nome: 'José de Alencar',
    nascimento: 1829,
    falecimento: 1877
  },
  {
    _id: ObjectId('667c1e50b2232a5bebcc898c'),
    nome: 'Joaquim Manuel de Macedo',
    nascimento: 1820,
    falecimento: 1882
  }
]
```

Algoritmo 6 - consulta e resultado 2

- 3) Selecionar editoras fundadas após o ano 2000.

```
biblioteca> db.editoras.find({ fundacao: { $gt: 2000 } });  
[  
  {  
    _id: ObjectId('667c4705b2232a5bebcc8998'),  
    nome: 'Editora Intrínseca',  
    fundacao: 2003  
  }  
]
```

Algoritmo 7 - consulta e resultado 3

- 4) Consultar os títulos dos livros

```
biblioteca> db.livros.find({}, { titulo: 1, _id: 0 })  
  
  titulo: 'Dom Casmurro' },  
  titulo: 'Memórias Póstumas de Brás Cubas' },  
  titulo: 'O Guarani' },  
  titulo: 'Iracema' },  
  titulo: 'Grande Sertão: Veredas' },  
  titulo: 'Vidas Secas' },  
  titulo: 'A Moreninha' },  
  titulo: 'A Hora da Estrela' },  
  titulo: 'Capitães da Areia' },  
  titulo: 'Os Sertões' }  
  
biblioteca> _
```

Algoritmo 8 - consulta e resultado 4

- 5) Selecionar livros ordenados por ano de publicação (do mais antigo para o mais recente):


```

biblioteca> db.livros.find().sort({ ano: 1 })
[
  {
    _id: ObjectId('667c1872caf9b68cd9cc898e'),
    titulo: 'A Moreninha',
    autor: 'Joaquim Manuel de Macedo',
    ano: 1844
  },
  {
    _id: ObjectId('667c1872caf9b68cd9cc898a'),
    titulo: 'O Guarani',
    autor: 'José de Alencar',
    ano: 1857
  },
  {
    _id: ObjectId('667c1872caf9b68cd9cc898b'),
    titulo: 'Iracema',
    autor: 'José de Alencar',
    ano: 1865
  },
  {
    _id: ObjectId('667c1872caf9b68cd9cc8989'),
    titulo: 'Memórias Póstumas de Brás Cubas',
    autor: 'Machado de Assis',
    ano: 1881
  },
  {
    _id: ObjectId('667c1872caf9b68cd9cc8988'),
    titulo: 'Dom Casmurro',
    autor: 'Machado de Assis',
    ano: 1899
  },
]

```

```

{
  _id: ObjectId('667c1872caf9b68cd9cc8991'),
  titulo: 'Os Sertões',
  autor: 'Euclides da Cunha',
  ano: 1902
},
{
  _id: ObjectId('667c1872caf9b68cd9cc8990'),
  titulo: 'Capitães da Areia',
  autor: 'Jorge Amado',
  ano: 1937
},
{
  _id: ObjectId('667c1872caf9b68cd9cc898d'),
  titulo: 'Vidas Secas',
  autor: 'Graciliano Ramos',
  ano: 1938
},
{
  _id: ObjectId('667c1872caf9b68cd9cc898c'),
  titulo: 'Grande Sertão: Veredas',
  autor: 'João Guimarães Rosa',
  ano: 1956
},
{
  _id: ObjectId('667c1872caf9b68cd9cc898f'),
  titulo: 'A Hora da Estrela',
  autor: 'Clarice Lispector',
  ano: 1977
}
]
biblioteca>

```

Algoritmo 9 - consulta e resultado 5

6) Selecionar autores ordenados por nome em ordem alfabética:

```

biblioteca> db.autores.find().sort({ nome: 1 })
[
  {
    _id: ObjectId('667c1cdecaf9b68cd9cc899b'),
    nome: 'Carlos Drummond de Andrade',
    nascimento: 1902,
    falecimento: 1987
  },
  {
    _id: ObjectId('667c1e50b2232a5bebcc8991'),
    nome: 'Carlos Drummond de Andrade',
    nascimento: 1902,
    falecimento: 1987
  },
  {
    _id: ObjectId('667c1cdecaf9b68cd9cc8997'),
    nome: 'Clarice Lispector',
    nascimento: 1920,
    falecimento: 1977
  },
  {
    _id: ObjectId('667c1e50b2232a5bebcc898d'),
    nome: 'Clarice Lispector',
    nascimento: 1920,
    falecimento: 1977
  },
  {
    _id: ObjectId('667c1cdecaf9b68cd9cc8999'),
    nome: 'Euclides da Cunha',
    nascimento: 1866,
    falecimento: 1909
  },
]

```

Algoritmo 10 - consulta e resultado 6

7) Selecionar os 5 primeiros livros que foram cadastrados:

```

biblioteca> db.livros.find().limit(5)
[
  {
    _id: ObjectId('667c1872caf9b68cd9cc8988'),
    titulo: 'Dom Casmurro',
    autor: 'Machado de Assis',
    ano: 1899
  },
  {
    _id: ObjectId('667c1872caf9b68cd9cc8989'),
    titulo: 'Memórias Póstumas de Brás Cubas',
    autor: 'Machado de Assis',
    ano: 1881
  },
  {
    _id: ObjectId('667c1872caf9b68cd9cc898a'),
    titulo: 'O Guarani',
    autor: 'José de Alencar',
    ano: 1857
  },
  {
    _id: ObjectId('667c1872caf9b68cd9cc898b'),
    titulo: 'Iracema',
    autor: 'José de Alencar',
    ano: 1865
  },
  {
    _id: ObjectId('667c1872caf9b68cd9cc898c'),
    titulo: 'Grande Sertão: Veredas',
    autor: 'João Guimarães Rosa',
    ano: 1956
  }
]

```

Algoritmo 11 - consulta e resultado 7

8) Selecionar os 5 primeiros autores, pulando os primeiros 2 (paginação):

```
biblioteca> db.autores.find().skip(2).limit(5)
[
  {
    _id: ObjectId('667c1cdecaf9b68cd9cc8994'),
    nome: 'João Guimarães Rosa',
    nascimento: 1908,
    falecimento: 1967
  },
  {
    _id: ObjectId('667c1cdecaf9b68cd9cc8995'),
    nome: 'Graciliano Ramos',
    nascimento: 1892,
    falecimento: 1953
  },
  {
    _id: ObjectId('667c1cdecaf9b68cd9cc8996'),
    nome: 'Joaquim Manuel de Macedo',
    nascimento: 1820,
    falecimento: 1882
  },
  {
    _id: ObjectId('667c1cdecaf9b68cd9cc8997'),
    nome: 'Clarice Lispector',
    nascimento: 1920,
    falecimento: 1977
  },
  {
    _id: ObjectId('667c1cdecaf9b68cd9cc8998'),
    nome: 'Jorge Amado',
    nascimento: 1912,
    falecimento: 2001
  }
]
```

Algoritmo 12 - consulta e resultado 8

9) Selecionar as 3 primeiras editoras ordenadas por nome em ordem alfabética:

```
biblioteca> db.editoras.find().sort({ nome: 1 }).limit(3)
[
  {
    _id: ObjectId('667c4705b2232a5bebcc8992'),
    nome: 'Companhia das Letras',
    fundacao: 1986
  },
  {
    _id: ObjectId('667c4705b2232a5bebcc8994'),
    nome: 'Editora Globo',
    fundacao: 1929
  },
  {
    _id: ObjectId('667c4705b2232a5bebcc8998'),
    nome: 'Editora Intrínseca',
    fundacao: 2003
  }
]
```

Algoritmo 13 - consulta e resultado 9

10) Selecionar autores que nasceram antes de 1860 ou faleceram após 1900:

```
biblioteca> db.autores.find({
...   $and: [
...     { nascimento: { $lt: 1860 } },
...     { falecimento: { $gt: 1900 } }
...   ]
... })
[
  {
    _id: ObjectId('667c1cdecaf9b68cd9cc8992'),
    nome: 'Machado de Assis',
    nascimento: 1839,
    falecimento: 1908
  },
  {
    _id: ObjectId('667c1e50b2232a5bebcc8988'),
    nome: 'Machado de Assis',
    nascimento: 1839,
    falecimento: 1908
  }
]
```

Algoritmo 14 - consulta e resultado 10

Essas 10 (dez) consultas cobrem uma variedade de consultas que podem ser realizadas no banco de dados de uma biblioteca. Cada consulta demonstra como filtrar, ordenar, projetar e paginar os resultados em MongoDB. Deste modo pode-se demonstrar a grande variedade de recursos que esse sistema possui.

4 CONCLUSÃO

Ao desenvolver-se esta pesquisa e implementar o NoSQL usando o mongoDB. Concluiu-se que esse oferece diversas vantagens que o tornam uma escolha atraente para muitos projetos observou-se que sua flexibilidade de esquema permite armazenar dados sem uma estrutura fechada a pré-requisitos, facilitando ajustes e evoluções no modelo de dados conforme as necessidades do projeto mudam. O MongoDB é incrivelmente escalável, suportando a escala tanto horizontal quanto vertical, o que é essencial para aplicações que precisam lidar com grandes volumes de dados e alto tráfego. Além disso, sua performance é otimizada para operações de leitura e escrita, graças à capacidade de criar índices variados e complexos. Outra característica observada foi a facilidade de integração com outras tecnologias, proporcionando um desenvolvimento mais ágil e eficiente. O MongoDB também oferece ótimas funcionalidades de replicação e alta disponibilidade, garantindo que os dados estejam sempre acessíveis e protegidos contra falhas. Por fim, suas capacidades avançadas de agregação e suporte a consultas complexas permitem a realização de análises de dados diretamente no banco, sem a necessidade de ferramentas externas. Essas vantagens combinadas fazem do MongoDB uma solução poderosa para construir aplicações modernas e escaláveis características cada vez mais solicitadas pelas companhias empresariais.

Neste projeto, criamos um banco de dados MongoDB chamado biblioteca com coleções para livros, autores e editoras, configurando collation para o idioma português e inserindo dados de exemplo. Realizamos diversas consultas para demonstrar as funcionalidades do MongoDB. Para projetos futuros, é crucial investir em uma modelagem de dados adequada, equilibrando normalização e desnormalização, criar índices apropriados para melhorar a performance, considerar o uso de sharding para grandes volumes de dados, implementar estratégias de backup e manutenção regular, configurar controle de acesso e criptografia para segurança, e documentar detalhadamente o sistema. Além disso, adicionar funcionalidades como sistemas de recomendação, integrar com outras ferramentas e plataformas, usar análise de dados para insights, e automatizar tarefas repetitivas são passos importantes para expandir e otimizar o sistema, garantindo que ele seja robusto, escalável e seguro.

REFERÊNCIAS

STROZZI, C. **NoSQL**: a non-SQL RDBMS. 1998.

Disponível em: <http://www.strozzi.it/cgi-bin/CSA/tw7//en_US/nosql/Home%20Page>. Acesso em: 22 jun. 2024

CLAUDINO, M . **Mapeamentos conceituais entre os modelos Relacional e NoSQL: uma abordagem comparativa**. 2015.

Disponível em: <https://periodicos.ifpb.edu.br/index.php/principia/article/view/495>
Acesso em: 23 jun 2024.

FREITAS, C. M. **Mapeamentos Conceituais entre Modelo Relacional e Estruturas NoSQL**: um estudo de caso com documentos. 2015. 105 f. Dissertação (Mestrado em Ciência da Computação) – Centro de Informática, Universidade Federal de Pernambuco, Recife, 2015

PRESSMAN, Roger. **Engenharia de Software: Uma Abordagem Profissional**. 7 ed. Porto Alegre: Book Man, 2010.