

# INFOESTE 2024

---

Explorando MongoDB: Um Guia Prático para Iniciantes

# Vinicius Vasconcelos

Professor | Desenvolvedor  
Back-end





Objetivo

## Métricas de sucesso...

- Oferecer um curso rápido e introdutório;
- Ampliar o conhecimento das pessoas participantes;
- Promover a interação entre as pessoas participantes de forma descontraída, educativa e envolvente;
- Incentivar o networking.



Ferramentas

# Tecnologias utilizadas no curso!

- [MongoDB](#);
- [MongoDB Compass](#).



# Bancos de Dados

# O que são banco de dados?

Banco de dados é uma estrutura organizada que pode ser facilmente acessada, gerenciada e atualizada, através de uma linguagem especializada.

Componentes de um banco de dados:

- **Dados:** A informação armazenada;
- **SGBD (Sistema de Gerenciamento de Banco de Dados):** Software que gerencia o banco de dados;
- **Client:** Pessoas ou aplicações que interagem com o banco de dados.



# Tipos de bancos de dados...

- **Relacionais:** Utilizam tabelas para armazenar dados e são baseados relacionamentos;
- **Não relacionais:** Incluem vários tipos e alguns não incluem relacionamento em seu core.

# Bancos não relacionais

Os 4 tipos...

## Documentos (MongoDB)

Armazenam dados na forma de documentos (JSON, BSON, ...)

## Chave e valor (Redis)

Armazenam dados como pares de chave-valor, onde cada chave é única e associada a um valor.

## Colunares (Cassandra)

Armazenam dados em colunas em vez de linhas, o que permite consultas rápidas em grandes volumes de dados.

## Grafo (Neo4j)

Armazenam dados na forma de grafos, com nós e arestas representando entidades e seus relacionamentos.

---

# Colunares

- **Casos de Uso:** Análise de dados, big data, sistemas OLAP e outros;
- **Vantagens:** Excelente performance para leitura e escrita de grandes volumes de dados;
- **Desvantagens:** Pode ser mais complexo de gerenciar e configurar.

Imagem retirada do Google Images



# Grafo

- **Casos de Uso:** Redes sociais, sistemas de recomendação, análise de fraudes e outros
- **Vantagens:** Ideal para dados altamente conectados e consultas complexas de relacionamento;
- **Desvantagens:** Pode ser menos eficiente para dados tabulares simples.

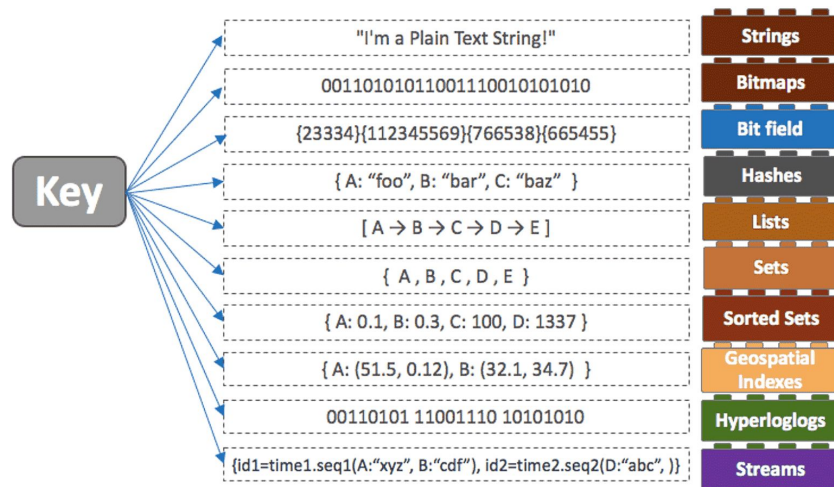
Imagem retirada do Google Images



# Chave e valor

- **Casos de Uso:** Cache, sessões de usuário, armazenamento de configurações e outros;
- **Vantagens:** Simplicidade e alta performance;
- **Desvantagens:** Limitado em termos de consultas complexas e relacionamentos entre dados.

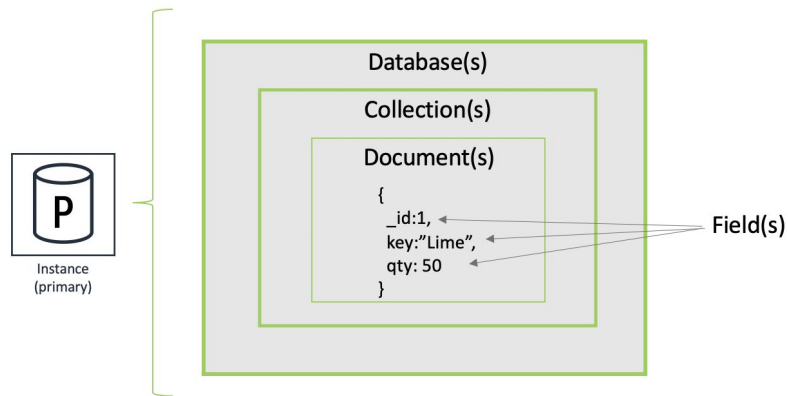
Imagem retirada do Google Images



# Documentos

- **Casos de Uso:** Aplicações web, gerenciamento de conteúdo e outros;
- **Vantagens:** Flexibilidade de esquema, fácil de escalar horizontalmente;
- **Desvantagens:** Pode ser menos eficiente para consultas complexas que envolvem múltiplos documentos.

Imagem retirada do Google Images



# Exercitando...

Pausa para o Kahoot!





MongoDB



# O que é o MongoDB?

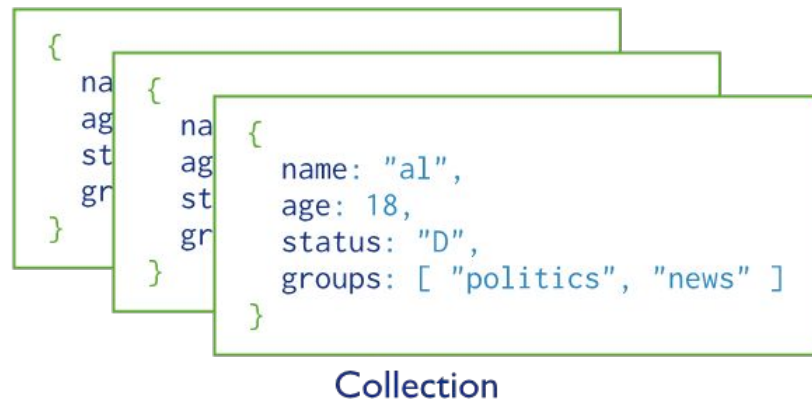
**MongoDB** é um banco de dados de documentos, projetado para facilitar o desenvolvimento e o dimensionamento de aplicativos.

[Documentação oficial.](#)



# Collections

O MongoDB armazena documentos em **coleções**. Coleções são análogas às tabelas do banco de dados relacional.



# Document

Um registro no MongoDB é um **documento**, que é uma estrutura de dados composta de pares de campos e valores. Os documentos MongoDB são semelhantes aos objetos JSON .

```
{  
  name: "sue",  
  age: 26,  
  status: "A",  
  groups: [ "news", "sports" ]  
}
```



← field: value  
← field: value  
← field: value  
← field: value



# Principais comandos - CRUD

# Criando base de dados!

Use o comando **use** seguido pelo nome da base de dados que deseja criar. Se a base de dados não existir, ela será criada automaticamente quando você inserir dados.

Exemplo: **use minhaBaseDeDados**

# Inserindo registros!

As operações de **inserção** adicionam novos **documentos** a uma **coleção**. Se a coleção não existir, ela é criada automaticamente.

MongoDB fornece os seguintes métodos para inserir documentos em uma coleção:

- **db.collection.insertOne()**
- **db.collection.insertMany()**

*\*Todas operações de gravação no MongoDB são atômicas no nível de um único documento.*

```
db.users.insertOne(  ← collection
{
  name: "sue",        ← field: value
  age: 26,            ← field: value
  status: "pending"   ← field: value
}                    } document
)
```

# Pesquisando registros!

Operações de **leitura** recuperam documentos de uma coleção.

MongoDB fornece o seguinte método para ler documentos de uma coleção:

- **db.collection.find()**

*\*Você pode especificar filtros de consulta ou critérios que identifiquem os documentos a serem devolvidos.*

```
db.users.find(  
  { age: { $gt: 18 } },  
  { name: 1, address: 1 }  
) .limit(5)
```

← collection  
← query criteria  
← projection  
← cursor modifier

# Alterando registros!

As operações de **atualização** modificam documentos existentes em uma coleção.

MongoDB fornece os seguintes métodos para atualizar documentos de uma coleção:

- **db.collection.updateOne()**
- **db.collection.updateMany()**

*\*Todas as operações de gravação no MongoDB são atômicas no nível de um único documento.*

*\*Você pode especificar critérios ou filtros que identificam os documentos a serem atualizados. Estes filtros utilizam a mesma sintaxe que as operações de leitura.*

```
db.users.updateMany(  
  { age: { $lt: 18 } },  
  { $set: { status: "reject" } }  
)
```

← collection  
← update filter  
← update action



# Deletando registros!

As operações de **exclusão** removem documentos de uma coleção.

MongoDB fornece os seguintes métodos para excluir documentos de uma coleta:

- **db.collection.deleteOne()**
- **db.collection.deleteMany()**

\*Todas as operações de gravação no MongoDB são atômicas no nível de um único documento.

\*Você pode especificar critérios ou filtros que identificam os documentos a serem removidos. Esses filtros usam a mesma sintaxe que as operações de leitura.

```
db.users.deleteMany(  
  { status: "reject" }  
)
```



collection  
delete filter



# Operadores de consulta e projeção

## Veja a [documentação oficial](#)!

O MongoDB provê inúmeros operadores para auxílio em suas consultas, dentre eles estão separados em algumas classes:

- **Comparação:** \$eq, \$ne, \$gt, \$lt, \$gte, \$lte;
- **Lógica:** \$and, \$or, \$not, \$nor;
- **Elemento:** \$exists, \$type;
- **Avaliação:** \$regex, \$expr, \$mod, \$text;
- **Arrays:** \$all, \$elemMatch, \$size;
- **Projeção:** \$, \$elemMatch, \$meta, \$slice.

# Obrigado



GitHub: [vinicius-vasconcelos](https://github.com/vinicius-vasconcelos)



LinkedIn: [vinicius-vasconcelos-8828a416b](https://www.linkedin.com/in/vinicius-vasconcelos-8828a416b)