

Introdução a Banco de NoSQL

Pâmela Apolinário Borges

Engenheira de Software

@pamelaapborges - Linkedin

Objetivo Geral

O objetivo geral do curso é fornecer uma introdução aos Bancos de Dados não relacionais e desenvolver habilidades na criação, modelagem e consulta no MongoDB

Pré-requisitos

- ✓ Computador com acesso a internet
- ✓ Instalação do Compass

Conteúdo Programático

- Conceitos básicos dos bancos de dados não relacionais
- Visão geral dos tipos de NoSQL: Key-Value, Document, Column, Graph

Conteúdo Programático

- ❑ Introdução ao MongoDB: Configuração no Atlas
- ❑ Modelagem de dados usando documentos
- ❑ Operações CRUD
- ❑ Operadores Lógicos e de comparação
- ❑ Projeção, ordenação, limitação e paginação

Conteúdo Programático

- ❑ Introdução ao Redis
- ❑ Operações Basicas no Redis

Conceitos básicos dos bancos de dados não relacionais

Introdução aos Bancos de Dados Não Relacionais

O que é um Banco de Dados não relacional?

- Termo correto: **NOT Only SQL**
- Não seguem modelo de tabelas e relacionamentos
- Projetados para lidar com **alto volume de dados**, alta escalabilidade
- Alta **flexibilidade** na estrutura de dados
- Eles são amplamente utilizados em cenários onde a consistência imediata dos dados não é crítica

Diferenças

SQL	NoSQL
Modelo de dados fixo	Modelo de dados flexivel
Escalabilidade vertical (hardware)	Escalabilidade horizontal
Transações ACID 100%	Transações ACID ausentes total ou parcial
Línguagem de consulta SQL	Cada SGBD tem sua própria

Vantagens dos bancos de dados NoSQL

- Flexibilidade na modelagem
- Alta escalabilidade
- Melhor desempenho em cenário de consulta intensiva
- Tolerância a falhas

Desvantagens dos bancos de dados NoSQL

- ❑ Menor consistência de dados imediata
- ❑ Menor suporte a consultas complexas ** depende do SGBD

Links Úteis

- **Referências:**
 - <https://www.oracle.com/br/database/nosql/what-is-nosql>

Visão geral dos tipos de NoSQL

Introdução aos Bancos de Dados Não Relacionais

Tipos

- Key-Value
- Documento
- Coluna
- Grafos
- entre outros...

Key-Value > Chave Valor

Armazena dados como pares de chave e valor, onde cada chave é um identificador único para acessar o valor correspondente

Exemplo de SGBD: Redis, Riak, Amazon DynamoDB

Uso: Um site pode usar um banco de dados Redis para armazenar informações de sessão de usuário

Document > Documento

Armazenam dados em documentos semiestruturados, geralmente em formato JSON ou BSON

Exemplo de SGBD: MongoDB, Couchbase, Apache CouchDB

Uso: Um catálogo de e-commerce pode usar o MongoDB para armazenar informações de produtos, como nome, descrição, preço e atributos adicionais.

Coluna

Armazenam dados em formato de colunas, o que permite alta escalabilidade e eficiência em determinados tipos de consultas

Exemplo de SGBD: Apache Cassandra, ScyllaDB, HBase

Uso: Um sistema de registro de aplicativos pode usar o Apache Cassandra para armazenar registros de log.

Grafo

Armazenar e consultar dados interconectados, onde os relacionamentos entre os dados são tão importantes quanto os próprios dados

Exemplo de SGBD: Neo4j, Amazon Neptune, JanusGraph

Uso: Uma rede social pode usar o Neo4j para armazenar os perfis dos usuários e suas conexões, permitindo consultas eficientes para encontrar amigos em comum.

Links Úteis

- **Referências:**
 - <https://www.oracle.com/br/database/nosql/what-is-nosql>

Introdução ao MongoDB

Introdução aos Bancos de Dados Não Relacionais

O que é o MongoDB

- Banco de dados NoSQL orientado a documentos.
- Grandes volumes de dados, escalabilidade horizontal e modelagem flexível.
- Não exige um esquema
- Permite que os documentos sejam armazenados em formato BSON (Binary JSON), proporcionando uma estrutura semiestruturada.

Vantagens

- Flexibilidade na modelagem de dados.
- Escalabilidade horizontal para lidar com grandes volumes de dados.
- Consultas ricas e suporte a consultas complexas.
- Alta disponibilidade e tolerância a falhas.
- Comunidade ativa e recursos de suporte.

Desvantagens

- ❑ Menor consistência imediata em comparação com bancos de dados relacionais.
- ❑ Consultas complexas podem exigir um maior conhecimento e planejamento adequado.
- ❑ Maior consumo de espaço de armazenamento em comparação com bancos de dados relacionais devido à flexibilidade dos documentos.

Onde o MongoDB é usado

- ❑ Aplicações web: Onde a flexibilidade e a escalabilidade são cruciais para lidar com volumes variáveis de dados.
- ❑ Análise de big data: Análise de grandes volumes de dados não estruturados ou semiestruturados, fornecendo uma plataforma para armazenar e processar esses dados.
- ❑ Armazenamento de dados semiestruturados: Permite a inserção de documentos com estruturas diferentes em uma mesma coleção.

Onde o MongoDB é usado

- ❑ Casos de uso de geolocalização: Com suas funcionalidades de consulta geoespacial, é adequado para casos de uso que envolvem dados baseados em localização, como aplicativos de mapeamento e rastreamento.

Links Úteis

- **Referências:**
 - <https://www.mongodb.com/docs/manual/introduction/>

Instalação e configuração do MongoDB (Atlas)

Introdução aos Bancos de Dados Não Relacionais

MongoDB Atlas

❑ <https://cloud.mongodb.com/>

Criando sua conta

[Products](#)[Solutions](#)[Resources](#)[Company](#)[Pricing](#)

MongoDB Cloud Services

MongoDB cloud services consist of a comprehensive suite of data products that accelerate and simplify how you build with data for any application. With Atlas Database (the Database-as-a-Service for MongoDB), Search, and Data Federation, you can serve any class of workload through a common API. Additionally, bidirectional sync between Atlas and Realm allows you to extend your cloud backend to the edge and mobile devices.



Try Free

Sign up

See what Atlas is capable of for free

First Name*

Last Name*

Company

Email*

Password*

I agree to the [Terms of Service](#) and [Privacy Policy](#).

Create your Atlas account

Sign in



Accept Privacy Policy & Terms of Service

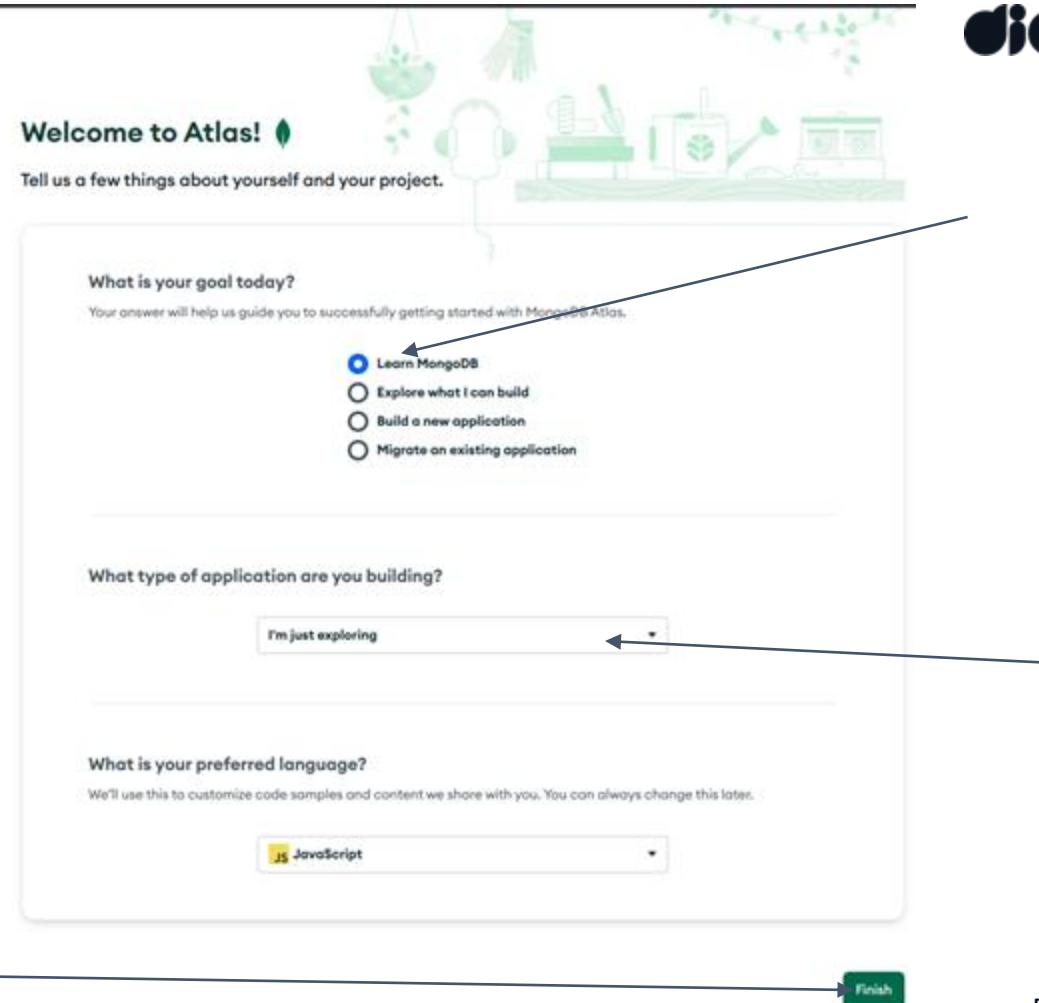
Please acknowledge the following terms and conditions to finish creating your account.

I accept the [Privacy Policy](#) and the [Terms of Service](#)

[Cancel Signup](#)

[Submit](#)

Configurando a conta



Welcome to Atlas! 🌱

Tell us a few things about yourself and your project.

What is your goal today?

Your answer will help us guide you to successfully getting started with MongoDB Atlas.

- Learn MongoDB
- Explore what I can build
- Build a new application
- Migrate an existing application

What type of application are you building?

I'm just exploring

What is your preferred language?

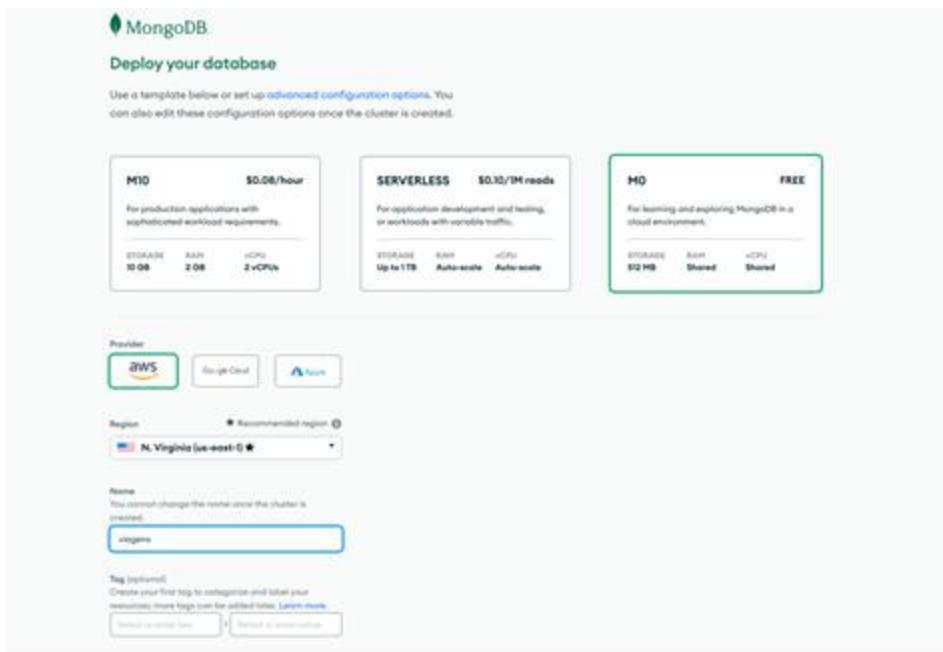
We'll use this to customize code samples and content we share with you. You can always change this later.

js JavaScript

Finish

[32]

Criando o Database



Criando o usuário

The screenshot shows the MongoDB Cloud interface for creating a database user. On the left, a sidebar lists 'DEPLOYMENT' (Database, Data Lake), 'SERVICES' (Triggers, Data API, Data Federation, Search, Stream Processing), and 'SECURITY' (Quickstart, Backup, Database Access, Network Access, Advanced). A 'New On Atlas' button is highlighted with a red arrow. The main area is titled 'Security Quickstart' and displays a message: 'To access data stored in Atlas, you'll need to create users and set up network security controls.' Below this, a question asks 'How would you like to authenticate your connection?' with two options: 'Username and Password' (highlighted with a green border) and 'Certificate'. A blue info box states: 'We autogenerated a username and password for your first database user in this project using your MongoDB Cloud registration information.' A large callout box provides instructions: 'Create a database user using a username and password. Users will be given the read and write to any database privilege by default. You can update these permissions and/or create additional users later. Ensure these credentials are different to your MongoDB Cloud username and password.' It includes fields for 'Username' (admin), 'Password' (empty field), and 'Autogenerate Secure Password' (checkbox). A 'Create User' button is at the bottom, also highlighted with a red arrow. A 'Copy' button is next to it.

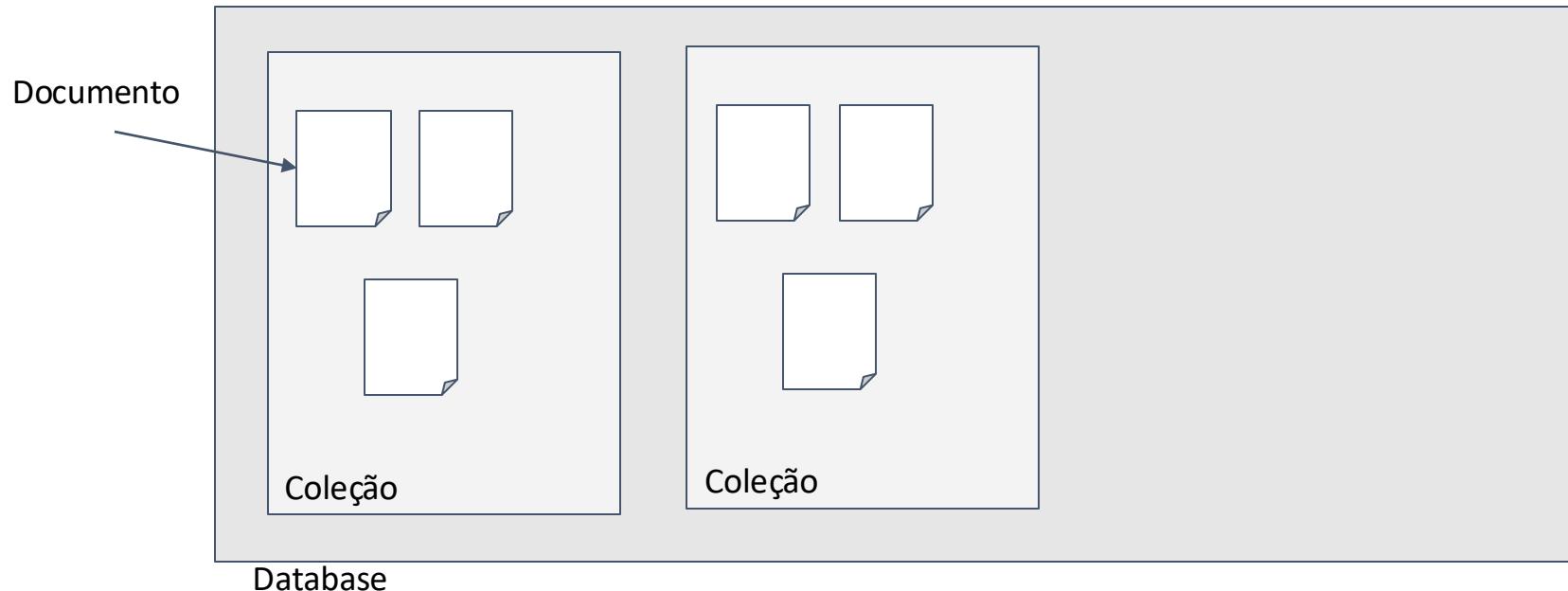
Links Úteis

- **Referências:**
 - <https://www.mongodb.com/docs/atlas/getting-started/>

Modelagem de dados usando documentos

Introdução aos Bancos de Dados Não Relacionais

Estrutura do MongoDB



Coleções

- Agrupamento lógico de documentos
- Não exige esquema ou que os documentos tenham a mesma estrutura

Características:

- Os nomes das coleções devem seguir algumas regras:
- Devem começar com uma letra ou um underscore (_).
- Podem conter letras, números ou underscores.
- Não podem ser vazios.
- Não podem ter mais de 64 bytes de comprimento.

Documentos:

- ❑ São armazenados em documentos BSON (Binary JSON), que são estruturas flexíveis e semiestruturadas.
- ❑ Cada documento possui um identificador único chamado "`_id`"
- ❑ É composto por pares de chaves e valores.

Documentos:

- ❑ Tamanho máximo: Cada documento no MongoDB pode ter um tamanho máximo de 16 MB
- ❑ Aninhamento de documentos
- ❑ Flexibilidade na evolução do esquema

Tipos de Dados Simples

- String
- Number
- Boolean
- Date
- Null
- ObjectId

Tipos de Dados Complexas

- Array
- Documento Embutido (Embedded Document)
- Referência (Reference)
- GeoJSON

Estrutura de um documento

```
{  
  _id: ObjectId(""),  
  "nome_campo": "valor_campo",  
  ...  
}
```

Modelagem da estrutura do Usuário e Destinos

<https://jsonformatter.curiousconcept.com/>

Links Úteis

- **Referências:**

- <https://www.mongodb.com/docs/manual/reference/bson-types/>
- <https://www.mongodb.com/docs/manual/reference/geojson/#std-label-geojson-point>

Estratégias de modelagem de dados eficientes e escaláveis

Introdução aos Bancos de Dados Não Relacionais

Modelagem orientada por consultas

- ❑ A modelagem de dados no MongoDB deve ser orientada pelas consultas que serão realizadas com mais frequência

Inner Documents

No MongoDB, é comum **denormalizar** os dados para evitar operações de junção (join) custosas. Isso significa que os dados relacionados podem ser armazenados juntos em um único documento, em vez de serem distribuídos em várias coleções.

Inner Documents

No MongoDB, é comum **denormalizar** os dados para evitar operações de junção (join) custosas. Isso significa que os dados relacionados podem ser armazenados juntos em um único documento, em vez de serem distribuídos em várias coleções.

Modelar usuário com estratégia desnormalizada

<https://jsonformatter.curiousconcept.com/>

Quando usar

- Os dados aninhados são específicos para o documento pai.
- Os dados aninhados são sempre acessados juntamente com o documento pai.
- A cardinalidade do relacionamento é um-para-muitos (um usuário pode ter várias reservas).

Quando NÃO usar

- Se os dados aninhados precisarem ser consultados e atualizados independentemente do documento pai, é mais adequado utilizar coleções separadas.

Referências

- ❑ Forma de relacionar os documentos entre si.

Modelar usuário com estratégia de referência

<https://jsonformatter.curiousconcept.com/>

Quando usar

- Os dados têm seu próprio significado e podem ser acessados independentemente do documento pai.
- Os dados têm uma cardinalidade mais alta (por exemplo, vários usuários podem ter reservas).

Quando NÃO usar

- Se os dados aninhados precisarem ser consultados e atualizados independentemente do documento pai, é mais adequado utilizar coleções separadas.

Links Úteis

- **Referências:**
 - <https://www.luiztools.com.br/post/padroes-para-modelagem-de-dados-documentos-em-mongodb/>

Operações no MongoDB

Introdução aos Bancos de Dados Não Relacionais

Instalação do Compass

- ❑ <https://www.mongodb.com/docs/compass/master/install/>

Criando um DataBase

```
use {{nome_do_banco}}
```

Enquanto o database não tiver uma collection ele não será apresentado na lista

Criando uma collection

```
db.createCollection("usuarios")
```

```
db.createCollection("destinos")
```

Inserindo Documentos

```
db.usuarios.insertOne({ });
db.usuarios.insertMany([{ }]);
```

Consultando Documentos

- db.usuarios.find({})
- db.usuarios.findOne({});
- db.usuarios.findOneAndUpdate({ }, {});
db.usuarios.findOneAndDelete({});

Atualizando Documentos

- db.usuarios.updateOne()
- db.usuarios.updateMany()
- db.usuarios.replaceOne()

Operadores de Update

- \$inc
- \$push
- \$set
- \$unset
- \$rename

<https://www.mongodb.com/docs/manual/reference/operator/update/>

Excluindo Documentos

- db.usuarios.deleteOne({ });
- db.usuarios.deleteMany({ });

Código

<https://github.com/pamelaborges/dio-db-nosql>

Links Úteis

- **Referências:**

- <https://www.mongodb.com/docs/manual/reference/method/db.collection.find/>
- <https://www.mongodb.com/docs/manual/reference/method/db.collection.findAndModify/>
- <https://www.mongodb.com/docs/manual/reference/method/db.collection.findOne/>
- <https://www.mongodb.com/docs/v6.0/tutorial/insert-documents/>

Consultas simples: Operadores

Introdução aos Bancos de Dados Não Relacionais

Igualdade

Realizar consultas baseadas em um valor específico para um campo.

```
db.usuarios.find({ "endereco.cidade": "São Paulo" })
```

Operadores Lógicos

Realizar consultas baseadas em um valor específico para um campo.

- \$and
- \$or
- \$not

Operadores Comparação

- \$eq: ==
- \$ne: !=
- \$gt: >
- \$gte: >=
- \$lt: <
- \$lte: <=
- \$in: []
- \$nin: []

Projeções

Definir quais campos devem ser retornados em uma consulta.

Ordenação

Ordenar os resultados de uma consulta com base em um ou mais campos.

Limitação

Limitar o número de documentos retornados em uma consulta.

Paginação

```
db.usuarios.find().skip(10).limit(5)
```

Código

<https://github.com/pamelaborges/dio-db-nosql>

Links Úteis

- **Referências:**

- <https://www.mongodb.com/docs/manual/reference/operator/query/>

Breve apresentação do Redis

Introdução aos Bancos de Dados Não Relacionais

O que é o Redis?

O Redis é um sistema de armazenamento de dados em memória de alto desempenho.

Principais Características do Redis

- Armazenamento em Memória
- Estrutura de Dados Versátil
- Operações Atômicas
- Cache de Alto Desempenho
- Pub/Sub (Publicação/Aassinatura)

Principais Utilizações do Redis

- Cache de Dados
- Filas de Mensagens
- Contagem de Acessos e Estatísticas em Tempo Real
- Gerenciamento de Sessões
- Cache de Resultados de Consultas

Principais Comandos

- SET
- GET
- DEL
- EXISTS
- KEYS
- INCR
- DECR

Exemplos

- ❑ <https://try.redis.io/>

Links Úteis

- **Referências:**

- <https://redis.io/docs/>

Dúvidas?

> Fórum/Artigos - <https://web.dio.me/articles>