



# Semantix

## **MongoDB - Básico**

Aula 4

Quem sou eu?

Eu sou Rodrigo Augusto Rebouças.

Engenheiro de dados da Semantix  
Instrutor do Semantix Academy

Você pode me encontrar em:  
[rodrigo.augusto@semantix.com.br](mailto:rodrigo.augusto@semantix.com.br)



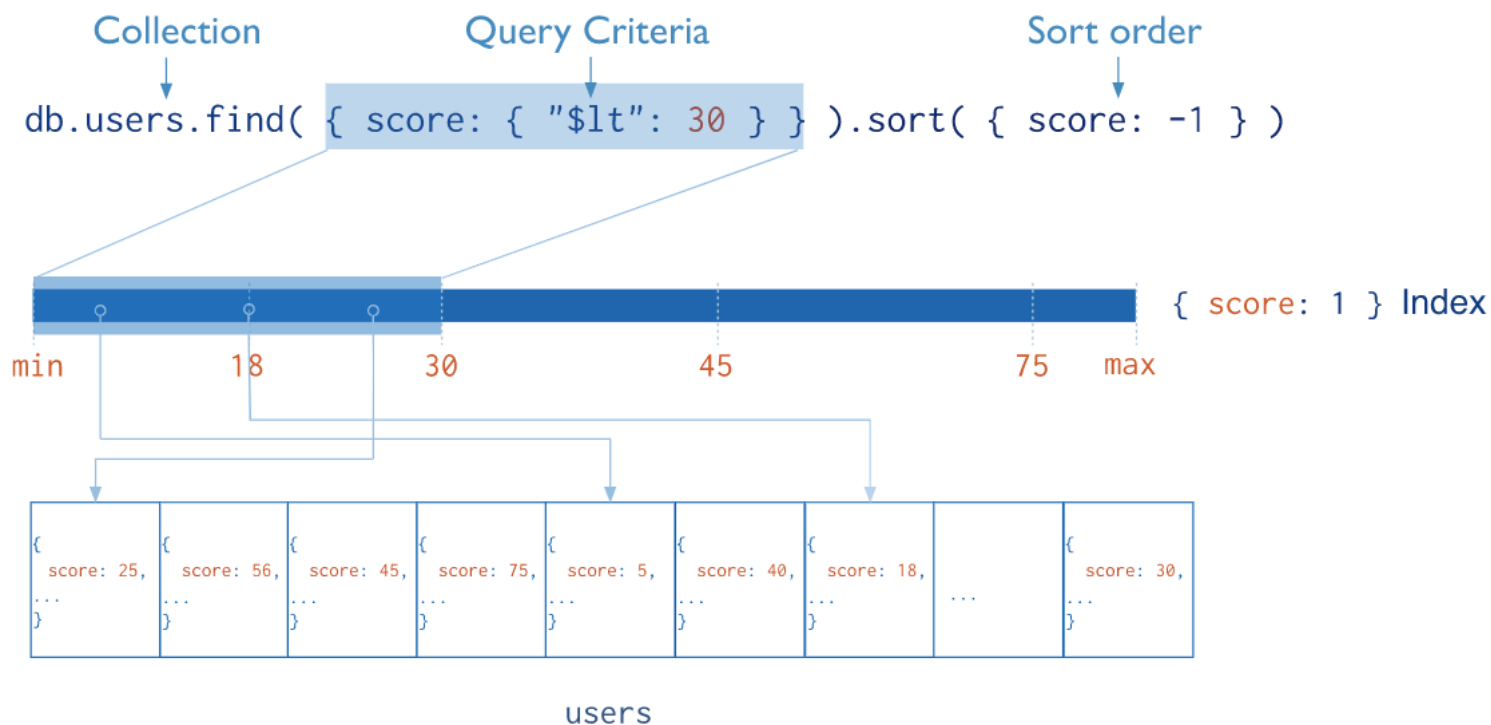


# Índices



# Index MongoDB

- São estruturas de dados especiais
  - Armazena o valor de um atributo específico ou conjunto de atributos, ordenado pelo valor do atributo
  - Estrutura de dados B-tree
- Execução eficiente de consultas no MongoDB





# Index MongoDB

- Index Default
  - MongoDB cria um index exclusivo no atributo **\_id**
  - Recomenda-se sempre usar esse default
    - Aplicação deve garantir a exclusividade dos valores no atributo **\_id** para evitar erros
- Criação de Index
  - Método `createIndex`
    - Irá criar um índice se ainda não existir um índice com a mesma especificação
    - Sintaxe:  

```
db.<nomeCollection>.createIndex( {<key>, <opções>} )
```

      - Key - `<atributo>:<valor>`
        - Valor
          - 1 – Ordenação Ascendente
          - 1 – Ordenação Descendente

# Criação e Visualização - Index

## ○ Exemplos

- Criação de index

```
db.cliente.createIndex({nome: 1})
```

- Visualizar indexes

```
db.cliente.getIndexes()
```

```
> db.cliente.createIndex({nome: 1})
{
  "createdCollectionAutomatically" : false,
  "numIndexesBefore" : 1,
  "numIndexesAfter" : 2,
  "ok" : 1
}
```

```
> db.cliente.getIndexes()
[
  {
    "v" : 2,
    "key" : {
      "_id" : 1
    },
    "name" : "_id_"
  },
  {
    "v" : 2,
    "key" : {
      "nome" : 1
    },
    "name" : "nome_1"
  }
]
```

# Criação e Visualização - Index

## ○ Nome Padrão do Index

- <atributo>\_<valor>\_ <atributo>\_<valor> ...

- Ex.

- db.cliente.createIndex({nome: 1, item: -1})

- Nome do Index: nome\_1\_item:\_-1

- db.cliente.createIndex(  
    {nome: 1, item: -1},  
    {name: "query itens"}  
)

## ○ Index Exclusivo

- Ex.

- db.cliente.createIndex( { user\_id: 1 }, { unique: true } )

- db.cliente.createIndex( { user\_id: 1, nome: 1 }, { unique: true } )

# Remover Index

- Excluir um Index
  - Sintaxe:
    - `db.<nomeCollection>.dropIndex({<key>})`
  - Ex.
    - `db.cliente.dropIndex({nome: 1})`
- Excluir todos os indexes de uma collection
  - Sintaxe:
    - `db.<nomeCollection>.dropIndexes()`
  - Ex.
    - `db.cliente.dropIndexes()`
- Alterar um index
  - Remover e criar novamente





# Consultas com Index



# Método Hint

- Forçar o Otimizador de consultas do MongoDB fazer uso de um índice específico
  - Sintaxe:
    - `db.<nomeCollection>.find({<filtro>}).hint({<key>})`
  - Ex.
    - `db.cliente.find().hint({nome: 1})`
- Como confirmar se a consulta faz uso do índice?
  - Método explain

# Método Explain

- Plano de execução
  - Compreender e otimizar consultas
  - Verificar stage "queryPlanner.winningPlan"
    - COLLSCAN: Collection Scan
    - IXSCAN: Scan com Index keys
    - FETCH : Recuperar documentos
    - SHARD\_MERGE: Junção de shards
    - SHARDING\_FILTER: Filtrar documentos órfãos em shards

- Ex.

- `db.cliente.find().explain()`

```
"winningPlan" : {  
  "stage" : "COLLSCAN",  
  "direction" : "forward"  
},
```

- `db.cliente.find().hint({nome: 1}).explain()`

```
"inputStage" : {  
  "stage" : "IXSCAN",  
  "keyPattern" : {  
    "nome" : 1  
  },  
  "indexName" : "nome_index",  
}
```

# Exercícios - Index

1. Mostrar todos os documentos da collection produto do banco de dados seu nome
2. Criar o index “query\_produto” para pesquisar o campo nome do produto em ordem alfabética
3. Pesquisar todos os índices da collection produto
4. Pesquisar todos os documentos da collection produto
5. Visualizar o plano de execução do exercício 4
6. Pesquisar todos os documentos da collection produto, com uso da index “query\_produto”
7. Visualizar o plano de execução do exercício 7
8. Remover o index “query\_produto”
9. Pesquisar todos os índices da collection produto



## Consultas com Regex



# Método `Regex`

- Aplicar expressão regulares nas consultas
  - Sintaxe:
    - `db.<nomeCollection>.find({ <field>: { $regex: /pattern/, $options: '<options>' } })`
    - `db.<nomeCollection>.find({{ <field>: { $regex: 'pattern', $options: '<options>' } }})`
    - `db.<nomeCollection>.find({{ <field>: { $regex: /pattern/<options> } }})`
- Options
  - `i` – Ignorar case-sensitive
  - `m` – Combinar várias linhas
    - Incluir as ancoras `^` no início e `$` no final



# Método `Regex`

○ Ex.:

- Retornar os clientes com o nome lucas, de forma não case-sensitive
  - `db.cliente.find({nome: {$regex: "lucas", $options: "i"}})`
  - `db.cliente.find({nome: {$regex: /lucas/, $options: "i "}})`
  - `db.cliente.find({nome: {$regex: /lucas/i}})`
- Retornar os clientes da cidade de São Paulo, pesquisando por `sao paulo`
  - `db.cliente.find({nome: {$regex: "s.o paulo", $options: "i"}})`
- Retornar os clientes das cidades que começam com “São”
  - `db.cliente.find({nome: {$regex: "^são", $options: "i "}})`
- Retornar os cpf que contenham letras
  - `db.cliente.find({cpf: {$regex: "[a-z]"}})`

# Exercícios – Consultas com Regex

1. Mostrar todos os documentos da collection produto do banco de dados seu nome
2. Buscar os documentos com o atributo nome, que contenham a palavra “cpu”
3. Buscar os documentos com o atributo nome, que começam por “hd” e apresentar os campos nome e qtd
4. Buscar os documentos com o atributo descricao.armazenamento, que terminam com “GB” ou “gb” e apresentar os campos nome e descricao
5. Buscar os documentos com o atributo nome, que contenha a palavra memória, ignorando a letra “o”
6. Buscar os documentos com o atributo qtd que contenham valores com letras, ao invés de números.
7. Buscar os documentos com o atributo descricao.sistema, que tenha exatamente a palavra “Windows”



# Interface Mongo Express

# Apresentação Prática

## ○ Acesso:

- <http://localhost:8081/>

localhost:8081















Mongo Express Database ▾

Mongo Express

Databases

Database Name

Create Database

 View	admin	 Del
 View	config	 Del
 View	escola	 Del
 View	local	 Del
 View	rodrigo	 Del
 View	semantix	 Del
 View	test	 Del

Server Status

Hostname	30163275a20b	MongoDB Version	4.4.0
Uptime	32109 seconds	Server Time	Thu, 13 Aug 2020 16:45:33 GMT

# Exercícios – Mongo Express

Todas as questões devem ser realizadas através do Mongo Express

1. Criar a collection cliente no banco de dados seu nome
2. Inserir os seguintes documentos:
  - nome: Rodrigo, cidade: São José dos Campos, data\_cadastro: 10/08/2020
  - nome: João, cidade: São Paulo, data\_cadastro: 05/08/2020
3. Renomear a collection para clientes
4. Buscar os documentos da cidade de São Paulo
5. Buscar os documentos da cidade de São Paulo e apresentar apenas o nome e a cidade
6. Atualizar o documento com nome João para cidade: Rio de Janeiro
7. Criar um index para o campo cidade em ordem alfabética
8. Deletar o documento com o nome João
9. Deletar a collection clientes



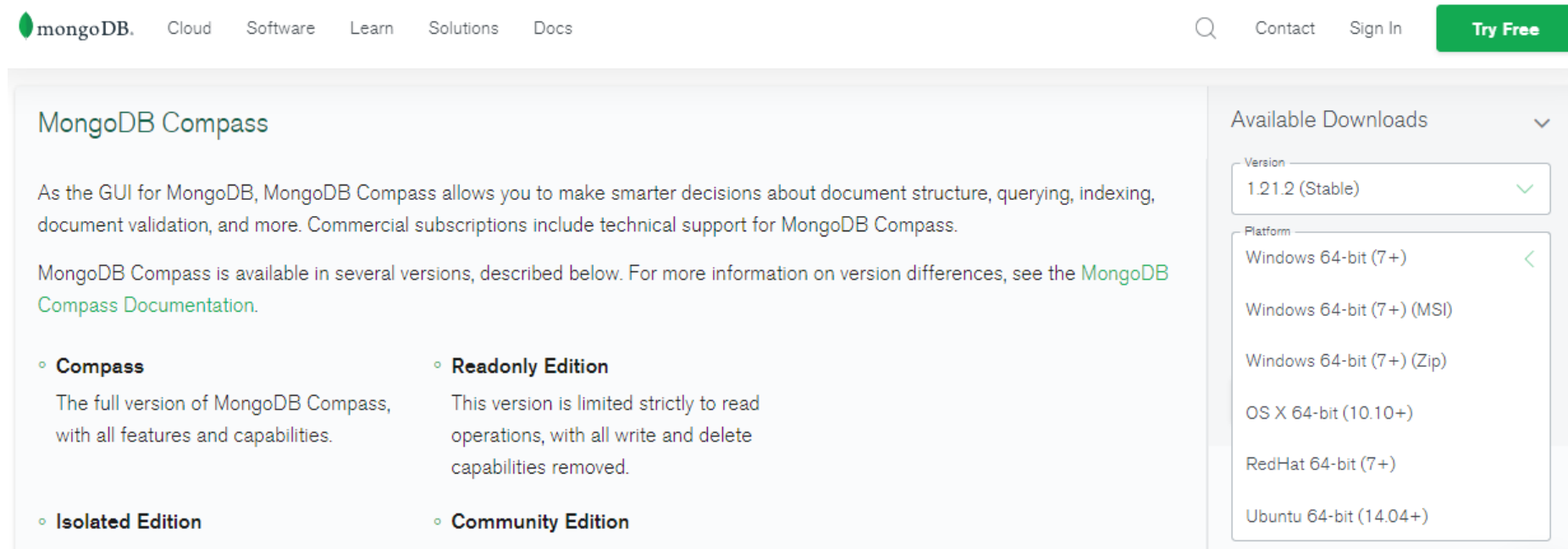


# Instalação MongoDB Compass



# MongoDB Compass

- GUI do MongoDB
  - Conexão
    - Localmente
    - Cluster no Atlas de MongoDB
  - Serviço na Cloud de BD
- Link download  
<https://docs.mongodb.com/compass/current/>



The screenshot shows the MongoDB Compass download page. The header includes the MongoDB logo and navigation links: Cloud, Software, Learn, Solutions, and Docs. On the right, there are links for Contact, Sign In, and a green 'Try Free' button. The main content area is titled 'MongoDB Compass' and describes it as the GUI for MongoDB. It mentions that commercial subscriptions include technical support. Below this, it states that MongoDB Compass is available in several versions and provides a link to the MongoDB Compass Documentation. The page is divided into two columns. The left column lists four editions: Compass, Readonly Edition, Isolated Edition, and Community Edition. The right column, titled 'Available Downloads', shows a dropdown menu for Version (1.21.2 (Stable)) and a dropdown menu for Platform (Windows 64-bit (7+), Windows 64-bit (7+) (MSI), Windows 64-bit (7+) (Zip), OS X 64-bit (10.10+), RedHat 64-bit (7+), Ubuntu 64-bit (14.04+)).

**MongoDB Compass**

As the GUI for MongoDB, MongoDB Compass allows you to make smarter decisions about document structure, querying, indexing, document validation, and more. Commercial subscriptions include technical support for MongoDB Compass.

MongoDB Compass is available in several versions, described below. For more information on version differences, see the [MongoDB Compass Documentation](#).

- **Compass**  
The full version of MongoDB Compass, with all features and capabilities.
- **Readonly Edition**  
This version is limited strictly to read operations, with all write and delete capabilities removed.
- **Isolated Edition**
- **Community Edition**

**Available Downloads**

Version: 1.21.2 (Stable) ✓

Platform: Windows 64-bit (7+) <

Windows 64-bit (7+) (MSI)

Windows 64-bit (7+) (Zip)

OS X 64-bit (10.10+)

RedHat 64-bit (7+)

Ubuntu 64-bit (14.04+)

# MongoDB Compass

## ○ Interface conexão

MongoDB Compass - Connect

Connect View Help

**New Connection**

★ Favorites

🔄 Recents

3 MINUTES AGO  
localhost:27017

### New Connection ☆ FAVORITE

[Paste connection string](#)

Hostname

More Options

Hostname

localhost

Port

27017

SRV Record

☐

Authentication

Username / Password ▼

Username

mongoadmin

Password

.....|

Authentication Database ⓘ

admin

**CONNECT**

# Apresentação Prática

MongoDB Compass - localhost:27017

Connect View Collection Help

Local

8 DBS 10 COLLECTIONS

☆ FAVORITE

HOST  
localhost:27017

CLUSTER  
Standalone

EDITION  
MongoDB 4.4.0 Community

Q Filter your data

- > admin
- > config
- > escola
- > local
- > rodrigo
- > semantix
- > test
- > teste
- ✓ treinamento
- usuários ...

Databases Performance

CREATE DATABASE

Database Name ^	Storage Size	Collections	Indexes	
admin	32.0KB	0	1	
config	36.0KB	0	2	
escola	500.0KB	2	2	
local	36.0KB	1	1	
rodrigo	36.0KB	1	1	
semantix	24.0KB	2	2	
test	92.0KB	3	5	
teste	4.0KB	1	1	

# Exercícios – MongoDB Compass

Todas as questões devem ser realizadas através do MongoDB Compass

1. Criar a collection cliente no banco de dados seu nome
2. Inserir os seguintes documentos:
  - nome: Rodrigo, cidade: São José dos Campos, data\_cadastro: 10/08/2020
  - nome: João, cidade: São Paulo, data\_cadastro: 05/08/2020
3. Renomear a collection para clientes
4. Buscar os documentos da cidade de São Paulo
5. Buscar os documentos da cidade de São Paulo e apresentar apenas o nome e a cidade
6. Atualizar o documento com nome João para cidade: Rio de Janeiro
7. Criar um index para o campo cidade em ordem alfabética
8. Deletar o documento com o nome João
9. Deletar a collection clientes



# Semantix

## Obrigado!

Alguma pergunta?



Você pode me encontrar em:  
[rodrigo.augusto@semantix.com.br](mailto:rodrigo.augusto@semantix.com.br)

**GET SMARTER**