

# Introdução ao mongoDB®

Prof. Antonio Guardado

# O que é

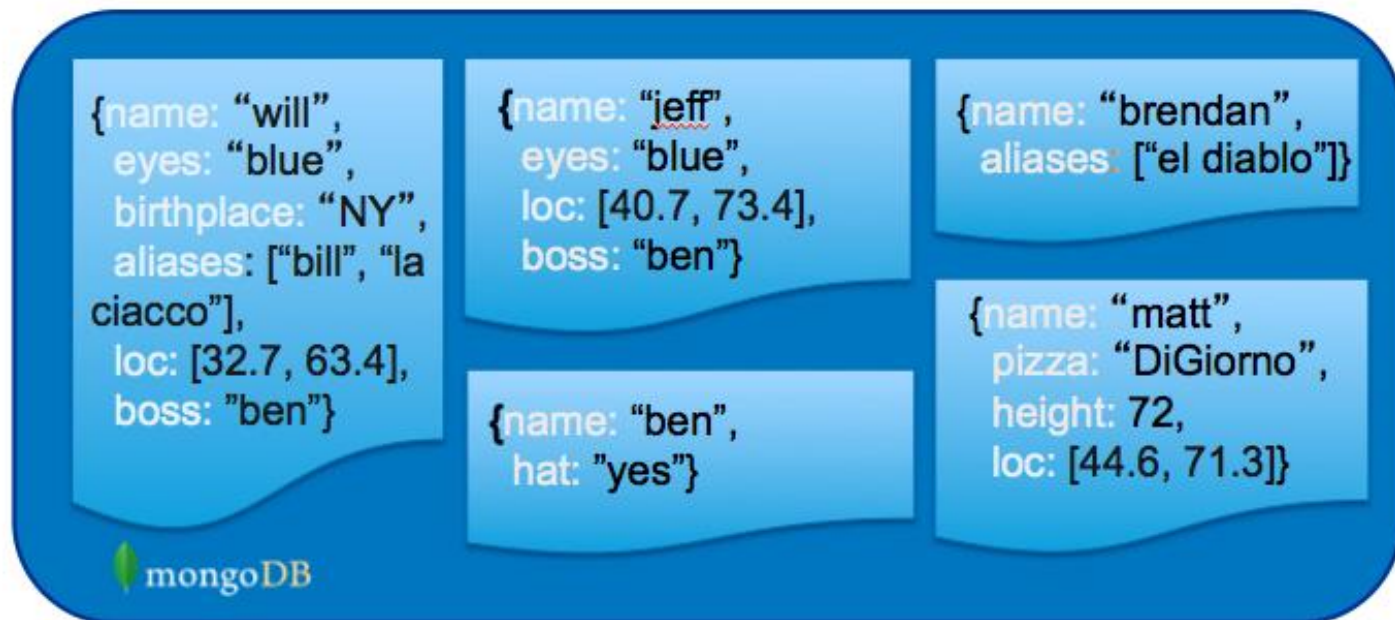
- ▶ O MongoDB é um SGBD NOSQL open-source e orientado a documentos.
- ▶ Alguns de seus diferenciais são:
  - ▶ Alto desempenho: documentos **embutidos** e índices atuando sobre eles;
  - ▶ Rica linguagem de consulta: permite operações **CRUD**, **agregações** de dados, busca por **texto** e consultas **geoespaciais**;
  - ▶ Alta disponibilidade: *replica set*;
  - ▶ Escalabilidade horizontal: **sharding** (fragmentação)

# Origem

- ▶ Nome originário de “Hum**mongous**” (enorme) & Huge Data
- ▶ Escrito em C++, desenvolvido em 2009
- ▶ Criadores : 10Gen antigo Doubleclick , Kevin P. Ryan, Dwight Merriman (CTO e fundador) e Eliot Horowitz
- ▶ Motivação : sérios problemas de escalabilidade em BDs relacionais para aplicativos WEB
- ▶ Documentação
- ▶ <https://docs.mongodb.com/manual/tutorial/>
- ▶ Instalação
- ▶ <https://www.mongodb.com/try/download/community>

# Estrutura

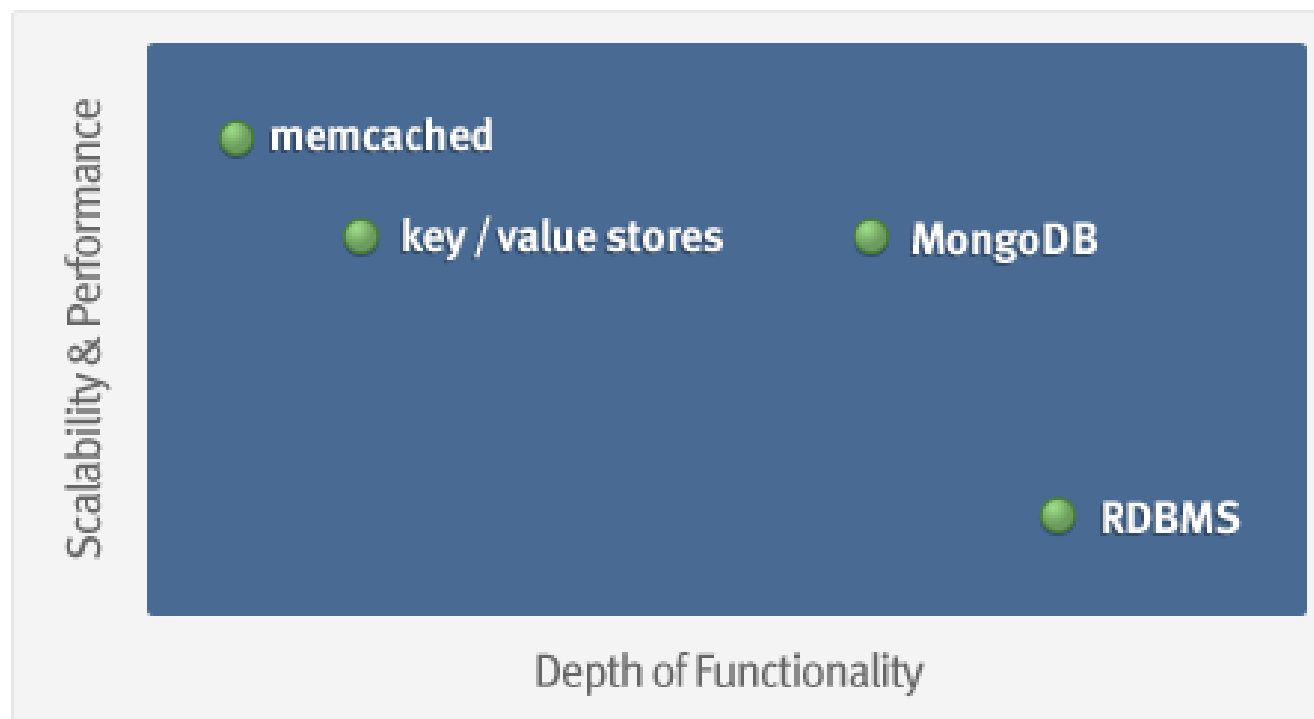
- ▶ Ao invés de armazenar os dados em tabelas e linhas como em um BD Relacional, os dados são armazenados em **documentos baseados em JSON** com esquemas dinâmicos (schema-free ou schemaless) .



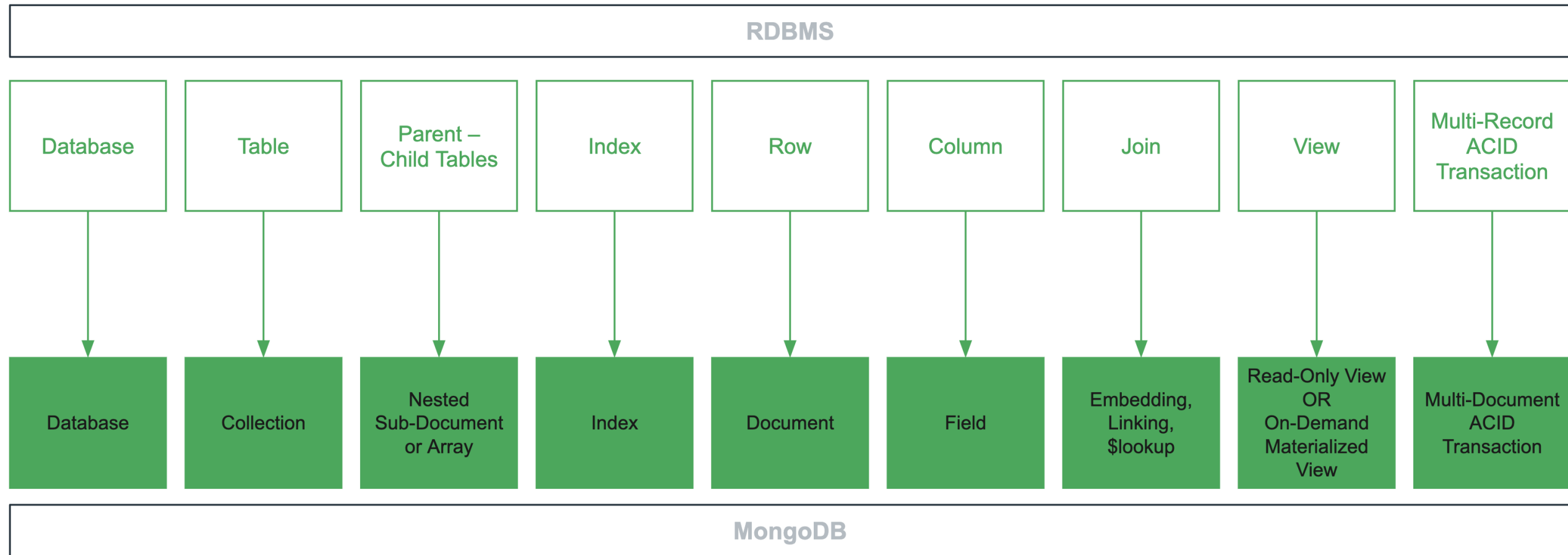
```
> db.user.findOne({age:39})
{
  "_id" : ObjectId("5114e0bd42..."),
  "first" : "John",
  "last" : "Doe",
  "age" : 39,
  "interests" : [
    "Reading",
    "Mountain Biking"
  ],
  "favorites": {
    "color": "Blue",
    "sport": "Soccer"
  }
}
```

# Objetivo

- Ser uma ponte entre BDs chave-valor ( mais rápidos e escaláveis) e BDs relacionais, que possuem funcionalidade mais rica



# Modelo de Dados - Termos comparados com BD Relacional



# Modelo de Dados

MongoDB organiza os dados em coleções de documentos.

- Cada documento possui um atributo identificador (\_id) e uma **quantidade** qualquer de outros atributos.
- Não é necessário (Mas é possível) especificar o ID dos documentos!
- Não é necessário especificar o tipo dos atributos!
- Documentos diferentes que fazem parte de uma mesma coleção podem ter atributos diferentes!

Nome do campo

Valor do campo

Um documento

Vetor de outros objetos JSON

```
{
  "_id": 55,
  "País": "Brasil",
  "Região": "América do Sul",
  "Populacao": 201032714,
  "PrincipaisCidades": [
    {
      "NomeCidade": "São Paulo",
      "Populacao": 1182876,
    },
    {
      "NomeCidade": "Rio de Janeiro",
      "Populacao": 6323037,
    }
  ]
}
```

# Modelo de Dados

Documentos aninhados ou embutidos

```
{
  "ClienteId" : 12,
  "DataPedido": "01/04/2018",
  "PedidoID": 123,
  "ItensPedido":
  [
    {
      "ItensPedidoID": 234,
      "Produto":
      {
        "ProdutoID": 23,
        "DescricaoProduto": "teste",
        "PrecoProduto": 2.00
      },
      "Quantidade": 50,
      "ValorTotal": 100.00
    }
  ]
}
```

Formato BSON (JSON binário) permite tipos de dados como float, integer, binary, char, data, etc.

Offset (h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	
00000000	54	68	69	73	20	77	69	6C	6C	20	62	65	20	61	20	34	This will be a 4
00000010	30	47	42	20	62	79	74	65	20	73	74	72	65	61	6D	21	OGB byte stream!
00000020	64	00	00	00	03	48	65	61	64	65	72	00	4E	00	00	00	d....Header.N...
00000030	03	53	75	62	48	65	61	64	65	72	31	00	21	00	00	00	.SubHeader1.!...
00000040	02	4E	61	6D	65	00	05	00	00	00	42	6F	6E	64	00	10	.Name.....Bond..
00000050	4C	69	63	65	6E	73	65	00	07	00	00	00	00	03	53	75	License.....Su
00000060	62	48	65	61	64	65	72	32	00	10	00	00	00	08	49	73	bHeader2.....Is
00000070	41	63	74	69	76	65	00	01	00	00	0A	50	61	79	6C	6F	Active.....Paylo
00000080	61	64	00	00													ad..

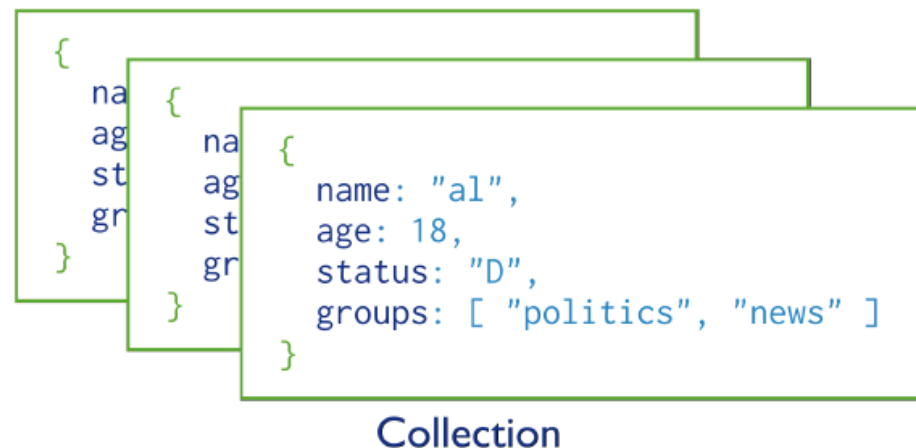


# Modelo de Dados

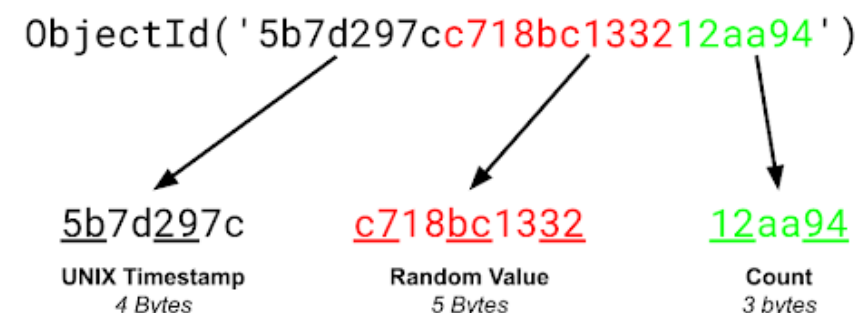
um documento  $\Leftrightarrow$  uma tupla



uma coleção  $\Leftrightarrow$  uma tabela



Coleção é um grupo de documentos similares.  
Dentro de uma coleção cada documento tem um único Id, dado pelo SGBD



Diferente de BDs Relacionais  
não existem restrições de  
integridade no MongoDB

# CRUD

## ► Create

- `db.collection.insert( <document> )`
- `db.collection.save( <document> )`
- `db.collection.update( <query>, <update>, { upsert: true } )`

## ► Read

- `db.collection.find( <query>, <projection> )`
- `db.collection.findOne( <query>, <projection> )`

## ► Update

- `db.collection.update( <query>, <update>, <options> )`

## ► Delete

- `db.collection.remove( <query>, <justOne> )`

# CRUD - Insert

- ▶ Para criar uma coleção, basta inserir um documento nela.
- ▶ Existem duas operações de inserção no MongoDB:
  - ▶ Inserção de um único documento: **insertOne** ou **insert**
    - ▶ Recebe como parâmetro um único documento.
  - ▶ Inserção de múltiplos documentos de uma só vez: **insertMany**
    - ▶ Recebe como parâmetro um vetor de documentos.

```
db.aluno.insertOne( {
  nome: 'Joao da Silva',
  sexo: 'Masculino', fones:
  [991234567, 996789001]
  } )
```

↓

```
db.aluno.insertMany ( [ {primeiro_nome: 'Maria', sobrenome :
  'Cardoso', cpf: 229876543, endereço : { rua: 'Rua Frei Joao',
  número : '59', bairro : 'Ipiranga', cidade : 'São Paulo', UF : 'SP' } },
  {nome_completo : 'Givanildo dos Santos', rg: '98765-X', sexo :
    'Masculino' } ] )
```

↑ vetor

## CRUD - Read (find)

- ▶ O MongoDB possui dois métodos principais para retornar informações de documentos.
  - ▶ O método **find()** retorna um ponteiro para todos os documentos que atendem aos critérios especificados.
  - ▶ O método **findOne()** retorna um único documento que atende aos critérios especificados.
- ▶ Caso exista mais de um documento atendendo aos critérios, o método **findOne()** retorna apenas o primeiro

```
db.aluno.find(
  { cpf:
    {$ne:229876543,
    sexo: 'Masculino'
    } } )
```

```
db.aluno.findOne( { $or:[ { nome_completo: /silva/i } ,
  { nome:/silva/i }, { sobrenome:/silva/i } ] },
  {sexo: {$not: /masc/i}})
```

# CRUD - Update

- ▶ O MongoDB possui três métodos para atualização de dados em um documento.
- ▶ Os métodos **updateOne()** e **updateMany()** localizam o documento segundo os critérios especificados e fazem as alterações descritas.
- ▶ Diferença: **quantidade** de documentos afetada.
  - ▶ Enquanto o **updateOne()** afeta somente um documento que atenda os critérios, o **updateMany()** afeta todos.
- ▶ O método **replaceOne()** localiza um único documento que atenda aos critérios especificados e o substitui por um novo documento.
- ▶ O atributo **\_id** do documento permanece o mesmo.

```
db.aluno.updateOne
({sexo:/masc/i},
  {$set :
    {nacionalidade: 'Brasil'
     } } )
```

```
db.aluno.updateMany
({sexo:/masc/i},
  {$set : {nacionalidade:
    'Brasil' } } )
```

# CRUD - Delete

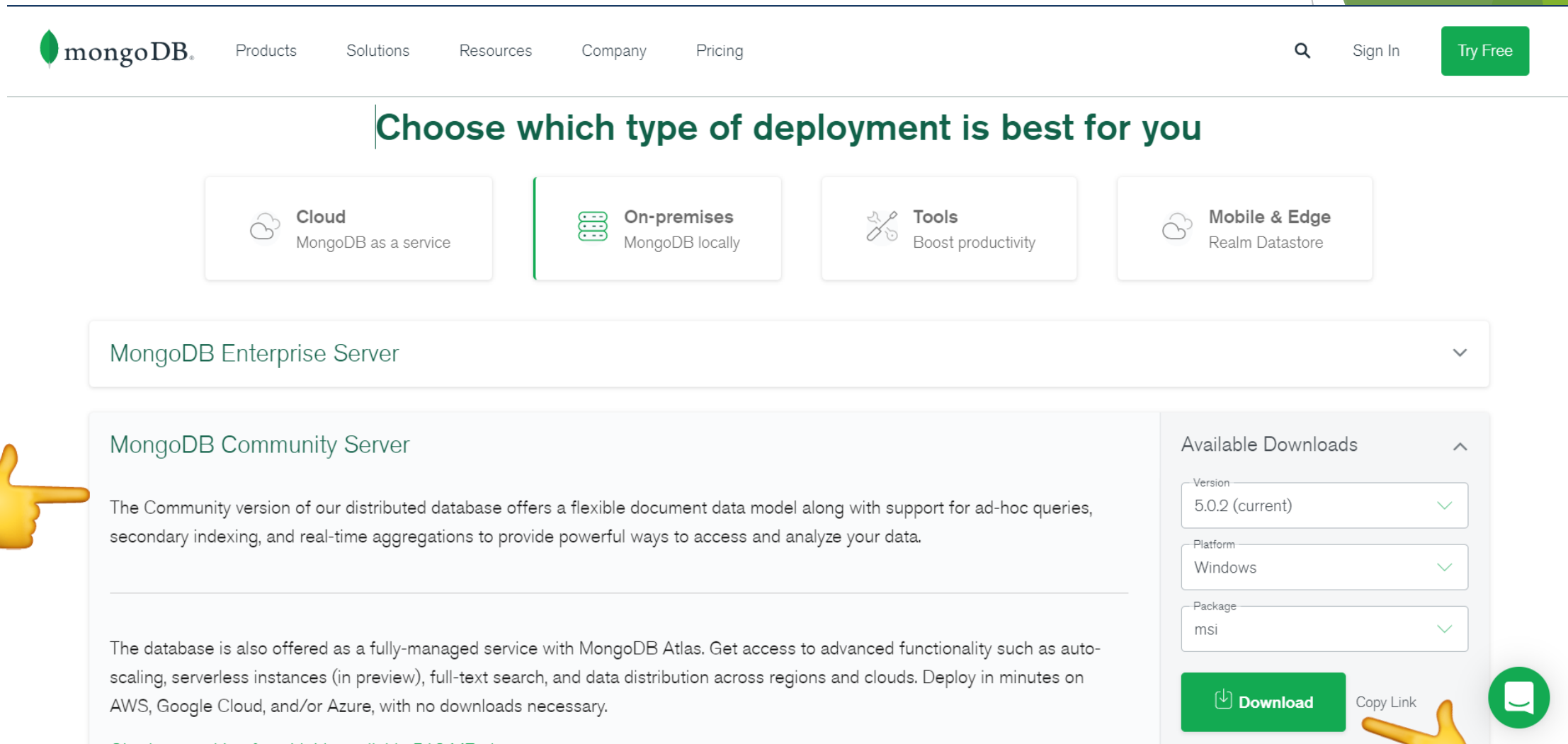
- ▶ O MongoDB possui dois métodos para a remoção de documentos.
- ▶ Os métodos **deleteOne()** e **deleteMany()** localizam o documento segundo os critérios especificados e o removem da base de dados.
- ▶ Diferença: **quantidade** de documentos afetada.
  - ▶ Enquanto o **deleteOne()** afeta somente **um documento** que atenda os critérios, o **deleteMany()** afeta **todos**.

```
db.aluno.deleteOne(  
  {primeiro_nome:  
    'Maria'} );
```

```
db.aluno.deleteMany(  
  {primeiro_nome:  
    'Maria'} );
```

# Instalação

<https://www.mongodb.com/try/download/community>



The screenshot shows the MongoDB website's deployment selection page. At the top, the navigation bar includes the MongoDB logo, links for Products, Solutions, Resources, Company, and Pricing, a search icon, a Sign In link, and a green Try Free button. The main heading is "Choose which type of deployment is best for you". Below this are four cards: Cloud (MongoDB as a service), On-premises (MongoDB locally), Tools (Boost productivity), and Mobile & Edge (Realm Datastore). The On-premises card is highlighted with a green border. Below these cards is a dropdown menu currently showing "MongoDB Enterprise Server". The "MongoDB Community Server" option is selected and highlighted in light blue, with a yellow hand icon pointing to it. The description for the Community Server states it offers a flexible document data model with support for ad-hoc queries, secondary indexing, and real-time aggregations. It also mentions that the database is offered as a fully-managed service with MongoDB Atlas. To the right of the description is a section titled "Available Downloads" with a dropdown arrow. It contains three dropdown menus: Version (5.0.2 (current)), Platform (Windows), and Package (msi). Below these is a green Download button with a download icon, a Copy Link button, and a green chat icon. A yellow hand icon points to the Download button.

**Cloud**  
MongoDB as a service

**On-premises**  
MongoDB locally

**Tools**  
Boost productivity

**Mobile & Edge**  
Realm Datastore

MongoDB Enterprise Server

**MongoDB Community Server**

The Community version of our distributed database offers a flexible document data model along with support for ad-hoc queries, secondary indexing, and real-time aggregations to provide powerful ways to access and analyze your data.

The database is also offered as a fully-managed service with MongoDB Atlas. Get access to advanced functionality such as auto-scaling, serverless instances (in preview), full-text search, and data distribution across regions and clouds. Deploy in minutes on AWS, Google Cloud, and/or Azure, with no downloads necessary.

**Available Downloads**

Version  
5.0.2 (current)

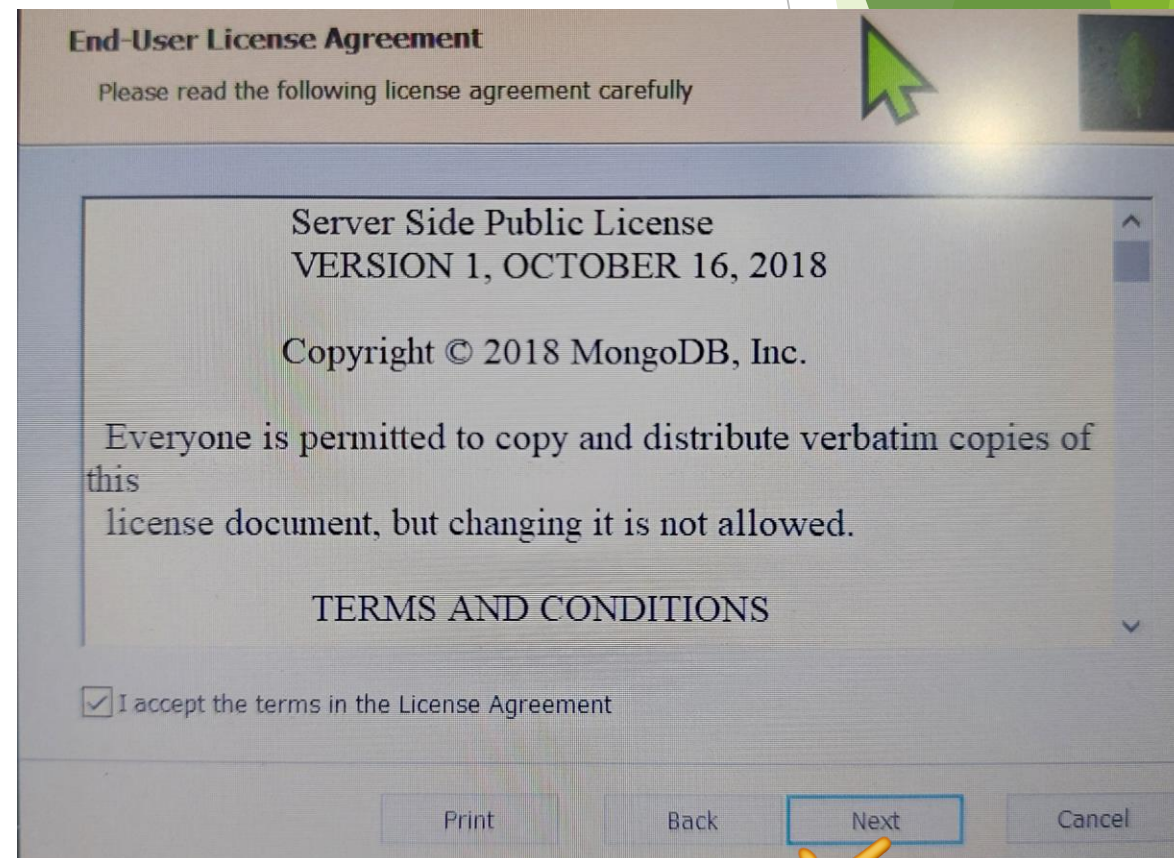
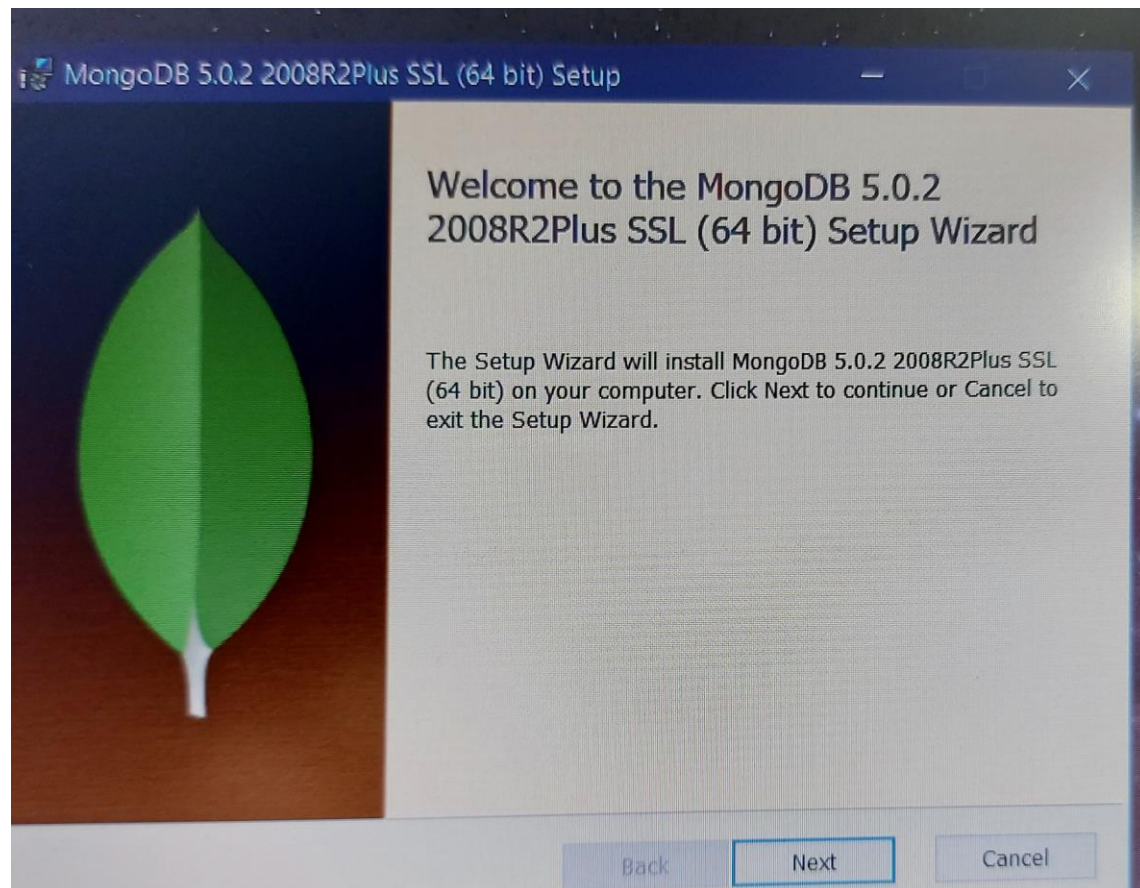
Platform  
Windows

Package  
msi

**Download** Copy Link

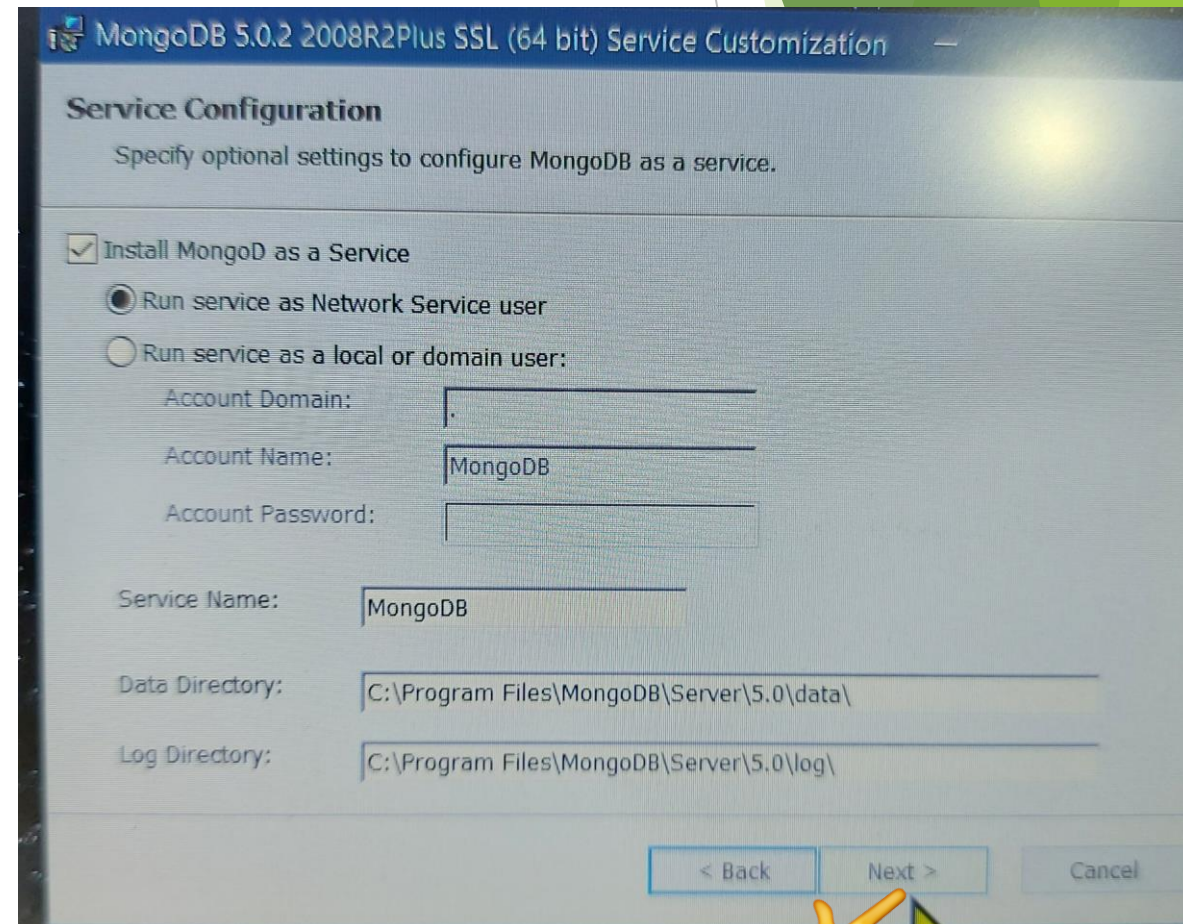
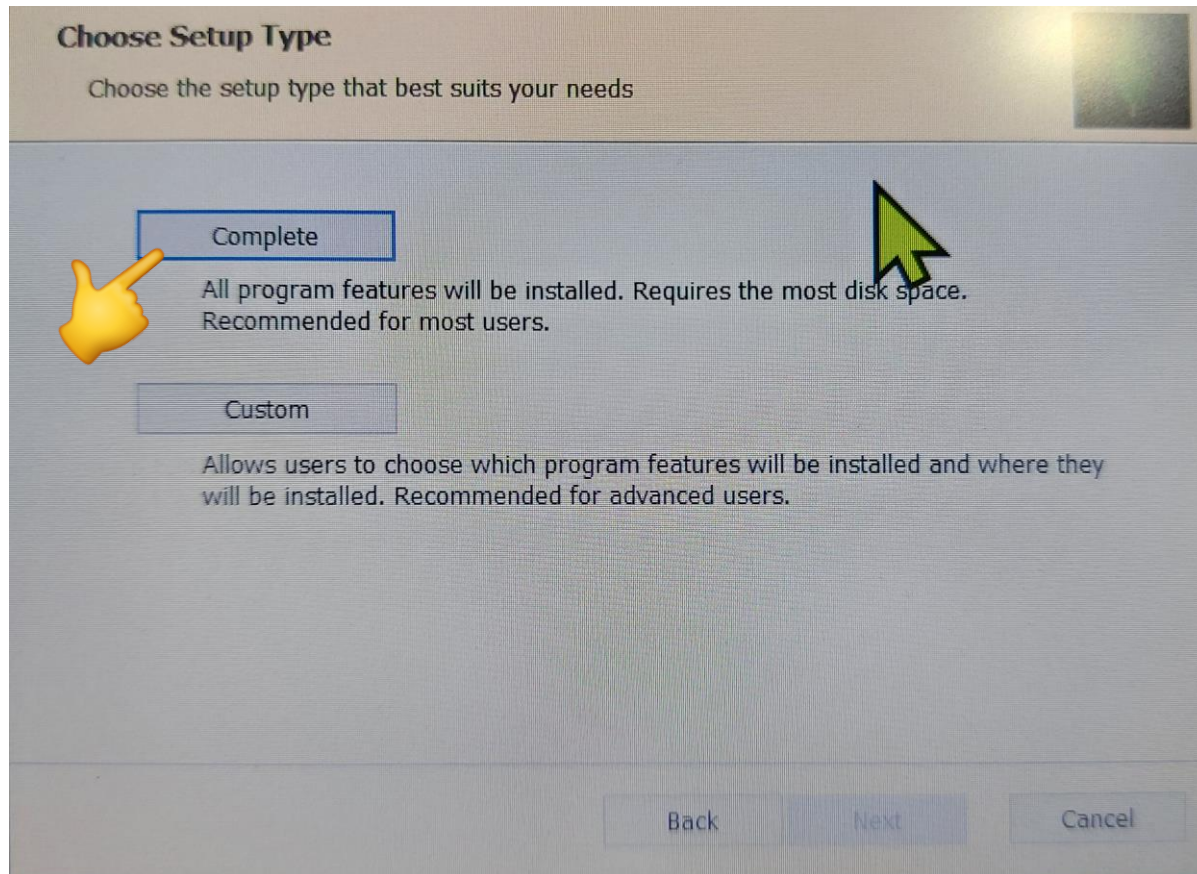


# Instalação

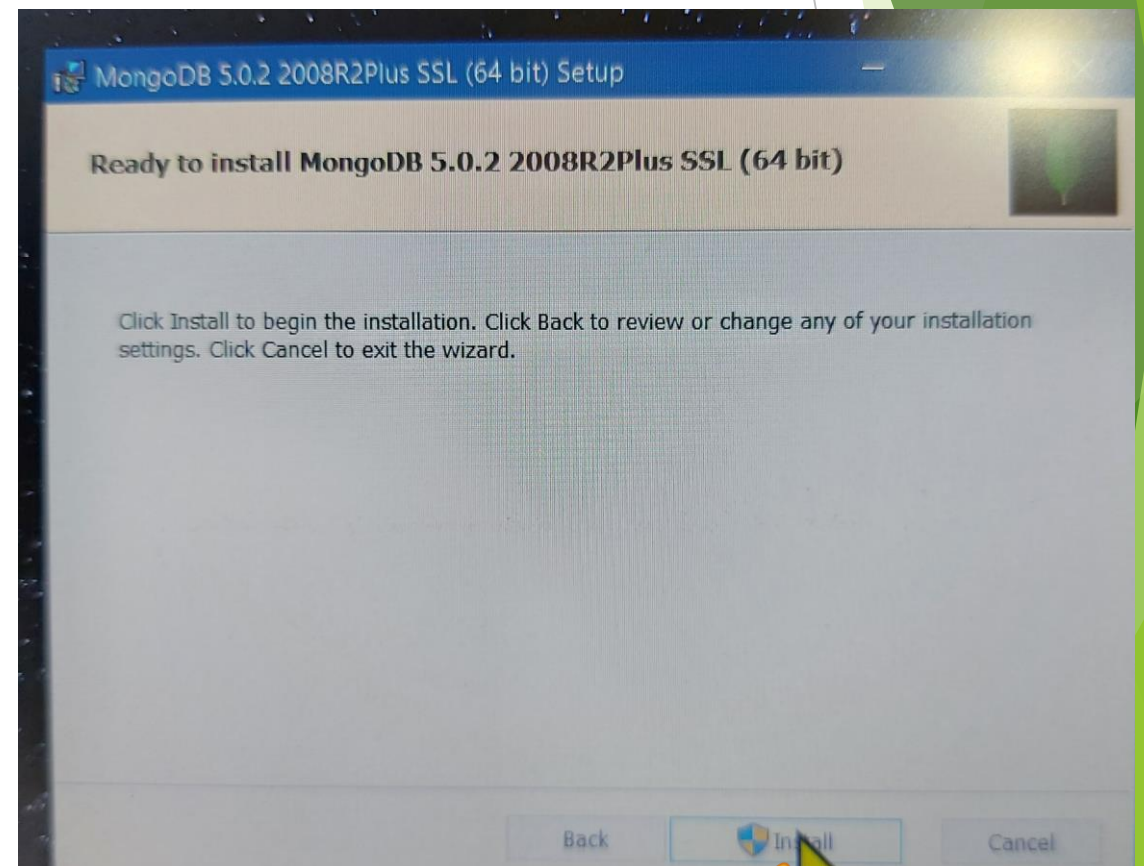
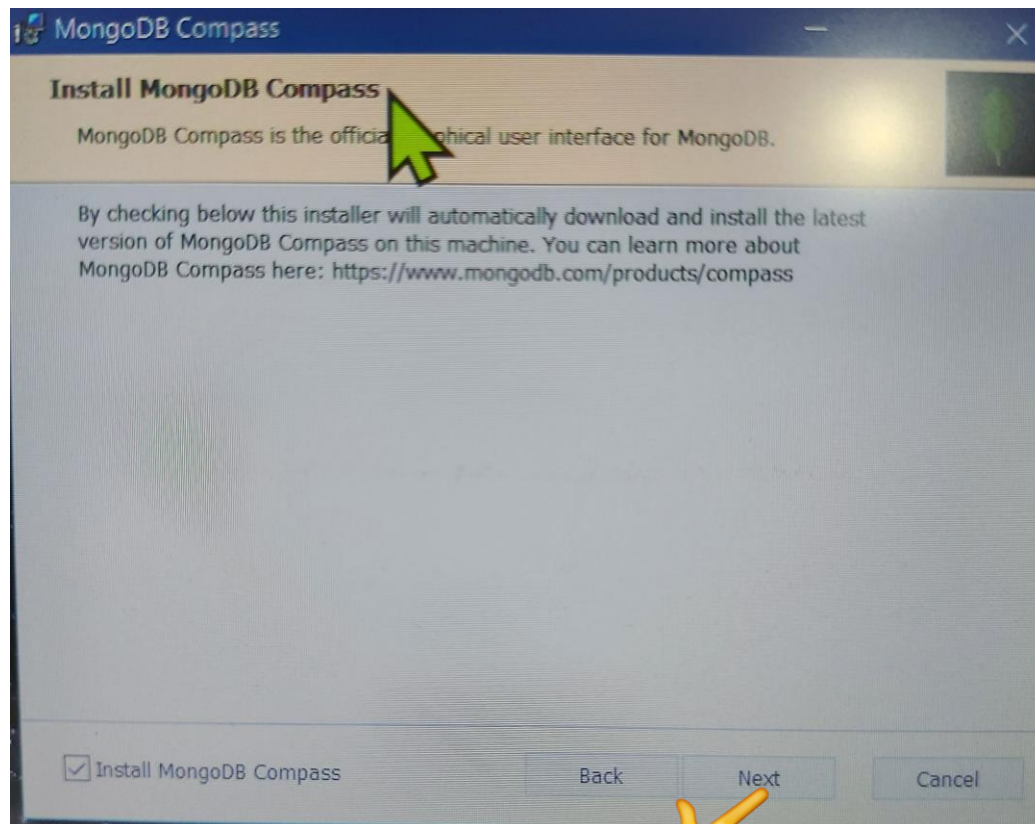




# Instalação



# Instalação



# Instalação

Nosso próximo passo será configurar o ambiente do MongoDB. É necessário criar um diretório de dados para armazenar todos os dados da instância. O caminho padrão usado pelo MongoDB é um diretório com o nome de `\data\db`. Criaremos esse diretório da seguinte forma:

1. Crie o diretório data dentro do diretório raiz C:\
2. Dentro do diretório data criado, crie um diretório de nome db - Você terá então a estrutura de diretórios C:\data\db\
3. Abra um prompt de comandos (cmd.exe)
4. Digite o comando a seguir:

**SET PATH=%PATH%;C:\Program Files\MongoDB\Server\5.0\bin**

5. Digite o comando a seguir:

**mongod C:\data\db\**



# GUI's - Compass

⚡ New Connection

★ Favorites

🕒 Recents

## New Connection

Fill in connection fields individually

Paste your connection string (SRV or Standard ⓘ)

e.g. mongodb+srv://username:password@cluster0-jtpxd.mongodb.net/admin

CONNECT

**New to Compass and don't have a cluster?**

If you don't already have a cluster, you can create one for free using [MongoDB Atlas](#).

CREATE FREE CLUSTER

- **How do I find my connection string in Atlas?**

If you have an Atlas cluster, go to the Cluster view. Click the 'Connect' button for the cluster to which you wish to connect.

[See example](#)
- **How do I format my connection string?**

[See example](#)

**mongodb://localhost:27017**

MongoDB Compass - localhost:27017

Connect View Help

Local

3 DBS 1 COLLECTIONS

☆ FAVORITE

HOST  
localhost:27017

CLUSTER  
Standalone

EDITION  
MongoDB 5.0.2 Community

Filter your data

admin

config

local

Databases Performance

CREATE DATABASE

Database Name	Storage Size	Collections	Indexes
admin	20.0KB	0	1
config	36.0KB	0	2
local	36.0KB	1	1

> \_MONGOSH

Windows taskbar: Digite aqui para pesquisar, File Explorer, Chrome, Excel, Word, PowerPoint, Task Manager, Settings, WhatsApp, Paint, Edge, Teams, MongoDB Compass, System Tray (Network, Volume, Date/Time: POR PTB2 18:21 29/08/2021)

Podemos criar um Database e coleção pelas interfaces

×

## Create Database

**Database Name**

**Collection Name**

☐ **Capped Collection**  
 Fixed-size collections that support high-throughput operations that insert and retrieve documents based on insertion order. ⓘ

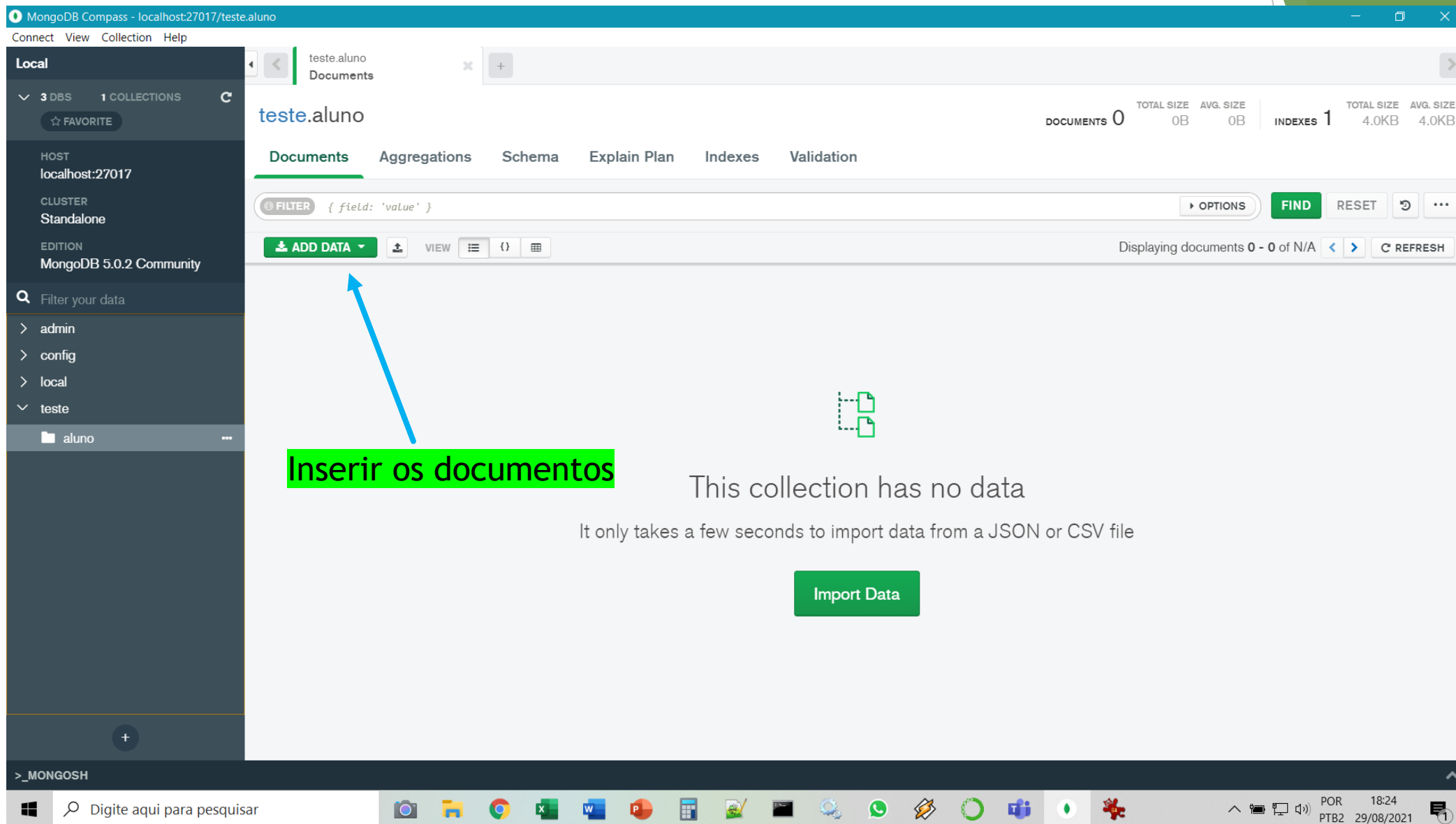
☐ **Use Custom Collation**  
 Collation allows users to specify language-specific rules for string comparison, such as rules for lettercase and accent marks. ⓘ

☐ **Time-Series**  
 Time-series collections efficiently store sequences of measurements over a period of time.

ⓘ Before MongoDB can save your new database, a collection name must also be specified at the time of creation. [More Information](#)

Cancel

Create Database



MongoDB Compass - localhost:27017/teste.aluno

Connect View Collection Help

Local

3 DBS 1 COLLECTIONS

☆ FAVORITE

HOST  
localhost:27017

CLUSTER  
Standalone

EDITION  
MongoDB 5.0.2 Community

Filter your data

- > admin
- > config
- > local
- > teste
  - aluno

teste.aluno Documents

DOCUMENTS 0 TOTAL SIZE 0B AVG. SIZE 0B INDEXES 1 TOTAL SIZE 4.0KB AVG. SIZE 4.0KB

Documents Aggregations Schema Explain Plan Indexes Validation

FILTER { field: 'value' }

ADD DATA

VIEW

Displaying documents 0 - 0 of N/A

REFRESH

Inserir os documentos

This collection has no data

It only takes a few seconds to import data from a JSON or CSV file

Import Data

>\_MONGOSH

Windows taskbar: Digite aqui para pesquisar, various application icons, system tray with date/time (29/08/2021 18:24).

MongoDB Compass - localhost:27017/teste.aluno

Connect View Collection Help

Local

3 DBS 1 COLLECTIONS

☆ FAVORITE

HOST  
localhost:27017

CLUSTER  
Standalone

EDITION  
MongoDB 5.0.2 Community

Filter your data

- > admin
- > config
- > local
- > teste
  - aluno

teste.aluno Documents

DOCUMENTS 0 TOTAL SIZE 0B AVG. SIZE 0B INDEXES 1 TOTAL SIZE 4.0KB AVG. SIZE 4.0KB

Documents Aggregations Schema Explain Plan Indexes Validation

FILTER { field: 'value' }

ADD DATA VIEW

Displaying documents 0 - 0 of N/A

REFRESH

This collection has no data

It only takes a few seconds to import data from a JSON or CSV file

```
>_MONGOSH
test>db.aluno.insertOne ({RA: 123, nome: "João Paulo da Glória"})
```

Ou usar o Shell com comandos de linha