Sumário

[Conceitos 2](#_Toc531000920)

[Tipos de Dados 3](#_Toc531000921)

[Utilização 4](#_Toc531000922)

[Importando dados 5](#_Toc531000923)

[Comandos Gerais 5](#_Toc531000924)

[Seleção – R (read) 6](#_Toc531000925)

[Filtros 6](#_Toc531000926)

[Operadores 7](#_Toc531000927)

[AND 7](#_Toc531000928)

[OR 8](#_Toc531000929)

[IN 9](#_Toc531000930)

[Exercícios 9](#_Toc531000931)

[LIKE 11](#_Toc531000932)

[Exercícios 12](#_Toc531000933)

[$gt e $lt 13](#_Toc531000934)

[Exercícios 14](#_Toc531000935)

[Inserção – C (create) 15](#_Toc531000936)

[Seleção – EXTRA 15](#_Toc531000937)

[Timestamp – ID 16](#_Toc531000938)

[Exercícios 16](#_Toc531000939)

[Atualização – U (update) 17](#_Toc531000940)

[Exercícios 17](#_Toc531000941)

[Deleção – D (delete) 19](#_Toc531000942)

[Exercícios 20](#_Toc531000943)

[Desafio 21](#_Toc531000944)

[Conteúdo adicional 22](#_Toc531000945)

[Instalação 22](#_Toc531000946)

[Autenticação 26](#_Toc531000947)

# Conceitos

O MongoDB é um document database (banco de dados de documentos) com informações no formato JSON. Um documento pode exibir um valor simples, como por exemplo, o nome da pessoa, um número, data ou uma lista de valores.

Esses documentos são agrupados em coleções. E um conjunto de coleções (vamos pensar em diferentes planilhas) forma um database (banco de dados).

Esquema flexível para persistência de dados.

# Tipos de Dados

<https://docs.mongodb.com/manual/core/document/>

<https://data-flair.training/blogs/mongodb-data-types/>

<https://docs.mongodb.com/manual/reference/bson-types/>

# Utilização

<https://stackoverflow.com/questions/4883045/mongodb-difference-between-running-mongo-and-mongod-databases>

Criar pasta *C:/data/db* – por padrão é a que o MongoDB utiliza para rodar o servidor

Ir até a pasta de instalação – *‘C:\Program Files\MongoDB\Server\4.0\bin’*

Executar **mongod.exe**

**mongod** basicamente faz com o que processo inicie.

mongo faz com que você se conecte com uma instância especifica do mongo.

Caso queira direcionar o caminho em que deseja armazenar as suas informações:

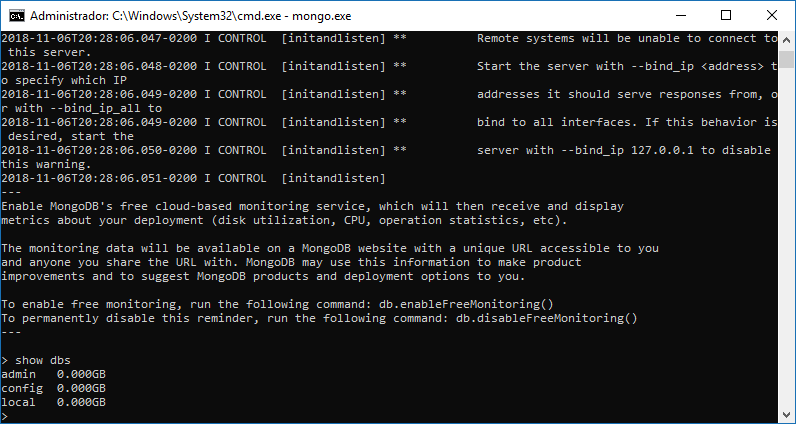
**mongod.exe --dbpath=”C:\Program Files\MongoDB\Server\4.0\data”**

Abra outro CMD/Prompt de Comando e dentro da pasta: ‘C:\Program Files\MongoDB\Server\4.0\bin’ execute:

mongo.exe

Podemos também utilizar: ./mongo localhost:27017/dbname

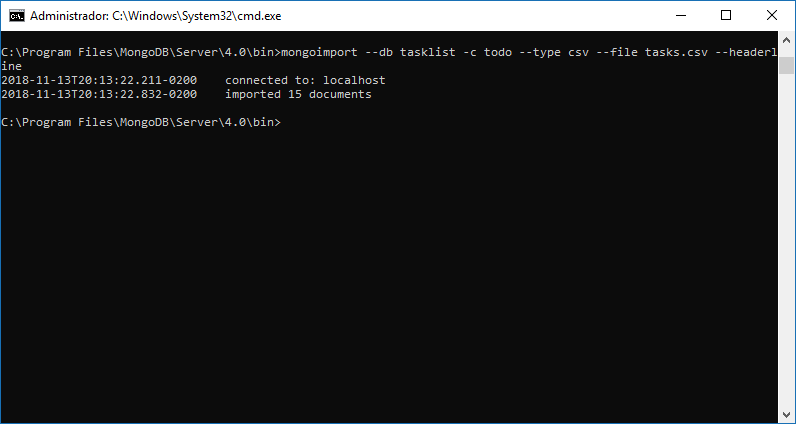
Por padrão, estes são os bancos criados:



# Importando dados

**mongoimport** serve para importar dados de um determinado documento (Excel, JSON, CSV) para o banco de dados do mongodb.

**mongoimport --db tasklist -c todo --type csv --file tasks.csv --headerline**



**use tasklist**

# Comandos Gerais

**show dbs** – mostrar todos os bancos de dados

**show collections** – mostrar todas as coleções

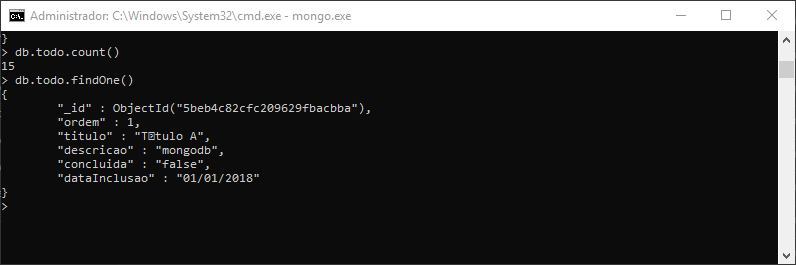
**use tasklist** – colocar o banco de dados em uso

**db.dropDatabase()** – deletar todo o banco de dados

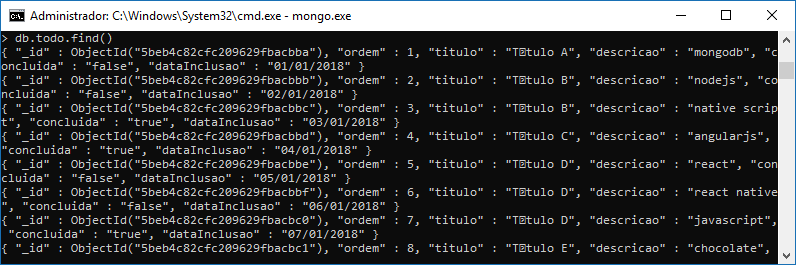
**db.todo.count()** – contar a quantidade de registros inserida

# Seleção – R (read)

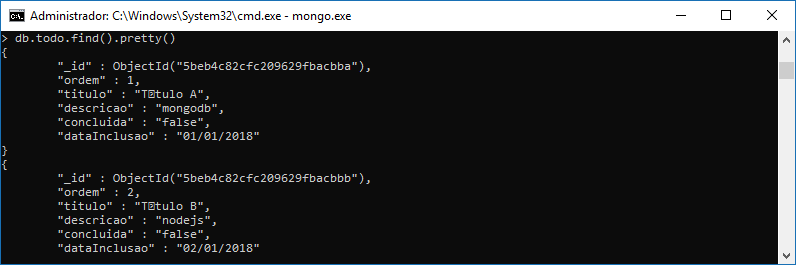
**db.todo.findOne()** – seleciona o primeiro registro



**db.todo.find()** – seleciona todos os registros da coleção selecionada



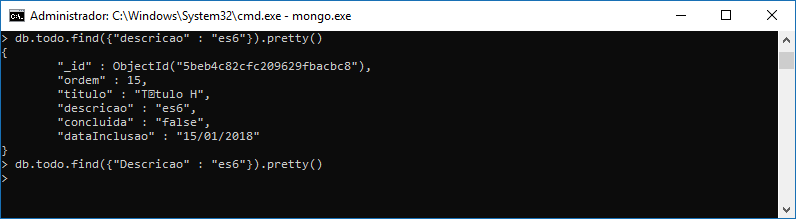
**db.todo.find().pretty()** – busca todos os registros da coleção com uma melhor apresentação



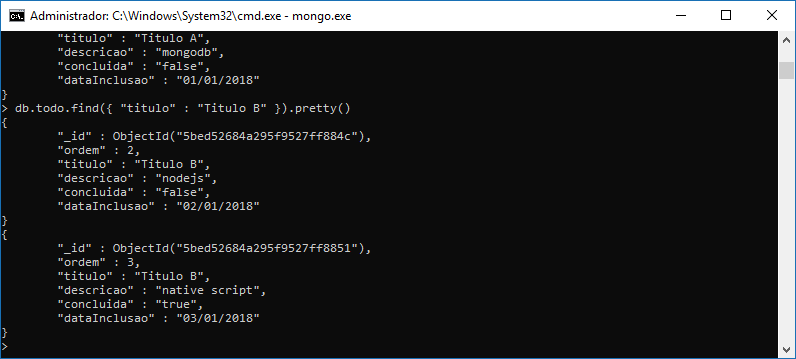
## Filtros

**db.todo.find({ “descricao” : “es6” }).pretty() –** Realizando filtro. Busca apenas a descrição aonde o valor será igual a es6.

Ele é case sensitive. Isso significa que, caso você digite Descricao no valor da chave, o mesmo não irá ser encontrado.

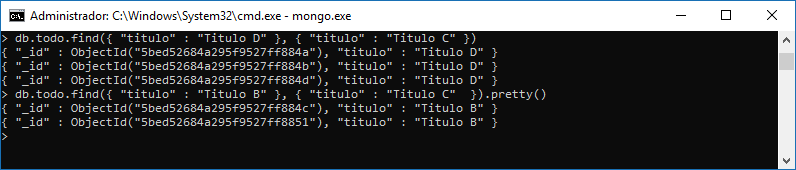


**db.todo.find({ "titulo" : "Titulo B" }).pretty()**



## Operadores

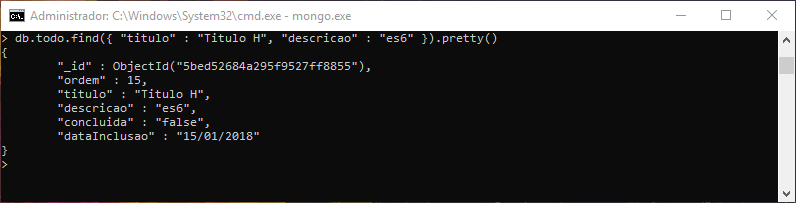
**db.todo.find({ "titulo" : "Titulo B" }, { "titulo" : "Titulo C" }).pretty() –** Quando são colocadas duas condições simultaneamente, ele irá considerar apenas o primeiro registro da lista.



### AND

Trazer todos os registros cujo título é igual a ‘Titulo H’ e descrição é igual a ‘es6’.

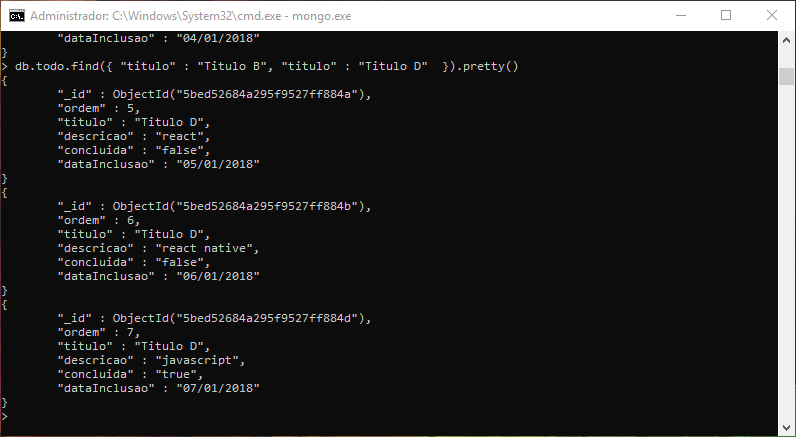
**db.todo.find({ "titulo" : "Titulo H", "descricao" : "es6" }).pretty()**



### OR

Porém, caso quiséssemos trabalhar com a mesma chave correspondente, aonde o título fosse A *ou* B, por exemplo, se utilizássemos a mesma condição, nos traria a lista incorreta.

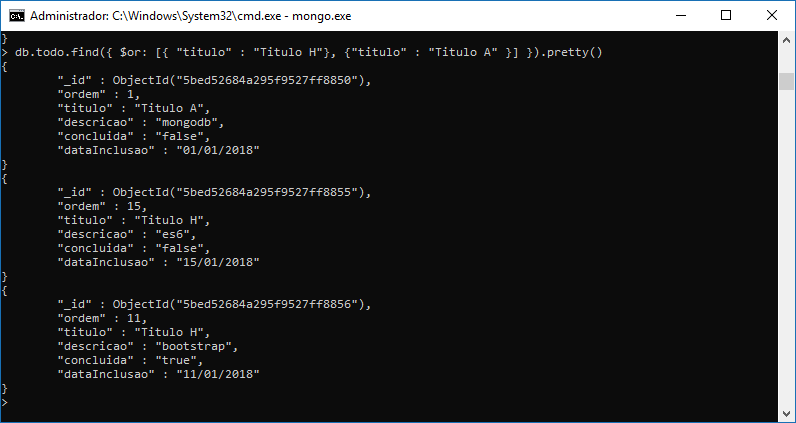
**db.todo.find({ "titulo" : "Titulo B", "titulo" : "Titulo D" }).pretty()**



Ele considera apenas o último registro. Para realizar essa condição, devemos incluir o operador $or. Para nos dizer que queremos um valor *ou* outro.

**db.todo.find({ $or: [{ "titulo" : "Titulo H"}, {"titulo" : "Titulo A" }] }).pretty()**

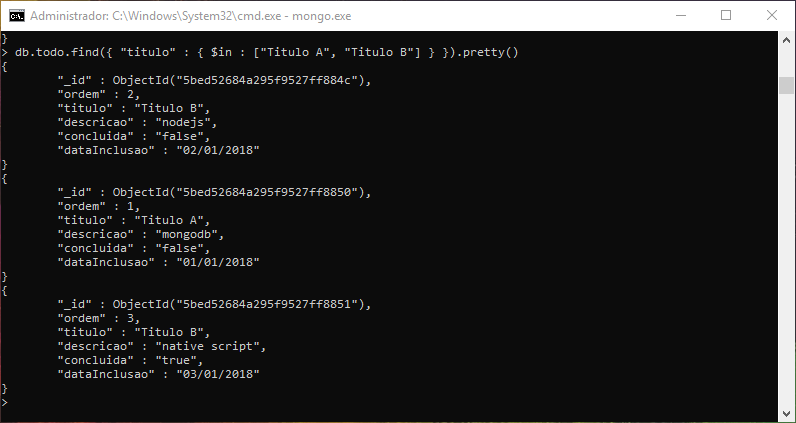
Lembrando: não esquecer de incluir as chaves e colchetes correspondentes.



### IN

Trabalhando com o operador IN, nós podemos colocar dois valores para uma única chave, por exemplo:

**db.todo.find({ "titulo" : { $in : ["Titulo A", "Titulo B"] } }).pretty()**



### Exercícios

1. Selecione todos os registros.

**db.todo.find()**

1. Selecione todos os registros e deixe a com apresentação melhor.

**db.todo.find().pretty()**

1. Selecione todos os registros em que o título seja igual a ‘Titulo A’.

**db.todo.find( { “titulo” : “Titulo A” } ).pretty()**

1. Selecione todos os registros em que o título seja igual a ‘Titulo C’.

**db.todo.find({ “titulo” : “Titulo C” }).pretty()**

1. Selecione todos os registros em que a descrição seja “mongodb”.

**db.todo.find({ “descricao” : “mongodb” })**

1. Selecione todos os registros onde o título seja igual a 'Titulo B' e a descrição 'nodejs'. Após, o Titulo seja 'Título K' e a descrição seja 'html'.

**db.todo.find({"titulo" : "Titulo B" , "descricao" : "nodejs"}).pretty()**

**db.todo.find({ "titulo" : "Titulo K", "descricao" : "html" }).pretty()**

1. Selecione todos os registros em que o título seja igual a 'Titulo B' OU 'Titulo C'.

**db.todo.find({ $or: [{ "titulo" : "Titulo B"} , { "titulo" : "Titulo C" }] }).pretty()**

1. Pegue o exercício acima e inclua também aonde a descrição seja igual 'javascript'.

**db.todo.find({ $or: [{ "titulo" : "Titulo B"} , { "titulo" : "Titulo C" }, {"descricao": "javascript"} ] }).pretty()**

1. Trabalhe com um operador diferente de 'OU' para trazer todos os registros aonde as descrições sejam iguais a 'mongodb', 'css' ou 'jquery'.

**db.todo.find({ $or : [{ "descricao" : "mongodb" }, { "descricao" : "css" }, { "descricao" : "jquery" }] }).pretty()**

**db.todo.find( { "descricao" :{ $in : [ "mongodb", "css", "jquery" ] } } ).pretty()**

Utilizando os dois operadores juntos. $or e $in.

1. Procure todas as tarefas cujo título seja igual a A ou a descrição seja igual a es6 ou bootstrap.

**db.todo.find({**

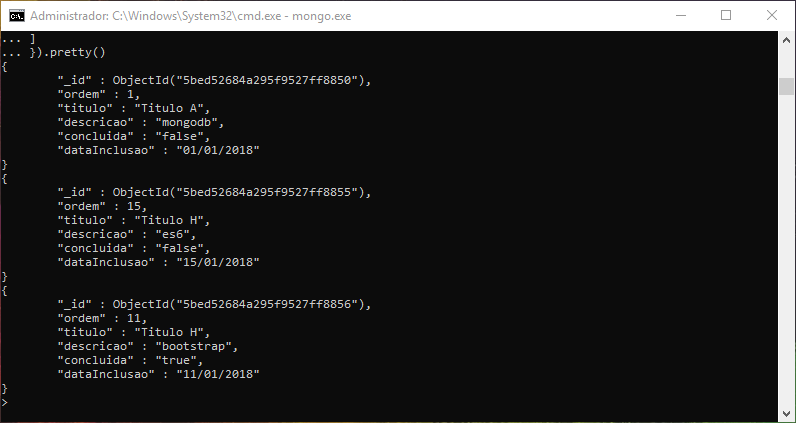
**$or : [**

**{ "titulo" : "Titulo A" },**

**{ "descricao" : { $in : [ "es6", "bootstrap" ] } }**

**]**

**}).pretty()**

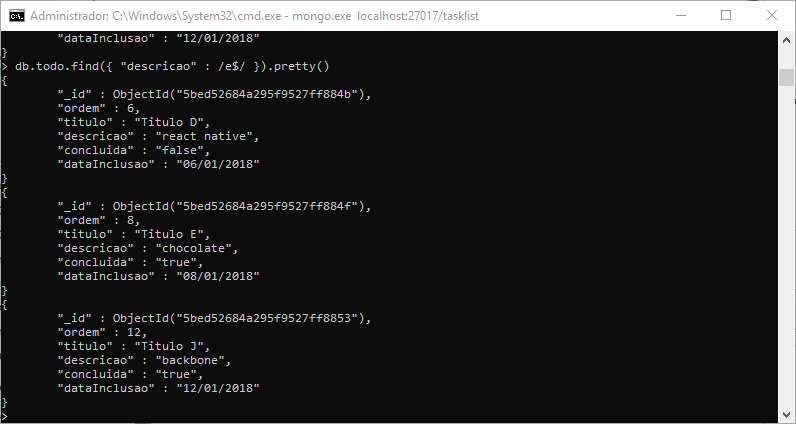


### LIKE

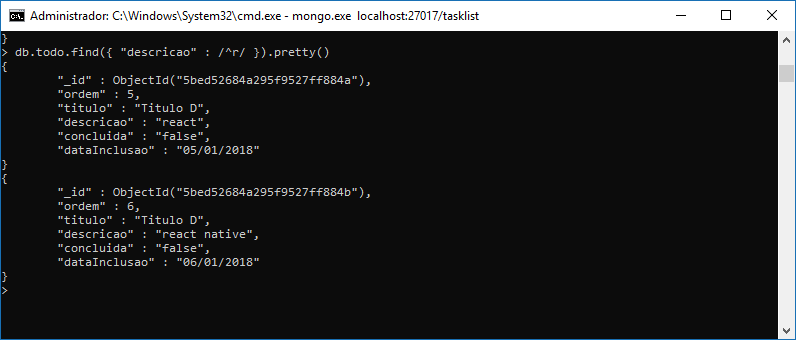
**db.todo.find({ "descricao" : /e/ }).pretty() –** uma descrição que contenha um e em qualquer parte.



**db.todo.find({ "descricao" : /e$/ }).pretty() –** uma descrição que termine com a letra e



**db.todo.find({ "descricao" : /^r/ }).pretty()** – uma descrição que comece com a letra r



### Exercícios

1. Selecione todos os registros em que as descrições contenham em qualquer parte da palavra, 'js'.

**db.todo.find({ "descricao" : /js/ }).pretty()**

1. Selecione todos os registros em que as descrições terminem com a letra 't'.

**db.todo.find({ "descricao" : /t$/ }).pretty()**

1. Selecione todos os registros em que as descrições terminem com a letra 'l'.

**db.todo.find({ "descricao" : /l$/ }).pretty()**

1. Selecione todos os registros em que as descrições terminem com a letra 't' ou 'l'.

**db.todo.find({ "descricao" : {$in: [/l$/, /t$/] }}).pretty()**

**db.todo.find({ $or : [ { "descricao" : /l$/ }, { "descricao" : /t$/ } ] }).pretty()**

1. Selecione todos os registros em que as descrições comecem com a letra 'b'.

**db.todo.find( { "descricao" : /^b/ } ).pretty()**

1. Selecione todos os registros em que as descrições comecem com a letra 'j' ou 'h'.

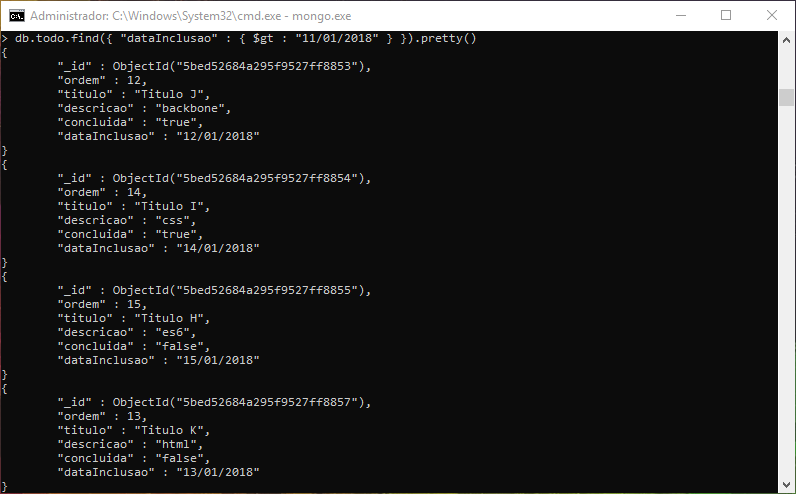
**db.todo.find({ $or : [ { "descricao" : /^j/}, {"descricao" : /^h/ } ] }).pretty()**

**db.todo.find( { "descricao" : { $in : [ /^j/, /^h/ ] } } ).pretty()**

### $gt e $lt

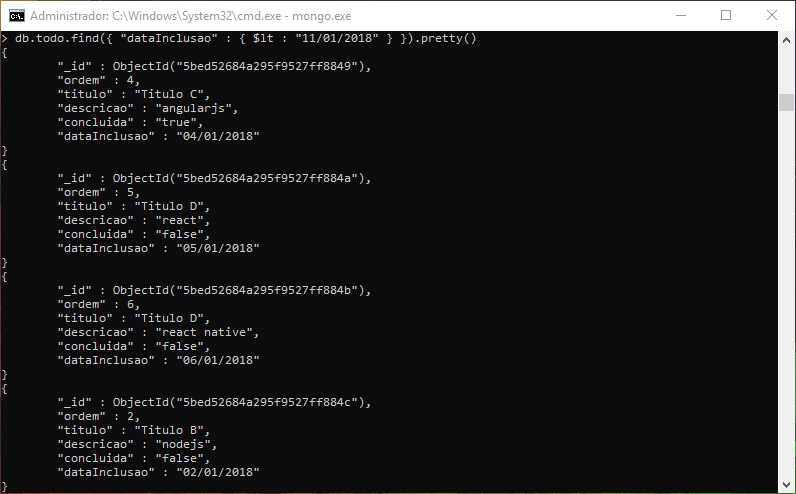
Todas as tarefas cuja data seja superior a 11/01/2018.

**db.todo.find({ "dataInclusao" : { $gt : "11/01/2018" } }).pretty()**



Todas as tarefas cuja data seja inferior a 11/01/2018.

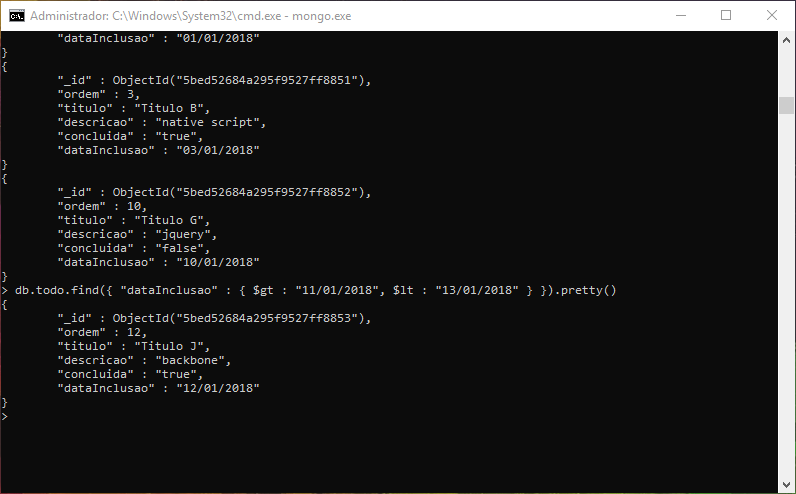
**db.todo.find({ "dataInclusao" : { $lt : "11/01/2018" } }).pretty()**



### Exercícios

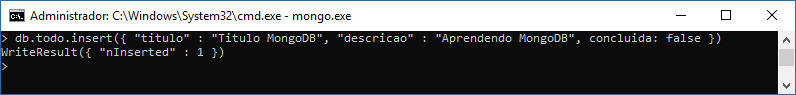
Traga somente os registros aonde o valor da data seja superior a 11/01/2018 e inferior a 13/01/2018.

**db.todo.find({ "dataInclusao" : { $gt : "11/01/2018", $lt : "13/01/2018" } }).pretty()**



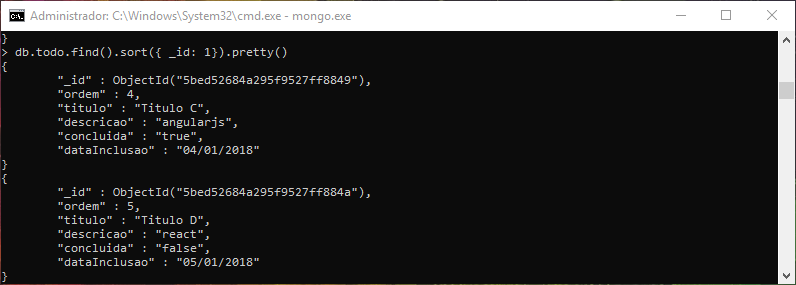
# Inserção – C (create)

**db.todo.insert({ "titulo" : "Titulo MongoDB", "descricao" : "Aprendendo MongoDB", concluida: false })**

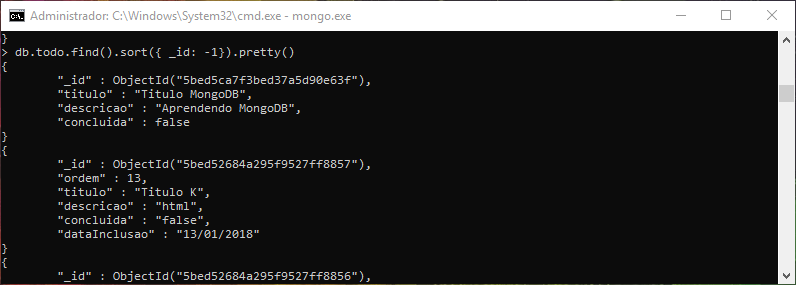


## Seleção – EXTRA

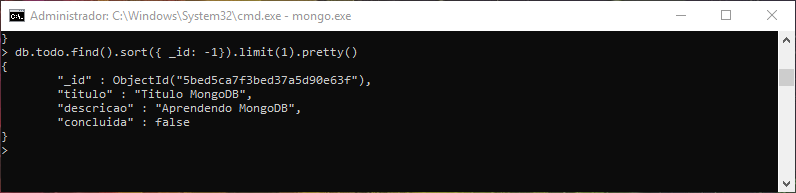
**db.todo.find().sort({ \_id: 1}).pretty()**



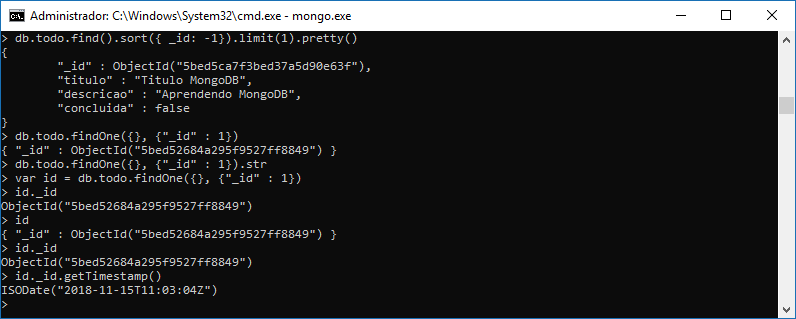
**db.todo.find().sort({ \_id: -1}).pretty()**



**db.todo.find().sort({ \_id: -1}).limit(1).pretty()**



## Timestamp – ID



### Exercícios

1. Insira 5 novos registros conforme descrições abaixo:

Titulo = Titulo U

Descrição = Java

Concluída = verdadeiro

Titulo = Titulo O

Descrição = CSharp

Concluída = falso

Titulo = Titulo P

Descrição = Firebase

Titulo = Titulo Y

Descrição = Microsoft

Titulo = Titulo R

Descrição = Oracle

Data de Inclusão = data de hoje

# Atualização – U (update)

Caso eu faça isso, sem settar os valores que eu quero no campo, o documento inteiro será atualizado com apenas o campo que foi passado.

**db.todo.update(**

**{ "titulo" : "Titulo MongoDB" },**

**{ "descricao" : "Alterando a descricao" }**

**)**

**db.todo.update(**

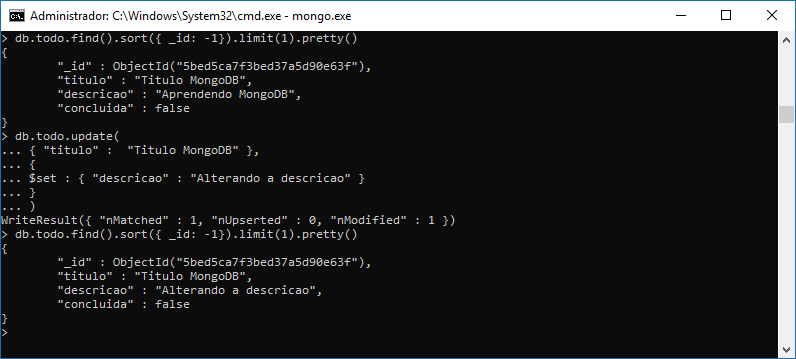
**{ "titulo" : "Titulo MongoDB" },**

**{**

**$set : { "descricao" : "Alterando a descricao" }**

**}**

**)**



### Exercícios

1. Atualize os registros criados no exercício anterior, com os filtros abaixo (não se esqueça de buscar pelo \_id):

Titulo = Titulo Uava

Descrição = Java Best

Concluída = verdadeiro

Titulo = Titulo OSharp

Descrição = CShark

Concluída = falso

Titulo = Titulo Pirebase

Descrição = Fireb4se

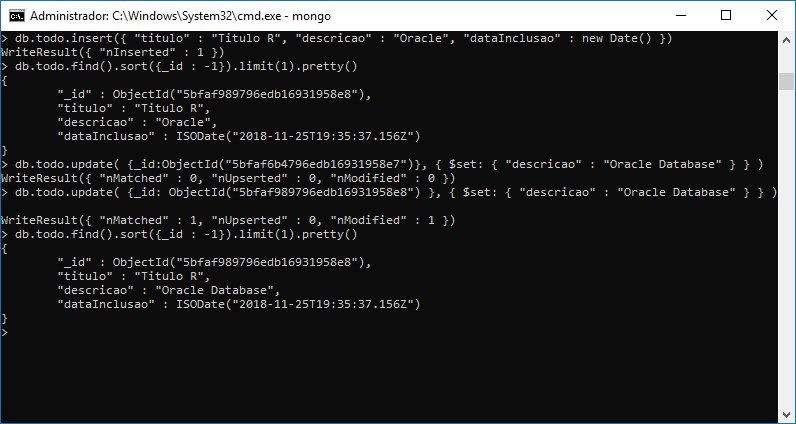
Titulo = Titulo YMicrosoft

Descrição = MicroSOFT

Titulo = Titulo Roracle

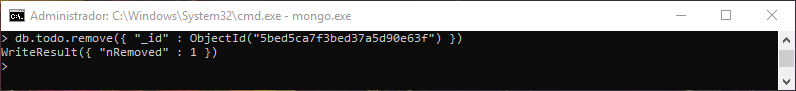
Descrição = Oracle Inc.

Data de Inclusão = data de hoje



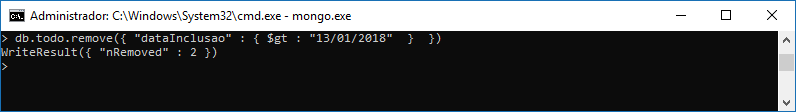
# Deleção – D (delete)

**db.todo.remove({ "\_id" : ObjectId("5bed5ca7f3bed37a5d90e63f") })**



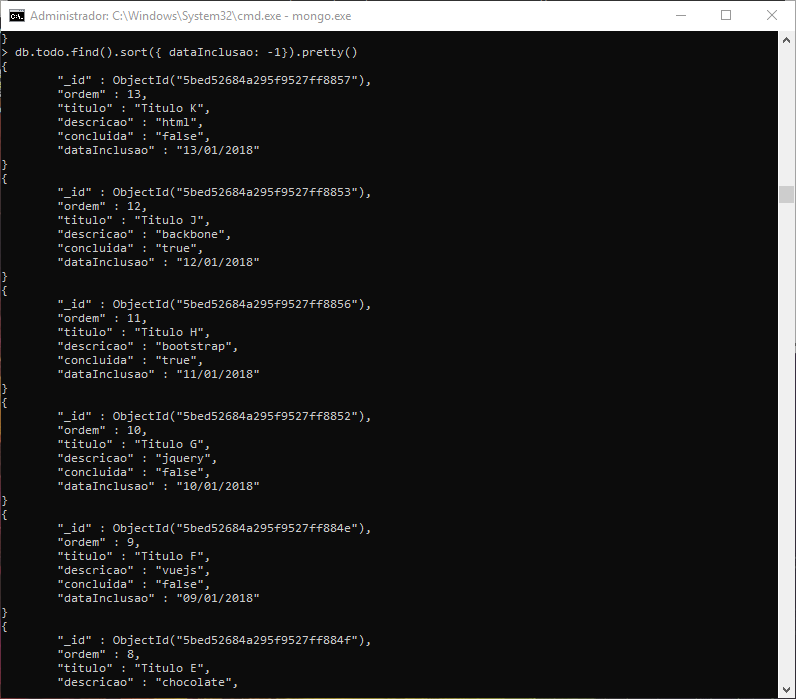
Deletando com filtros

**db.todo.remove({ "dataInclusao" : { $gt : "13/01/2018" } })**



Ordenando por data de inclusão

**db.todo.find().sort({ dataInclusao: -1}).pretty()**



### Exercícios

1. Exclua os últimos 5 itens criados e atualizados nos itens anteriores com as seguintes condições:

Os dois primeiros pelo \_id.

Titulo = Titulo Uava

Descrição = Java Best

Concluída = verdadeiro

Titulo = Titulo OSharp

Descrição = CShark

Concluída = falso

Faça o filtro pelo título no item abaixo.

Titulo = Titulo Pirebase

Descrição = Fireb4se

Faça o filtro pela descrição, nos itens que seguem:

Titulo = Titulo YMicrosoft

Descrição = MicroSOFT

Titulo = Titulo Roracle

Descrição = Oracle Inc.

Data de Inclusão = data de hoje

# Desafio

Você receberá um arquivo com o formato JSON contendo alguns personagens de quadrinhos e/ou desenhos animados e você deverá realizar as alterações que foram solicitadas abaixo pelo cliente.

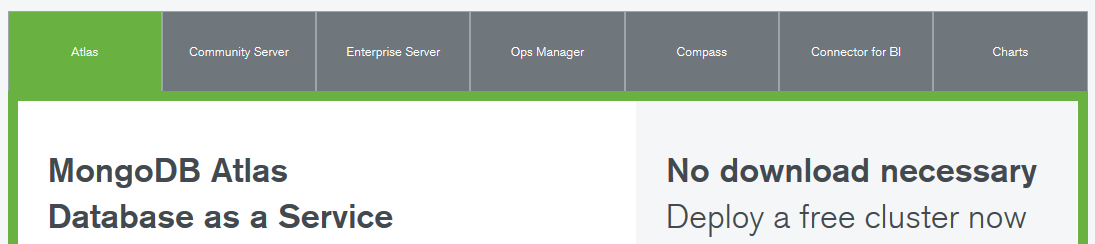
1. Selecione todos os registros.
   1. **db.personagens.find().pretty()**
2. Selecione apenas o primeiro registro.
   1. **db.personagens.findOne()**
   2. **db.personagens.find().sort({\_id:1}).limit(1).pretty()**
3. Selecione todos os registros em que o nome seja igual a ‘Carla’ ou a quantidade de gatos seja igual a 2.
   1. **db.personagens.find( {$or : [ {"Nome" : "Carla" }, {"Qtd Gatos" : 2} ]}).pretty()**
4. Selecione todos os registros em que o local de nascimento seja igual a Roma e necessariamente que a quantidade de cachorros seja igual a 3.
   1. **db.personagens.find( {"Local" : "Roma", "Qtd Cachorros" : 3} ).pretty()**
5. Selecione todos os registros em que a quantidade de gatos seja igual a 3 ou a quantidade de cachorros seja igual a 6.
   1. **db.personagens.find({ $or: [ {"Qtd Gatos" : 3}, { "Qtd Cachorros" : 6 } ] }).pretty()**
6. Selecione todos os registros em que o nome comece com C.
   1. **db.personagens.find({ "Nome" : /^C/ }).pretty()**
7. Selecione todos os registros em que o nome comece com H.
   1. **db.personagens.find({ "Nome" : /^H/ }).pretty()**
8. Selecione todos os registros em que o nome termine com a.
   1. **db.personagens.find({ "Nome" : /a$/ }).pretty()**
9. Selecione todos os registros em que o nome contenha s.
   1. **db.personagens.find({ "Nome" : /s/ }).pretty()**
10. Selecione todos os registros em que o nome contenha ‘e’ ou ‘o’.
    1. **db.personagens.find( { $or : [{ "Nome" : /e/}, { "Nome": /o/ } ]}).pretty()**
11. Selecione todos os registros em que a ordem seja maior do que 10.
    1. **db.personagens.find({ "Ordem" : { $gt : 10 } }).pretty()**
12. Selecione todos os registros em que a quantidade de gatos seja maior do que 2 ou menor do que 6.
    1. **db.personagens.find({ "Qtd Gatos" : { $gt : 2, $lt : 6 } }).pretty()**
13. Insira 3 novos super-heróis.
14. Delete 1 super-herói em que o nome seja igual a Thor.
15. Atualize a quantidade de gatos em que o usuário com o registro de nome ‘Carla’ possui. Atualize a quantidade de gatos de ‘0’ para ‘7’.
16. Selecione todos os registros que foram trabalhados e exporte para um arquivo JSON.

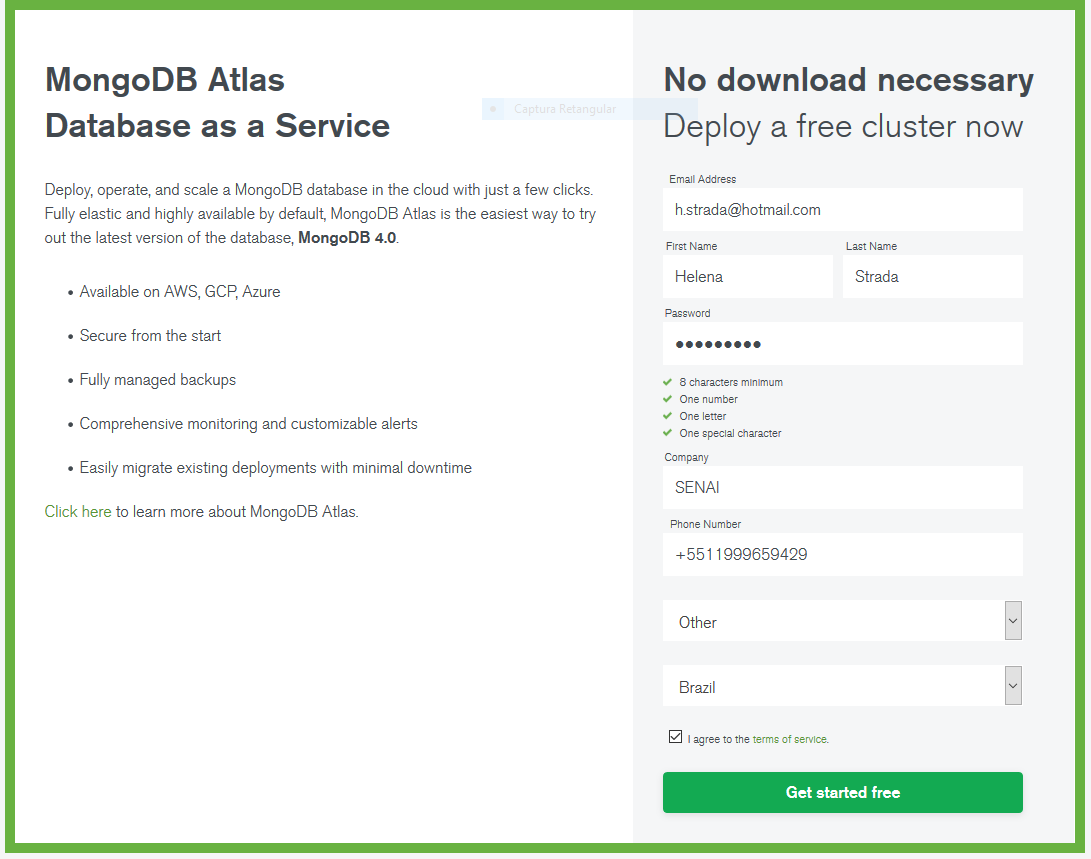
# Conteúdo adicional

## Instalação

MongoDB Atlas

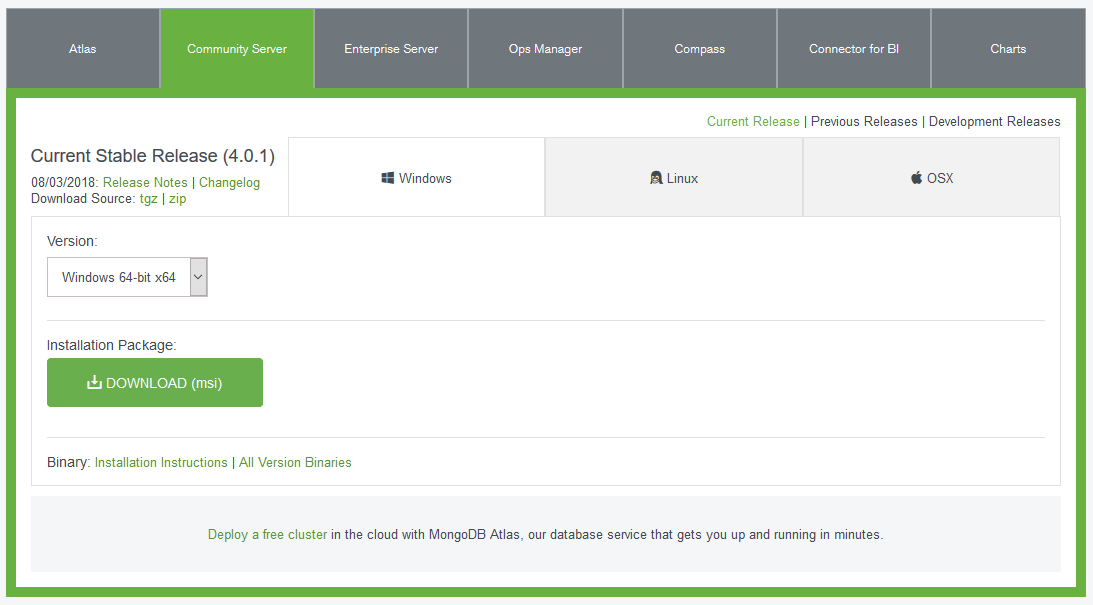
<https://www.mongodb.com/download-center?jmp=nav#atlas>

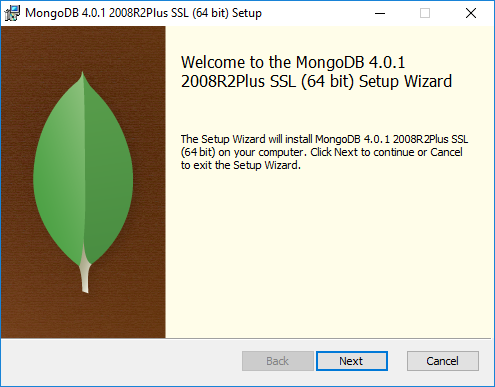


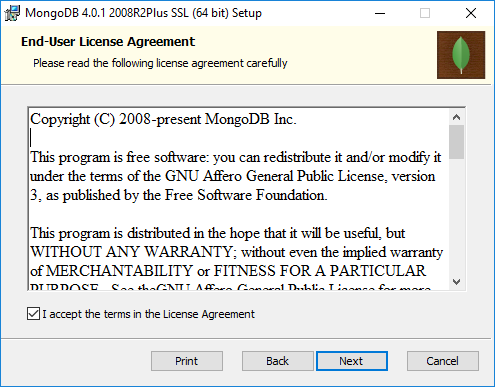


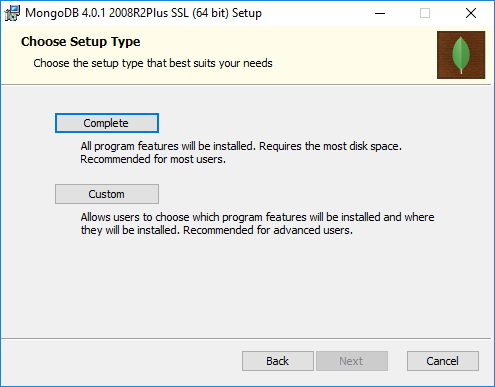
MongoDB Community Server

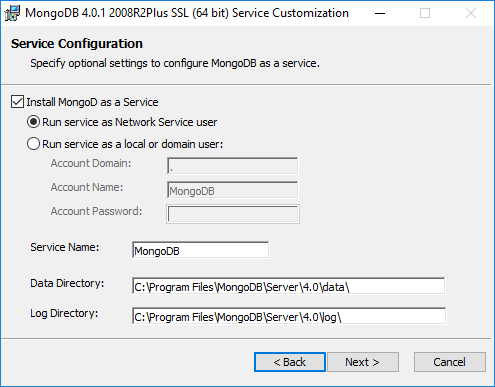
<https://www.mongodb.com/dr/fastdl.mongodb.org/win32/mongodb-win32-x86_64-2008plus-ssl-4.0.1-signed.msi/download>

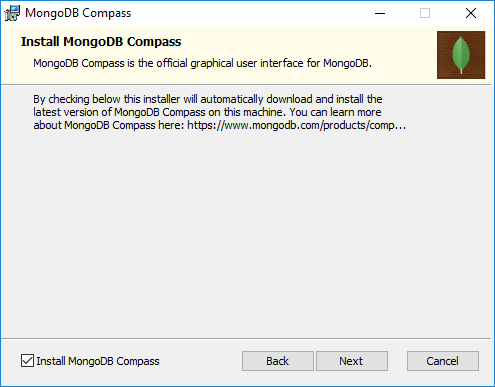


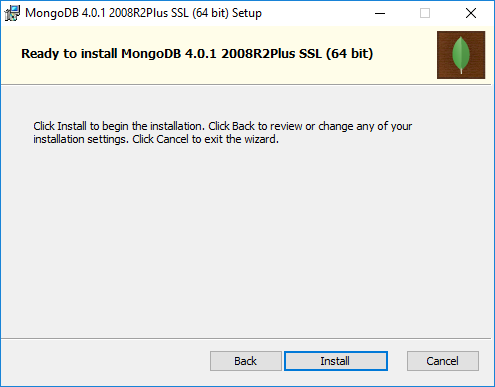


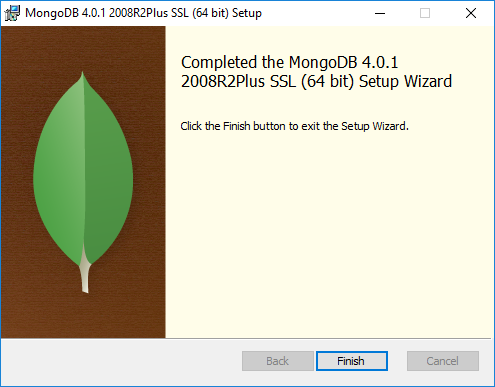












## Autenticação

<https://docs.mongodb.com/manual/tutorial/enable-authentication/>