

MongoDB





01

MongoDB

02

Instalação

SGBD

Os Sistemas de Gerenciamento de Banco de Dados são softwares utilizados para gerir como estrutura de armazenamento dos dados, permite realizar manipulações, bem como controlar como uso de utilização dos bancos de dados. Aqui nesse curso iremos focar no MongoDB para trabalhar com nossos dados!



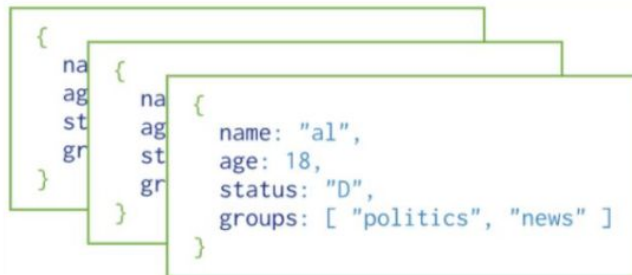
MongoDB

MongoDB é um banco de dados não relacional (noSQL) orientado a documentos no formato JSON. Ele é opensource e possui alta performance e flexibilidade. Ao contrário de um banco de dados relacional (SQL), ele não possui como restrição a necessidade de ter as tabelas e colunas baseadas, permitindo que um documento represente toda a informação necessária, com todos os dados que você queira, no formato de um JSON. Esses documentos são agrupados em coleções , que em conjunto, forma um banco de dados (banco de dados).

MongoDB

```
{  
  name: "al",  
  age: 18,  
  status: "D",  
  groups: [ "politics", "news" ]  
}
```

Documento



Collections (Coleções)



Banco de Dados

Instalação MongoDB

Aqui vamos focar na instalação do MongoDB no Windows, mas caso seja necessário a instalação no Linux ou MacOS, basta seguir os links abaixo:

Linux: <https://docs.mongodb.com/manual/administration/install-on-linux/>

MacOS: <https://docs.mongodb.com/manual/tutorial/install-mongodb-on-os-x/> (É importante lembrar de criar a pasta data/db dentro do diretório /Users/nomeUsuario)

ps: Caso seu MacOS seja de uma versão mais antiga (ex: Sierra), deve-se rodar esses comandos no terminal para instalar:

```
$ brew tap mongodb/brew
```

```
$ brew install mongodb-community@4.2
```

E então usar os comandos abaixo para iniciar e parar o servidor, respectivamente:

```
$ brew services start mongodb-community@4.2
```

```
$ brew services stop mongodb-community@4.2
```

Em versões Mais Atuais fazer mac, Podemos subir o Servidor utilizando o Comando:

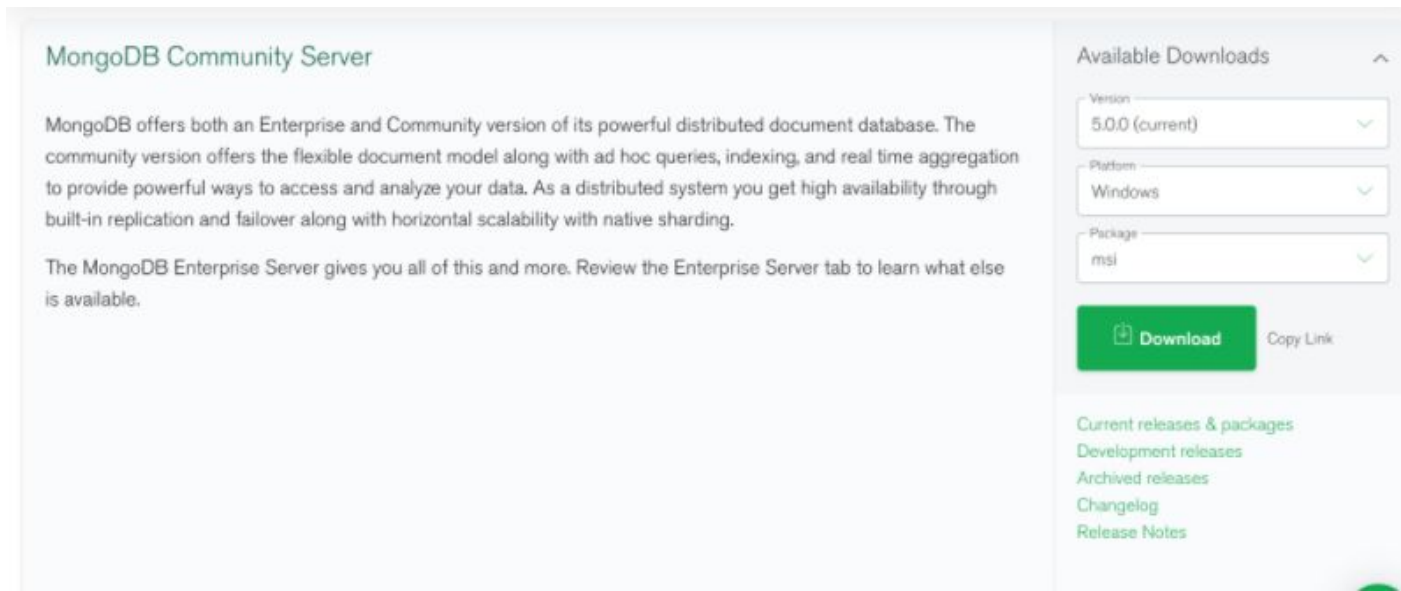
```
mongod --dbpath=/Users/nomeUsuario/data/db
```

Instalação MongoDB

Windows

Para instalar o MongoDB no Windows, deve-se acessar o link:

<https://www.mongodb.com/try/download/community> , selecionar corretamente as comboboxes, conforme a imagem abaixo, e em seguida, pressionar o botão de Download.



The screenshot shows the 'MongoDB Community Server' download page. On the left, there is descriptive text about the community and enterprise versions. On the right, under the 'Available Downloads' section, there are three dropdown menus: 'Version' set to '5.0.0 (current)', 'Platform' set to 'Windows', and 'Package' set to 'msi'. Below these is a green 'Download' button and a 'Copy Link' option. At the bottom right, there are links for 'Current releases & packages', 'Development releases', 'Archived releases', 'Changelog', and 'Release Notes'.

MongoDB Community Server

MongoDB offers both an Enterprise and Community version of its powerful distributed document database. The community version offers the flexible document model along with ad hoc queries, indexing, and real time aggregation to provide powerful ways to access and analyze your data. As a distributed system you get high availability through built-in replication and failover along with horizontal scalability with native sharding.


The MongoDB Enterprise Server gives you all of this and more. Review the Enterprise Server tab to learn what else is available.

Available Downloads

Version: 5.0.0 (current) ▼

Platform: Windows ▼

Package: msi ▼

 **Download** Copy Link

[Current releases & packages](#)
[Development releases](#)
[Archived releases](#)
[Changelog](#)
[Release Notes](#)

Instalação MongoDB

MongoDB Community Server

MongoDB offers both an Enterprise and Community version of its powerful distributed document database. The community version offers the flexible document model along with ad hoc queries, indexing, and real time aggregation to provide powerful ways to access and analyze your data. As a distributed system you get high availability through built-in replication and failover along with horizontal scalability with native sharding.

The MongoDB Enterprise Server gives you all of this and more. Review the Enterprise Server tab to learn what else is available.

Available Downloads

Version	5.0.0 (current)	✓
Platform	Windows	✓
Package	msi	✓

[Download](#)[Copy Link](#)[Current releases & packages](#)[Development releases](#)[Archived releases](#)[Changelog](#)[Release Notes](#)

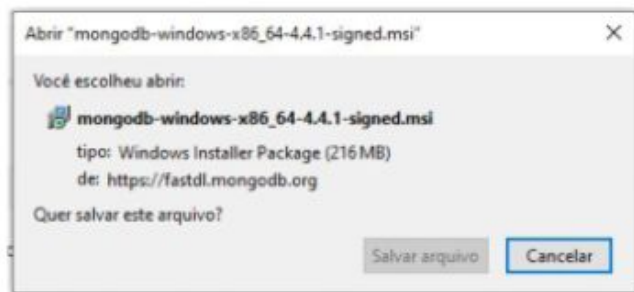
Instalação MongoDB

Feito isso, deve-se seguir os passos a seguir:

Arquivo:



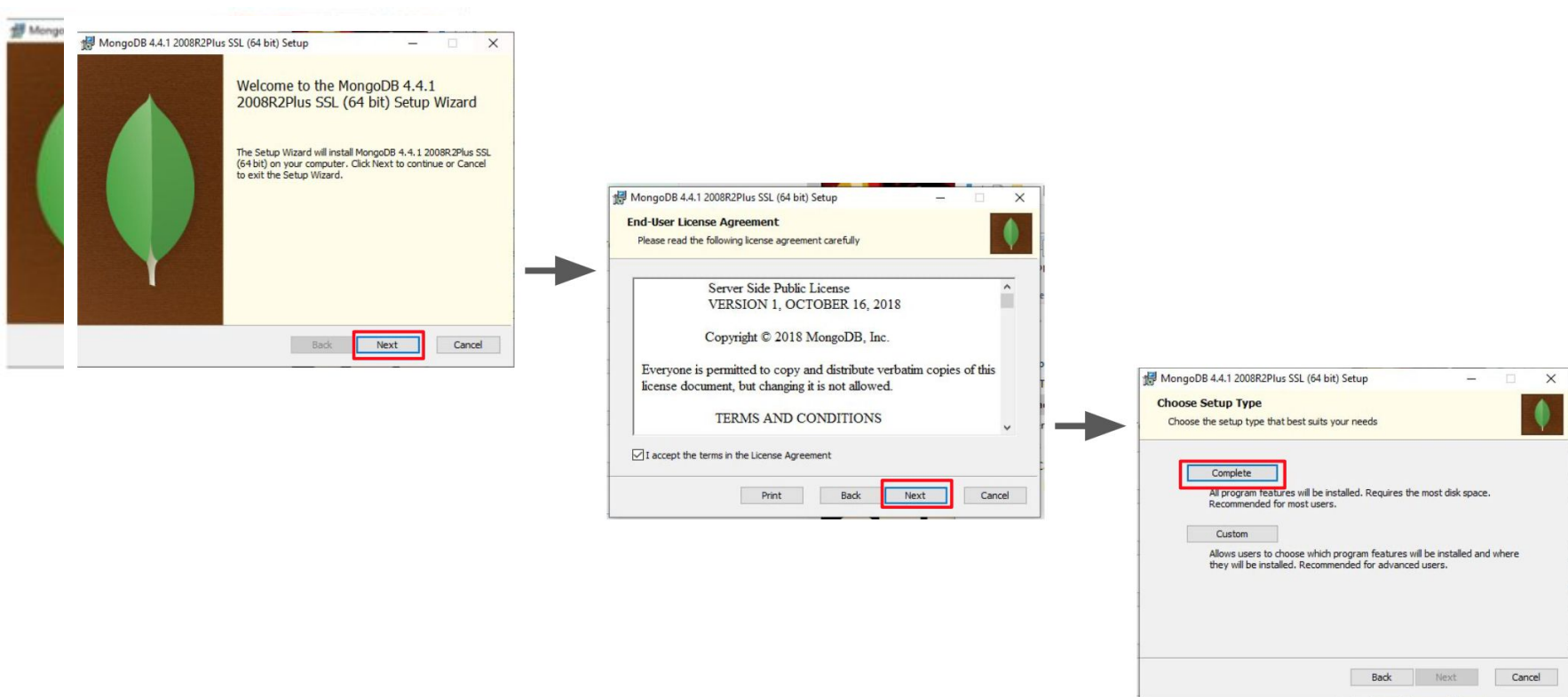
mongodb-win32
-x86_64-2012plus
-4.2.0-signed.msi



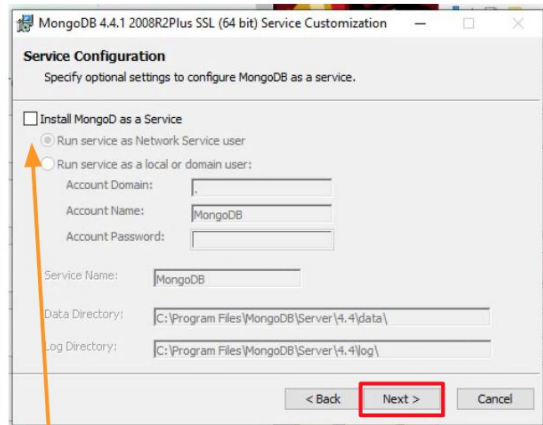
O diretório padrão de instalação
é: C:\Program
Files\MongoDB\Server\{versão
instalada}

Obs.: esse diretório default é
automaticamente
criado durante a instalação.

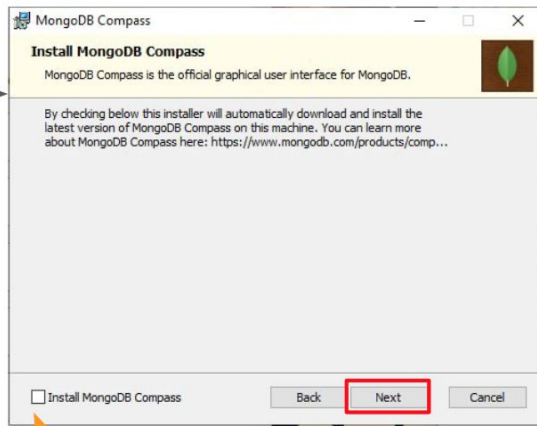
Instalação MongoDB



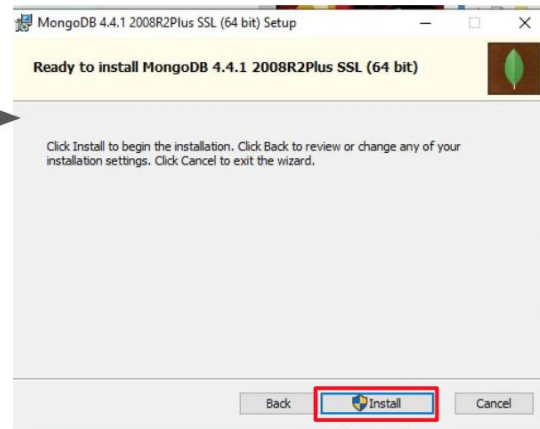
Instalação MongoDB



Desmarcar a opção: Install MongoDB as a Service, pois não temos nenhum servidor existente, iremos utilizar a nossa máquina como servidor por enquanto.

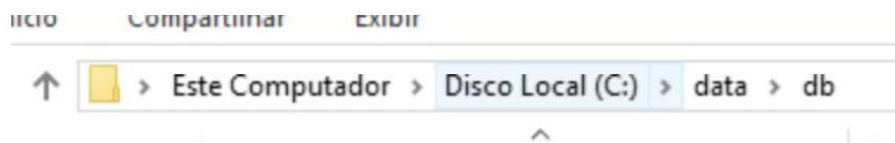


Desmarcar a opção Install MongoDB Compass.

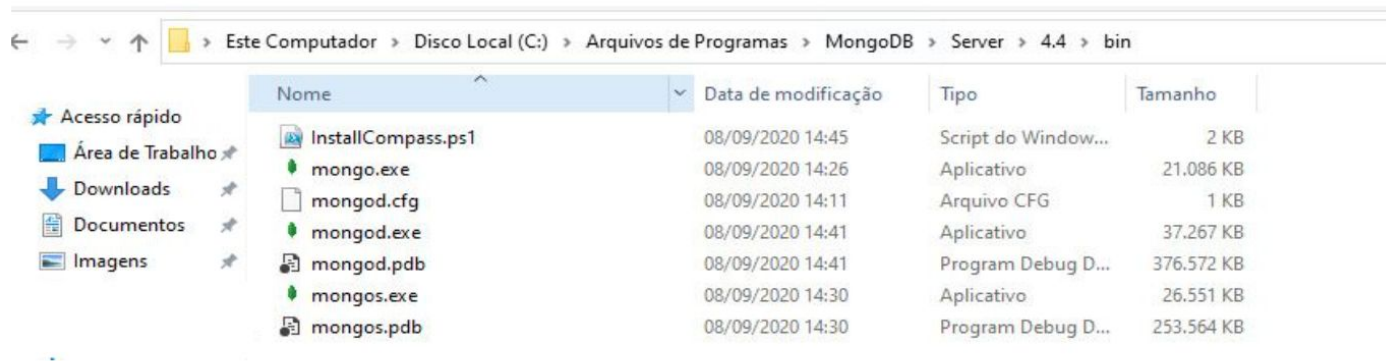


Instalação MongoDB

O diretório padrão para armazenamento dos bancos de dados é “C:\data\db. **É necessário criá-lo.**

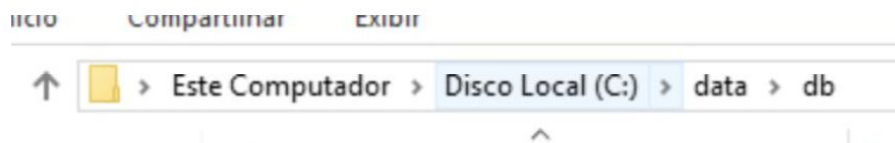


Repositório onde foi instalado o MongoDB:

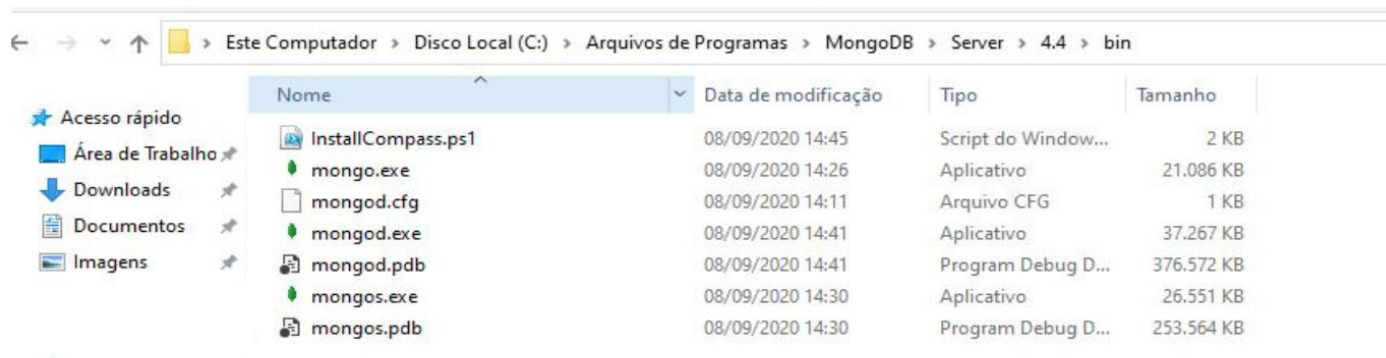


Instalação MongoDB

O diretório padrão para armazenamento dos bancos de dados é “C:\data\db. **É necessário criá-lo.**



Repositório onde foi instalado o MongoDB:

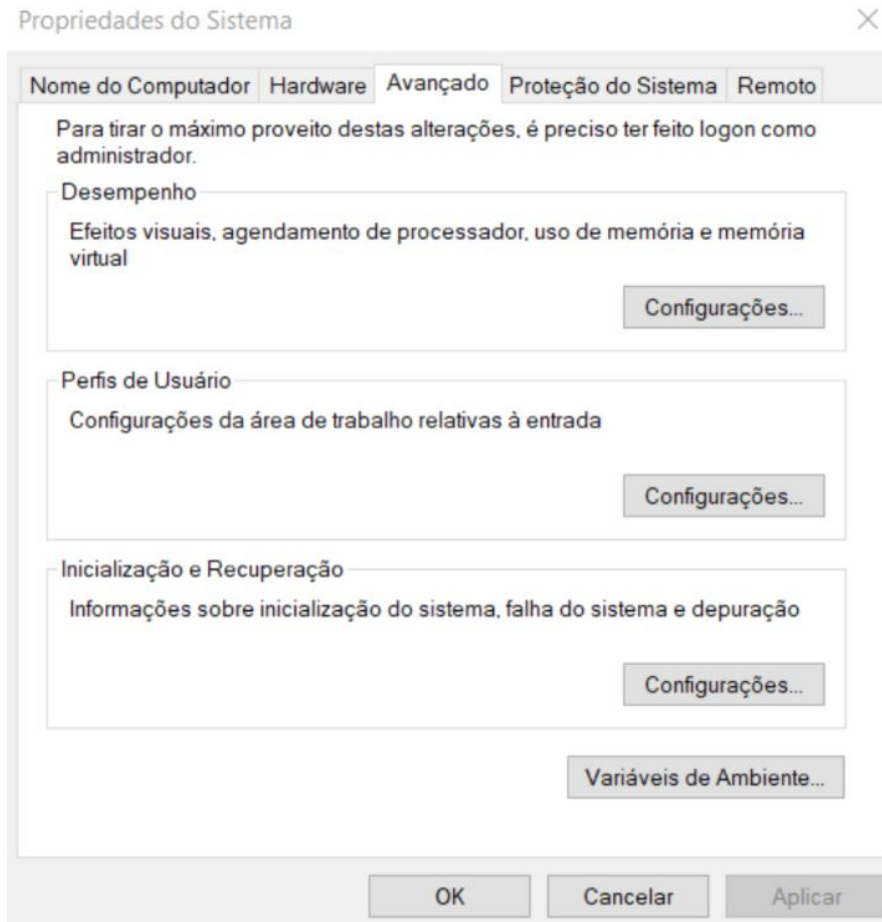


Instalação MongoDB

Devemos acessar o painel de controle para adicionar variáveis de ambiente. Para isso devemos clicar em editar as variáveis de ambiente do sistema e em seguida na janela que for aberta, clicar em Variáveis de Ambiente...

Instalação MongoDB

Ao clicar em Variáveis de Ambiente, irá abrir uma janela onde deverá ser adicionado o caminho da pasta bin de onde está instalado o seu mongoDB. Exemplo, adicionar o caminho C:\Program Files\MongoDB\Server\5.0\bin



Inicializando o servidor

Para iniciar o servidor, abra o terminal e navegue, por meio dele, até a pasta `C:\Program Files\MongoDB\Server\5.0\bin`. Para isso, digite no terminal `cd C:\Program Files\MongoDB\Server\5.0\bin` e pressione "enter". Com isso você estará na pasta desejada, e deverá então digitar `mongod` e pressionar enter. Feito isso, o servidor deverá ficar de pé.

Utilizando o MongoDB

Com o servidor rodando em um terminal, deixamos esse terminal aberto e abrimos um novo para trabalharmos com o banco de dados. No novo terminal que abrimos, sem fechar o que está rodando o servidor, devemos navegar até a pasta bin do mongo, como fizemos para subir o servidor, digitando `cd C:\Program Files\MongoDB\Server\5.0\bin`

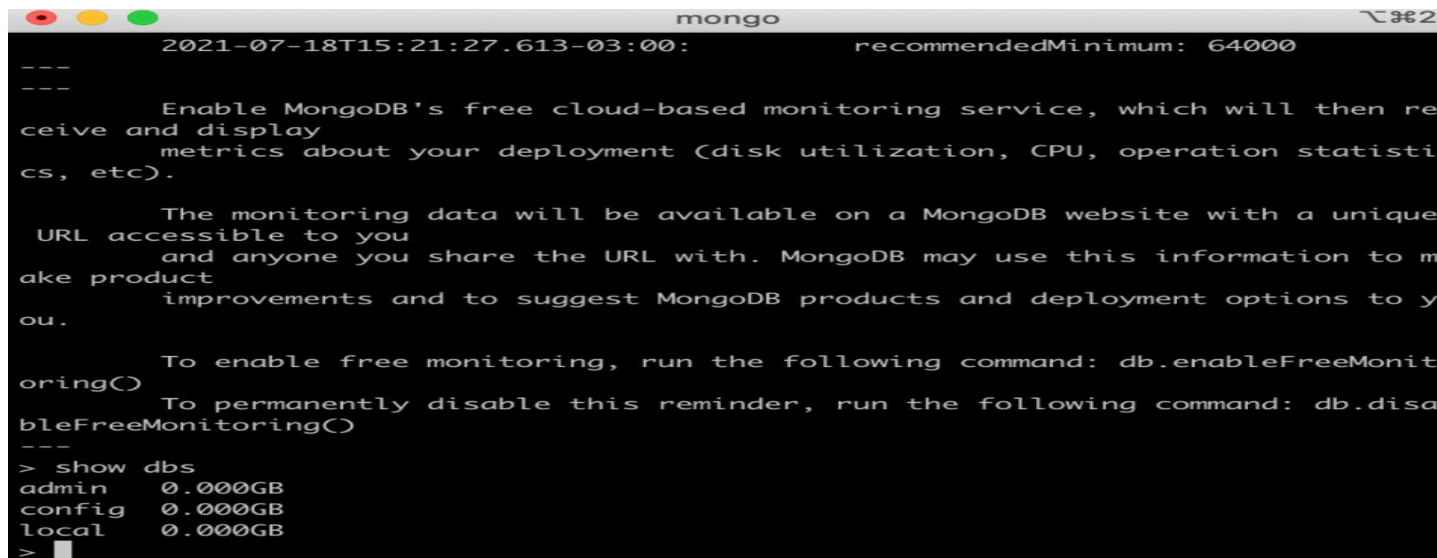
Já dentro da pasta bin no terminal, digitar `mongo` e pressionar enter.

Feito isso, já poderemos começar a trabalhar com o mongo!

Utilizando o MongoDB

Databases

Para começar, vamos visualizar as bases de dados (databases) existentes, utilizando o comando `show dbs`



```
mongo
2021-07-18T15:21:27.613-03:00: recommendedMinimum: 64000
---
---
  Enable MongoDB's free cloud-based monitoring service, which will then re
ceive and display
  metrics about your deployment (disk utilization, CPU, operation statisti
cs, etc).

  The monitoring data will be available on a MongoDB website with a unique
URL accessible to you
  and anyone you share the URL with. MongoDB may use this information to m
ake product
  improvements and to suggest MongoDB products and deployment options to y
ou.

  To enable free monitoring, run the following command: db.enableFreeMonit
oring()
  To permanently disable this reminder, run the following command: db.disa
bleFreeMonitoring()
---
> show dbs
admin    0.000GB
config  0.000GB
local    0.000GB
>
```

Utilizando o MongoDB

Vamos criar um novo database para trabalharmos? Para isso utilize o comando **use gamapan**. Esse comando irá selecionar um database ou criar, caso esse não exista. Com nosso database criado podemos agora trabalhar em cima dele e criar nossas collections!

Clínica Gama Pan Anatomy

Problema

A clínica Gama Pan Anatomy é um pouco "cringe", então costuma agendar as consultas e dados do paciente no papel e caneta. Com o passar do tempo e com a clínica crescendo está ficando insustentável manter o que temos hoje. Toda vez que precisamos de qualquer informação do paciente ou mesmo das consultas do médico demoramos muito para achar o que precisamos. Além disso, temos dados super importantes em anotações, como prontuário e todo um histórico de pacientes. Se perdermos uma folha ou alguma dessas cair em mãos erradas, podemos perder algo muito importante e sensível. Com todo esse cenário caótico, a clínica pediu nossa ajuda para organizar esses dados para melhorar o dia a dia deles e dos pacientes.

Clínica Gama Pan Anatomy

Collections

Antes de criar collections no nosso novo database reprograma, vamos recordar do nosso de documento que fizemos para consultas da Clínica Gama Pan Anatomy?

Clínica Gama Pan Anatomy

Como podemos armazenar essas informações no nosso banco de dados mongo? Para isso vamos criar uma collection chamada **Consultas** para armazenar essas informações. Para criar nossa coleção de consultas (collection), utilizaremos o comando **db.createCollection('Consultas')**.

Para visualizar todas as collections criadas no nosso database podemos utilizar o comando **show collections**.

Caso não queiramos mais nossa collection Consultas, podemos utilizar o comando **db.Consultas.drop()** para deletá-la.

MongoDB Parte 2



Consultas no MongoDB

Agora que temos a nossa base de dados populada, podemos efetuar algumas consultas que poderemos necessitar em algumas situações:

Como saber em quais consultas o médico receitou Xarope? Podemos consultar de duas maneiras:

1. Quando queremos retornar todos os documentos que contém apenas o texto exato Xarope na prescrição (caso esteja escrito 20ml de Xarope antes de dormir **não será retornado nessa consulta**): `db.getCollection('Consultas').find({prescicoes: 'Xarope'})`
2. Quando queremos retornar todos os documentos que contém o texto Xarope na prescrição (caso esteja escrito 20ml de Xarope antes de dormir **nessa consulta irá retornar, além de retornar também caso esteja escrito apenas Xarope**): `db.getCollection('Consultas').find({prescicoes: /. *Xarope.*/})`

Menor que

- Como saber quais consultas ocorreram antes das 11h do dia 12/07/2021?

```
db.getCollection('Consultas').find({dataHora: {$lt:
ISODate('2021-07-12 11:00:00.000Z')} })).
```

 Nesse caso serão retornados 2 documentos de consultas no dia 12/07/2021 às 10h.

Menor ou igual

- Como saber quais consultas ocorreram até das 11h (inclusive 11h) do dia 12/07/2021?

```
db.getCollection('Consultas').find({dataHora: {$lte:
ISODate('2021-07-12 11:00:00.000Z')} })).
```

 Nesse caso serão retornados 2 documentos de consultas no dia 12/07/2021 às 10h e 1 com consulta às 11h nesse dia.

Maior que

- Como saber quais consultas ocorreram depois das 11h do dia 12/07/2021?

`db.getCollection('Consultas').find({dataHora: {$gt: ISODate('2021-07-12 11:00:00.000Z')}})`. Nesse caso serão retornados 2 documentos de consultas, um com consulta no dia 19/07/2021 às 11h e um com consulta no dia 20/07/2021 às 9h.

Maior ou igual

- Como saber quais consultas ocorreram depois das 11h (inclusive o das 11h) do dia 12/07/2021? `db.getCollection('Consultas').find({dataHora: {$gte: ISODate('2021-07-12 11:00:00.000Z')}})`. Nesse caso serão retornados 3 documentos de consultas, um com consulta no dia 12/07/2021 às 11h, um com consulta no dia 19/07/2021 às 11h, um com consulta no dia 20/07/2021 às 9h.

Diferente

- Quero ver todas as consultas que ocorreram sem ser a consulta das 11h do dia 12/07/2021: `db.getCollection('Consultas').find({dataHora: {$ne: ISODate('2021-07-12 11:00:00.000Z')}})`. Nesse caso serão retornados 4 documentos, que são todos menos a consulta do dia 12/07/2021 às 11h.

AND

- Quero ver todas as consultas com pacientes que tenham a palavra da no nome E que também tenham endereço na Rua dos Bobos:
`db.getCollection('Consultas').find({"paciente.nome": /. *da.*/, "paciente.endereco": /. *Rua dos Bobos.*/})`. Nesse caso serão retornados 2 documentos que possuem o paciente com essas informações.

OR

- Quero ver todas as consultas com pacientes que tenham a palavra da no nome e OU que também tenham endereço na Rua dos Bobos:
`db.getCollection('Consultas').find({$or:[{"paciente.nome": /. *da. */}, {"paciente.endereco": /. *Rua dos Bobos. */}]}).` **Nesse caso serão retornados 3 documentos que possuem o paciente com essas informações.**

Atualizando um documento

Quero incluir o valor da consulta da paciente Rita da Silva. Como posso fazer isso? No Robo 3T posso clicar com o direito e escolher a opção `Update Documents`. Feito isso ele irá exibir um json já com um template para você utilizar para alterar o documento:

```
db.getCollection('Consultas').update(  
  // query  
  {  
    "key" : "value"  
  },  
  // update  
  {  
  },  
  // options  
  {  
    "multi" : false, // update only one document  
    "upsert" : false // insert a new document, if no existing document match the query  
  }  
);
```

Na primeiro json que aparece na imagem { "key" : "value" } (onde tem o comentário //query) corresponde ao filtro que vamos fazer para a alteração que iremos setar onde tem o comentário //update. Em outras palavras, você diz que você quer alterar tal coisa onde tem o //update quando o documento tiver { "key" : "value" } (onde tem o comentário //query). Por exemplo, vamos incluir o valor da consulta nas consultas da paciente Rita da Silva:

```
db.getCollection('Consultas').update(
  { "paciente.nome" : "Rita da Silva" },
  { $set:
    {
      "valor": 100
    }
  }
);
```



Poderíamos incluir a parte de `\options` da imagem se quiséssemos:

- Utilizando o `multi`, passando valor `true` alteraríamos todos os documentos com a query `{ "key" : "value" }` que especificamos. Caso informemos valor `false` alteraríamos apenas um documento com essa query `{ "key" : "value" }`. O valor padrão caso não informemos o `multi` é `false`.
- Utilizando o `upsert`, podemos informar o valor `true` caso queiramos inserir um novo documento caso não exista nenhum documento que satisfaça a query `{ "key" : "value" }`. O valor padrão caso não informemos o `upsert` é `false`.

Excluindo um documento

Para excluir um documento pelo Robo 3T basta clicar em cima dele com o botão direito e escolher a opção `Delete Document` ou usar o comando

`db.Consultas.remove({selecao})`, onde `selecao` é a condição que você quer passar para a deleção. Exemplo: quero deletar todas as consultas que possui valor da consulta

100: `db.Consultas.remove({"valor" : 100})`. Caso queiramos deletar apenas um único registro nessa condição podemos fazer essa remoção dessa forma:

```
db.Consultas.remove({"valor": 100},{justOne: true})
```


Projeção do documento

Caso eu queira buscar todos os documentos, porém a única informação que quero que retorne seja a `dataHora` da consulta, como fazer? Podemos utilizar a seguinte seleção:

```
db.getCollection('Consultas').find({}, {"dataHora" : 1, "_id" : 0}).
```

Nesse caso utilizamos o número `1` para informar que queremos que esse atributo esteja no resultado, porém utilizamos o número `0` informando que não queremos que apareça no resultado da busca. Então na nossa listagem somente virá retornando o atributo `dataHora` das consultas.

Limitando resultado de consultas

Caso eu queira retornar, por exemplo, no máximo 2 consultas na minha busca eu posso utilizar o limit: `db.getCollection('Consultas').find({}).limit(2)`

Pulando resultado de consultas

Caso eu queira retornar as consultas mas remover alguns registros da lista (ex: 3 registros), não os mostrando: `db.getCollection('Consultas').find().skip(3)`

Ordenando resultado de consultas

Caso queiramos ordenar o resultado das consultas por algum atributo, por exemplo, pelo nome do médico, podemos fazer a seguinte busca:

```
db.getCollection('Consultas').find().sort({"medico.nome": 1})
```

Esse número 1 indica que estamos ordenando do primeiro para o último, no caso do nome, da letra A -> Z do alfabeto. Caso fosse interessante ordenar ao contrário, da letra Z -> A, colocaríamos -1 no lugar.

- Caso queiramos o resultado das consultas pelo nome do médico (A -> Z) e ordenar da última consulta para a primeira, considerando o atributo

```
dataHora:db.getCollection('Consultas').find({}).sort({"medico.nome": 1, "dataHora": -1})
```

Resumo de comandos básicos Mongo

1. Base de Dados

- Exibir existentes - `show dbs`
- Selecionar para uso (e criar, caso não exista) - `use nome-do-database`
- Excluir base selecionada - `db.dropDatabase()`

2. Coleção

- Criar coleção de documentos -
`db.createCollection('nome-da-collection')` (Mongo é case sensitive)
- Exibir todas as coleções - `show collections`
- Apagar coleção de documentos - `db.nomedacollection.drop()`

3. Documentos

- Inserir - `db.nomedacollection.insert(documento)`
- Consultar - `db.nomedacollection.find({selecao})`
 - Igualdade - `{<key>:<value>}`
 - Menor que - `{<key>:{$lt:<value>}}`
 - Menor ou igual - `{<key>:{$lte:<value>}}`
 - Maior que - `{<key>:{$gt:<value>}}`
 - Maior ou igual - `{<key>:{$gte:<value>}}`
 - Diferente - `{<key>:{$ne:<value>}}`
 - AND - `{<key>:<value>, <key>:<value>}`
 - OR - `$or:[{<key>:<value>},{<key>:<value>}]`
- Atualizar - `db.nomedacollection.update({selecao}, {$set:{campos-atualizados}})`
- Excluir - `db.nomedacollection.remove({selecao})`(Considerar exclusão de seleção, apenas um e todos)
- Projetar - `db.nomedacollection.find({selecao},{<key>:1})`
- Limitar - `db.nomedacollection.find().limit(numero)`
- "Pular" - `db.nomedacollection.find().skip(numero)`
- Ordenar - `db.nomedacollection.find().sort({<key>:1})`(Considerar crescente e decrescente, e combinações)

VAMOS PRATICAR?

Atividade MongoDB

Agora que já sabemos o que é um banco de dados, já sabemos como instalar um e já conhecemos os comandos para utilizá-lo, podemos e devemos exercitar! O que podemos criar de novo? O que podemos atualizar? O que podemos buscar de informação nesse banco de dados?

Espero que tenha gostado da atividade e o segredo é praticar !!! Quanto mais exercícios fizer, melhor :)

Atividade MongoDB

Você deve criar um banco de dados novo (database) e uma coleção com um nome pertinente, de acordo com os dados e tema que você escolher. Os seguintes comandos devem ser feitos e entregues:

- Inserção de documentos
- Atualização de documentos
- Exclusão de documentos
- Consulta com projeção
- Consulta utilizando combinação entre os seletores
- Consulta paginada e ordenada (utilizar ignore, limitar e classificar)