

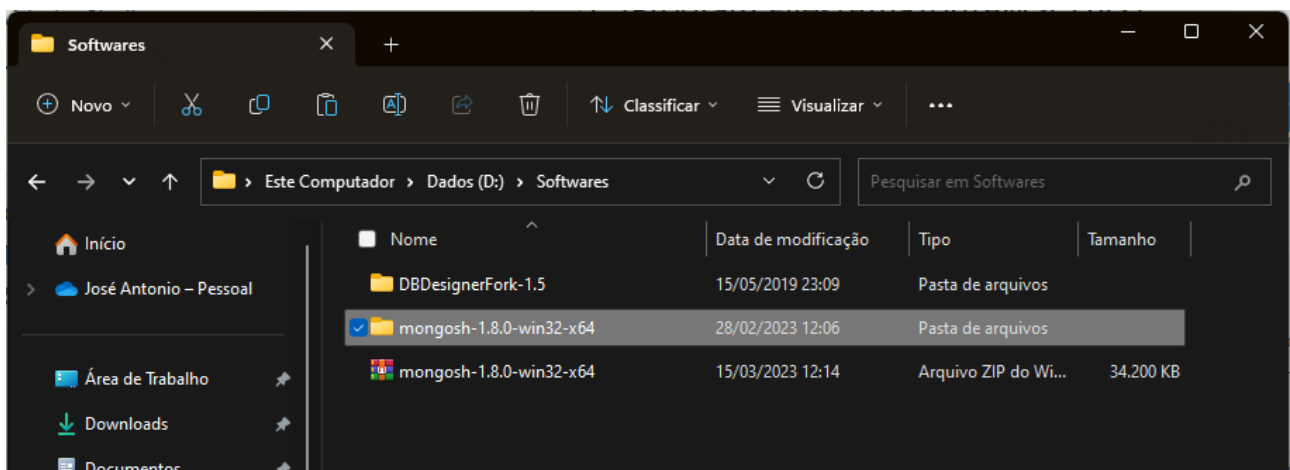
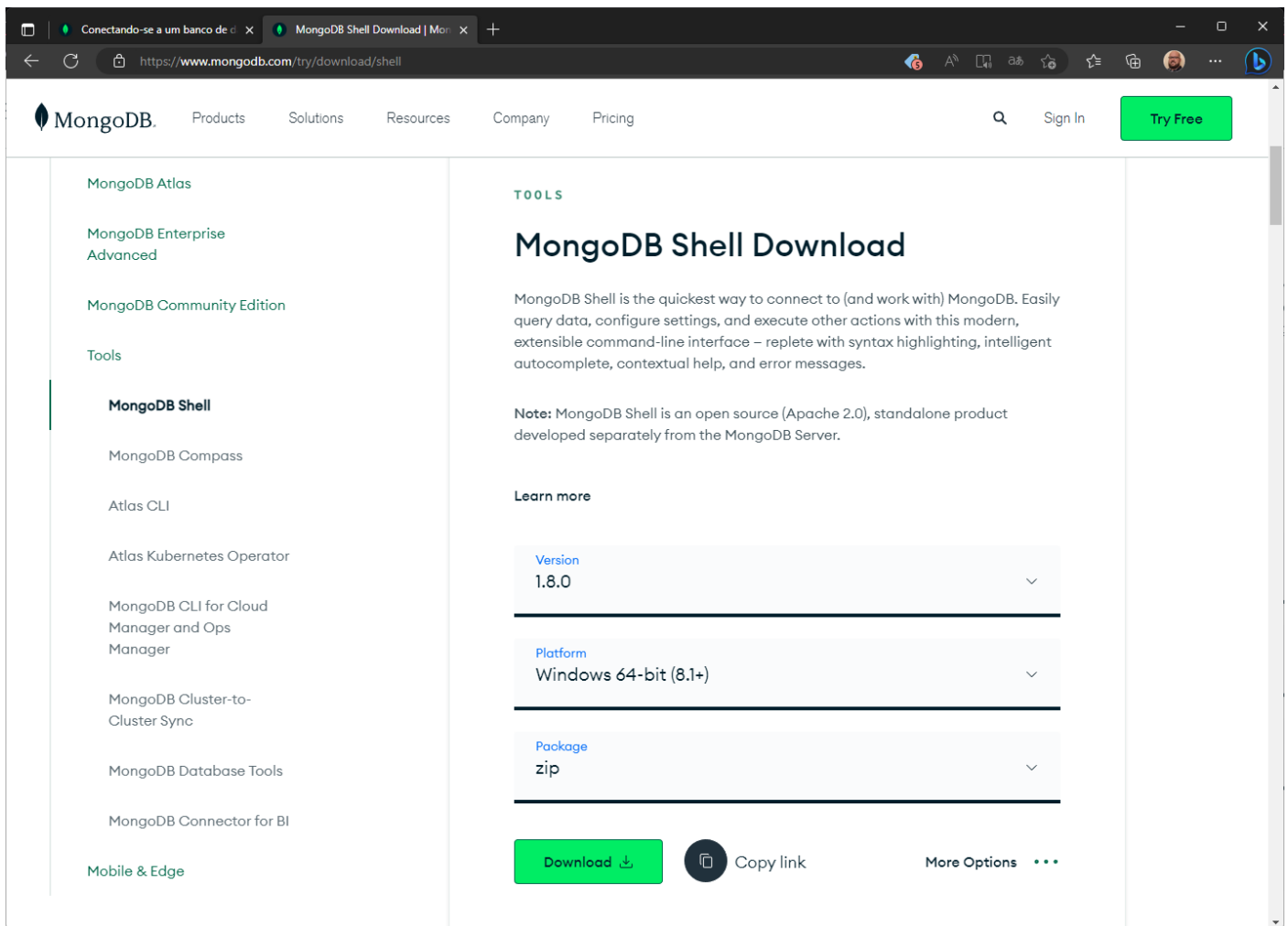
# PROJETO HEROES - MONGODB

Prof. Ms. José Antonio Gallo Junior

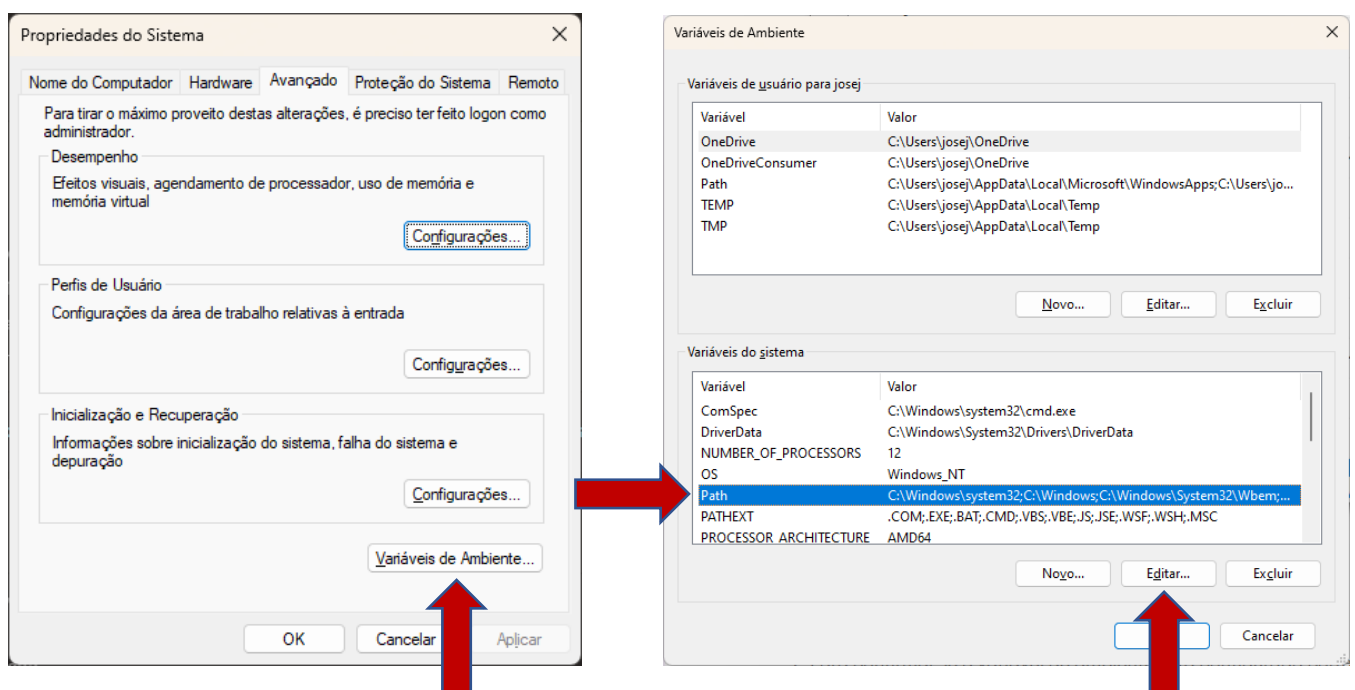
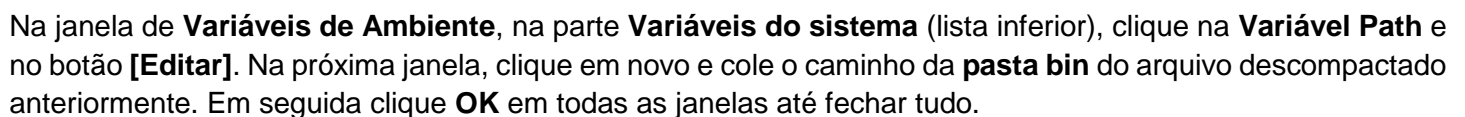
## INSTALANDO O AMBIENTE DE DESENVOLVIMENTO

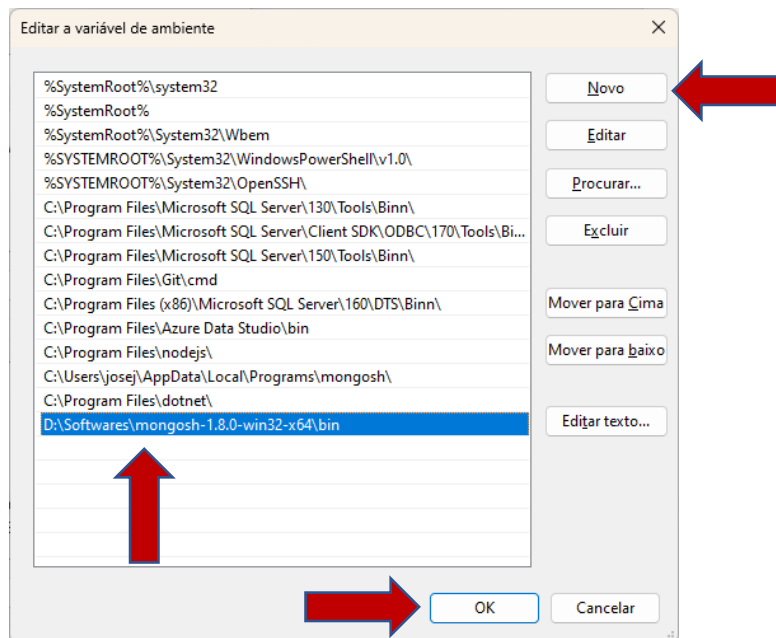
### Instalação do Mongo Shell

Acesse o site <https://www.mongodb.com/try/download/shell> e localize a seção de download do **MongoDB Shell**. Em seguida clique no botão **[Download]**. Ao término do download, mova o arquivo zip para o diretório **raiz do C:** e extraia a pasta do arquivo zip (clique com o botão direito do mouse sobre o arquivo e escolha a opção extrair aqui no **Winrar**).



Clique com o botão direito do mouse sobre o símbolo do **Windows** e escolha a opção Sistema. Em seguida clique em **Configurações avançadas do sistema** e na janela que se abrirá, **Variáveis de Ambiente...**





Para confirmar se a variável de ambiente está configurada corretamente, abra um prompt de comando e digite o comando **mongosh --help**. Se o **PATH** estiver configurado corretamente, uma lista de comandos válidos será exibida, conforme mostra abaixo:

```
C:\Windows\system32\cmd.e: X + v
C:\Users\josej>mongosh --help

$ mongosh [options] [db address] [file names (ending in .js or .mongodb)]

Options:
  -h, --help                Show this usage information
  -f, --file [arg]          Load the specified mongosh script
  --host [arg]              Server to connect to
  --port [arg]              Port to connect to
  --version                 Show version information
  --verbose                 Increase the verbosity of the output of the shell
  --quiet                  Silence output from the shell during the connection process
  --shell                   Run the shell after executing files
  --nodb                    Don't connect to mongod on startup - no 'db address' [arg] expected
  --norc                    Will not run the '.mongoshrc.js' file on start up
  --eval [arg]              Evaluate javascript
  --json[=canonical|relaxed] Print result of --eval as Extended JSON, including errors
  --retryWrites[=true|false] Automatically retry write operations upon transient network errors (Default: true)

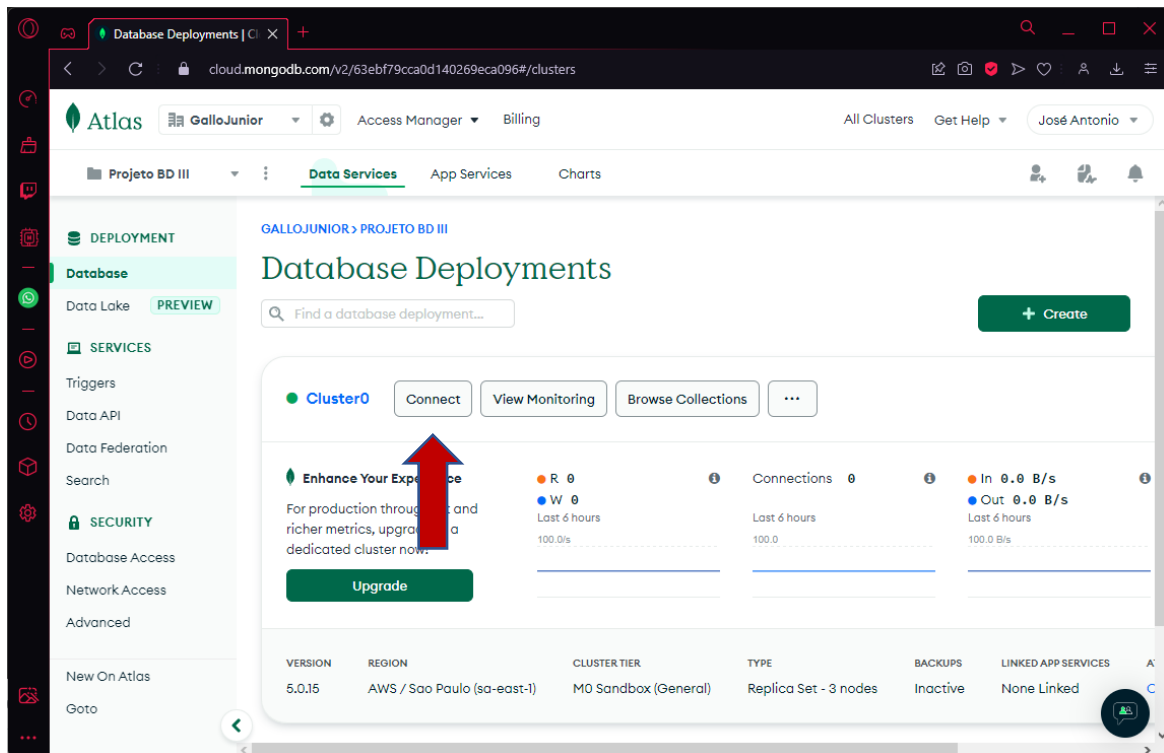
Authentication Options:
  -u, --username [arg]      Username for authentication
  -p, --password [arg]      Password for authentication
  --authenticationDatabase [arg] User source (defaults to dbname)
  --authenticationMechanism [arg] Authentication mechanism
  --awsIamSessionToken [arg] AWS IAM Temporary Session Token ID
  --gssapiServiceName [arg] Service name to use when authenticating using GSSAPI/Kerberos
  --sspiHostnameCanonicalization [arg] Specify the SSPI hostname canonicalization (none or forward, available on Windows)
  --sspiRealmOverride [arg] Specify the SSPI server realm (available on Windows)

TLS Options:
  --tls                     Use TLS for all connections
  --tlsCertificateKeyFile [arg] PEM certificate/key file for TLS
  --tlsCertificateKeyFilePassword [arg] Password for key in PEM file for TLS
  --tlsCAFile [arg]         Certificate Authority file for TLS
  --tlsAllowInvalidHostnames Allow connections to servers with non-matching hostnames
  --tlsAllowInvalidCertificates Allow connections to servers with invalid certificates
  --tlsCertificateSelector [arg] TLS Certificate in system store (Windows and macOS only)
  --tlsCRLFile [arg]         Specifies the .pem file that contains the Certificate Revocation List
  --tlsDisabledProtocols [arg] Comma separated list of TLS protocols to disable [TLS1_0,TLS1_1,TLS1_2]
  --tlsUseSystemCA           Load the operating system trusted certificate list
  --tlsFIPSMode              Enable the system TLS library's FIPS mode

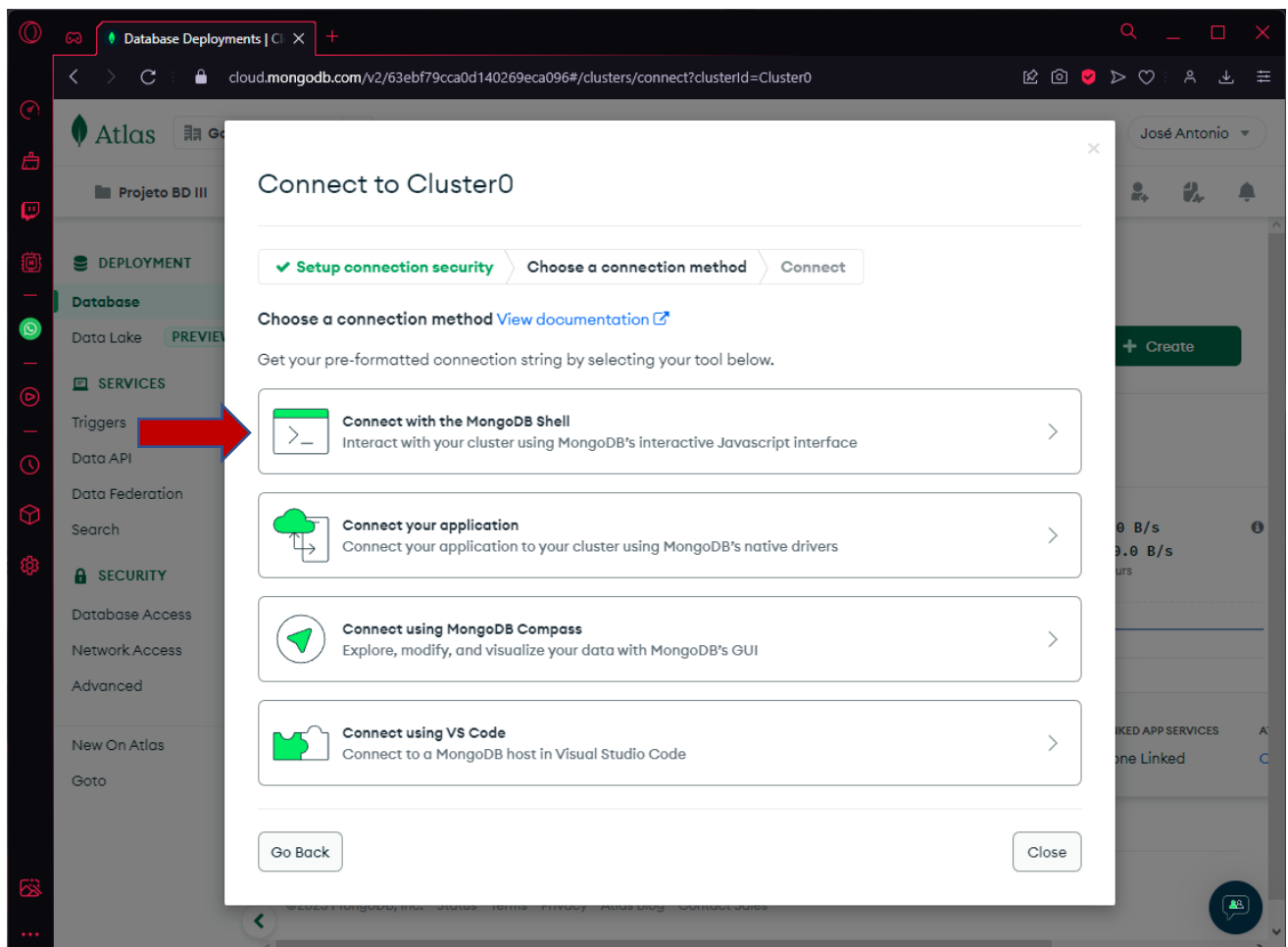
API version options:
  --apiVersion [arg]        Specifies the API version to connect with
```

# CONECTANDO AO BANCO DE DADOS

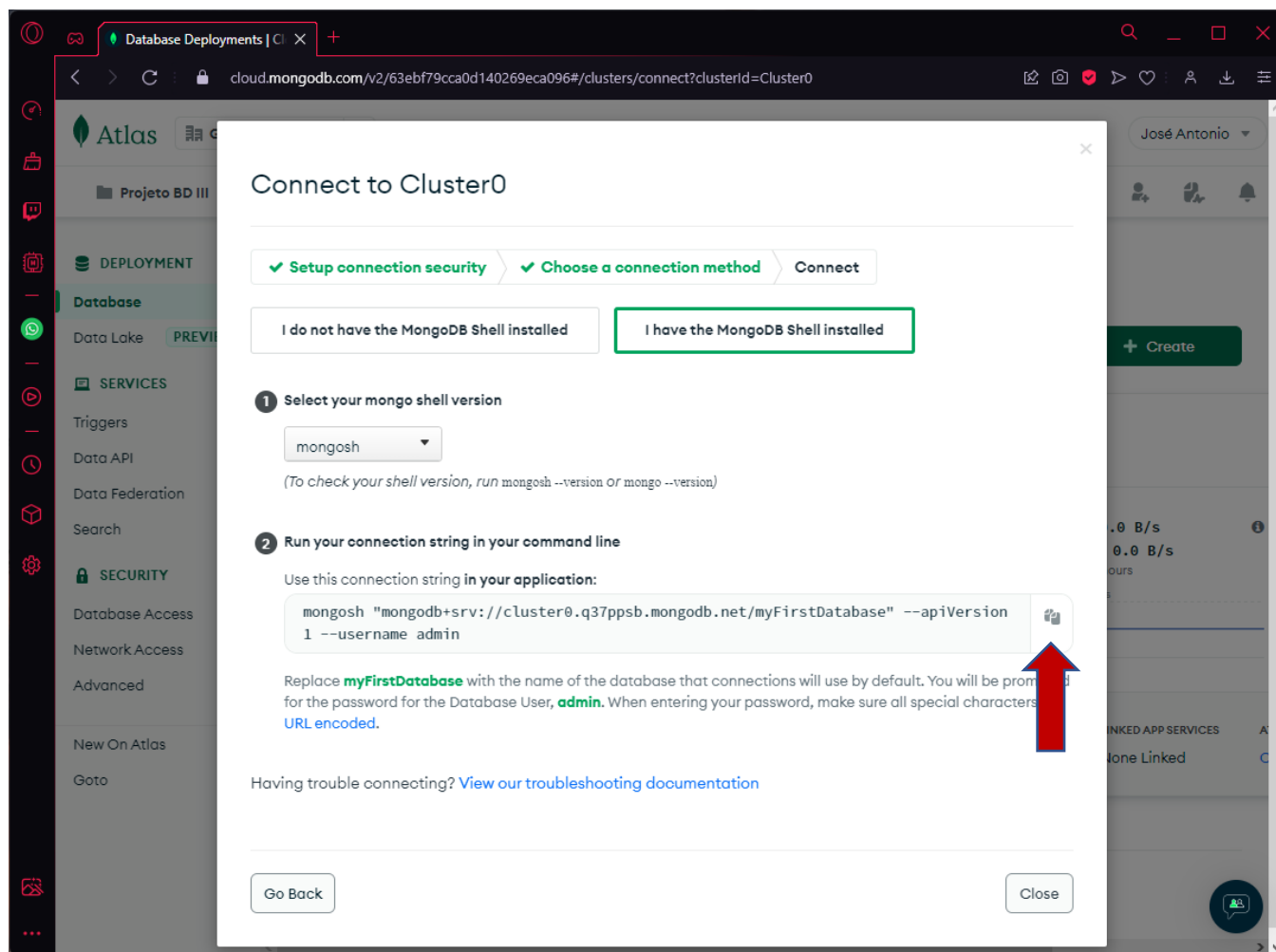
Acesse o site do **MongoDB** e faça login em sua conta. Em seguida clique no botão **Connect**, conforme mostra a figura abaixo:



Agora clique na primeira opção, “**Connect with the MongoDB Shell**”.



Em seguida clique no botão para copiar a **connection string**,



Abra um **terminal (CMD)**, se preferir pode utilizar o terminal do **Visual Studio Code**, ou outro de sua preferência, e cole o comando:

```
Cmder
D:\Softwares\cmder_mini
λ mongosh "mongodb+srv://cluster0.q37ppsb.mongodb.net/myFirstDatabase" --apiVersion 1 --username admin
Enter password:
```

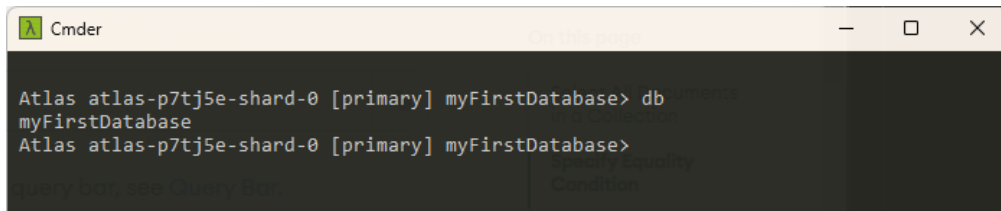
Em seguida será solicitada a senha do seu usuário criado no mongo. Após inserir a senha e pressionar **Enter**, você está pronto para manipular seus bancos de dados através de linhas de comando.

```
Cmder
D:\Softwares\cmder_mini
λ mongosh "mongodb+srv://cluster0.q37ppsb.mongodb.net/myFirstDatabase" --apiVersion 1 --username admin
Enter password: *****
Current Mongosh Log ID: 643ffdaee4f51686de8bc3ad
Connecting to: mongodb+srv://<credentials>@cluster0.q37ppsb.mongodb.net/myFirstDatabase?appName=mongosh+1.7.1
Using MongoDB: 6.0.5 (API Version 1)
Using Mongosh: 1.7.1
For mongosh info see: https://docs.mongodb.com/mongosh-shell/
Atlas atlas-p7tj5e-shard-0 [primary] myFirstDatabase> |
```

# CRIANDO UM BANCO DE DADOS

Para manipular bancos de dados no terminal, temos a disposição alguns comandos.

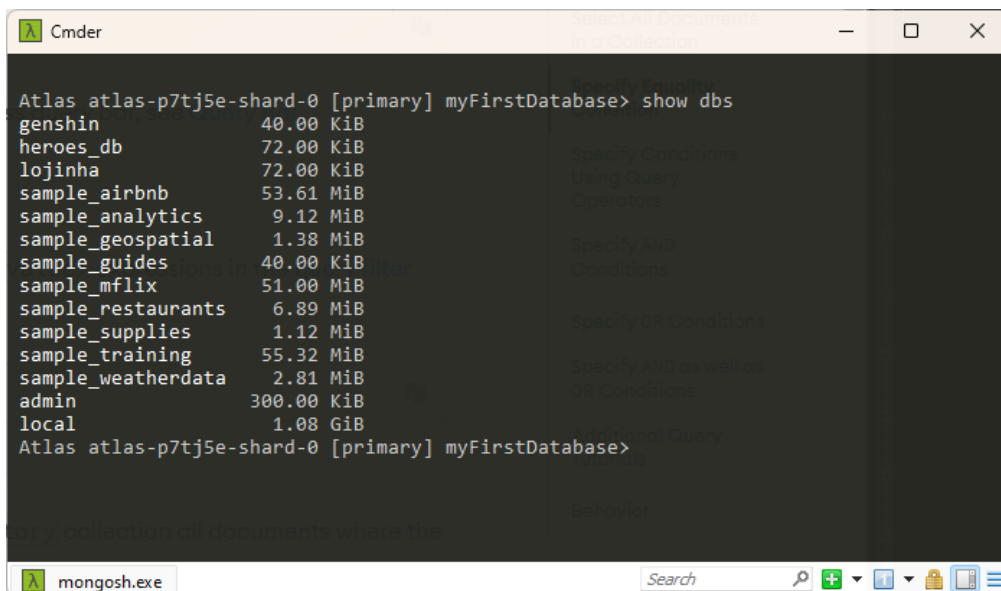
1. Verificar qual o banco de dados atualmente selecionado: **db**



```
Atlas atlas-p7tj5e-shard-0 [primary] myFirstDatabase> db
myFirstDatabase
Atlas atlas-p7tj5e-shard-0 [primary] myFirstDatabase>
```

**Nota:** É importante sempre verificar qual é o banco de dados corrente, para não realizar operações no banco de dados errado

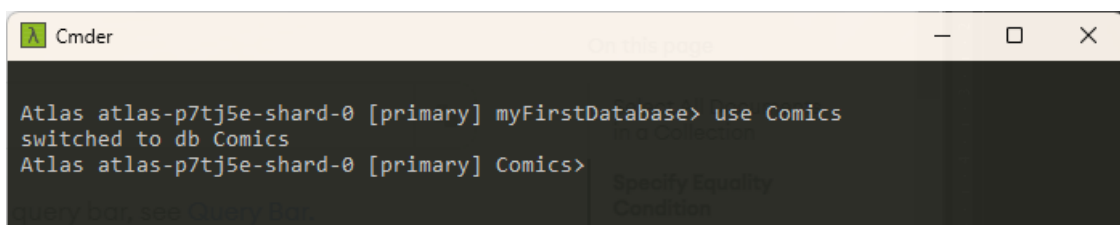
2. Exibir os bancos de dados existentes no **Cluster**: **show dbs**



```
Atlas atlas-p7tj5e-shard-0 [primary] myFirstDatabase> show dbs
genshin          40.00 KiB
heroes_db        72.00 KiB
lojinha          72.00 KiB
sample_airbnb    53.61 MiB
sample_analytics  9.12 MiB
sample_geospatial 1.38 MiB
sample_guides    40.00 KiB
sample_mflix     51.00 MiB
sample_restaurants 6.89 MiB
sample_supplies  1.12 MiB
sample_training  55.32 MiB
sample_weatherdata 2.81 MiB
admin            300.00 KiB
local            1.08 GiB
Atlas atlas-p7tj5e-shard-0 [primary] myFirstDatabase>
```

3. Criar ou acessar um banco de dados: **use <nome do banco>**

Criando o banco de dados que será utilizado para esse material: **use Comics**



```
Atlas atlas-p7tj5e-shard-0 [primary] myFirstDatabase> use Comics
switched to db Comics
Atlas atlas-p7tj5e-shard-0 [primary] Comics>
```

**Nota 1:** Lembre-se que o comando use, cria o banco de dados se ele ainda não existir.

**Nota 2:** Após criar um banco de dados, caso você utilize o comando **show dbs**, o banco não será exibido, isso acontece porque esse banco só passa a realmente existir após a criação de pelo menos uma coleção dentro dele.

# OPERAÇÕES CRUD

**CRUD** (acrônimo do inglês para **Create**, **Read**, **Update** e **Delete**) são as quatro operações básicas do desenvolvimento de uma aplicação, sendo utilizadas em bases de dados fornecidas aos utilizadores de um sistema.

## Operação de Criação – CREATE

Operações para Criar ou Inserir permitem adicionar novos Documentos (**Documents**) em uma Coleção (**Collection**). Se a Coleção não existe no momento, as operações de Inserção criarão a Coleção.

O **MongoDB** fornece os seguintes métodos para inserir documentos em uma coleção:

- `db.collection.insertOne()`
- `db.collection.insertMany()`

```
db.users.insertOne(  ← collection
{
  name: "sue",        ← field: value
  age: 26,            ← field: value
  status: "pending"   ← field: value
}                    } document
)
```

### Inserir um único documento

O comando **db.collection.insertOne()** insere um único documento em uma coleção.

O exemplo a seguir insere um novo documento na coleção. Se o documento não especificar um **campo \_id**, o **MongoDB** adiciona o campo com um valor do tipo **ObjectId** para o novo documento.

```
db.herois.insertOne({
  nome: "James Howlett",
  identidade: "Wolverine",
  universo: "Marvel",
  especie: "Mutante",
  tipo: "Herói",
  primeiraAparicao: "Novembro de 1974",
  poderes: ["Regeneração", "Sentidos Aguçados", "Garras"],
  afiliacoes: ["X-Force", "X-Men", "Novos Vingadores", "Clã Yashida", "Arma-X"],
  usaCapa: false,
  caracteristicas: {
    idade: 137,
    genero: "Masculino",
    altura: 1.60,
    peso: 136.1,
    olhos: "Azuis",
    cabelo: "Preto"
  },
  imagem: "https://static.wikia.nocookie.net/marvel/images/1/16/James_Howlett_%28Earth-616%29_from_Wolverine_Vol_7_1_Silva_Variant_cover_001.jpg/revision/latest?cb=20210828064850&path-prefix=pt-br"
})
```



O comando **insertOne()** retorna um documento que inclui o valor do campo **\_id** do documento recém-inserido. Executando o comando acima, teremos o resultado apresentado na imagem abaixo:

```
Atlas atlas-p7tj5e-shard-0 [primary] Comics> db.herois.insertOne({
...   nome: "James Howlett",
...   identidade: "Wolverine",
...   universo: "Marvel",
...   especie: "Mutante",
...   primeiraAparicao: "Novembro de 1974",
...   poderes: ["Regeneração", "Sentidos Aguçados", "Garras"],
...   afiliacoes: ["X-Force", "X-Men", "Novos Vingadores", "Clã Yashida", "Arma-X"],
...   usaCapa: false,
...   caracteristicas: {
...     idade: 137,
...     genero: "Masculino",
...     altura: 1.60,
...     peso: 136.1,
...     olhos: "Azuis",
...     cabelo: "Preto"
...   },
...   imagem: "https://static.wikia.nocookie.net/marvel/images/1/16/James_Howlett_%28Earth-616%29_from_Wolverine_Vol_7_1_Silva_Variant_cover_001.jpg/revision/latest?cb=20210828064850&path-prefix=pt-br"
... })
{
  acknowledged: true,
  insertedId: ObjectId("64402c7f79710171e7b43c3b")
}
Atlas atlas-p7tj5e-shard-0 [primary] Comics> |
```

Para recuperar o documento que você acabou de inserir, podemos usar o comando **find()** para consultar a coleção:

```
Atlas atlas-p7tj5e-shard-0 [primary] Comics> db.herois.find()
[
  {
    _id: ObjectId("64402c7f79710171e7b43c3b"),
    nome: 'James Howlett',
    identidade: 'Wolverine',
    universo: 'Marvel',
    especie: 'Mutante',
    primeiraAparicao: 'Novembro de 1974',
    poderes: [ 'Regeneração', 'Sentidos Aguçados', 'Garras' ],
    afiliacoes: [ 'X-Force', 'X-Men', 'Novos Vingadores', 'Clã Yashida', 'Arma-X' ],
    usaCapa: false,
    caracteristicas: {
      idade: 137,
      genero: 'Masculino',
      altura: 1.6,
      peso: 136.1,
      olhos: 'Azuis',
      cabelo: 'Preto'
    },
    imagem: 'https://static.wikia.nocookie.net/marvel/images/1/16/James_Howlett_%28Earth-616%29_from_Wolverine_Vol_7_1_Silva_Variant_cover_001.jpg/revision/latest?cb=20210828064850&path-prefix=pt-br'
  }
]
Atlas atlas-p7tj5e-shard-0 [primary] Comics> |
```



## Inserir vários documentos

O comando **db.collection.insertMany()** pode inserir vários documentos em uma coleção. Para isso devemos passar um vetor de documentos para o método.

O exemplo a seguir insere três novos documentos na coleção **herois**. Se os documentos não especificarem um campo **\_id**, o **MongoDB** adicionará o campo com um valor do tipo **ObjectId** para cada documento.

```
db.herois.insertMany([
  {
    nome: "Peter Parker",
    identidade: "Homem-Aranha",
    universo: "Marvel",
    especie: "Humano Geneticamente Modificado",
    primeiraAparicao: "Agosto de 1962",
    poderes: ["Agilidade", "Força", "Sentido Aranha"],
    afiliacoes: ["Vingadores", "Novos Vingadores", "Defensores"],
    usaCapa: false,
    caracteristicas: {
      idade: 33,
      genero: "Masculino",
      altura: 1.78,
      peso: 75.75,
      olhos: "Preto",
      cabelo: "Castanho"
    },
    imagem: "https://static.wikia.nocookie.net/marveldatabase/images/d/da/Amazing_Spider-Man_Vol_5_15_Textless.jpg/revision/latest?cb=20181121213108"
  },
  {
    nome: "Wade Winston Wilson",
    identidade: "Deadpool",
    universo: "Marvel",
    especie: "Humano Geneticamente Modificado",
    primeiraAparicao: "Fevereiro de 1991",
    poderes: ["Regeneração", "Imortalidade", "Habilidades Marciais"],
    afiliacoes: ["Agência X", "X-Force", "X-Men"],
    usaCapa: false,
    caracteristicas: {
      idade: 40,
      genero: "Masculino",
      altura: 1.88,
      peso: 95.25,
      olhos: "Castanho",
      cabelo: "Careca"
    },
    imagem: "https://static.wikia.nocookie.net/marveldatabase/images/1/16/Deadpool_Vol_9_1_Nakayama_Virgin_Variant.jpg/revision/latest?cb=20221209095203"
  },
  {
    nome: "Robert Bruce Banner",
    identidade: "Hulk",
    universo: "Marvel",
    especie: "Humano Geneticamente Modificado",
    primeiraAparicao: "Fevereiro de 1991",
    poderes: ["Super-Força", "Regeneração", "Resistência"],
    afiliacoes: ["Vingadores", "Incríveis Hulks", "Novo Quarteto Fantástico"],
    usaCapa: false,
    caracteristicas: {
      idade: 50,
```

```

    genero: "Masculino",
    altura: 1.80,
    peso: 76.2,
    olhos: "Castanho",
    cabelo: "Castanho"
  },
  imagem:
    "https://static.wikia.nocookie.net/marvel/images/d/dc/Immortal_Hulk_20_Unknown_Comic_Books_Exclusive_Virgin_Variant.jpg/revision/latest?cb=20190907034513&path-prefix=pt-br"
}
])

```

O comando **insertMany()** retorna um documento que inclui os valores do campo **\_id** dos documentos recém-inseridos. Executando o comando acima, teremos o resultado apresentado na imagem abaixo:

```

Atlas atlas-p7tj5e-shard-0 [primary] Comics>
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("64402cc579710171e7b43c3c"),
    '1': ObjectId("64402cc579710171e7b43c3d"),
    '2': ObjectId("64402cc579710171e7b43c3e")
  }
}

```

Para recuperar os documentos inseridos, podemos usar o comando **find()** para consultar a coleção:

```

Atlas atlas-p7tj5e-shard-0 [primary] Comics> db.herois.find({}, { identidade: 1 }).pretty()
[
  {
    _id: ObjectId("64402c7f79710171e7b43c3b"),
    identidade: 'Wolverine'
  },
  {
    _id: ObjectId("64402cc579710171e7b43c3c"),
    identidade: 'Homem-Aranha'
  },
  {
    _id: ObjectId("64402cc579710171e7b43c3d"),
    identidade: 'Deadpool'
  },
  {
    _id: ObjectId("64402cc579710171e7b43c3e"),
    identidade: 'Hulk'
  }
]

```

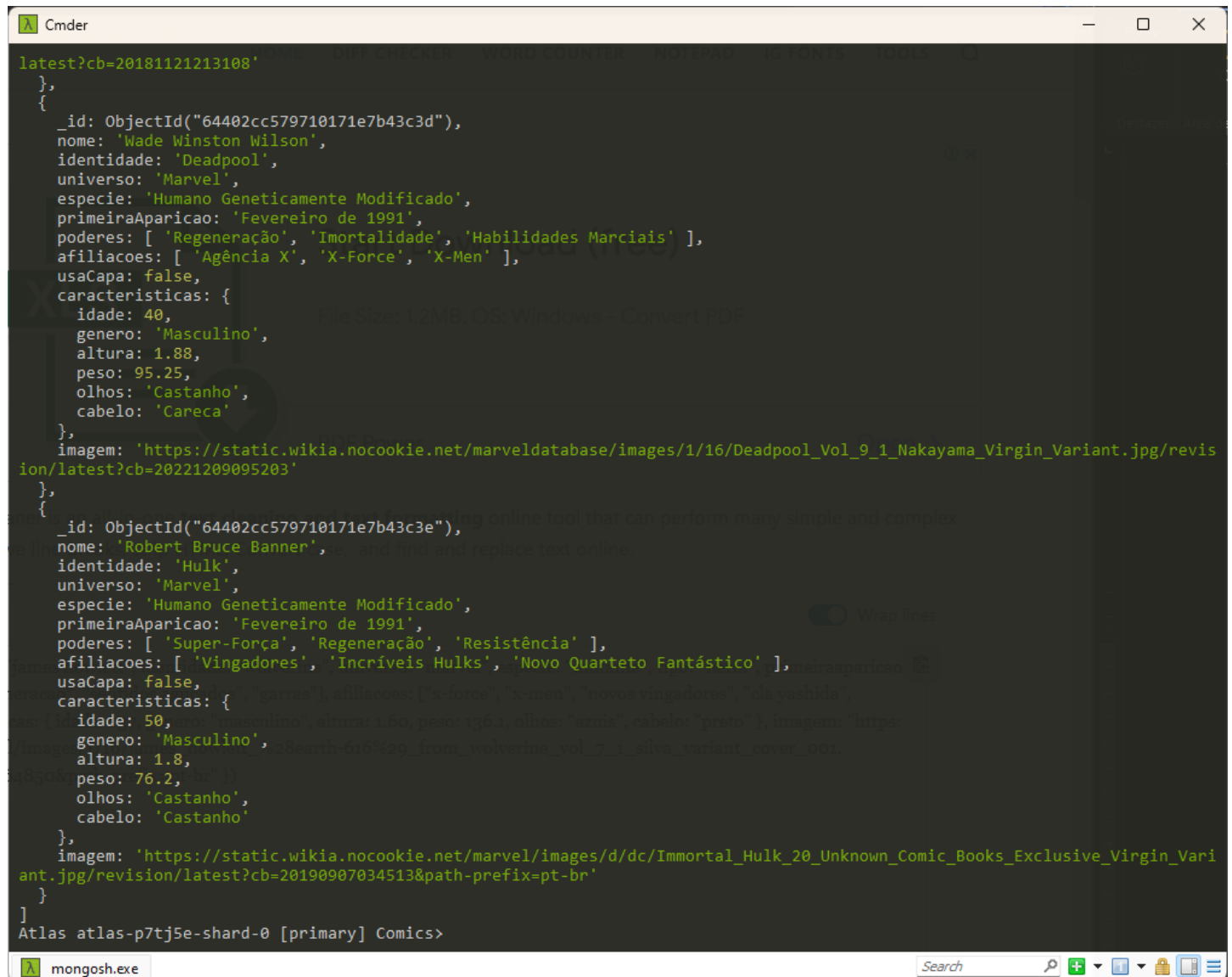
O documento inserido no **find( {}, { identidade: 1 })** especifica que queremos apenas o campo **identidade**.

Vale ressaltar que outros comandos também podem ser utilizados para incluir dados como:

- **db.collection.updateOne()** quando usado com a opção **upsert: true**.
- **db.collection.updateMany()** quando usado com a opção **upsert: true**.
- **db.collection.findAndModify()** quando usado com a opção **upsert: true**.
- **db.collection.findOneAndUpdate()** quando usado com a opção **upsert: true**.
- **db.collection.findOneAndReplace()** quando usado com a opção **upsert: true**.
- **db.collection.bulkWrite()**.

## Operações de Leitura - READ

Para selecionar todos os documentos em uma coleção, passamos um documento vazio como filtro da consulta. O documento de parâmetro determina os critérios da consulta:



```
latest?cb=20181121213108'
},
{
  _id: ObjectId("64402cc579710171e7b43c3d"),
  nome: 'Wade Winston Wilson',
  identidade: 'Deadpool',
  universo: 'Marvel',
  especie: 'Humano Geneticamente Modificado',
  primeiraAparicao: 'Fevereiro de 1991',
  poderes: [ 'Regeneração', 'Imortalidade', 'Habilidades Marciais' ],
  afiliacoes: [ 'Agência X', 'X-Force', 'X-Men' ],
  usaCapa: false,
  caracteristicas: {
    idade: 40,
    genero: 'Masculino',
    altura: 1.88,
    peso: 95.25,
    olhos: 'Castanho',
    cabelo: 'Careca'
  },
  imagem: 'https://static.wikia.nocookie.net/marveldatabase/images/1/16/Deadpool_Vol_9_1_Nakayama_Virgin_Variant.jpg/revision/latest?cb=20221209095203'
},
{
  _id: ObjectId("64402cc579710171e7b43c3e"),
  nome: 'Robert Bruce Banner',
  identidade: 'Hulk',
  universo: 'Marvel',
  especie: 'Humano Geneticamente Modificado',
  primeiraAparicao: 'Fevereiro de 1991',
  poderes: [ 'Super-Força', 'Regeneração', 'Resistência' ],
  afiliacoes: [ 'Vingadores', 'Incríveis Hulks', 'Novo Quarteto Fantástico' ],
  usaCapa: false,
  caracteristicas: {
    idade: 50,
    genero: 'Masculino',
    altura: 1.60,
    peso: 136.1,
    olhos: 'azuis',
    cabelo: 'preto'
  },
  imagem: 'https://static.wikia.nocookie.net/marvel/images/d/dc/Immortal_Hulk_20_Unknown_Comic_Books_Exclusive_Virgin_Variant.jpg/revision/latest?cb=20190907034513&path-prefix=pt-br'
}
]
Atlas atlas-p7tj5e-shard-0 [primary] Comics>
```

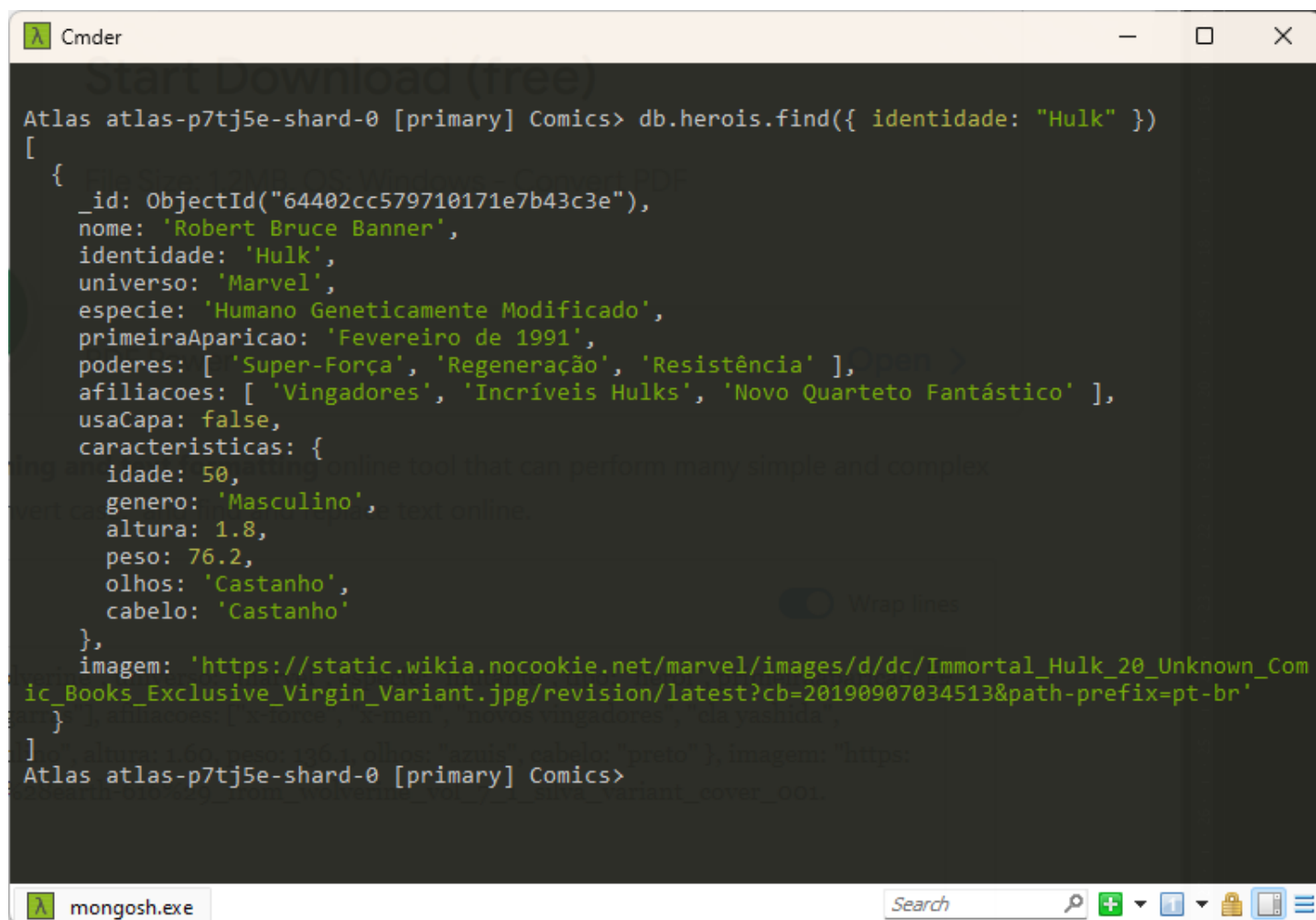
Este tipo de consulta, onde não especificamos um parâmetro é o correspondente no **SQL** ao:

```
SELECT * FROM herois
```

## Operações de Leitura – Operador de Igualdade

Para especificar condições de igualdade, usamos a expressão { <campo> : <valor> } como documento de filtro da consulta:

```
db.herois.find({ identidade: "Hulk" })
```



```
Atlas atlas-p7tj5e-shard-0 [primary] Comics> db.herois.find({ identidade: "Hulk" })
[
  {
    _id: ObjectId("64402cc579710171e7b43c3e"),
    nome: 'Robert Bruce Banner',
    identidade: 'Hulk',
    universo: 'Marvel',
    especie: 'Humano Geneticamente Modificado',
    primeiraAparicao: 'Fevereiro de 1991',
    poderes: [ 'Super-Força', 'Regeneração', 'Resistência' ],
    afiliacoes: [ 'Vingadores', 'Incríveis Hulks', 'Novo Quarteto Fantástico' ],
    usaCapa: false,
    caracteristicas: {
      idade: 50,
      genero: 'Masculino',
      altura: 1.8,
      peso: 76.2,
      olhos: 'Castanho',
      cabelo: 'Castanho'
    },
    imagem: 'https://static.wikia.nocookie.net/marvel/images/d/dc/Immortal_Hulk_20_Unknown_Comic_Books_Exclusive_Virgin_Variant.jpg/revision/latest?cb=20190907034513&path-prefix=pt-br'
  }
]
Atlas atlas-p7tj5e-shard-0 [primary] Comics>
```

Este tipo de consulta, onde não especificamos um parâmetro é o correspondente no **SQL** ao:

```
SELECT * FROM herois WHERE identidade = "Hulk"
```

## Operação de Leitura – Operadores de Condição

Um documento de filtro de consulta pode usar operadores de consulta para especificar condições sob a seguinte forma: { <campo> : { <operador>: <valor> } }

A consulta a seguir recupera todos os documentos da coleção onde a identidade do herói for igual a **Deadpool** ou **Wolverine**.

```
db.herois.find( { identidade: { $in: ["Deadpool", "Wolverine"] } })
```

Este tipo de consulta, onde não especificamos um parâmetro é o correspondente no **SQL** ao:

```
SELECT * FROM herois WHERE identidade in ("Deadpool", "Wolverine")
```

## Operação de Leitura – Operadores de Condição AND

Uma consulta composta pode especificar condições para mais de um campo nos documentos da coleção. Implicitamente, uma conjunção lógica **AND** conecta as cláusulas de uma consulta composta para que a consulta selecione os documentos na coleção que correspondam a todas as condições.

O exemplo a seguir filtra todos os documentos da coleção **herois** onde **useCapa** estiver **falso** e que sejam do **universo Marvel**:

```
db.herois.find({ usaCapa: false, universo: "Marvel" })
```

Este tipo de consulta, onde não especificamos um parâmetro é o correspondente no **SQL** ao:

```
SELECT * FROM herois WHERE usaCapa is false AND universo = "Marvel"
```

## Operação de Leitura – Operadores de Condição OR

Usando o operador **\$or**, podemos especificar uma consulta combinada com diversas cláusulas com a conjunção lógica **OR** de forma que a seleção dos documentos da coleção tenha ao menos uma condição correspondente.

O exemplo a seguir filtra todos os documentos da coleção **herois** onde **usaCapa** estiver **true** e que a **especie** tenha o valor **Mutante**:

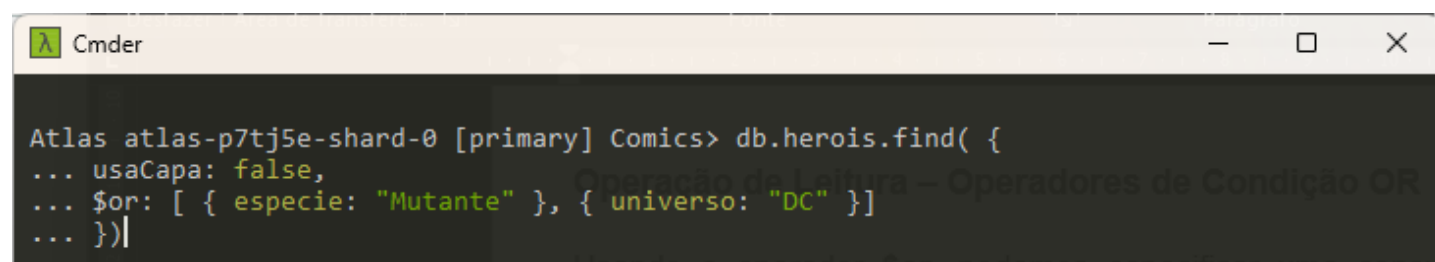
```
db.herois.find({ $or: [ { usaCapa: true }, { especie: "Mutante" } ] })
```

Este tipo de consulta, onde não especificamos um parâmetro é o correspondente no **SQL** ao:

```
SELECT * FROM herois WHERE usaCapa is true OR especie = "Mutante"
```

## Operação de Leitura – Operadores de Condição AND e OR Combinados

No exemplo a seguir, vamos combinar os documentos de seleção para buscar todos os documentos que possuam o campo **usaCapa** como **falso**, e que sejam da **especie "Mutante"** ou do **universo "DC"**:



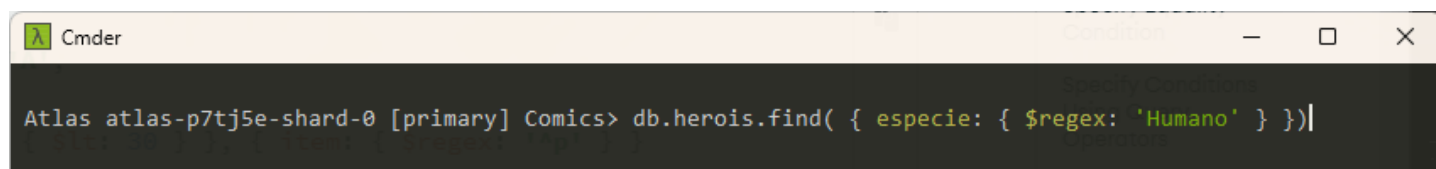
```
Atlas atlas-p7tj5e-shard-0 [primary] Comics> db.herois.find( {  
... usaCapa: false,  
... $or: [ { especie: "Mutante" }, { universo: "DC" } ]  
... } )|
```

Este tipo de consulta, onde não especificamos um parâmetro é o correspondente no **SQL** ao:

```
SELECT * FROM heróis WHERE usaCapa is false AND (especie = "Mutante" OR universo = "DC")
```

## Operação de Leitura – Operador Regex

É possível pesquisar dentro de um texto utilizando expressões regulares (regex), desta forma podemos solicitar todos os heróis que possuem em **espécie** o valor **Humano** ou **derivados (Humanos Geneticamente Modificados, entre outros)**.



```
Atlas atlas-p7tj5e-shard-0 [primary] Comics> db.herois.find( { especie: { $regex: 'Humano' } } )
```

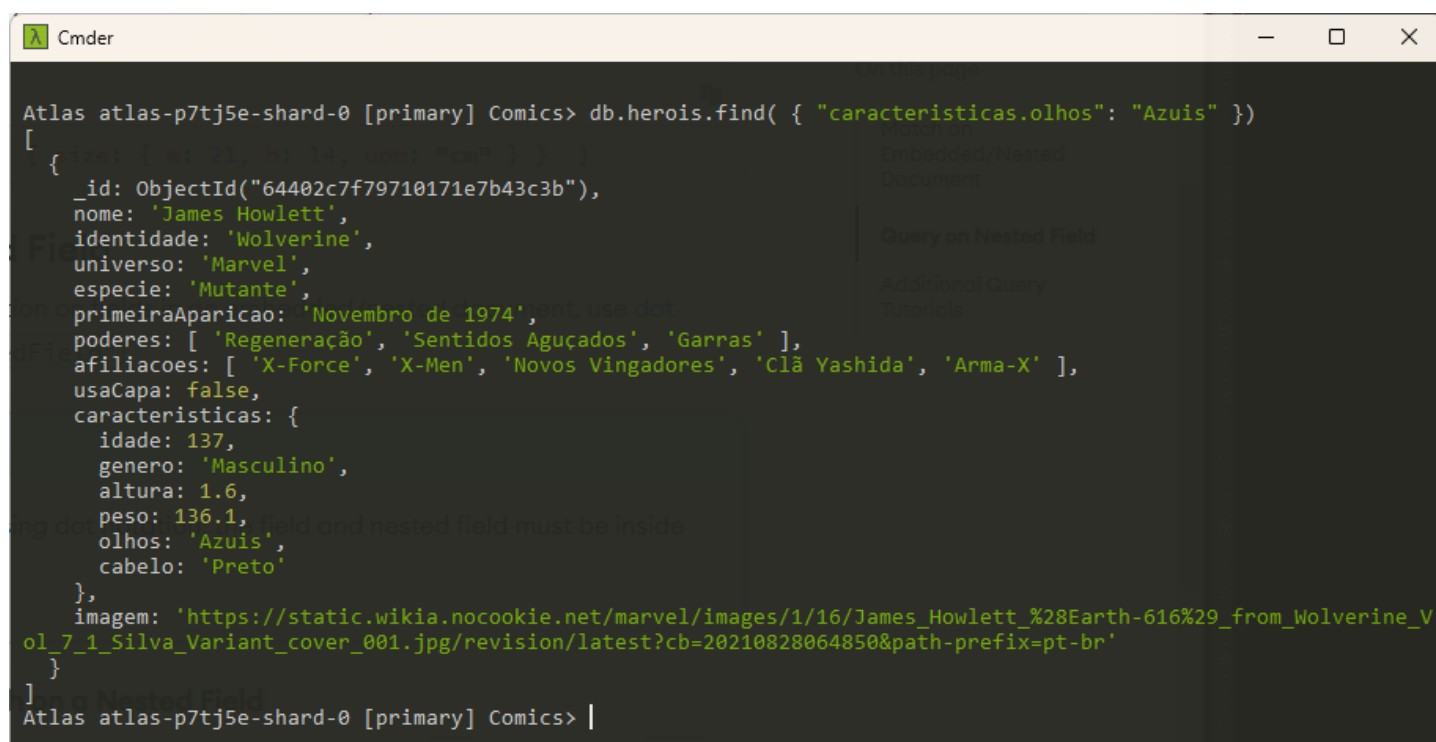
Este tipo de consulta, onde não especificamos um parâmetro é o correspondente no **SQL** ao:

```
SELECT * FROM heróis WHERE especie like "%Humano%"
```

## Operação de Leitura – Pesquisando em Campos de Documentos Embedded

Também podemos aplicar os filtros e operadores em campos que estão incorporados (**Embedded**) através de outros documentos. No caso do nosso exemplo de heróis temos o campo características que é um documento, com seus próprios campos.

Para pesquisar em campos de documentos incorporados, precisamos usar a notação “<campos>.<campos-incorporado>” inclusive com o uso das aspas.



```
Atlas atlas-p7tj5e-shard-0 [primary] Comics> db.herois.find( { "caracteristicas.olhos": "Azuis" })
[
  {
    _id: ObjectId("64402c7f79710171e7b43c3b"),
    nome: 'James Howlett',
    identidade: 'Wolverine',
    universo: 'Marvel',
    especie: 'Mutante',
    primeiraAparicao: 'Novembro de 1974',
    poderes: [ 'Regeneração', 'Sentidos Aguçados', 'Garras' ],
    afiliacoes: [ 'X-Force', 'X-Men', 'Novos Vingadores', 'Clã Yashida', 'Arma-X' ],
    usaCapa: false,
    caracteristicas: {
      idade: 137,
      genero: 'Masculino',
      altura: 1.6,
      peso: 136.1,
      olhos: 'Azuis',
      cabelo: 'Preto'
    },
    imagem: 'https://static.wikia.nocookie.net/marvel/images/1/16/James_Howlett_%28Earth-616%29_from_Wolverine_Vol_7_1_Silva_Variant_cover_001.jpg/revision/latest?cb=20210828064850&path-prefix=pt-br'
  }
]
```

## Operação de Leitura – Operadores de Comparação Numérica

**Operador \$gt (greater than – maior que).** Permite pesquisar campos com valores maiores que o informado.

**Exemplo:** Exibir os heróis com mais de 50 anos de idade:



```
Atlas atlas-p7tj5e-shard-0 [primary] Comics> db.herois.find( { "caracteristicas.idade": { $gt: 50 } })
[
  {
    _id: ObjectId("64402c7f79710171e7b43c3b"),
    nome: 'James Howlett',
    identidade: 'Wolverine',
    universo: 'Marvel',
    especie: 'Mutante',
    primeiraAparicao: 'Novembro de 1974',
    poderes: [ 'Regeneração', 'Sentidos Aguçados', 'Garras' ],
    afiliacoes: [ 'X-Force', 'X-Men', 'Novos Vingadores', 'Clã Yashida', 'Arma-X' ],
    usaCapa: false,
    caracteristicas: {
      idade: 137,
      genero: 'Masculino',
      altura: 1.6,
      peso: 136.1,
      olhos: 'Azuis',
      cabelo: 'Preto'
    },
    imagem: 'https://static.wikia.nocookie.net/marvel/images/1/16/James_Howlett_%28Earth-616%29_from_Wolverine_Vol_7_1_Silva_Variant_cover_001.jpg/revision/latest?cb=20210828064850&path-prefix=pt-br'
  }
]
Atlas atlas-p7tj5e-shard-0 [primary] Comics>
```

No caso, foi exibido apenas o **Wolverine** que possui 137 anos, porém o **Hulk** possui **50 anos** exatamente, neste caso temos outro operador.

**Operador \$gte (greater than or equal – maior que ou igual).** Permite pesquisar campos com valores maiores ou iguais ao informado. **Exemplo:** Exibir os heróis que tenham 50 anos ou mais de idade:



```
Atlas atlas-p7tj5e-shard-0 [primary] Comics> db.herois.find( { "caracteristicas.idade": { $gte: 50 } }, { identidade: 1, "caracteristicas.idade": 1 })
[
  {
    _id: ObjectId("64402c7f79710171e7b43c3b"),
    identidade: 'Wolverine',
    caracteristicas: { idade: 137 }
  },
  {
    _id: ObjectId("64402cc579710171e7b43c3e"),
    identidade: 'Hulk',
    caracteristicas: { idade: 50 }
  }
]
Atlas atlas-p7tj5e-shard-0 [primary] Comics>
```

Usamos o segundo documento para especificar no resultado apenas a identidade e a idade.



**Operador \$lt (lesser than – menor que).** Permite pesquisar campos com valores menores que o informado. **Exemplo:** Exibir os heróis com menos de 40 anos de idade:

```
Atlas atlas-p7tj5e-shard-0 [primary] Comics> db.herois.find( { "caracteristicas.idade": { $lt: 40 } })
[
  {
    _id: ObjectId("64402cc579710171e7b43c3c"),
    nome: 'Peter Parker',
    identidade: 'Homem-Aranha',
    universo: 'Marvel',
    especie: 'Humano Geneticamente Modificado',
    primeiraAparicao: 'Agosto de 1962',
    poderes: [ 'Agilidade', 'Força', 'Sentido Aranha' ],
    afiliacoes: [ 'Vingadores', 'Novos Vingadores', 'Defensores' ],
    usaCapa: false,
    caracteristicas: {
      idade: 33,
      genero: 'Masculino',
      altura: 1.78,
      peso: 75.75,
      olhos: 'Preto',
      cabelo: 'Castanho'
    },
    imagem: 'https://static.wikia.nocookie.net/marveldatabase/images/d/da/Amazing_Spider-Man_Vol_5_15_T
extless.jpg/revision/latest?cb=20181121213108'
  }
]
Atlas atlas-p7tj5e-shard-0 [primary] Comics> |
```

No caso, foi exibido apenas o **Homem-Aranha** que possui 33 anos, porém o **Deadpool** possui **40 anos** exatamente, neste caso temos outro operador.

**Operador \$lte (lesser than or equal – menor que ou igual).** Permite pesquisar campos com valores menores ou iguais ao informado. **Exemplo:** Exibir os heróis que tenham 40 anos ou menos de idade:

```
Atlas atlas-p7tj5e-shard-0 [primary] Comics> db.herois.find( { "caracteristicas.idade": { $lte: 40 } },
{ identidade: 1, "caracteristicas.idade": 1 })
[
  {
    _id: ObjectId("64402cc579710171e7b43c3c"),
    identidade: 'Homem-Aranha',
    caracteristicas: { idade: 33 }
  },
  {
    _id: ObjectId("64402cc579710171e7b43c3d"),
    identidade: 'Deadpool',
    caracteristicas: { idade: 40 }
  }
]
Atlas atlas-p7tj5e-shard-0 [primary] Comics> |
```

Usamos o segundo documento para especificar no resultado apenas a identidade e a idade.

## **Operação de Leitura – Pesquisa em vetores**

Para pesquisas em campos do tipo vetor, é necessário