

## Bootcamp IGTI: Engenheiro de Dados

### Trabalho Prático

<b>Módulo 02</b>	<b>Armazenamento de Dados</b>
------------------	-------------------------------

#### Objetivos

Exercitar os seguintes conceitos trabalhados no módulo:

- ✓ Banco de dados NoSQL.

#### Enunciado

Vamos utilizar um arquivo de entrada para ser carregado no MongoDB. Esse arquivo tem informações sobre livros diversos. O objetivo é carregar esse arquivo e exercitar alguns comandos no MongoDB.

Basicamente, vamos carregar o conteúdo de um arquivo json em uma collection e, em seguida, praticar alguns comandos no MongoDB.

#### Atividades

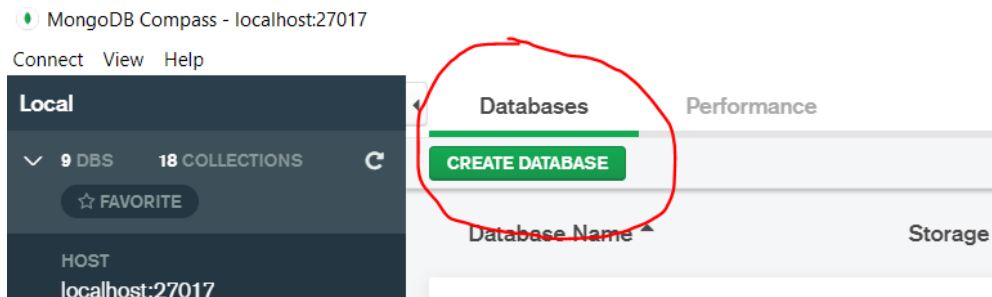
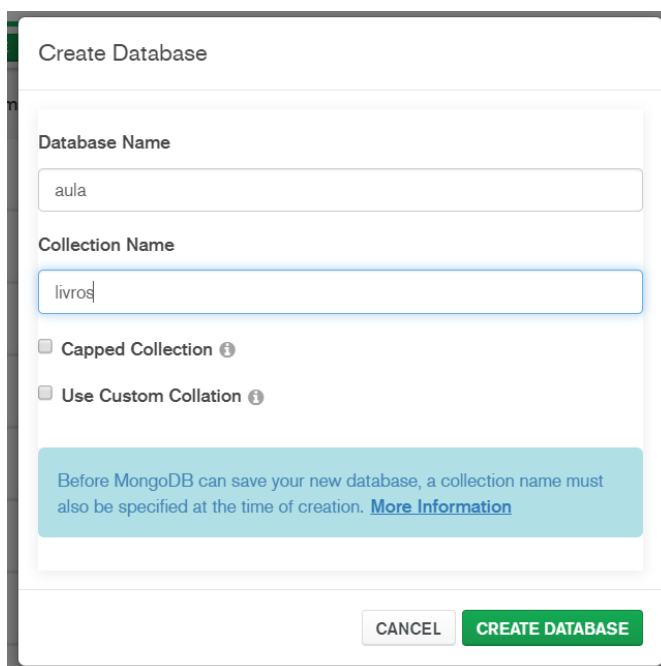
O primeiro passo é baixar o arquivo *“books.json”* do link abaixo ou pegá-los na plataforma do IGTI:

<https://drive.google.com/drive/u/1/folders/1GRFZlhjHFbzhcjDwq7EQXwH7fFj235QF>

- Você precisará criar um database chamado “aula”.
- Você precisará criar uma collection chamada “livros”.
- Vamos carregar o arquivo books.json na collection “livros”. Você pode fazer essa carga pelo MongoDBCompass ou pelo prompt de comando do MongoDB.

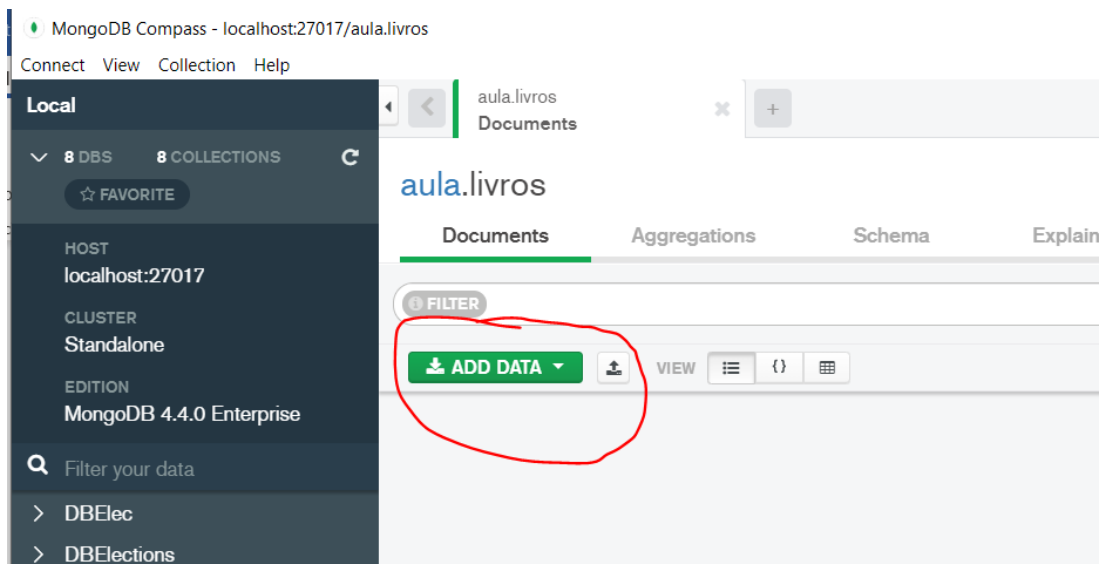
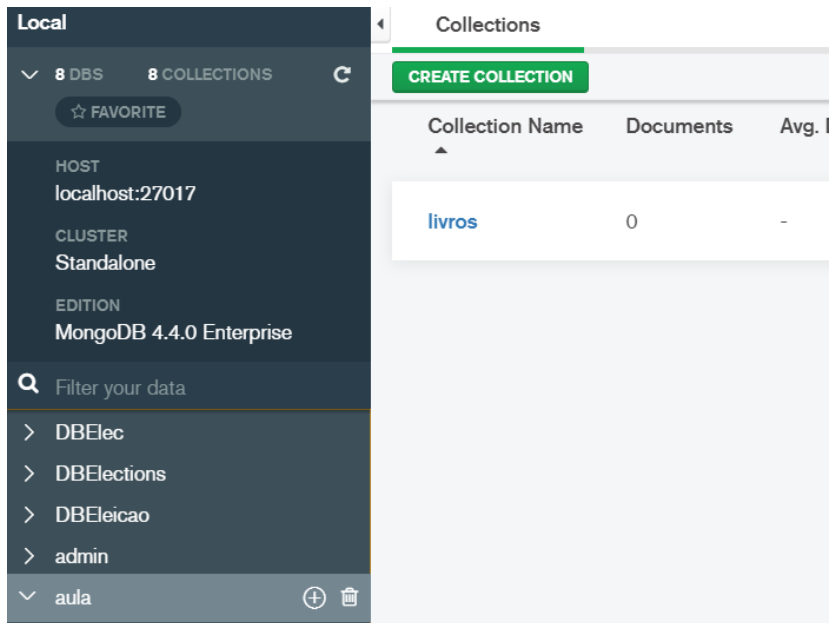
## Carga pelo MongoDBCompass

É possível criar o database e a collection.

The 'Create Database' dialog box is shown. It contains two input fields: 'Database Name' with the value 'aula' and 'Collection Name' with the value 'livros'. There are two checkboxes: 'Capped Collection' and 'Use Custom Collation', both of which are unchecked. A blue informational message states: 'Before MongoDB can save your new database, a collection name must also be specified at the time of creation. [More Information](#)'. At the bottom, there are two buttons: 'CANCEL' and 'CREATE DATABASE'.

Carga do json na collections: dentro do database, localize a collection e clique nela. Será aberta uma janela mostrando a collection aula.livro com um botão “add data”, que servirá para carregar o arquivo json.



## Carga pelo prompt de comando do MongoDB

Você precisará usar o executável *mongoimport.exe* que fica localizando dentro da pasta “bin” na instalação do MongoDB.

A partir da versão 4.4.0 do MongoDB, o *mongoimport* é lançado separadamente do *MongoDB Server*. Então, você precisa fazer uma instalação independente para o *mongoimport*.

Baixe em [Acesse o Centro de Download](#);

[https://www.mongodb.com/try/download/database-tools?tck=docs\\_databasetools](https://www.mongodb.com/try/download/database-tools?tck=docs_databasetools)

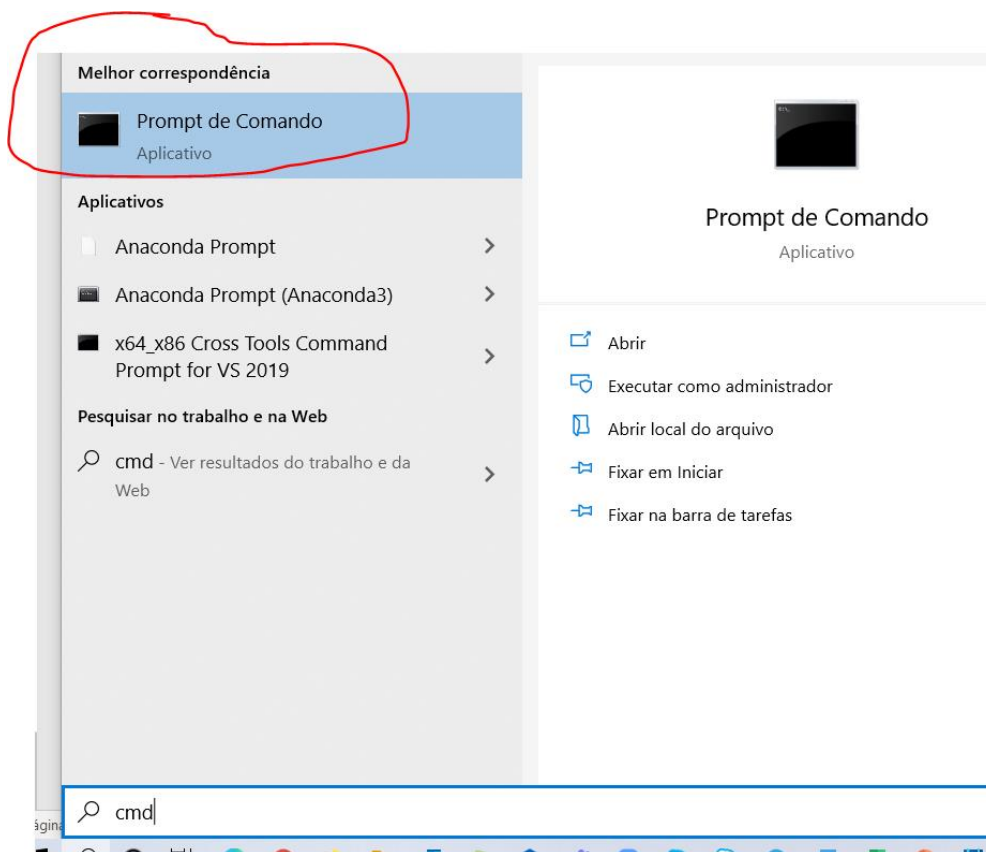
- Clique em Tools;
- Selecione plataforma (windows) e o package (zip)
- Clique em Download;

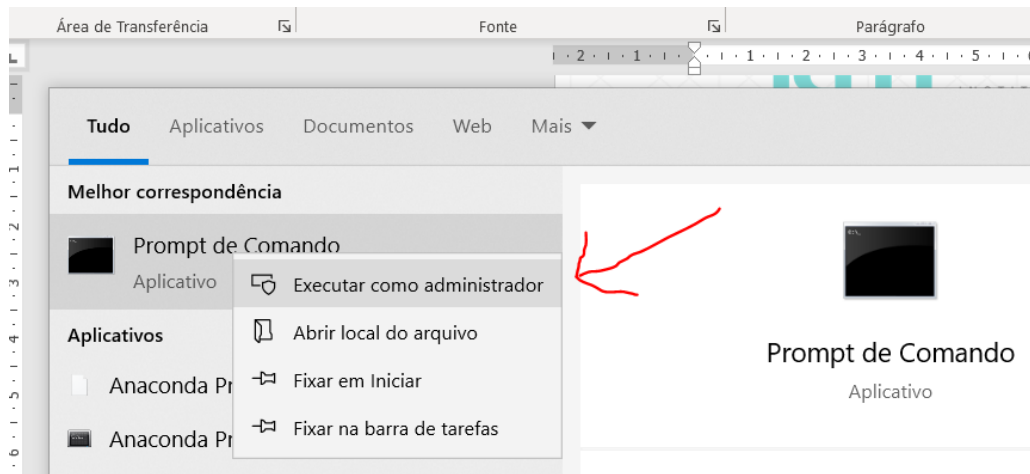
Extraia os arquivos do zip. São arquivos do tipo exe que você irá copiar para a pasta bin no diretório onde está o MongoDB (por exemplo: C:\Program Files\MongoDB\Server\4.4\bin).

## Interface de comando do MongoDB

O próximo passo é executar a interface de comando que nos permitirá interagir com o MongoDB.

Basta clicar na lupa à esquerda na barra de tarefas do Windows e digitar cmd ou command. Clique em cima dele com o botão direito para abrir como administrador.





- Abrir como administrador o prompt de comando.
- Mudar para o diretório "bin" onde o MongoDB está instalado (CD\Program Files\MongoDB\Server\4.4\bin).
- Digitar: **mongo**.

## Interface de comando do mongoimport

Vamos abrir outro terminal de comandos que nos permitirá interagir com o mongoimport.

Basta clicar na lupa à esquerda na barra de tarefas do Windows e digitar cmd ou command. Clique em cima dele com o botão direito para abrir como administrador.

- Abrir como administrador o prompt de comando.
- Mudar para o diretório "bin" onde o MongoDB está instalado (CD\Program Files\MongoDB\Server\4.4\bin).
- Digitar o seguinte comando. Nele, eu considere que o arquivo json estaria no diretório d:\temp. Você precisa considerar o caminho onde está armazenado o seu arquivo json.

```
mongoimport --db="aula" --collection="livros" --file="d:\temp\MongoDB\books.json"
```

```
D:\Program Files\MongoDB\Server\4.4\bin>mongoimport --db="aula" --collection="livros" --file="d:\Meus Documentos\Aulas\IGTI\Bootcamp EngDados\MongoDB\books.json"
2020-11-26T22:22:29.201-0300 connected to: mongodb://localhost/
2020-11-26T22:22:29.332-0300 431 document(s) imported successfully. 0 document(s) failed to import.
```

Vá no prompt do MongoDB e faça as práticas abaixo. **Anote todos os resultados,** pois você vai precisar deles ao responder o trabalho prático.

Digite os comandos conforme instruções abaixo.

1. Acesso ao database aula:

***use aula***

2. Checagem se há collections no database aula:

***show collections***

3. Checagem se a collections livros foi carregada:

***db.livros.find().count()***

O resultado será um número de títulos. **Anote esse valor.**

4. Você precisa consultar os livros com a tag isbn menor ou igual a "1000000000". Repare que essa tag tem um conteúdo string, e não numérico. O comando abaixo vai retornar a lista de livros. Para facilitar, você pode usar ***.pretty()***. Para facilitar mais ainda, você pode usar ***.count()***.

***db.livros.find({isbn:{\$lte: "1000000000"}})*** **Anote esse valor.**

5. **Agora, vamos consultar os livros com a tag isbn menor ou igual a "1617200000".**  
**Anote esse valor.**

6. Ainda na consulta dos livros com a tag isbn menor ou igual a "1617200000", recupere apenas os nomes dos livros.

Para facilitar, você pode usar o comando ***pretty()*** ou, para facilitar ainda mais, use algo similar ao comando ***db.customers.find({ }, {"name":1, "age":1})*** e, nesse caso, tire o ***pretty()***.

**Nessa lista retornada anote o título do livro começado com a letra G.**

7. Execute o comando abaixo para inserir mais 4 livros

```
db.livros.insertMany([
```

```
{ "title" : "Saci Pererê", "isbn" : "100", "pageCount" : 0, "publishedDate": ISODate("2000-10-01T07:00:00Z"), "status" : "PUBLISH", "authors" : [ ], "categories" : [ ] },
```

```
{ "title" : "A Cuca", "isbn" : "99", "pageCount" : 0, "publishedDate": ISODate("2000-10-01T07:00:00Z"), "status" : "PUBLISH", "authors" : [ ], "categories" : [ ] },
```

```
{ "title" : "Curupira", "isbn" : "98", "pageCount" : 0, "publishedDate": ISODate("2000-10-01T07:00:00Z"), "status" : "PUBLISH", "authors" : [ ], "categories" : [ ] },
```

```
{ "title" : "Jeca Tatu", "isbn" : "97", "pageCount" : 0, "publishedDate": ISODate("2000-10-01T07:00:00Z"), "status" : "PUBLISH", "authors" : [ ], "categories" : [ ] }
```

```
])
```

8. Execute o comando para saber quantos livros tem a chave isbn menor ou igual a "100000" (\$lte: "100000"). Lembre se de usar o ".count()" para facilitar o resultado.

**Anote o valor retornado.**

9. Considere o comando do passo anterior (8) – feito para saber quantos livros tem a chave **isbn** menor ou igual a "100000" (\$lte: "100000"). Quais são os 2 primeiros livros da lista? Dica: use o comando pretty() e limit().

**Anote o nome dos dois livros retornados.**

10. O que faz o comando abaixo? Ele mostra quantos livros, e porquê isso?

```
db.livros.find({isbn:{$lte: "100000"}}).pretty().skip(2)
```

**Anote o nome dos livros retornados.**

11. O que faz o comando abaixo? Ele mostra quantos livros, e porquê isso? Observe o que há em comum nos títulos dos livros.

```
db.livros.find({title: /Windows/}).count()
```

**Anote o número de livros retornados.**

12. No comando find vamos acrescentar o comando **sort descendente** (-1) para a chave **pageCount**.

Utilize o comando **limit = 2**, pois o que quero saber é quais são os dois menores valores para a chave **pageCount**.

A dica é usar o `find({ }, {"pageCount":1, "_id":0})` para trazer apenas a chave `pageCount` desejada.

**Anote os dois valores retornados para a chave “pageCount”.**



**Respostas Finais**

Os alunos deverão desenvolver a prática e, depois, responder às questões objetivas.