

# mongoDB

ELISA DE FÁTIMA ANDRADE SOARES  
([efas@cin.ufpe.br](mailto:efas@cin.ufpe.br))

# Resumo da Aula 1

- Saber instalar o **mongoDB** e criar um banco de dados;
- Criar e remover coleções e dbs através do Shell;
- Conseguir conectar e manipular dados através do **Shell**;
- Ter uma ideia a respeito do funcionamento do **Compass**;
- Poder **criar coleções** e **inserir** documentos na base de dados
- Conseguir consultar e filtrar **documentos**.

# MongoDB

- Origem: Humongous;
- Significa **Gigantesco**.

Alguém conhece MongoDB?

Sim! Já trabalhou?

# MongoDB

- O que é?

Banco de dados (BD) não relacional orientado a **documentos**

- Não relacional?
- Não existe a necessidade de criar uma estrutura de tabelas e dados antes de começar a inserir informações

# MongoDB

- Criado pelos fundadores da **DoubleClick**
- Empresa de anúncios da internet comprada pelo Google
- Criada em 2007 como “10gen”. Em 2013 foi alterado o nome para MongoDB Inc;
- É um dos bancos NoSQL mais populares do mundo;
- Baixa curva de aprendizagem;
- Erros (Tentar implementar pensando relacional).

# MongoDB

## Estrutura Documento:

```
{  
  title: "Avengers: Endgame",  
  director: "Anthony Russo",  
  year: 2016,  
  actors: ["Robert Downey Jr", "Chris Evans", "DonCheadle"],  
  imdb: {  
    id: "tt5921884",  
    rating: 8.5,  
    votes: 15160  
  }  
}
```

# Instalando o MongoDB

Select the server you would like to run:

## MongoDB Community Server

FEATURE RICH. DEVELOPER READY.

## MongoDB Enterprise Server

ADVANCED FEATURES. PERFORMANCE GRADE.

Version

4.0.10 (current release)

OS

Windows 64-bit x64

Package

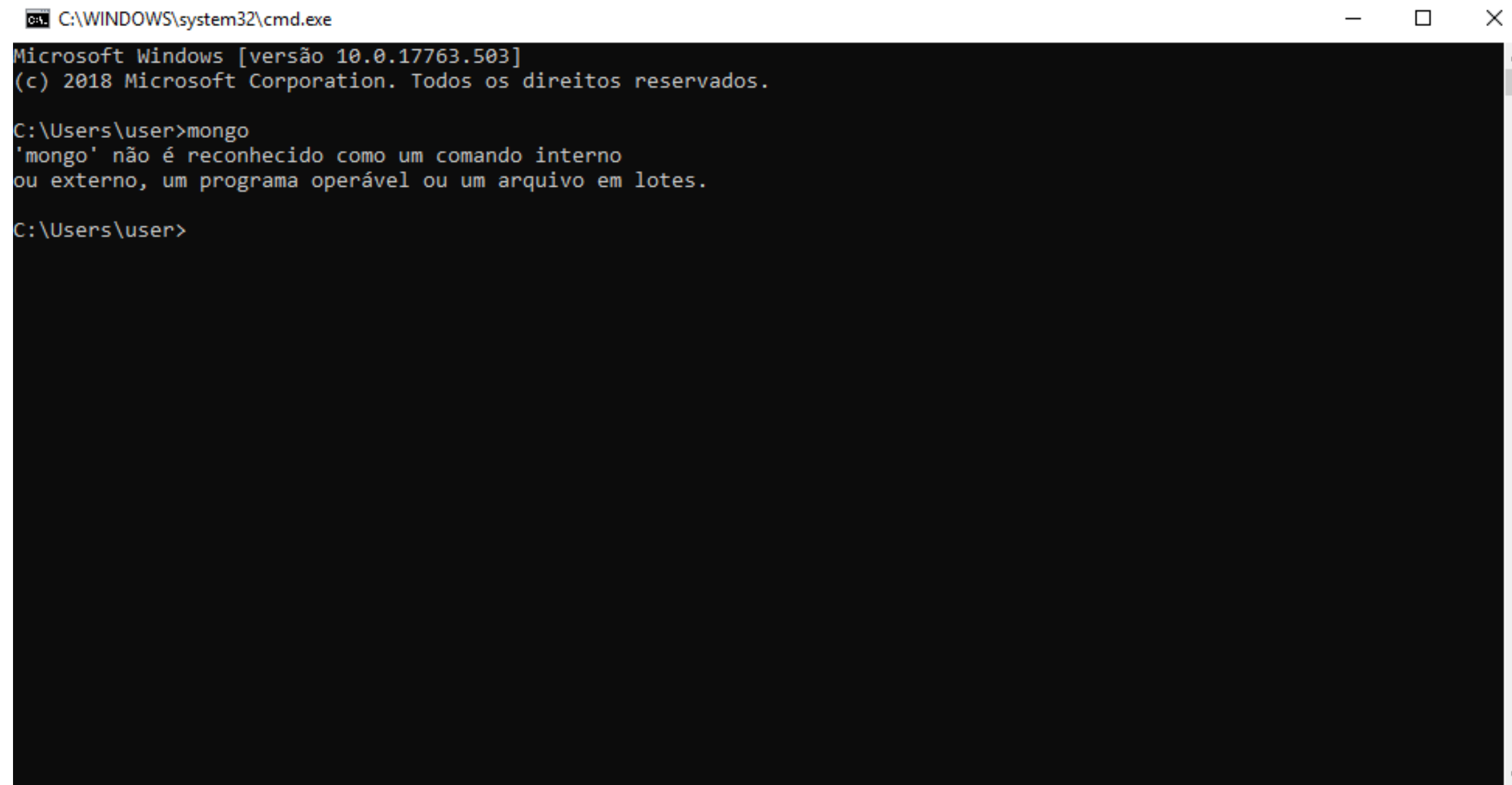
MSI

Download

[https://fastdl.mongodb.org/win32/mongodb-win32-x86\\_64-2008plus-ssl-4.0.10-signed.msi](https://fastdl.mongodb.org/win32/mongodb-win32-x86_64-2008plus-ssl-4.0.10-signed.msi)

- Release notes
- Changelog
- All version binaries
- Installation instructions
- Download source (tgz)
- Download source (zip)

# WINDOWS



```
cmd C:\WINDOWS\system32\cmd.exe
Microsoft Windows [versão 10.0.17763.503]
(c) 2018 Microsoft Corporation. Todos os direitos reservados.

C:\Users\user>mongo
'mongo' não é reconhecido como um comando interno
ou externo, um programa operável ou um arquivo em lotes.

C:\Users\user>
```



# WINDOWS

Após o processo de download:

Configurar a variável PATH para a pasta onde foi instalado!!!

- 1) Descobrir onde o mongo foi instalado;
- 2) Configurar a variável PATH para a pasta onde foi instalado.



Dúvida??? → <https://www.youtube.com/watch?v=kgpTIUoET3w>

# MAC

Através do gerenciador de pacotes Homebrew

- 1) No terminal, executar **brew tap mongodb/brew**
- 2) Executar `brew install mongodb-community@4.0`
- 3) De preferência, baixar e instalar o Compass
- 4) Ir no arquivo `.bash_profile` na sua pasta raiz e adicionar o diretório na variável `PATH`:

`.../MongoDB/mongodb-osx-x86_64-4.0.9/bin`

Você pode iniciar o servidor através do comando:

`mongod --config /usr/local/etc/mongod.conf`

Lembre-se de baixar e instalar o **Compass**!

# LINUX

Através do gerenciador de pacotes Homebrew

- 1) No terminal, executar **brew tap mongodb/brew**
- 2) Executar `brew install mongodb-community@4.0`
- 3) De preferência, baixar e instalar o Compass
- 4) Ir no arquivo `.bash_profile` na sua pasta raiz e adicionar o diretório na variável PATH:

`.../MongoDB/mongodb-osx-x86_64-4.0.9/bin`

Você pode iniciar o servidor através do comando:

`mongod --config /usr/local/etc/mongod.conf`

Lembre-se de baixar e instalar o **Compass**!

# Utilizando o Compass

- Abrir o **Compass**
  - Conectar ao localhost
    - Criar um BD;
    - Criar uma coleção;
    - Realizar uma inserção.

# Carregando dados

- Baixar o arquivo **loadMovieDetailsDataSet.js**



GOOGLE CLASSROOM

- Coloque-a em um caminho de fácil acesso
- Abrir o prompt de comando ou terminal

`cd <pasta do arquivo>`

- local: rodar o comando **mongo**
- Rodar o comando **load("loadMovieDetailsDataset.js")**

# Conferindo o Carregamento

- Abrir o **Compass**
  - Conectar ao localhost
  - Selecionar **video.movieDetails**

The screenshot shows the MongoDB Compass interface. On the left, the 'My Cluster' sidebar lists databases and collections. The 'video' database is expanded, showing the 'movieDetails' collection. The main panel displays the 'video.movieDetails' collection with 2.3k documents, 1.5MB total size, and 688B average size. The 'Documents' tab is active, showing a list of documents. The first document is displayed in JSON format:

```
{
  "_id": ObjectId("5cdc17e76ba069e4ab283b1e"),
  "title": "Once Upon a Time in the West",
  "year": 1968,
  "rated": "PG-13",
  "runtime": 175,
  "countries": Array,
  "genres": Array,
  "director": "Sergio Leone"
}
```

# Conferindo o Carregamento

- Visão **Schema**
  - Selecionar video.movieDetails e navegar para a aba Schema.

My Cluster

4 DBS2 COLLECTIONS

filter

> admin

> config

> local

> video

movieDetails

localhost:27017STANDALONE

video.movieDetails

DOCUMENTS 2.3kTOTAL SIZE 1.5MBAVG. SIZE 688BINDEXES 1TOTAL SIZE 32.0KB AVG. SIZE 32.0KB

DocumentsAggregations

Sample of 1000 documents (4)

\_id

objectid

actors

array

string

> awards

document

countries

array

string

Calendar

SMTWTFSS

0:006:0012:0018:0023:00

Timeline

inserted: 20%

Summary

Array lengths

min: 0 average: 3.29 max: 4

Document with 3 nested fields.

Array lengths

min: 0 average: 1.17 max: 6

Nome qualificado  
database.collection

Lista de campos e tipos  
Esquema flexível

Estatísticas  
Quantidade de docs, tamanho, índices

Resumo do conjunto de dados  
Baseado em amostra



# Utilizando o Compass



## Atividade Prática!

5 Min?

- Quantos documentos em video.movieDetails correspondem ao filtro {genres: "Comedy"}?
- Quantos filmes na coleção de video.movieDetails foram dirigidos por Robert Rodriguez. Em termos mais preciso, quantos documentos na coleção video.movieDetails têm o valor : "Robert Rodriguez" para o campo director?

# Criando dbs

- **Criando uma nova db**
- Primeiro alternar para a db que você quer criar
  - **use db**
- Depois, criar uma coleção nessa db
  - `db.createCollection("nome", opções...)`
- Em seguida, verifique se foi tudo criado através de
  - `show dbs`
  - `show collections`

# Inserção

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

# Inserção

- **InsertOne ({documento})**
  - método insere somente um documento
- O campo **\_id** sempre está presente nos documentos
  - É como se fosse uma chave primária (identificar único)
- Não é necessário especificar um valor, o mongo já atribui um por padrão.
  - Mas é possível!
- Os valores atribuídos automaticamente são do tipo **objectId**

# Inserindo documentos em uma coleção



## Atividade Prática!

10 Min?

- Inserir o filme Aquarius na coleção movieDetails via **shell**
  - director: Kleber Mendonça Filho
  - year: 2016
  - actors: Sônia Braga, Maeve Jinkings, Irandhir Santos
  - imdb:
    - id: tt5221584
    - rating: 7.5
    - votes: 15160

# Inserindo documentos em uma coleção

- **InsertMany([{documento1}, {documento2}...], {ordered:true/false})**
  - Método insere múltiplos documentos

# Inserindo documentos em uma coleção



## Atividade Prática!

10 Min?

- Se a coleção video.myMovies estiver vazia no momento, quantos documentos seriam inseridos pela seguinte chamada com insertMany()?

```
db.myMovies.insertMany(  
  [  
    {  
      "_id" : "tt0084726",  
      "title" : "Star Trek II: The Wrath of Khan",  
      "year" : 1982,  
      "type" : "movie"  
    },  
    {  
      "_id" : "tt0796366",  
      "title" : "Star Trek",  
      "year" : 2009,  
      "type" : "movie"  
    },  
    {  
      "_id" : "tt0084726",  
      "title" : "Star Trek II: The Wrath of Khan",  
      "year" : 1982,  
      "type" : "movie"  
    },  
    {  
      "_id" : "tt1408101",  
      "title" : "Star Trek Into Darkness",  
      "year" : 2013,  
      "type" : "movie"  
    },  
    {  
      "_id" : "tt0117731",  
      "title" : "Star Trek: First Contact",  
      "year" : 1996,  
      "type" : "movie"  
    }  
  ],  
  {  
    ordered: false  
  }  
);
```

# Busca

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



# Buscando documentos em uma coleção

- Método **find({"campo": "valor"})**
  - Busca registros que satisfazem aos critérios postos
  - **find( )** = SELECT \* FROM ...
- Para utilizar o find() em **nested objects** podemos utilizar a notação de pontos
  - db.movieDetails.find({"**imdb.id**" : "tt0064116"})

```
plot: "Epic story of a mysterious stranger with a harmonica who joins forces ..."  
poster: "http://ia.media-imdb.com/images/M/MV5BMTEyODQzNDkzNjVeQTJJeQWpwZ15BbWU4..."  
✓ imdb: Object  
    id: "tt0064116"  
    rating: 8.6  
    votes: 201283  
> tomato: Object  
    metacritic: 80  
> awards: Object  
    type: "movie"
```

# Buscando documentos em uma coleção

- Filtrando **arrays**

- Podemos filtrar os documentos baseados em arrays de diversas maneiras:

- Correspondência com o **array completo**

- `db.movieDetails.find({actors: ["Matt Damon", "Jessica Chastain", "Kristen Wiig", "Jeff Daniels"]})`

- Correspondências com **algum elemento** do Array

- `db.movieDetails.find({actors: "Matt Damon"})`

- Valores em uma **posição específica**

- `db.movieDetails.find({"actors.0": "Matt Damon"})`

# Buscando documentos em uma coleção

- **Trabalhando com o cursor**
- A query mostra os primeiros 20 resultados
- Podemos iterar sobre o cursor utilizando o comando **it**
- Podemos formatar os resultados utilizando o método **.pretty()** ao final do comando
- Podemos também usar o método **.count()** para retornar a quantidade de registros encontrados pelo find .
- Também é possível ordenar os resultados através do **.sort()**
  - 1 para ordem crescente, -1 para ordem decrescente

# Buscando documentos em uma coleção



## Atividade Prática!

10 Min?

- Quantos filmes possuem Jeff Bridges no elenco?
- Quantos filmes possuem Jeff Bridges como protagonista?
- Qual o filme teve o seguinte elenco?

```
✓ actors: Array  
  0: "Leonardo DiCaprio"  
  1: "Mark Ruffalo"  
  2: "Ben Kingsley"  
  3: "Max von Sydow"
```

# Buscando documentos em uma coleção

- **Projeção**
- Nem sempre queremos retornar todos os documentos em um find
- Podemos utilizar um segundo argumento na função para realizar uma projeção, ou seja, selecionar **somente alguns campos**
- `db.movieDetails.find({genres: ["Mystery","Thriller"]}, {title: 1})`
  - 1: retornar o campo; 0: não retornar
- A projeção tem alguns comportamentos diferentes dependendo dos parâmetros utilizados

# Buscando documentos em uma coleção

- Execute as seguintes consultas e observe os resultados:
  - `db.movieDetails.find({genres: ["Mystery", "Thriller"]}, {title: 1})`
  - `db.movieDetails.find({genres: ["Mystery", "Thriller"]}, {title: 1, _id: 0})`
  - `db.movieDetails.find({genres: ["Mystery", "Thriller"]}, {title: 0, _id: 0})`
  - `db.movieDetails.find({genres: ["Mystery", "Thriller"]}, {title: 1, genres: 0})`

# Buscando documentos em uma coleção

- **\$lt** - Retorna documentos com valores menores que o valor especificado
- `db.colecao.find({ "campo" : { $lt: valor } } );`
- **\$lte** - Retorna documentos com valores menores ou igual que o valor especificado.
- `db.colecao.find({ "campo" : { $lte: valor } } );`
- **\$gt** – Retorna documentos com valores maiores que o valor especificado
- `db.colecao.find({ "campo" : { $gt: valor } } );`
- **\$gte** - Retorna documentos com valores maiores ou igual que o valor especificado.
- `db.colecao.find({ "campo" : { $gte: valor } } );`

# Buscando documentos em uma coleção

- **Operadores de comparação**
- Até agora só buscamos elementos que apresentavam uma correspondência exata com algum valor
- Muitas vezes, contudo, será necessário comparar valores de formas diferentes:
- `db.movieDetails.find({metacritic:{$gt:80}})`
- `db.movieDetails.find({metacritic:{$lt:30}})`
- `db.movieDetails.find({metacritic:{$gte:80}})`
- `db.movieDetails.find({metacritic:{$lte:30}})`
- Esses operadores funcionam em **arrays** da mesma forma que as igualdades



# Operadores de Comparação

- **\$in**
  - Retorna os valores que encontram correspondência com algum dos elementos fornecidos
  - `db.movieDetails.find({writers:{$in:["Ethan Coen", "Joel Coen"]}})`
- **\$nin** - not in
  - Comportamento oposto ao \$in
- **\$eq**
  - Operador de igualdade, equivalente a `{campo:valor}`
- **\$ne**
  - "Not equal"
  - Retorna aqueles documentos cujo valor do campo é diferente do fornecido.
  - Também retorna os documentos que **não possuem o campo** ou que possuem o valor **null**

# Operadores Lógicos

- **\$or**
  - `db.movieDetails.find({$or: [{writers: "Ethan Coen"}, {writers: "Joel Coen"}]})`
- **\$and**
  - `db.movieDetails.find({$and: [{"awards.wins": 2}, {"awards.nominations": 2}]})`
- **\$not**
  - `db.movieDetails.find({writers: {$not: {$eq: "elisa" }}}) = $ne`
- **\$nor**
  - `db.movieDetails.find({$nor: [{actors: "Elisa"}, {writers: "Valeria"}]})`

# Buscando documentos em uma coleção



## Atividade Prática!

15 Min?

- Quantos filmes em que Mark Hamill atuou não foram dirigidos por George Lucas?
- Quantos filmes desde 2010 possuem tomatometer ou metacritic acima de 75?
- Há mais filmes longos (a partir de 90 minutos) na década de 80 ou de 90?
- Sem utilizar o operador **\$or**, verifique quantos filmes possuíam classificação “PG-13” ou “R”.

# Operadores de Elemento

- “Perguntam” a respeito do esquema dos documentos
- **\$exists**
  - `db.movieDetails.find({metacritic: {$exists: true}})`
- **\$type**
  - `db.movieDetails.find({metacritic: {$type: “double”}})`

# Operadores em Arrays

- **\$all**

- Retorna aqueles documentos cujo array contém todos os elementos especificados

- db.movieDetails.find({genres: {\$all: ["Comedy", "Drama"]}})

- **\$size**

- Retorna todos documentos cujo array contém a quantidade de elementos definida

- db.movieDetails.find({countries: {\$size: 1}})

- db.movieDetails.find({countries: {\$size: 3}})

# Buscando documentos em uma coleção



## Atividade Prática!

15 Min?

- Antes de tudo, carregue o arquivo **loadReviewsDataset.js**
- Na **coleção reviews**:
  - Quantos documentos da coleção reviews possuem o campo rating do tipo int?
  - Quantos documentos da coleção reviews não possuem o campo text definido?
- Na **coleção movieDetails**:
  - Quais os títulos dos filmes que têm Harrison Ford e Mark Hamill no elenco, em qualquer ordem?
  - Com os operadores que já vimos, levante quantos filmes possuem mais de um autor (campo writers).

# OBRIGADA!



[efas@cin.ufpe.br](mailto:efas@cin.ufpe.br)