



## Mãoz na massa: Analisando um pouco os dados

Chegou a hora de você executar o que foi visto na aula! Para isso, execute os passos listados abaixo.

- 1) Para restaurar a base de dados e coleção do MongoDB, primeiramente disponibilize o seu servidor, por exemplo:

```
start /b mongod --dbpath D:\MongoDB\data\db --logpath D:\MongoDB\log.txt --oplogSize 50 --smallfiles
```

- 2) Baixe [aqui](https://s3.amazonaws.com/caelum-online-public/787-data-science-coleta-de-dados/03/clima bson) (<https://s3.amazonaws.com/caelum-online-public/787-data-science-coleta-de-dados/03/clima bson>) o arquivo **clima bson**, e restaure a coleção através deste arquivo, por exemplo:

```
mongorestore --db dbclima --collection clima "D:\MongoDB\bkp_clima\dbclima\clima bson"
```

- 3) Restaurada a coleção, você pode fazer consultas à mesma, por exemplo, para ver quantos documentos há na coleção:

```
mongo dbclima --eval "db.clima.count()" --quiet
```

- 4) Para identificar o nome dos campos que estão sendo utilizados no MongoDB e no Wunderground, selecione o primeiro documento da coleção **clima** e visualize-o. Através do **Robo 3T (Robomongo)**, faça:

```
db.clima.findOne()
```

E no **Wunderground**, procure pelos dados históricos do documento retornado pela consulta acima. Caso você queira, pode acessar os dados desse documento diretamente [aqui](https://www.wunderground.com/history/airport/KIGQ/2004/10/31/DailyHistory.html?req_city=Tinley+Park&req_state=IL&req_statename=Illinois&reqdb.zip=60477&reqdb.magic=1&reqdb.wmo=99999) ([https://www.wunderground.com/history/airport/KIGQ/2004/10/31/DailyHistory.html?req\\_city=Tinley+Park&req\\_state=IL&req\\_statename=Illinois&reqdb.zip=60477&reqdb.magic=1&reqdb.wmo=99999](https://www.wunderground.com/history/airport/KIGQ/2004/10/31/DailyHistory.html?req_city=Tinley+Park&req_state=IL&req_statename=Illinois&reqdb.zip=60477&reqdb.magic=1&reqdb.wmo=99999)).

## Analisando os dados

- 5) Agora, analise os dados, fazendo algumas consultas no MongoDB, através do **Robo 3T (Robomongo)**. Primeiramente, conte quantos documentos existem na coleção:

```
db.clima.count()
```

- 6) Na próxima consulta, veja as cidades dos documentos com ids entre **8442** e **8447**:

```
db.clima.find (  
  { $and : [ { posicao : { $gte : 8442 } }, { posicao : { $lte : 8447 } } ] },  
  { _id : 0, posicao : 1, estado : 1, cidade : 1 }  
)
```

Perceba a ausência do documento de id 8445.

7) Agora, veja os últimos 10 documentos gravados na coleção:

```
db.clima.find(
  { },
  { _id : 0 , posicao : 1, cidade : 1, estado : 1, "history.date" : 1}
).sort ( { posicao : -1 }). limit (10)
```

8) Você também consegue ver quantos documentos não possuem histórico preenchido:

```
db.clima.count()
db.clima.find (
  { history : { $exists : 1 } },
  { posicao : 1, estado : 1 }
).count()
```

Um exemplo de documento sem histórico é o de id **8464**, cidade **Parma Heights**, estado **OH**, e data **20151027**.

9) Ainda mais, veja os anos com mais observações:

```
db.clima.aggregate ( [
  { $match: { history : { $exists : 1 } } }, // eliminando quem não tem história
  { $group : { _id : "$history.utcdate.year", quantos : { $sum : 1} } },
  { $match : {quantos: {$gt: 500} } },
  { $sort: {quantos: -1} }
] )
```

10) Continuando com a análise, veja quantos dados de clima há em cada estado:

```
db.clima.aggregate ( [
  { $group : { _id : "$estado", quantos : { $sum : 1} } },
  { $sort: {quantos: -1} }
])
```

11) Agora, veja a quantidade de dados de clima por cidade:

```
db.clima.aggregate ( [
  { $group : { _id : {estado: "$estado", cidade: "$cidade"}, quantos : { $sum : 1} } },
  { $sort: {quantos: -1} }
])
```

12) Visualize agora se choveu ou não, nos documentos com histórico, de Phoenix, Arizona:

```
db.clima.find (
  { history : { $exists : 1 }, estado : "AZ", cidade : "Phoenix" },
  { posicao : 1, estado : 1, "$history.utcdate.year" : 1, "history.dailysummary.rain" : 1 }
)
```

13) Por fim, em quantos dias choveu em Phoenix, Arizona?

```
db.clima.find (
  { "history.dailysummary.rain" : "1", estado : "AZ", cidade : "Phoenix" },
  { posicao : 1, estado : 1 }
).count()
```