

Ordenado e limitando os dados

Buscando com desigualdade e buscando um único elemento

Vamos explorar os dados de nossos alunos. Aprenderemos diversas informações sobre as *query* do *MongoDB*. Vamos buscar os alunos com as notas mais altas. Se quisermos buscar alunos que possuam uma nota específica, por exemplo, 8.5, podemos usar uma *query* com o campo "notas" e o `db.alunos.find` :

```
db.alunos.find(
  {
    "notas" : 8.5
  })
```

Agora, para encontrar alunos, por exemplo, com notas maiores do que cinco usaremos o `find` e teremos que passar um objeto de *javascript*. Usaremos um operador, o greater than, o `gt`. Ficaremos com:

```
db.alunos.find({
  notas : { $gt : 5 }
})
```

Basta uma das notas ser maior do que 5 que ele nos mostrará o aluno.

Primeiro, passamos um valor fixo. Segundo, passamos um conjunto de coisas, isto é, um dicionário de operações que é rodado para cada um dos valores.

Vamos adicionar mais dois alunos, o "André" e a "Lúcia", com suas respectivas informações. Usaremos o `insert` :

```
db.alunos.insert({
  nome : "André",
  data_nascimento : new Date(1991,01,25),
  curso : {
    nome : "Matemática"
  },
  notas : [ 7, 5, 9, 4.5 ]
})

db.alunos.insert({
  nome : "Lúcia",
  data_nascimento : new Date(1984,07,17),
  curso : {
    nome : "Matemática"
  },
  notas : [ 8, 9.5, 10 ]
})
```

O `find` traz um registro de diversas coisas. Se quisermos visualizar apenas uma coisa podemos usar o `findOne`. Se quisermos encontrar um aluno cuja nota seja maior do que 5 digitaremos o seguinte:

```
db.alunos.findOne({  
  notas : { $gt : 5}  
})
```

Buscando ordenado e limitando a quantidade de elementos do resultado

Temos nossa lista de alunos que podemos consultar usando o `db.alunos.find()` .

Para pedir que essa lista venha em ordem alfabética podemos usar o `sort` para ordenar os elementos. Passaremos um objeto dizendo para organizar por campo `"nome"` e que essa ordem seja crescente `1` :

```
db.alunos.find().sort({"nome" : 1})
```

Rodando isso teremos os nomes em ordem alfabética, podemos, ainda utilizar o `-1` para que os nomes venham na ordem contrária:

```
db.alunos.find().sort({"nome" : -1})
```

Além de ordenar podemos, também, pedir apenas os três primeiros nomes. Isto é, limitar o número de alunos que queremos visualizar:

```
db.alunos.find().sort({"nome" : 1}).limit(3)
```

Quando queremos buscar algo com uma ordem específica usamos o `sort` . Quando queremos que essa ordem seja limitada, utilizamos o `limit` para delimitar o tamanho que bem entendermos.

Temos, ainda, diversos operadores que podemos utilizar. O importante é compreender que trabalhamos com objetos de tipo *javascript* para os quais podemos falar que desejamos analisar uma parte do objeto, documento ou que gostaríamos de alterar determinada parte do objeto e etc.