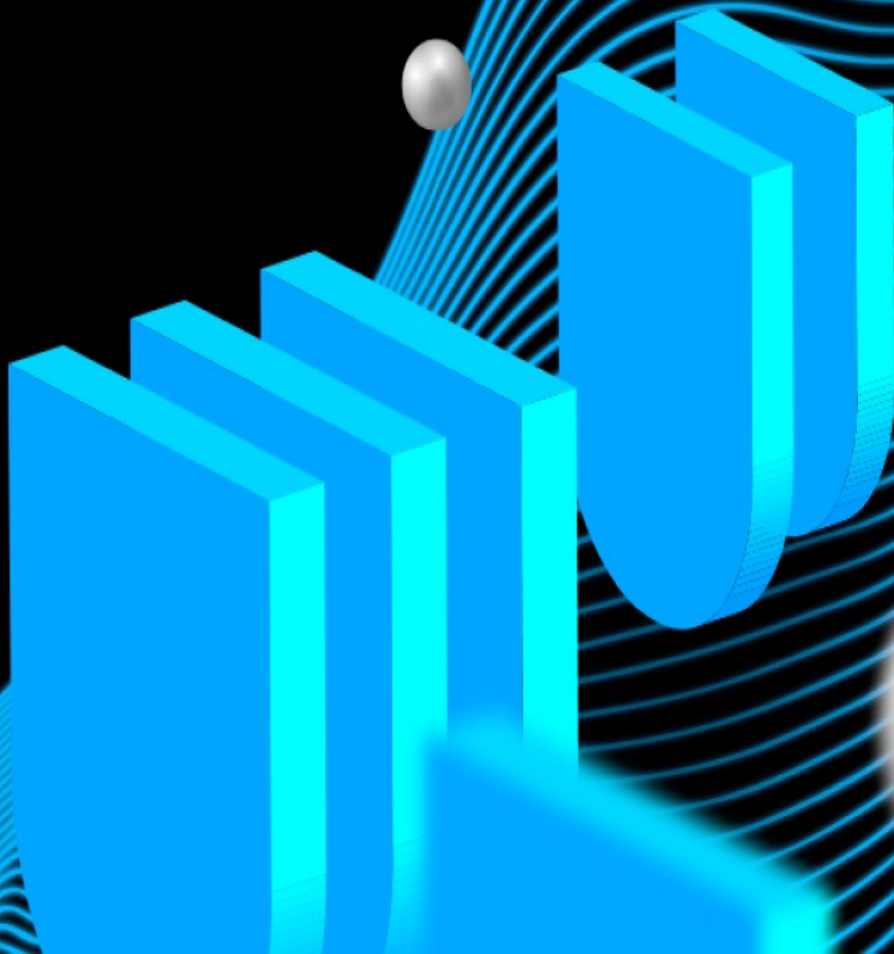


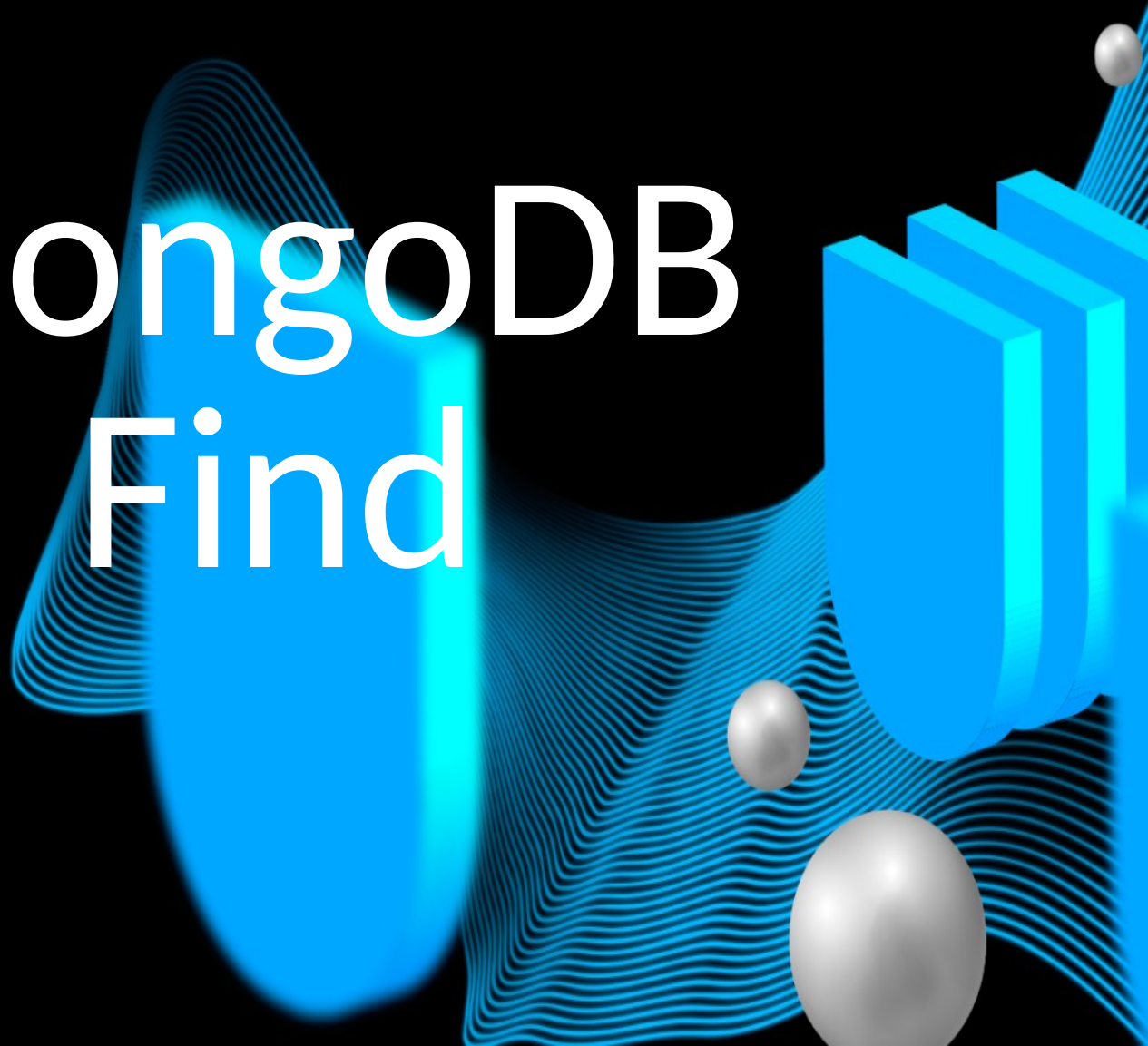


**UNITAU**  
Universidade de Taubaté





# MongoDB Find



# Pesquisando Documentos



Para recuperar documentos é utilizado o método **db.find()**.

**db.collection.find(query, projection)**



# Pesquisando Documentos



Para recuperar a coleção inteira, o método deve ser usado sem parâmetros:

```
> db.produtos.find()  
{ "_id" : 1, "item" : "caderno", "quantidade" :  
20 }  
{ "_id" : 2, "item" : "cartao", "quantidade" :  
15, "minimo" : 5 }  
{ "_id" : 3, "item" : "envelope", "quantidade" :  
20, "minimo" : 10 }  
{ "_id" : 4, "item" : "selos", "quantidade" : 30,  
"minimo" : null }  
{ "_id" : 5, "item" : "lapis", "quantidade" : 50,  
"minimo" : 30 }  
{ "_id" : 6, "item" : "borracha", "quantidade" :  
25, "minimo" : 10 }
```

# Pesquisando Documentos



Para pesquisar documentos com um campo com um determinado valor é utilizada a seguinte expressão:

```
> db.produtos.find( {minimo: 10} )  
{ "_id" : 3, "item" : "envelope", "quantidade" :  
20, "minimo" : 10 }  
{ "_id" : 6, "item" : "borracha", "quantidade" :  
25, "minimo" : 10 }
```

# Pesquisando Documentos



Comparar um campo com o valor **null** também retorna documentos que não tenham o campo:

```
> db.produtos.find( {minimo: null} )  
{ "_id" : 1, "item" : "caderno", "quantidade" :  
20 }  
{ "_id" : 4, "item" : "selos", "quantidade" : 30,  
"minimo" : null }
```



# Pesquisando Documentos



O MongoDB possui os seguintes operadores de comparação:

- **\$eq** - igual
- **\$ne** - diferente
- **\$gt** - maior
- **\$gte** - maior ou igual
- **\$lt** - menor
- **\$lte** - menor ou igual
- **\$in** - valor igual a um dos elementos do array
- **\$nin** - valor diferente de todos elementos do array

# \$eq



```
{ <field>: { $eq: <value> } }
```

```
> db.produtos.find( {quantidade: {$eq: 20}} )  
{ "_id" : 1, "item" : "caderno", "quantidade" :  
20 }  
{ "_id" : 3, "item" : "envelope", "quantidade" :  
20, "minimo" : 10 }
```



# \$ne



```
{field: {$ne: value} }
```

```
> db.produtos.find( {minimo: {$ne: 10}} )  
{ "_id" : 1, "item" : "caderno", "quantidade" :  
20 }  
{ "_id" : 2, "item" : "cartao", "quantidade" :  
15, "minimo" : 5 }  
{ "_id" : 4, "item" : "selos", "quantidade" : 30,  
"minimo" : null }  
{ "_id" : 5, "item" : "lapis", "quantidade" : 50,  
"minimo" : 30 }
```

O operador **\$ne** também retorna **true** se o campo não existir.

# \$gt



```
{field: {$gt: value} }
```

```
> db.produtos.find( {quantidade: {$gt: 30}} )  
{ "_id" : 5, "item" : "lapis", "quantidade" : 50,  
  "minimo" : 30 }
```

# \$gte



```
{field: {$gte: value} }
```

```
> db.produtos.find( {quantidade: {$gte: 30}} )  
{ "_id" : 4, "item" : "selos", "quantidade" : 30,  
  "minimo" : null }  
{ "_id" : 5, "item" : "lapis", "quantidade" : 50,  
  "minimo" : 30 }
```



# \$lt



```
{field: {$lt: value} }
```

```
> db.produtos.find( {quantidade: {$lt: 20}} )  
{ "_id" : 2, "item" : "cartao", "quantidade" :  
15, "minimo" : 5 }
```

# \$lte



```
{ field: { $lte: value} }
```

```
> db.produtos.find( {quantidade: {$lte: 20}} )  
{ "_id" : 1, "item" : "caderno", "quantidade" :  
20 }  
{ "_id" : 2, "item" : "cartao", "quantidade" :  
15, "minimo" : 5 }  
{ "_id" : 3, "item" : "envelope", "quantidade" :  
20, "minimo" : 10 }
```

# \$in



```
{ field: { $in: [<value1>, <value2>, ... <valueN>] } }
```

```
> db.produtos.find( {quantidade: {$in: [20, 30, 50]}} )  
{ "_id" : 1, "item" : "caderno", "quantidade" : 20 }  
{ "_id" : 3, "item" : "envelope", "quantidade" : 20, "minimo" : 10 }  
{ "_id" : 4, "item" : "selos", "quantidade" : 30, "minimo" : null }  
{ "_id" : 5, "item" : "lapis", "quantidade" : 50, "minimo" : 30 }
```



# \$nin



```
{ field: { $nin: [ <value1>, <value2> ...  
<valueN> ] } }
```

```
> db.produtos.find( {minimo: {$nin: [10, 30]}} )  
{ "_id" : 1, "item" : "caderno", "quantidade" :  
20 }  
{ "_id" : 2, "item" : "cartao", "quantidade" :  
15, "minimo" : 5 }  
{ "_id" : 4, "item" : "selos", "quantidade" : 30,  
"minimo" : null }
```

O operador **\$nin** também retorna **true** se o campo não existir.

# Pesquisando por Faixa



Para pesquisar documentos com um campo dentro de uma faixa de valores deve ser usada a expressão:

```
> db.produtos.find( {quantidade: {$gte: 20, $lte: 30}} )
```

```
{ "_id" : 1, "item" : "caderno", "quantidade" : 20 }
```

```
{ "_id" : 3, "item" : "envelope", "quantidade" : 20, "minimo" : 10 }
```

```
{ "_id" : 4, "item" : "selos", "quantidade" : 30, "minimo" : null }
```

```
{ "_id" : 6, "item" : "borracha", "quantidade" : 25, "minimo" : 10 }
```

# Encadeando Condições



Para encadear condições de pesquisa com o operador lógico **AND** implícito, basta encadear as condições com vírgulas:

```
> db.notas.find( {disciplina: "portugues", nota:  
{$gt: 80}} )
```

```
{ "_id" : 6, "nome" : "Andre", "disciplina" :  
"portugues", "nota" : 89 }
```

```
{ "_id" : 9, "nome" : "Ana", "disciplina" :  
"portugues", "nota" : 90 }
```

```
{ "_id" : 10, "nome" : "Paula", "disciplina" :  
"portugues", "nota" : 85 }
```



# Operadores Lógicos



Além do **AND** implícito, também é possível utilizar operadores lógicos entre as condições da pesquisa:

**\$and**

**\$or**

**\$nor**

**\$not**

Exceto pelo operador **\$not**, os operadores lógicos são aplicados a um array de condições.

# \$and



```
{ $and: [ { <expression1> },  
{ <expression2> } , ... , { <expressionN> } ] }
```

```
> db.notas.find( {$and: [{disciplina:  
"portugues"}, {nota: {$gt: 80}}]} )  
{ "_id" : 6, "nome" : "Andre", "disciplina" :  
"portugues", "nota" : 89 }  
{ "_id" : 9, "nome" : "Ana", "disciplina" :  
"portugues", "nota" : 90 }  
{ "_id" : 10, "nome" : "Paula", "disciplina" :  
"portugues", "nota" : 85 }
```

# \$or



```
{ $or: [ { <expression1> },  
{ <expression2> }, ... , { <expressionN> } ] }
```

```
> db.produtos.find( {$or: [{quantidade: {$lt:  
20}}, {quantidade: {$gt: 30}}, {quantidade: 25}]}  
)  
{ "_id" : 2, "item" : "cartao", "quantidade" :  
15, "minimo" : 5 }  
{ "_id" : 5, "item" : "lapis", "quantidade" : 50,  
"minimo" : 30 }  
{ "_id" : 6, "item" : "borracha", "quantidade" :  
25, "minimo" : 10 }
```



# \$nor



```
{ $nor: [ { <expression1> },  
{ <expression2> }, ... { <expressionN> } ] }
```

```
> db.produtos.find( {$nor: [{minimo: 10},  
{minimo: 30}]} )  
{ "_id" : 1, "item" : "caderno", "quantidade" :  
20 }  
{ "_id" : 2, "item" : "cartao", "quantidade" :  
15, "minimo" : 5 }  
{ "_id" : 4, "item" : "selos", "quantidade" : 30,  
"minimo" : null }
```

# \$not



```
{ field: { $not: { <operator-expression> } } }
```

```
> db.produtos.find( {minimo: {$not: {$eq: 10}}} )  
{ "_id" : 1, "item" : "caderno", "quantidade" :  
20 }  
{ "_id" : 2, "item" : "cartao", "quantidade" :  
15, "minimo" : 5 }  
{ "_id" : 4, "item" : "selos", "quantidade" : 30,  
"minimo" : null }  
{ "_id" : 5, "item" : "lapis", "quantidade" : 50,  
"minimo" : 30 }
```

**O operador `$not` deve ser aplicado a uma condição e não ao valor de um campo.**

# \$not



O uso do operador **\$not** usado em conjunto com **\$gt**, **\$gte**, **\$lt** e **\$lte** não é equivalente ao uso do operador relacional complementar pois também retorna true se o campo não existir.

```
> db.produtos.find( {minimo: {$not: {$gte: 10}}} )  
{ "_id" : 1, "item" : "caderno", "quantidade" : 20 }  
{ "_id" : 2, "item" : "cartao", "quantidade" : 15, "minimo" : 5 }  
{ "_id" : 4, "item" : "selos", "quantidade" : 30, "minimo" : null }
```

```
> db.produtos.find( {minimo: {$lt: 10}} )  
{ "_id" : 2, "item" : "cartao", "quantidade" : 15, "minimo" : 5 }
```



# \$exists



O operador **\$exists** verifica a existência de um campo no documento.

```
{ field: { $exists: <boolean> } }
```

```
> db.produtos.find( {minimo: {$exists: 1}} )
{ "_id" : 2, "item" : "cartao", "quantidade" :
15, "minimo" : 5 }
{ "_id" : 3, "item" : "envelope", "quantidade" :
20, "minimo" : 10 }
{ "_id" : 4, "item" : "selos", "quantidade" : 30,
"minimo" : null }
{ "_id" : 5, "item" : "lapis", "quantidade" : 50,
"minimo" : 30 }
{ "_id" : 6, "item" : "borracha", "quantidade" :
25, "minimo" : 10 }
```

# \$exists



```
> db.produtos.find( {minimo: {$exists: 0}} )  
{ "_id" : 1, "item" : "caderno", "quantidade" :  
20 }
```

# \$exists



O operador **\$exists** pode ser usado com outros operadores para evitar documentos onde o campo testado não exista:

```
> db.produtos.find( {minimo: {$not: {$eq: 10}}} )
{ "_id" : 1, "item" : "caderno", "quantidade" : 20 }
{ "_id" : 2, "item" : "cartao", "quantidade" : 15, "minimo" : 5 }
{ "_id" : 4, "item" : "selos", "quantidade" : 30, "minimo" : null }
{ "_id" : 5, "item" : "lapis", "quantidade" : 50, "minimo" : 30 }
```



# \$exists



```
> db.produtos.find( {minimo: {$not: {$eq: 10}},  
$exists: 1}} )  
{ "_id" : 2, "item" : "cartao", "quantidade" :  
15, "minimo" : 5 }  
{ "_id" : 4, "item" : "selos", "quantidade" : 30,  
"minimo" : null }  
{ "_id" : 5, "item" : "lapis", "quantidade" : 50,  
"minimo" : 30 }
```

# \$exists



```
> db.produtos.find( {minimo: {$eq: null}} )  
{ "_id" : 1, "item" : "caderno", "quantidade" :  
20 }  
{ "_id" : 4, "item" : "selos", "quantidade" : 30,  
"minimo" : null }  
  
> db.produtos.find( {minimo: {$eq: null , $exists:  
1}} )  
{ "_id" : 4, "item" : "selos", "quantidade" : 30,  
"minimo" : null }
```

# \$type



O operador **\$type** verifica se o tipo do campo corresponde a um tipo de dados BSON.

```
{ field: { $type: <BSON type> } }
```

Ou

```
{ field: { $type: [ <BSON type1> , <BSON  
type2>, ... ] } }
```



# \$type



```
> db.produtos.find( {minimo: {$type: "int"}} )
{ "_id" : 2, "item" : "cartao", "quantidade" :
15, "minimo" : 5 }
{ "_id" : 3, "item" : "envelope", "quantidade" :
20, "minimo" : 10 }
{ "_id" : 5, "item" : "lapis", "quantidade" : 50,
"minimo" : 30 }
{ "_id" : 6, "item" : "borracha", "quantidade" :
25, "minimo" : 10 }
```

# \$type



```
> db.produtos.find( {minimo: {$type: "null"}} )  
{ "_id" : 4, "item" : "selos", "quantidade" : 30,  
  "minimo" : null }
```

# \$expr



O operador **\$expr** permite o uso de expressões na consulta.

```
{ $expr: { <expression> } }
```

```
> db.megasena.find( {$expr: {$gt:
["$quina_ganhadores", "$quadra_ganhadores"]}} )
{ "_id" : 309, "data" : "31/10/2001", "dezenas" :
[ 39, 50, 25, 55, 11, 4 ], "sena_ganhadores" : 1,
"uf" : [ "RS" ], "sena_rateio" : 442899.06,
"quina_ganhadores" : 3001, "quina_rateio" :
122.53, "quadra_ganhadores" : 2997,
"quadra_rateio" : 122.69 }
```



# \$mod



O operador **\$mod** verifica se a divisão de um valor tem o resto especificado.

```
{ field: { $mod: [ divisor, remainder ] } }
```

```
> db.produtos.find( {quantidade: {$mod: [2,
0]}} )
{ "_id" : 1, "item" : "caderno", "quantidade" :
20 }
{ "_id" : 3, "item" : "envelope", "quantidade" :
20, "minimo" : 10 }
{ "_id" : 4, "item" : "selos", "quantidade" : 30,
"minimo" : null }
{ "_id" : 5, "item" : "lapis", "quantidade" : 50,
"minimo" : 30 }
```

# \$mod



```
> db.produtos.find( {quantidade: {$mod: [2,  
1]}} )  
{ "_id" : 2, "item" : "cartao", "quantidade" :  
15, "minimo" : 5 }  
{ "_id" : 6, "item" : "borracha", "quantidade" :  
25, "minimo" : 10 }
```

# \$regex



O operador **\$regex** permite a pesquisa por uma expressão regular.

```
{ <field>: { $regex: /pattern/, $options:  
'<options>' } }  
{ <field>: { $regex: 'pattern', $options:  
'<options>' } }  
{ <field>: { $regex: /pattern/<options> } }
```

```
> db.produtos.find( {item: {$regex: /o$/}} )  
{ "_id" : 1, "item" : "caderno", "quantidade" :  
20 }  
{ "_id" : 2, "item" : "cartao", "quantidade" :  
15, "minimo" : 5 }
```



# \$where



O operador **\$where** permite o uso de uma expressão ou uma função JavaScript na pesquisa.

```
> db.megasena.find( {$where: "this.uf.length > 10"} )
{ "_id" : 529, "data" : "14/01/2004", "dezenas" :
[ 56, 45, 13, 33, 38, 1 ], "sena_ganhadores" :
15, "uf" : [ "BA", "CE", "CE", "PB", "PB", "PB",
"PE", "PE", "PE", "PE", "PE", "PI", "PI", "RN",
"RN" ], "sena_rateio" : 348732.75,
"quina_ganhadores" : 87, "quina_rateio" :
8565.12, "quadra_ganhadores" : 4758,
"quadra_rateio" : 156.03 }
```

# \$where



```
{  "_id"      : 2000,  "data"      : "31/12/2017",  
  "dezenas"   : [ 6, 37, 34, 10, 3, 17 ],  
  "sena_ganhadores" : 17,  "uf"      : [ "BA", "BA",  
  "BA", "MG", "MG", "PA", "PR", "PR", "RJ", "RJ",  
  "SC", "SP", "SP", "SP", "SP", "SP", "SP" ],  
  "sena_rateio" : 18042279.04,  "quina_ganhadores" :  
4862,  "quina_rateio"      : 10565.03,  
  "quadra_ganhadores" : 173428,  "quadra_rateio" :  
423.12 }
```



# Projeção



É possível determinar quais campos devem ser retornados ou não no resultado da consulta. Após a condição de seleção, deve ser passada a lista dos campos a retornar ou a lista dos campos a não retornar na consulta.

Consultas com uma lista de campos a retornar incluem apenas os campos relacionados. A única exceção permitida é excluir o campo **\_id**. Se não for explicitamente excluído, o campo **\_id** sempre será retornado.

```
> db.produtos.find( {quantidade: 20} , {item: 1,
quantidade: 1} )
{ "_id" : 1, "item" : "caderno", "quantidade" :
20 }
{ "_id" : 3, "item" : "envelope", "quantidade" :
20 }
```



# Projeção



```
> db.produtos.find( {quantidade: 20} , {_id: 0,  
item: 1, quantidade: 1} )  
{ "item" : "caderno", "quantidade" : 20 }  
{ "item" : "envelope", "quantidade" : 20 }
```

# Projeção



**Em uma consulta com lista de campos a excluir, serão retornados todos os campos que não constarem da lista.**

```
> db.produtos.find( { } , {_id: 0, minimo: 0} )  
{ "item" : "caderno", "quantidade" : 20 }  
{ "item" : "cartao", "quantidade" : 15 }  
{ "item" : "envelope", "quantidade" : 20 }  
{ "item" : "selos", "quantidade" : 30 }  
{ "item" : "lapis", "quantidade" : 50 }  
{ "item" : "borracha", "quantidade" : 25 }
```

**Exceto pela exclusão do campo `_id` não é permitido misturar indicação de campos a incluir e a excluir na mesma consulta.**



# Exercícios



- 1) Relacionar o título e ISBN dos livros de 2010**
- 2) Relacionar o nome e nota da disciplina de matemática**
- 3) Encontrar os empregados que não se reportam ao Eliot**
- 4) Relacionar o assunto e autor dos artigos com 90 visualizações ou mais**
- 5) Relacionar o id, assunto e autor dos artigos do Ben, Jeff e Betty**
- 6) Encontrar as notas entre 70 e 90**
- 7) Encontrar as notas de matemática entre 70 e 90**
- 8) Encontrar as notas de matemática abaixo de 80 e acima de 90**
- 9) Encontrar os aeroportos que não sejam JFK, ORD ou LHR**
- 10) Encontrar os empregados que não se reportam ao Eliot, mas que se reportam a alguém**