

## Aula 12 – 2 - Criando *Collections* e inserindo dados

Depois da instalação do Banco de Dados Não-relacional MongoDB vamos criar as *collections* que são as “tabelas” desse modelo de BD e inserir os dados. Siga os passos abaixo:

1. Verifique se o MongoDB está em execução acessando a ferramenta “Serviços” no Windows.
2. Abra a ferramenta Compass e conecte-se no BD “localhost” clicando em “Fill in connection fields individually” e clicando em “connect”;
3. Primeiro, vamos criar uma *database* “ex1” e uma *collection* chamada “clientes”;
4. Clique em “ex1” e depois em “clientes”, agora vamos inserir clientes dentro dessa *collection*. Lembre-se que esse BD utiliza documentos JSON, então para isso clique no botão “Add Data” e escolha “Insert Document”. Coloque os códigos abaixo (um de cada vez):

```
{
  "nome": "Breno Dantas",
  "cpf": "87569832536",
  "telefone": "44987546333",
  "data_nasc": "2018-04-15",
  "logradouro": "Rua Amapodi",
  "numero": 3890,
  "bairro": "Jardim Belo",
  "cep": "87988230",
  "cidade": "Umuarama",
  "estado": "PR"
}
```

```
{
  "nome": "Pedro Firmino",
  "cpf": "99036541236",
  "telefone": "44991236655",
  "data_nasc": "2002-02-10",
  "logradouro": "Av. dos Patos",
  "numero": 32,
  "bairro": "Jardim Panorama",
  "cep": "87101360",
  "cidade": "Umuarama",
  "estado": "PR"
}
```

5. Para inserir vários documentos de uma vez é necessário colocar os objetos dentro de um array [ ].

### Exercício

Utilizando o BD “ex1”, crie uma nova *collection* “produtos” e faça 2 inserções de dados nessa *collection*, contendo os campos: nome, valor unitário, estoque, e um array com as fotos (links para as fotos dos produtos).

## Aula 12 – 3 - Executando Consultas NoSQL

Para executarmos consultas em Bancos de Dados Não-Relacionais existem funções que fazem a busca de dados dentro das *collections*. Uma dessas funções é o *find()* que possui várias opções e atributos que deixam a consulta mais refinada.

Para vermos isso em prática, siga os passos abaixo:

1. Clique no botão “\_MongoSH Beta”;
2. Para as consultas abaixo é necessário usar primeiro o comando “use ex1” para confirmar que está ativa a *database* e seguir com os comandos:

```
// Mostra todos os documentos com todos os campos da collection “produtos”
db.produtos.find({});

// Mostra todos os campos dos produtos que tenham o campo nome como o descrito
db.produtos.find({nome: "Camisa Polo Branca"});

// Mostra os campos nome e valor_unitário (não mostrando o id) dos produtos que tenham o
campo nome como o descrito
db.produtos.find({nome: "Camisa Polo Branca"}, {_id: 0, nome: 1, valor_unitário: 1});

// Mostra os campos nome e valor_unitário (não mostrando o id) dos produtos que tenham o
campo nome contendo o descrito (similar ao LIKE % em SQL)
db.produtos.find({nome: /Camisa Polo./}, {_id: 0, nome: 1, valor_unitário: 1});

// Mostra a contagem de documentos da collection
db.produtos.find().count();

// Mostra os documentos que tenham o campo valor_unitário igual a (equal) 100
db.produtos.find({valor_unitário: {$eq: 125}});

// Mostra os documentos que tenham o campo valor_unitário maior que (greater than) 100
db.produtos.find({valor_unitário: {$gt: 100}});

// Mostra os documentos que tenham o campo valor_unitário maior ou igual que (greater than
equal to) 90
db.produtos.find({valor_unitário: {$gte: 90}});

// Mostra os documentos que tenham o campo valor_unitário menor que (less than) 90
db.produtos.find({valor_unitário: {$lt: 90}});

// Mostra os documentos que tenham o campo valor_unitário menor ou igual que (less than
equal to) 90
db.produtos.find({valor_unitário: {$lte: 90}});

// Mostra os documentos que tenham o campo valor_unitário com os valores exatos de 90 e 125
db.produtos.find({valor_unitário: {$in: [90, 125]}});

// Mostra os documentos que tenham o campo valor_unitário sem os valores exatos 89.9 e 125
db.produtos.find({valor_unitário: {$nin: [89.9, 125]}});
```

**Exercícios**

Faça as seguintes consultas na *collection* “clientes”:

1. Mostre todos os clientes;
2. Mostre os clientes com o nome “Pedro Firmino”;
3. Mostre o nome e o CPF dos clientes com o nome que começa com “Breno”;
4. Mostre o nome e o telefone dos clientes com o telefone terminando com número “6333”;
5. Mostre quantos clientes moram no estado do “PR”;
6. Mostre quantos clientes não moram em “Umuarama”;