



Curso: Mestrado em Informática/ Mestrado em Bioinformática  
U.C.: Bases de Dados NoSQL

## Ficha de Exercícios 03 - PL04

Docente:	António Abelha / Cristiana Neto
Tema:	Introdução ao MongoDB
Turma:	Prática Laboratorial
Ano Letivo:	2023-2024 – 2º Semestre
Duração da aula:	2 horas

### 1. Customers

Recorrendo à mongoshell:

```
# docker exec -it <id-container-mongo> bash
# mongosh
```

- [1] Listar todas as bases de dados após a instalação do container com a imagem do MongoDB.
- [2] Criar uma base de dados denominada “customers”.
- [3] Verificar a criação da base de dados.
- [4] Criar uma coleção denominada “customers”.
- [5] Validar a criação da coleção.
- [6] Criar um cliente com os seguintes características:  
first\_name: "John", last\_name: "Doe", age: 30
- [7] Introduzir 2 clientes na coleção com uma só instrução criada com as seguintes características:  
first\_name: "Steven", last\_name: "Williams", gender: "male"  
first\_name: "Mary", last\_name: "Troy", age: 19
- [8] Introduzir mais um cliente com as seguintes características:  
first\_name: "Ric", last\_name: "Foe", address: {street: "4 main st", city: "Boston"}
- [9] Criar um cliente com as seguintes características:  
first\_name: "Ana", last\_name: "Durant", degree: ["phD", "Msc"],  
address: {street: "4 Square Garden", city: "New York"}, age: 32
- [10] Criar um cliente com as seguintes características:  
first\_name: "Natalia", last\_name: "Will", age: 44, gender: "female"
- [11] Listar todos os clientes.
- [12] Efetuar uma atualização ao cliente ‘Ric’, colocar idade 45.
- [13] Encontrar todos os clientes que tenham ‘Will’ no último nome.
- [14] Efetuar uma atualização ao cliente ‘Steven’, colocar idade 35.
- [15] Verificar se a idade da cliente ‘Ana’ é superior a 30 e se sim aumentar a idade em 10 anos.
- [16] O cliente ‘Ric’ quer que a sua idade seja removida da base de dados.



[17] Procurar um cliente com o primeiro nome: "Jimmy" e atualizar, ou criar, caso não exista, com as seguintes características.:

```
first_name: "Jimmy", last_name: "Connors", age: 25, gender: male
```

[18] Procurar todos os clientes com idade superior ou igual a 25.

[19] Procurar todos os clientes sexo masculino.

[20] Apagar o cliente cujo primeiro nome é "Mary".

[21] Encontrar os clientes com o nome "Ana" ou "Ric".

## 2. restaurants.json

Importe o ficheiro "[restaurants.json](#)" utilizando o Compass para uma coleção *restaurants*.

Considere:

```
{
  "address": {
    "building": "351",
    "coord": {
      "type": "Point",
      "coordinates": [
        -73.98513559999999,
        40.7676919
      ]
    },
    "street": "West 57 Street",
    "zipcode": "10019"
  },
  "borough": "Bronx",
  "cuisine": "Bakery",
  "grades": [
    { "date": { "$date": 1393804800000 }, "grade": "A", "score": 2 },
    { "date": { "$date": 1378857600000 }, "grade": "A", "score": 6 },
    { "date": { "$date": 1358985600000 }, "grade": "A", "score": 10 },
    { "date": { "$date": 1322006400000 }, "grade": "A", "score": 9 },
    { "date": { "$date": 1299715200000 }, "grade": "B", "score": 14 }
  ],
  "name": "Morris Park Bake Shop",
  "restaurant_id": "30075445"
}
```

- [1] liste todos os documentos na coleção *restaurants*.
- [2] liste apenas os campos *restaurant\_id*, *name*, *borough* e *cuisine* para todos os documentos na coleção.
- [3] liste os campos *restaurant\_id*, *name*, *borough* e *cuisine* para todos os documentos na coleção, mas que exclua o campo *\_id*.
- [4] liste os campos *restaurant\_id*, *name*, *borough* e *zipcode* para todos os documentos na coleção, mas que exclua o campo *\_id*.
- [5] liste os restaurantes que estão localizados no bairro (*borough*) "Bronx".
- [6] liste os primeiros 5 restaurantes que estão localizados no bairro (*borough*) "Bronx".
- [7] liste os 5 restaurantes após os primeiros 5 (do 6º ao 10º) que estão localizados no bairro (*borough*) "Bronx".
- [8] liste todos os restaurantes que têm pelo menos uma pontuação (*score*) maior que 90.



- [9] liste todos os restaurantes que têm uma pontuação (*score*) maior que 80 mas menor que 100.
- [10] liste todos os restaurantes que estão localizados numa latitude (*coordinates.0*) menor que -95.754168.
- [11] liste todos os restaurantes cujo tipo de cozinha (*cuisine*) não seja "American", que a sua pontuação (*score*) seja maior que 70 e a latitude (*address.coord.0*) menor que -65.754168, utilizando o operador \$and.
- [12] liste todos os restaurantes cujo tipo de cozinha (*cuisine*) não seja "American", que a sua pontuação (*score*) seja maior que 70 e a latitude (*address.coord.0*) menor que -65.754168.
- [13] liste todos os restaurantes cujo tipo de cozinha (*cuisine*) não seja do tipo "American" e que tenham atingido uma classificação (*grade*) de "A" mas que não pertençam ao bairro (*borough*) de "Brooklyn". Deverá ser apresentada de acordo com o tipo de cozinha (*cuisine*) em ordem descendente.
- [14] liste todos os restaurantes que pertençam ao bairro (*borough*) "Bronx" e cujo tipo de cozinha (*cuisine*) seja quer "American" quer "Chinese".
- [15] liste todos os restaurantes que contenham informação da rua (*address.street*).
- [16] liste todos os restaurantes de forma ascendente pelo tipo de cozinha (*cuisine*) e descendente pelo bairro (*borough*).
- [17] liste o *restaurant\_id*, *name*, *address* e localização geográfica (*coord*) para os restaurantes cujo segundo elemento do array da localização geográfica (*coord*) seja maior que 42 e até 52.
- [18] liste os restaurantes (*restaurante\_id*, *name*, *borough*, *cuisine*) que não conseguiram uma pontuação (*score*) maior que 10.
- [19] liste todos os restaurantes (*restaurante\_id*, *name*, *borough* e *cuisine*) que não pertencem ao bairro (*borough*) de "Staten Islan", ou "Queens" ou "Bronx" ou "Brooklin".
- [20] Atualize o campo "name" de um restaurante específico com o *restaurante\_id* igual a "30191841" para um novo nome, por exemplo, "NewName Restaurant".