



**Politécnico
Castelo Branco**

Escola Superior
de Tecnologia

MongoDB

Relatório da disciplina de Complementos de Bases de Dados

Trabalho realizado por:

- Gonçalo Santos, nº 20231487, 06/01/2025

Índice geral

1. Introduão	4
2. Criaão e Inserão.....	5
3. Consultas	9
3.1. Utilizaão de todas as condiões (Igual; Maior e Maior que; Menor e Menor que; Diferente; E/OU combinaões entre elas), pelo menos 5 consultas	9
3.2. Consultas com todos os operadores lgicos (AND; OR; NE; NOT; ... E/OU combinaões entre eles), pelo menos 5 consultas.....	10
3.3. Consultas com base em campos com formato array, pelo menos 2 consultas	12
4. Conclusão	13

Índice de Figuras

Figura 1 – Criação de base de dados no Studio3T.....	5
Figura 2 – Criação de uma <i>collection</i> no Studio3T	5
Figura 3 – Inserção de um documento único	6
Figura 4 – Inserção de documentos inseridos em simultâneo	6
Figura 5 – Ficheiro JSON	7
Figura 6 – Inserção no Studio3T dos dados via ficheiro JSON.....	7
Figura 7 – Ficheiro CSV.....	7
Figura 8 - Inserção no Studio3T dos dados via ficheiro CSV.....	7
Figura 9 – Inserção de documentos em formato <i>array</i>	8
Figura 10 – Resultado 1.....	9
Figura 11 – Resultado 2.....	9
Figura 12 – Resultado 3.....	9
Figura 13 – Resultado 4.....	10
Figura 14 – Resultado 5.....	10
Figura 15 – Resultado 7.....	11
Figura 16 – Resultado 8.....	11
Figura 17 – Resultado 9.....	11
Figura 18 – Resultado 10	12
Figura 19 – Resultado 11	12

1. Introduo

O MongoDB  uma base de dados NoSQL *open-source* amplamente utilizada, compatvel com os sistemas operativos Linux, Mac e Windows. Desenvolvido em C++, permite consultas e interaes num ambiente iterativo utilizando JavaScript. Classificado como uma base de dados orientada a documentos, armazena informaes no formato BSON, uma extenso do JSON, onde cada documento numa coleo equivale a uma linha numa tabela relacional. Entre as suas principais vantagens destacam-se a tolerncia a falhas, a escalabilidade e a distribuio eficiente dos dados.

Neste relatrio, ser demonstrada a criao de uma base de dados e de uma coleo, bem como a execuo de consultas utilizando o software Studio3T.

2. Criação e Inserção

1. Após criar uma base de dados denominada “CBD”

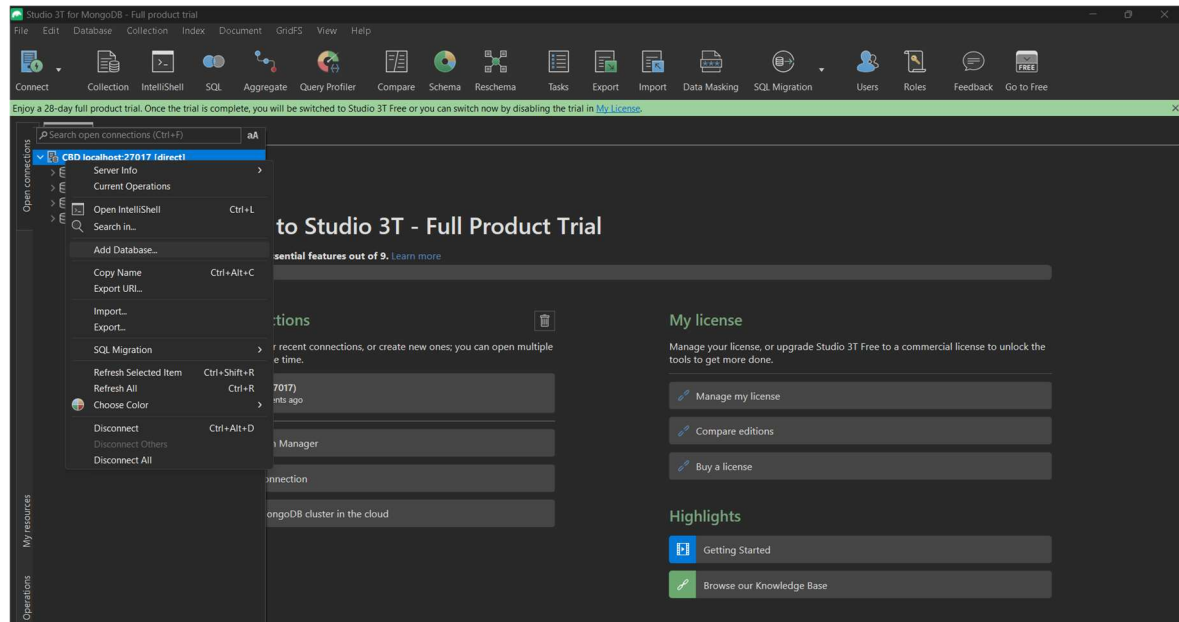


Figura 1 - Criação de base de dados no Studio3T

2. Criar uma *collection* chamada “Funcionario”

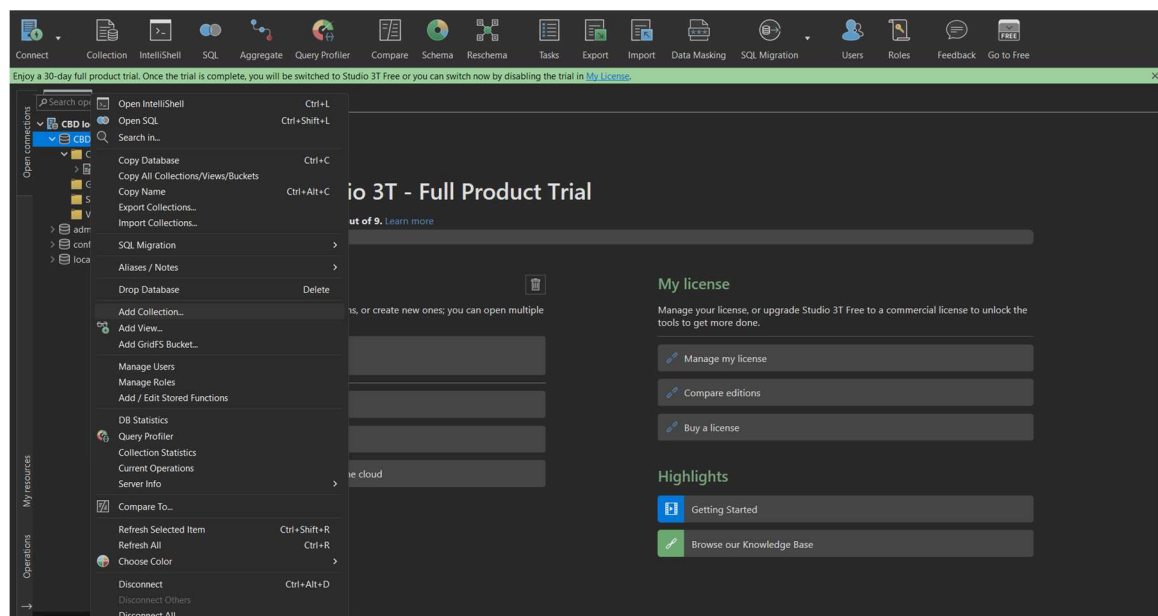


Figura 2 - Criação de uma *collection* no Studio3T

3. Criar os documentos

a) Inserção de um documento único

```
{  
  "id": 1,  
  "nome_funcionario": "André",  
  "data_nascimento": "01-01-2000",  
  "contatos_telefonicos": 919119101  
}
```

Figura 3 - Inserção de um documento único

b) Inserção de documentos inseridos em simultâneo

```
[  
  {  
    "id": 2,  
    "nome_funcionario": "Ana",  
    "data_nascimento": "02-01-2000",  
    "contatos_telefonicos": 919119102  
  },  
  {  
    "id": 3,  
    "nome_funcionario": "Bernardo",  
    "data_nascimento": "03-01-2000",  
    "contatos_telefonicos": 919119103  
  },  
  {  
    "id": 4,  
    "nome_funcionario": "Beatriz",  
    "data_nascimento": "03-01-2000",  
    "contatos_telefonicos": 919119104  
  }  
]
```

Figura 4 - Inserção de documentos inseridos em simultâneo

c) Inserção de documentos via ficheiro JSON

```
[
  { "id": 5, "nome_funcionario": "Carlos", "data_nascimento": "2000.01.05", "contatos_telefonicos": 919119115},
  { "id": 6, "nome_funcionario": "Catarina", "data_nascimento": "2000.01.06", "contatos_telefonicos": 919119116}
]
```

Figura 5 - Ficheiro JSON

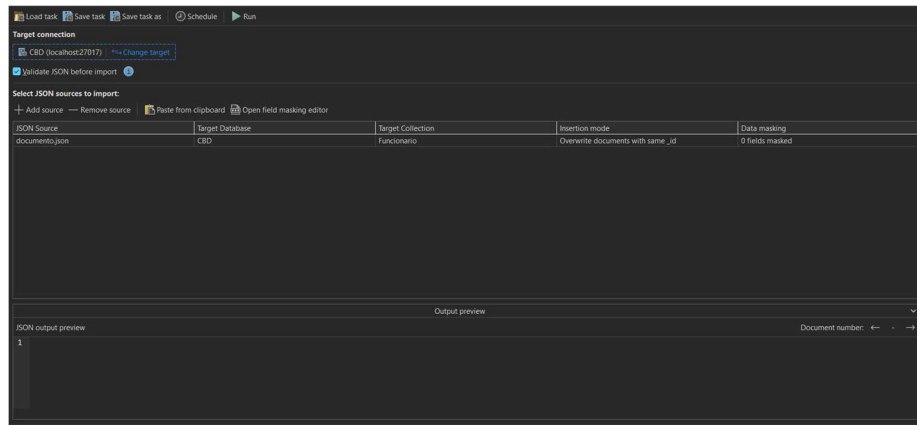


Figura 6 - Inserção no Studio3T dos dados via ficheiro JSON

d) Inserção de documentos via ficheiro CSV

id	nome_funcionario	data_nascimento	contatos_telefonicos
7	David	07-01-2000	919119117
8	Diana	08-01-2000	919119118

Figura 7 - Ficheiro CSV

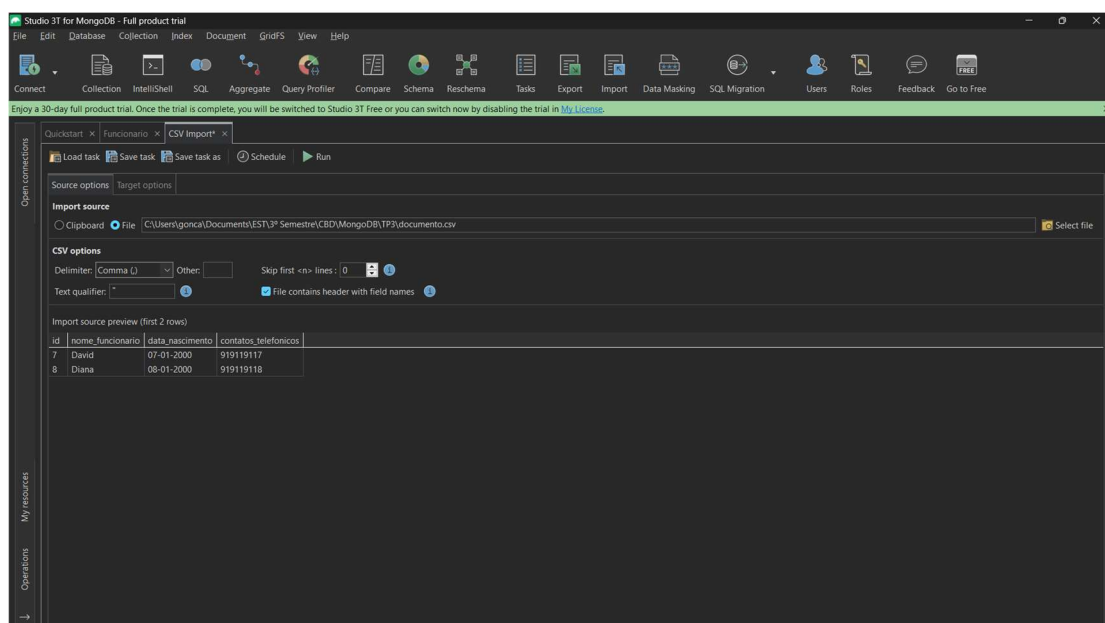


Figura 8 - Inserção no Studio3T dos dados via ficheiro CSV

e) Insero de documentos em formato *array*

```
[
  {
    "id": [9, 10, 11, 12],
    "nome_funcionario": ["Ernesto", "Eva", "Francisco", "Filipa"],
    "data_nascimento": ["09-01-2000", "10-01-2000", "11-01-2000", "12-01-2000"],
    "contatos_telefonicos": [919119119, 919119120, 919119121, 919119122]
  }
]
```

Figura 9 - Insero de documentos em formato *array*

3. Consultas

3.1. Utilização de todas as condições (Igual; Maior e Maior que; Menor e Menor que; Diferente; E/OU combinações entre elas), pelo menos 5 consultas

```
db.Funcionario.find({ nome_funcionario: { $eq: "André" } })
```

```
{
  "_id" : ObjectId("677c1b58ad94462bdfe9a904"),
  "id" : NumberInt(1),
  "nome_funcionario" : "André",
  "data_nascimento" : "01-01-2000",
  "contatos_telefonicos" : NumberInt(919119111)
}
```

Figura 10 - Resultado 1

```
db.Funcionario.find({ contatos_telefonicos: { $gt: 919119112 } })
```

_id	id	nome_funcionario	data_nascimento	contatos_telefonicos
677c1d27ad944...	3	Bernardo	03-01-2000	919119113
677c1d27ad944...	4	Beatriz	03-01-2000	919119114
677c1e07ad9446...	5	Carlos	05-01-2000	919119115
677c1e07ad9446...	6	Catarina	06-01-2000	919119116
677c20de4c78f8...	7	David	07-01-2000	919119117
677c20de4c78f8...	8	Diana	08-01-2000	919119118
677c23164c78f8...	[4 elements]	[4 elements]	[4 elements]	[4 elements]
677c239d4c78f8...	13	Gonçalo	13-01-2000	919119123
677c239d4c78f8...	14	Gabriela	14-01-2000	919119124

Figura 11 - Resultado 2

```
db.Funcionario.find({ data_nascimento: { $lt: "07-01-2000" } })
```

_id	id	nome_funcionario	data_nascimento	contatos_telefonicos
677c1b58ad944...	1	André	01-01-2000	919119111
677c1d27ad944...	2	Ana	02-01-2000	919119112
677c1d27ad944...	3	Bernardo	03-01-2000	919119113
677c1d27ad944...	4	Beatriz	03-01-2000	919119114
677c1e07ad9446...	5	Carlos	05-01-2000	919119115
677c1e07ad9446...	6	Catarina	06-01-2000	919119116

Figura 12 - Resultado 3

```
db.Funcionario.find({ id: { $ne: null } })
```

_id	id	nome_funcionario	data_nascimento	contatos_telefonicos
677c1b58ad944...	1	André	01-01-2000	919119111
677c1d27ad944...	2	Ana	02-01-2000	919119112
677c1d27ad944...	3	Bernardo	03-01-2000	919119113
677c1d27ad944...	4	Beatriz	03-01-2000	919119114
677c1e07ad9446...	5	Carlos	05-01-2000	919119115
677c1e07ad9446...	6	Catarina	06-01-2000	919119116
677c20de4c78f8...	7	David	07-01-2000	919119117
677c20de4c78f8...	8	Diana	08-01-2000	919119118
677c23164c78f8...	[4 elements]	[4 elements]	[4 elements]	[4 elements]
677c239d4c78f8...	13	Gonalo	13-01-2000	919119123
677c239d4c78f8...	14	Gabriela	14-01-2000	919119124

Figura 13 - Resultado 4

```
db.Funcionario.find({ $and: [{ contatos_telefonicos: { $gte: 919119112 } }, { nome_funcionario: "Beatriz" }] })
```

```
{
  "_id" : ObjectId("677c1d27ad94462bdf9e9a908"),
  "id" : NumberInt(4),
  "nome_funcionario" : "Beatriz",
  "data_nascimento" : "03-01-2000",
  "contatos_telefonicos" : NumberInt(919119114)
}
```

Figura 14 - Resultado 5

3.2. Consultas com todos os operadores lógicos (AND; OR; NE; NOT; ... E/OU combinaões entre eles), pelo menos 5 consultas

```
db.Funcionario.find({ $and: [{ nome_funcionario: "Carlos" }, { id: { $lt: 5 } }] })
```

Esta consulta no vai retornar nada, porque o id do funcionrio "Carlos"  igual a 5

```
db.Funcionario.find({ $or: [{ contatos_telefonicos: { $lte: 919119111 } }, { nome_funcionario: "Bernardo" }] })
```

```
{
  "_id" : ObjectId("677c1b58ad94462bdfe9a904"),
  "id" : NumberInt(1),
  "nome_funcionario" : "André",
  "data_nascimento" : "01-01-2000",
  "contatos_telefonicos" : NumberInt(919119111)
}
{
  "_id" : ObjectId("677c1d27ad94462bdfe9a907"),
  "id" : NumberInt(3),
  "nome_funcionario" : "Bernardo",
  "data_nascimento" : "03-01-2000",
  "contatos_telefonicos" : NumberInt(919119113)
}
```

Figura 15 - Resultado 7

```
db.Funcionario.find({ nome_funcionario: { $not: { $eq: "David" } } })
```

_id	id	nome_funcionario	data_nascimento	contatos_telefonicos
677c1b58ad944...	1	André	01-01-2000	919119111
677c1d27ad944...	2	Ana	02-01-2000	919119112
677c1d27ad944...	3	Bernardo	03-01-2000	919119113
677c1d27ad944...	4	Beatriz	03-01-2000	919119114
677c1e07ad9446...	5	Carlos	05-01-2000	919119115
677c1e07ad9446...	6	Catarina	06-01-2000	919119116
677c20de4c78f8...	8	Diana	08-01-2000	919119118
677c23164c78f8...	[4 elements]	[4 elements]	[4 elements]	[4 elements]
677c239d4c78f8...	13	Gonçalo	13-01-2000	919119123
677c239d4c78f8...	14	Gabriela	14-01-2000	919119124

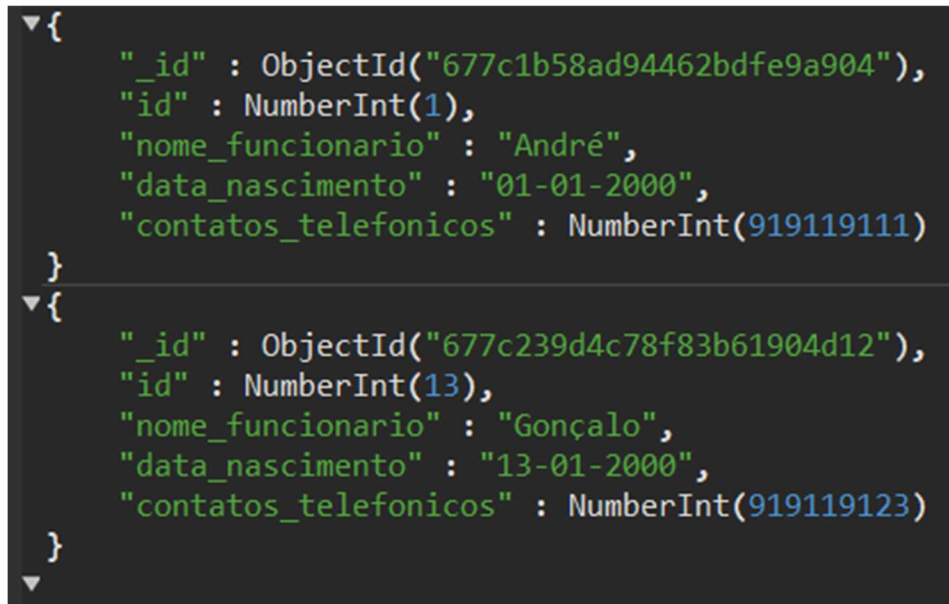
Figura 16 - Resultado 8

```
db.Funcionario.find({ contatos_telefonicos: { $ne: 919119113 } })
```

_id	id	nome_funcionario	data_nascimento	contatos_telefonicos
677c1b58ad944...	1	André	01-01-2000	919119111
677c1d27ad944...	2	Ana	02-01-2000	919119112
677c1d27ad944...	4	Beatriz	03-01-2000	919119114
677c1e07ad9446...	5	Carlos	05-01-2000	919119115
677c1e07ad9446...	6	Catarina	06-01-2000	919119116
677c20de4c78f8...	7	David	07-01-2000	919119117
677c20de4c78f8...	8	Diana	08-01-2000	919119118
677c23164c78f8...	[4 elements]	[4 elements]	[4 elements]	[4 elements]
677c239d4c78f8...	13	Gonçalo	13-01-2000	919119123
677c239d4c78f8...	14	Gabriela	14-01-2000	919119124

Figura 17 - Resultado 9

```
db.Funcionario.find({ $and: [{ contatos_telefonicos: { $ne: null } }, { $or: [{ nome_funcionario: "Gonçalo" }, { data_nascimento: { $lt: "02-01-2000" } } ] } ] })
```



```
{
  "_id" : ObjectId("677c1b58ad94462bdfe9a904"),
  "id" : NumberInt(1),
  "nome_funcionario" : "André",
  "data_nascimento" : "01-01-2000",
  "contatos_telefonicos" : NumberInt(919119111)
}
{
  "_id" : ObjectId("677c239d4c78f83b61904d12"),
  "id" : NumberInt(13),
  "nome_funcionario" : "Gonçalo",
  "data_nascimento" : "13-01-2000",
  "contatos_telefonicos" : NumberInt(919119123)
}
```

Figura 18 - Resultado 10

3.3. Consultas com base em campos com formato array, pelo menos 2 consultas

```
db.Funcionario.find({ contatos_telefonicos: { $size: 2 } })
```



```
{
  "_id" : ObjectId("677c239d4c78f83b61904d13"),
  "id" : NumberInt(14),
  "nome_funcionario" : "Gabriela",
  "contatos_telefonicos" : [
    NumberInt(919119124),
    NumberInt(919119125)
  ]
}
```

Figura 19 - Resultado 11

```
db.Funcionario.find({ id: { $size: 4 } }).count()
```

Este output irá retornar o resultado 1, porque existe apenas um documento com 4 id

4. Conclusão

Conclui-se que o MongoDB é uma solução robusta e eficiente para armazenamento e gestão de dados orientados a documentos, destacando-se pela sua flexibilidade, escalabilidade e tolerância a falhas. A utilização do formato BSON e o suporte a consultas em JavaScript tornam-no uma ferramenta intuitiva e poderosa para diferentes aplicações.

Durante este trabalho, foi possível criar uma base de dados, configurar uma coleção e executar consultas utilizando o Studio3T, evidenciando a praticidade e eficiência do MongoDB na manipulação de grandes volumes de dados. Assim, reforça-se a importância desta tecnologia para projetos que requerem desempenho elevado e estrutura de dados flexível.