

**MongoDB**

**Relatório da disciplina de Complementos de Bases de Dados**

**Trabalho realizado por:**

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

Índice geral

[1. Introdução 4](#_Toc187245777)

[2. Criação e Inserção 5](#_Toc187245778)

[3. Consultas 9](#_Toc187245779)

[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 9](#_Toc187245780)

[3.2. Consultas com todos os operadores lógicos (AND; OR; NE; NOT; … E/OU combinações entre eles), pelo menos 5 consultas 10](#_Toc187245781)

[3.3. Consultas com base em campos com formato array, pelo menos 2 consultas 12](#_Toc187245782)

[4. Conclusão 13](#_Toc187245783)

Índice de Figuras

[**Figura 1 –** Criação de base de dados no Studio3T 5](#_Toc187244792)

[**Figura 2 –** Criação de uma *collection* no Studio3T 5](#_Toc187244793)

[**Figura 3** – Inserção de um documento único 6](#_Toc187244794)

[**Figura 4 –** Inserção de documentos inseridos em simultâneo 6](#_Toc187244795)

[**Figura 5 –** Ficheiro JSON 7](#_Toc187244796)

[**Figura 6 –** Inserção no Studio3T dos dados via ficheiro JSON 7](#_Toc187244797)

[**Figura 7 –** Ficheiro CSV 7](#_Toc187244798)

[**Figura 8 -** Inserção no Studio3T dos dados via ficheiro CSV 7](#_Toc187244799)

[**Figura 9 –** Inserção de documentos em formato *array* 8](#_Toc187244800)

[**Figura 10 –** Resultado 1 9](#_Toc187244801)

[**Figura 11 –** Resultado 2 9](#_Toc187244802)

[**Figura 12 –** Resultado 3 9](#_Toc187244803)

[**Figura 13 –** Resultado 4 10](#_Toc187244804)

[**Figura 14 –** Resultado 5 10](#_Toc187244805)

[**Figura 15 –** Resultado 7 11](#_Toc187244806)

[**Figura 16 –** Resultado 8 11](#_Toc187244807)

[**Figura 17 –** Resultado 9 11](#_Toc187244808)

[**Figura 18 –** Resultado 10 12](#_Toc187244809)

[**Figura 19 –** Resultado 11 12](#_Toc187244810)

1. Introdução

O MongoDB é uma base de dados NoSQL *open-source* amplamente utilizada, compatível com os sistemas operativos Linux, Mac e Windows. Desenvolvido em C++, permite consultas e interações num ambiente iterativo utilizando JavaScript. Classificado como uma base de dados orientada a documentos, armazena informações no formato BSON, uma extensão do JSON, onde cada documento numa coleção equivale a uma linha numa tabela relacional. Entre as suas principais vantagens destacam-se a tolerância a falhas, a escalabilidade e a distribuição eficiente dos dados.

Neste relatório, será demonstrada a criação de uma base de dados e de uma coleção, bem como a execução de consultas utilizando o software Studio3T.

2. Criação e Inserção

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

Uma imagem com texto, captura de ecrã, software, Software de multimédia

Descrição gerada automaticamente

**Figura 1 –** Criação de base de dados no Studio3T

2. Criar uma *collection* chamada “Funcionario”

Uma imagem com texto, captura de ecrã, software, Software de multimédia

Descrição gerada automaticamente

**Figura 2 –** Criação de uma *collection* no Studio3T

3. Criar os documentos

a) Inserção de um documento único

Uma imagem com texto, captura de ecrã, Tipo de letra

Descrição gerada automaticamente

**Figura 3** – Inserção de um documento único

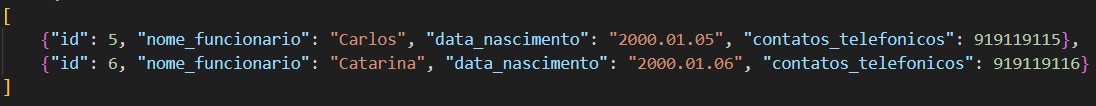
b) Inserção de documentos inseridos em simultâneo

Uma imagem com texto, captura de ecrã, software, Tipo de letra

Descrição gerada automaticamente

**Figura 4 –** Inserção de documentos inseridos em simultâneo

c) Inserção de documentos via ficheiro JSON



**Figura 5 –** Ficheiro JSON

Uma imagem com captura de ecrã, texto, software, Software de multimédia

Descrição gerada automaticamente

**Figura 6 –** Inserção no Studio3T dos dados via ficheiro JSON

d) Inserção de documentos via ficheiro CSV

Uma imagem com texto, captura de ecrã, Tipo de letra, file

Descrição gerada automaticamente

**Figura 7 –** Ficheiro CSV

Uma imagem com captura de ecrã, texto, software, Software de multimédia

Descrição gerada automaticamente

**Figura 8 -** Inserção no Studio3T dos dados via ficheiro CSV

e) Inserção de documentos em formato *array*

*Uma imagem com texto, captura de ecrã, Tipo de letra

Descrição gerada automaticamente*

**Figura 9 –** Inserção 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" **}** **})**

Uma imagem com texto, Tipo de letra, captura de ecrã

Descrição gerada automaticamente

**Figura 10 –** Resultado 1

db.Funcionario.find({ contatos\_telefonicos: { $gt: 919119112 } })

Uma imagem com texto, captura de ecrã, software, Software de multimédia

Descrição gerada automaticamente

**Figura 11 –** Resultado 2

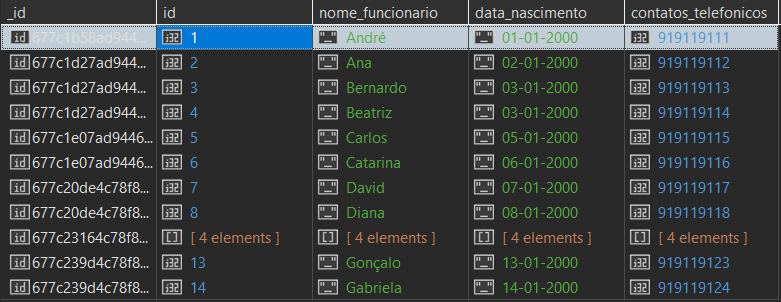
db.Funcionario.find({ data\_nascimento: { $lt: "07-01-2000" } })

Uma imagem com texto, software, Software de multimédia, Tipo de letra

Descrição gerada automaticamente

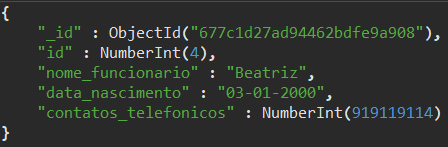
**Figura 12 –** Resultado 3

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



**Figura 13 –** Resultado 4

db.Funcionario.find({ $and: [{ contatos\_telefonicos: { $gte: 919119112 } }, { nome\_funcionario: "Beatriz" }] })



**Figura 14 –** Resultado 5

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

db.Funcionario.find**({** $and**:** **[{** nome\_funcionario**:** "Carlos" **},** **{** id**:** **{** $lt**:** 5 **}** **}]** **})**

Esta consulta não vai retornar nada, porque o id do funcionário “Carlos” é igual a 5

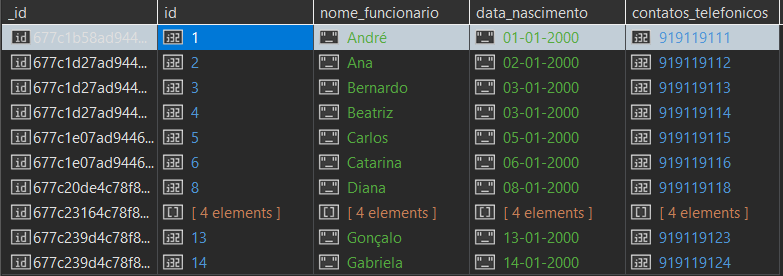
db.Funcionario.find**({** $or**:** **[{** contatos\_telefonicos**:** **{** $lte**:** 919119111 **}** **},** **{** nome\_funcionario**:** "Bernardo" **}]** **})**

Uma imagem com texto, captura de ecrã, Tipo de letra

Descrição gerada automaticamente

**Figura 15 –** Resultado 7

db.Funcionario.find**({** nome\_funcionario**:** **{** $not**:** **{** $eq**:** "David" **}** **}** **})**



**Figura 16 –** Resultado 8

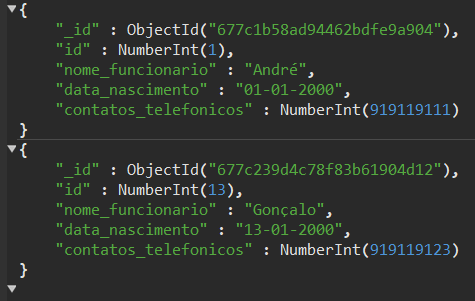
db.Funcionario.find**({** contatos\_telefonicos**:** **{** $ne**:** 919119113 **}** **})**

Uma imagem com texto, captura de ecrã, software, Software de multimédia

Descrição gerada automaticamente

**Figura 17 –** Resultado 9

db.Funcionario.find({ $and: [{ contatos\_telefonicos: { $ne: null } }, { $or: [{ nome\_funcionario: "Gonçalo" }, { data\_nascimento: { $lt: "02-01-2000" } }] } ] })



**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 **}** **})**

Uma imagem com texto, captura de ecrã, Tipo de letra

Descrição gerada automaticamente

**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.