

Tecnólogo Ciência de Dados

Guilherme Giacomini Teixeira

nosql e Mongodb:

Trabalho de Avaliação da Unidade 3 da Disciplina Banco de Dados em Nuvem

Balneário Camboriú – SC

2025

Balneário Camboriú - SC

2025

Guilherme Giacomini Teixeira

nosql e Mongodb:

Trabalho de Avaliação da Unidade 3 da Disciplina Banco de Dados em Nuvem

­­­­

Trabalho de avaliação da unidade 3 da Disciplina Banco de Dados em Nuvem apresentado como requisito parcial para a obtenção da média no curso Ciência de Dados.

Professor: Romulo de Almeida Neves  
Tutor: João Henrique Correia dos Santos

SUMÁRIO

[1 INTRODUÇÃO 3](#_Toc129809424)

[2 DESENVOLVIMENTO 4](#_Toc129809425)

[3 RESULTADOS](#_Toc129809426) 6

[4 CONCLUSÃO](#_Toc129809427) 7

[5 REFERÊNCIAS](#_Toc129809428) 8

# INTRODUÇÃO

Este relatório descreve a atividade prática da disciplina de **Banco de Dados em Nuvem**, na **Unidade 3**, focada em **MongoDB**. O objetivo principal é demonstrar o entendimento e a aplicação de operações **CRUD** (Create, Read, Update, Delete) em um banco de dados **NoSQL**.

A atividade consiste em atuar como desenvolvedor principal para a ***"Biblioteca Sabedoria Eterna"***. A tarefa é realizar uma série de operações para atualizar e otimizar a coleção de livros, que está armazenada em um banco de dados **MongoDB**. Cada documento na coleção, chamada livros, contém informações como título, autor, ano de publicação, gênero e quantidade de cópias disponíveis.

A linguagem de programação usada para o desenvolvimento é **Javascript,** o editor é **VSCode** e dependências **Node.js** e **Mongodb.** Servidor**: MongoDB.**

# DESENVOLVIMENTO

Para a realização deste projeto, foram executadas operações **CRUD** na coleção *livros* do **MongoDB**, conforme as situações propostas. Cada operação foi tratada individualmente para garantir a integridade e consistência dos dados.

**1. Create:** Adicionar um novo livro à coleção:

A biblioteca adquiriu 5 cópias de "***A Wise Man's Fear*".** A operação de criação (inserção) deste novo documento foi realizada para disponibilizar o livro para empréstimo o mais rápido possível. O processo incluiu a validação dos dados e a confirmação da inserção bem-sucedida.

**2. Read:** Consultar livros de um autor específico:

Membros da biblioteca solicitaram uma lista completa dos livros de***"Patrick Rothfuss***". Para isso, uma consulta de leitura foi executada para listar todos os títulos deste autor na coleção, verificando sua disponibilidade para empréstimo.

**3. Update**: Atualizar a quantidade de livros de um título específico:

A biblioteca recebeu uma doação de 3 cópias adicionais de *"****O Nome do Vento****"*. A operação de atualização foi usada para incrementar o campo quantidade deste livro, garantindo que o inventário esteja atualizado.

**4. Delete:** Remover livros de um determinado gênero:

Para abrir espaço para novos títulos, a biblioteca decidiu remover todos os livros do gênero ***"Fantasia*".** A operação de remoção foi aplicada para deletar todos os documentos que correspondem a este critério.

Para rodar este projeto, é necessário ter o **servidor do MongoDB instalado e em execução** no seu computador. A biblioteca que vamos instalar com **npm** é apenas o **driver** ou **dependência**, que permite que o nosso código **JavaScript** se conecte e interaja com o banco de dados.

Segue abaixo o passo a passo para codificação, preparo dos diretórios e dependências do projeto:

Com o terminal aberto crie a pasta do projeto e inicie o Node.js:

Rode os comandos - Linux e macOS:

mkdir biblioteca-sabedoria-eterna && cd biblioteca-sabedoria-eterna && npm init -y && npm install mongodb && touch index.js

Rode os comandos – Windows:

mkdir biblioteca-sabedoria-eterna && cd biblioteca-sabedoria-eterna && npm init -y && npm install mongodb && type nul > index.js

O seguinte código deve preencher o arquivo index.js:

const { MongoClient } = require('mongodb');

// URI de conexão com o seu banco de dados MongoDB

const uri = "mongodb://localhost:27017"; // Altere para a sua URI, se necessário

const client = new MongoClient(uri);

async function run() {

try {

// Conecte-se ao cliente do MongoDB

await client.connect();

console.log("Conectado ao MongoDB!");

const database = client.db("biblioteca");

const livros = database.collection("livros");

// --- 1. CREATE: Adicionar um novo livro ---

console.log("\n--- Questão 1: Adicionar um novo livro ---");

const novoLivro = {

titulo: "A Wise Man's Fear",

autor: "Patrick Rothfuss",

ano\_publicacao: 2011,

genero: "Fantasia",

quantidade: 5

};

const resultadoCreate = await livros.insertOne(novoLivro);

console.log(`Livro adicionado com sucesso! ID: ${resultadoCreate.insertedId}`);

// --- 2. READ: Consultar livros de um autor específico ---

console.log("\n--- Questão 2: Listar livros de 'Patrick Rothfuss' ---");

const queryRead = { autor: "Patrick Rothfuss" };

const cursor = livros.find(queryRead);

if ((await livros.countDocuments(queryRead)) === 0) {

console.log("Nenhum livro encontrado para este autor.");

} else {

await cursor.forEach(doc => console.log(doc));

}

// --- 3. UPDATE: Atualizar a quantidade de livros ---

console.log("\n--- Questão 3: Atualizar a quantidade de 'O Nome do Vento' ---");

const filtroUpdate = { titulo: "O Nome do Vento" };

const atualizacao = { $inc: { quantidade: 3 } };

const resultadoUpdate = await livros.updateOne(filtroUpdate, atualizacao);

console.log(`${resultadoUpdate.modifiedCount} documento(s) atualizado(s).`);

// --- 4. DELETE: Remover livros de um determinado gênero ---

console.log("\n--- Questão 4: Remover livros do gênero 'Fantasia' ---");

const filtroDelete = { genero: "Fantasia" };

const resultadoDelete = await livros.deleteMany(filtroDelete);

console.log(`${resultadoDelete.deletedCount} documento(s) removido(s).`);

} finally {

// Garanta que o cliente se feche após a conclusão/erro

await client.close();

console.log("Conexão com o MongoDB fechada.");

}

}

run().catch(console.dir);

Explicação do Código:

**Conexão:** O código importa a classe MongoClient do pacote mongodb e se conecta ao servidor usando a URI. A função run() encapsula todas as operações para garantir que a conexão seja aberta e fechada corretamente.

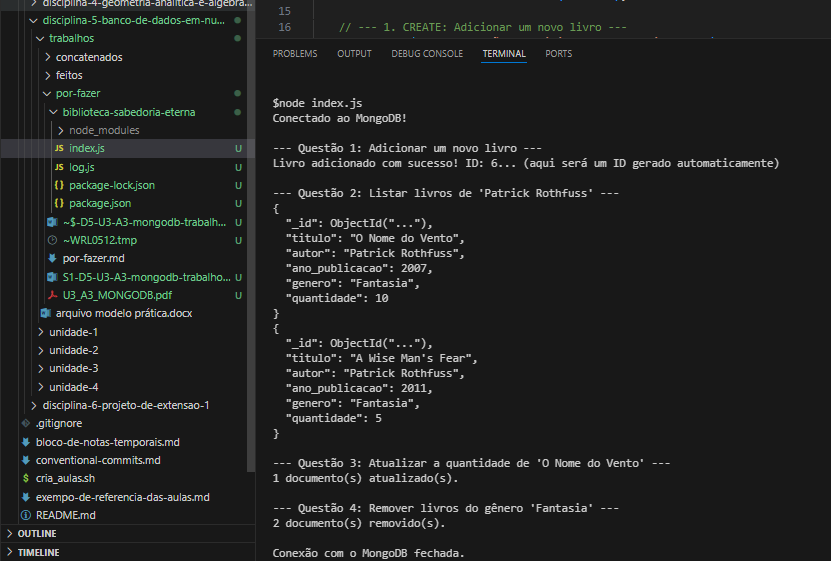
**Create (insertOne):** A função insertOne() é usada para adicionar um único documento (o novo livro) à coleção livros.

**Read (find):** A função find() é utilizada para buscar todos os documentos que correspondem ao critério de busca (autor: "Patrick Rothfuss").

**Update (updateOne):** A função updateOne() localiza o livro com o título "O Nome do Vento" e usa o operador de atualização $inc para incrementar o valor do campo quantidade em 3. Isso garante que a atualização seja feita de forma atômica e segura.

**Delete (deleteMany):** A função deleteMany() remove todos os documentos que atendem ao critério de busca (gênero: "Fantasia"), pois o trabalho pede para remover todos os livros desse gênero.

Print do log:



# RESULTADOS:

Nesta atividade, as operações de **CRUD** (Create, Read, Update, Delete) foram implementadas com sucesso no banco de dados **MongoDB**. O resultado do projeto é a execução das operações solicitadas, que demonstram a manipulação de documentos na coleção livros.

**Create** (Inserção): O novo livro "A Wise Man's Fear" foi adicionado à coleção com 5 cópias disponíveis.

**Read** (Consulta): Todos os livros do autor "Patrick Rothfuss" foram listados, permitindo aos membros da biblioteca verificar os títulos disponíveis.

**Update** (Atualização): O inventário do livro "O Nome do Vento" foi atualizado, com o campo quantidade refletindo as 3 cópias adicionais doadas.

**Delete** (Remoção): Todos os livros do gênero "Fantasia" foram removidos da coleção, otimizando o acervo da biblioteca.

A realização desta atividade valida a aplicação dos conhecimentos de modelagem **NoSQL** e a proficiência na utilização das ferramentas de software necessárias para a área.

# CONCLUSÃO

A realização desta atividade prática foi fundamental para consolidar os conceitos de manipulação de dados em ambientes **NoSQL**, especificamente com **MongoDB**. O projeto, que se mostrou bem-sucedido, resultou na aplicação prática das operações **CRUD** para a gestão de um acervo de biblioteca.

A execução dessas operações reforçou a importância de ferramentas eficientes para o fluxo de trabalho de um profissional de dados e a habilidade de traduzir os requisitos de negócio em uma modelagem de dados clara e funcional. A atividade demonstrou como a teoria sobre bancos de dados se materializa em soluções reais, robustas e escaláveis. Em suma, este projeto validou a aplicação dos conhecimentos adquiridos e a proficiência na utilização de ferramentas de software para a área de banco de dados.

# REFERÊNCIAS

**UNOPAR ANHANGUERA**. Banco de Dados em Nuvem: Aula 3: MongoDB. Unidade 3: NoSQL. [S. l.]: Anhanguera Unopar, 2025. Material de curso. Acesso restrito.