

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE  
SÃO PAULO – CAMPUS CAMPOS DO JORDÃO**

**BEATRIZ HELENA SILVA**

**PROJETO DE BANCO DE DADOS NoSQL**

**CAMPOS DO JORDÃO**

**2024**

## RESUMO

Tendo em vista, a versatilidade e relevância dos sistemas de bancos de dados para organizações e pensando em como criá-los de forma coerente que possa atender as necessidades atuais e futuras das organizações, o seguinte projeto tem como objetivo estudar a estruturação de um projeto de banco de dados relacional, bem como criar um projeto, aplicando as técnicas aqui estudadas, tendo em vista a importância e utilização desse modelo de banco de dados desde sua criação em 1970. É necessário que estudante de análise e desenvolvimento de sistemas, conheça as regras que regem a criação de projetos de banco de dados, bem como de sua implementação, dessa forma, faz-se necessário que o estudo seja aprofundado por meio de pesquisas e aplicado na forma de um projeto para a disciplina. Para elaboração do trabalho, deve-se responder quais as etapas de um projeto de banco de dados, como é sua implementação e quais os passos essenciais para criação desse sistema. A metodologia a ser aplicada, contemplará pesquisa teórica. Como bibliografia, utilizar-se-ão os materiais de aula, os livros de base elaborados por Heuser (2009) e Rob & Coronel (2011) em suas obras que tratam sobre os sistemas e projetos de banco de dados. Ao fim dessa pesquisa, espera-se ter um projeto passível de aplicação real para o negócio pretendido.

**Palavras-Chave:** Banco de Dados; Projetos; Banco de dados relacional; Sistemas de banco de dados;

## **ABSTRACT**

Considering the versatility and relevance of database systems for organizations, and aiming to create them coherently to a both current and future organizational needs, the following project aims to study the structuring of a relational database project. Additionally, it seeks to create a project by applying the techniques studied here, recognizing the importance and utilization of this database model since it is conception in 1970. It is essential for students of analysis and systems development to understand the rules governing database project creation and implementation. Therefore, in-depth research is necessary, followed by practical application in the form of a project for the course. To develop this work, we must address the stages of a database project, discuss its implementation, and outline the essential steps for creating such a system. The methodology applied will include theoretical research, drawing from course materials and foundational books by Heuser (2009) and Rob & Coronel (2011) that cover database systems and projects. Ultimately, the goal is to produce a project that can be practically applied to the intended business.”

**Keywords:** Database, Project, System of Relacional Database.

## LISTA DE FIGURAS

Figura 1: No SQL X SQL.....	10
Figura 2: DB No SQL.....	11
Figura 3: Diagrama de nós e relacionamentos.....	15

## **ACRÔNIMOS, SIGLAS E ABREVIações**

**IFSP** Instituto Federal de Educação, Ciência e Tecnologia de São Paulo

**SGBD** Sistema de Gerenciamento de Banco de Dados

**DER** Diagrama entidade-relacionamento

**RSGB** Sistema de Gerenciamento de Banco de Dados Relacional

# SUMÁRIO

LISTA DE FIGURAS.....	4
ACRÔNIMOS, SIGLAS E ABREVIACÕES.....	5
SUMÁRIO.....	6
1 INTRODUÇÃO.....	7
1.1 Objetivos.....	7
1.2 Justificativa.....	7
1.3 Aspectos Metodológicos.....	7
1.4 Aporte Teórico.....	8
2 SISTEMAS DE BANCO DE DADOS NÃO RELACIONAIS.....	9
2.1 Tipos de bancos de dados NoSQL.....	10
2.2 Exemplos de banco de dados não relacional.....	10
3 NEO4J.....	12
3.1 Criando um projeto.....	12
3.2 Criando o banco de dados.....	12
3.3 Exemplos de Consultas.....	13
3.3.1 Listar todas as pessoas.....	13
3.3.2 Encontrar os amigos de uma pessoa.....	14
3.3.3 Encontrar pessoas que têm interesse em 'Música Pop'.....	14
3.3.4 Recomendar amigos baseados em interesses comuns.....	14
3.4 Como adicionar e atualizar dados.....	14
3.4.1 Adicionar uma nova pessoa.....	14
3.4.2 Conectar uma pessoa a outra como amigo.....	14
3.4.3 Atualizar a idade de uma pessoa.....	14
3.4 Deletar Dados.....	15
3.5 Criando Constraints (Restrições).....	15
3.6 Criando Path (Caminho).....	15
3.7 Diagrama.....	15
4 CONCLUSÃO.....	16
BIBLIOGRAFIA.....	17

# **1 INTRODUÇÃO**

Este trabalho tem como objetivo entender e aplicar os conhecimentos adquiridos durante as aulas e por meio de pesquisas para criação de banco de dados NoSQL. Ao final, espera-se apresentar um projeto de banco de dados que possa ser implementado utilizando as ferramentas adequadas.

Além disso, pretende-se criar um projeto no decorrer do estudo, compreendendo os requisitos e as regras de negócio, bem como a construção do sistema baseada nesses aspectos.

## **1.1 Objetivos**

Com base apresentado, esse trabalho tem por objetivo:

- Apresentar a fundação teórica utilizada para criação de um BD não relacional;
- Explorar as etapas de criação de um projeto de um BD não relacional;
- Entender os requisitos e regras de negócio;
- Expor um projeto de BD não relacional para esse negócio.

## **1.2 Justificativa**

É essencial para os estudantes que aspiram a trabalhar com sistemas de banco de dados não relacionais em diversos negócios, compreender o passo a passo de sua criação e a fundamentação teórica antes de sua construção como sistema. Esse entendimento é crucial para construir sistemas que ofereçam bons resultados e poucas falhas, ou seja, que sejam capazes de atender às necessidades do negócio, ou, em outras palavras, aos requisitos e regras declarados. Também, é imprescindível que os alunos compreendam as diferenças entre banco de dados relacionais e não relacionais e como trabalhar com ambos.

## **1.3 Aspectos Metodológicos**

Para o seguinte estudo, utilizaram-se pesquisas bibliográficas de diversos tipos de fontes, o mesmo, para o que se refere à parte prática.

#### **1.4 Aporte Teórico**

Considera-se que as principais bases teóricas para o embasamento desse trabalho foram captadas por meio de sites e artigos online.



## **2 SISTEMAS DE BANCO DE DADOS NÃO RELACIONAIS**

De acordo com a Oracle, o banco de dados No SQL ou não relacional, são bancos que armazenam dados em um formato diferente das tabelas relacionais. Segundo o site da IBM Brasil, no lugar da estrutura tabular típica dos RDBMS, os bancos de dados NoSQL armazenam dados em estruturas de dados, como esse não exige um esquema, oferece rápida escalabilidade para gerenciar conjuntos de dados grandes e normalmente não estruturados.

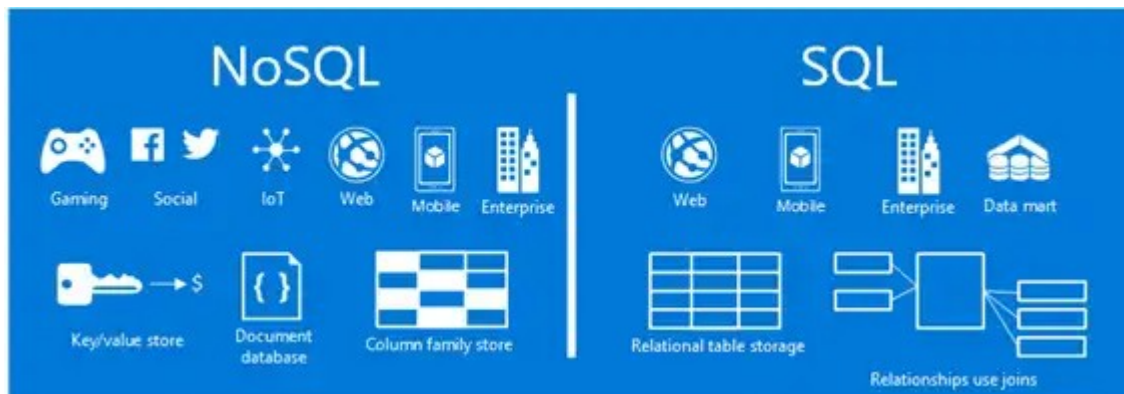
Também, o NoSQL é um tipo de banco de dados o qual é distribuído, ou seja, suas informações são copiadas e armazenadas em vários servidores, remotos ou locais, garantindo a disponibilidade e a confiabilidade dos dados. Se alguns dos dados ficarem offline, o restante do banco de dados poderá continuar a ser executado.

Os bancos de dados SQL são baseados em um modelo relacional, enquanto os bancos de dados NoSQL adotam um modelo não relacional. Os sistemas de gerenciamento de banco de dados relacional (RDBMS) utilizam a linguagem de consulta estruturada (SQL) para permitir que os usuários acessem e manipulem dados em tabelas altamente estruturadas. Exemplos de sistemas de banco de dados baseados nesse modelo incluem MS SQL Server, IBM DB2, Oracle e MySQL.

Os bancos de dados NoSQL normalmente dependem de dados desnormalizados, suportando os tipos de aplicativos que usam menos tabelas e os quais os relacionamentos de dados não são modelados usando referências, mas sim como registros (ou documentos) incorporados.

Os bancos de dados NoSQL funcionam incrivelmente bem com consultas em uma única tabela. No entanto, conforme a complexidade das consultas aumenta, os bancos de dados relacionais são uma escolha melhor. O banco de dados NoSQL normalmente não oferece junções complexas, subconsultas e aninhamento de consultas em uma cláusula WHERE.

Dentre os benefícios do sistema NoSQL e o motivo para sua crescente utilização, destacam-se a Flexibilidade, a Escalabilidade, o Alto desempenho, a Disponibilidade e o fato desse sistema ser Altamente Funcional.



*Figura 1: No SQL X SQL*

## 2.1 Tipos de bancos de dados NoSQL

Existem alguns tipos de banco de dados NoSQL, os quais pode-se citar: O tipo Valor Principal, o qual é o mais flexível de banco de dados NoSQL porque o aplicativo tem controle total sobre o que é armazenado no campo de valor sem quaisquer restrições. O tipo Documento, o qual também é conhecido como armazenamento de documentos ou bancos de dados orientados a documentos, esses bancos de dados são usados para armazenar, recuperar e gerenciar dados semiestruturados, sem que haja a necessidade de especificar quais campos um documento conterá.

O tipo Gráfico, o qual organiza os dados como nós e relacionamentos, que mostram as conexões entre os nós. Isso oferece suporte a uma representação de dados mais rica e completa. Bancos de dados gráficos são aplicados em redes sociais, sistemas de reserva e detecção de fraudes. O tipo Coluna Larga, que armazenam e gerenciam dados na forma de tabelas, linhas e colunas. Eles são amplamente implantados em aplicativos que requerem um formato de coluna para capturar dados sem esquema.

## 2.2 Exemplos de banco de dados não relacional

De acordo com a Awari alguns exemplos de banco de dados não relacional, englobam o Amazon DynamoDB, Bigtable, Cassandra, Hbase, Google Cloud Datastore, entre outros, são exemplos populares de bancos NoSQL colunares. Já no site Saphir encontram-se como exemplos os sistemas de banco de dados não

relacionais: Redis, Memcached, Hbase, Neo4j e MongoDB.



*Figura 2: DB No SQL*

### 3 NEO4J

De acordo com Alves (2019) o Neo4J é um banco de dados NoSQL orientado a grafos, o qual tem como prioridade, tratar os relacionamentos da melhor maneira, ou seja, na medida que os relacionamentos entre “nós” aumentam, sua capacidade de processamento continua estável, diferentemente de bancos de dados tradicionais. Ainda segundo Alves(2019), A linguagem do Neo4J é a cypherQuery, e a curva de aprendizagem não é tão grande.

Também, de acordo com a Saphir o Neo4J é uma implementação de código aberto e pode ser útil para casos de mineração de dados e reconhecimento de padrões.

#### 3.1 Criando um projeto

No NEO4J os dados são organizados por nós e relacionamentos, ou seja, um nó se liga ao outro por meio de um relacionamento. Para esse trabalho, considerar a construção de uma rede social simples, nas quais os usuários têm suas características, seus interesses e seus amigos.

Então, nesse caso:

Nós:

**Pessoa:** Contém atributos como nome, idade, e-mail, e cidade.

**Interesse:** Contém atributos como nome (ex.: Música Pop, rugby, Ficção Científica, etc).

**Relacionamentos:**

**AMIGO\_DE:** Representa amizade entre pessoas.

**TEM\_INTERESSE:** Conecta uma pessoa a um interesse.

#### 3.2 Criando o banco de dados

É necessário criar, primeiramente, os nós que contenham as informações do usuário, depois os nós que contém os interesses e, por fim, realizar as ligações que formam os relacionamentos.

```

1 // Criar nós de pessoas
2 CREATE (:Pessoa {nome: 'Alice Abacaxi', idade: 25, email:
3 aliceabacaxi@gmail.com, cidade: 'São Paulo'});
4 CREATE (:Pessoa {nome: 'João Jaca', idade: 30, email:
5 joaojaca@gmail.com, cidade: 'Rio de Janeiro'});
6 CREATE (:Pessoa {nome: 'Carol Carambola', idade: 22, email:
7 carolcarambols@gmail.com, cidade: 'Belo Horizonte'});
8
9 // Criar nós de interesses
10 CREATE (:Interesse {nome: 'Música Pop'});
11 CREATE (:Interesse {nome: 'Rugby'});
12 CREATE (:Interesse {nome: 'Ficção Científica'});
13
14 // Criar relacionamentos de amizade
15 MATCH (a:Pessoa {nome: 'Alice'}), (b:Pessoa {nome: 'João'})
16 CREATE (a)-[:AMIGO_DE]->(b);
17
18 MATCH (b:Pessoa {nome: 'Bob'}), (c:Pessoa {nome: 'Carol'})
19 CREATE (b)-[:AMIGO_DE]->(c);
20
21
22 // Conectar pessoas a interesses
23 MATCH (a:Pessoa {nome: 'Alice'}), (i:Interesse {nome: 'Música Pop'})
24 CREATE (a)-[:TEM_INTERESSE]->(i);
25
26 MATCH (b:Pessoa {nome: 'João'}), (i:Interesse {nome: 'Rugby'})
27 CREATE (b)-[:TEM_INTERESSE]->(i);
28
29 MATCH (c:Pessoa {nome: 'Carol'}), (i:Interesse {nome: 'Ficção
30 Científica'})
31 CREATE (c)-[:TEM_INTERESSE]->(i);

```

### 3.3 Exemplos de Consultas

#### 3.3.1 Listar todas as pessoas

```

1 MATCH (p:Pessoa)
2 RETURN p.nome, p.idade, p.cidade;

```

### 3.3.2 Encontrar os amigos de uma pessoa

```
1 MATCH (p:Pessoa)-[:AMIGO_DE]->(amigo:Pessoa)
2 WHERE p.nome = 'Alice'
3 RETURN amigo.nome;
```

### 3.3.3 Encontrar pessoas que têm interesse em 'Música Pop'

```
1 MATCH (p:Pessoa)-[:TEM_INTERESSE]->(i:Interesse)
2 WHERE i.nome = 'Música Pop'
3 RETURN p.nome, p.cidade;
```

### 3.3.4 Recomendar amigos baseados em interesses comuns

**Obs:** buscar as pessoas que compartilham interesses com a mesma pessoa.

```
1 MATCH (p:Pessoa)-[:TEM_INTERESSE]->(i:Interesse)-[:TEM_INTERESSE]-
2 (outra:Pessoa)
3 WHERE p.nome = 'Alice' AND NOT (p)-[:AMIGO_DE]->(outra)
4 RETURN outra.nome, i.nome;
```

## 3.4 Como adicionar e atualizar dados

Métodos para adicionar novos dados ou atualizar dados já existentes no DB.

### 3.4.1 Adicionar uma nova pessoa

```
1 CREATE (:Pessoa {nome: 'Marcos Melão', idade: 28, email:
2 marcosmelao@gmail.com, cidade: 'Curitiba'});
```

### 3.4.2 Conectar uma pessoa a outra como amigo

```
1 MATCH (p1:Pessoa {nome: 'Davi'}), (p2:Pessoa {nome: 'Alice'})
2 CREATE (p1)-[:AMIGO_DE]->(p2);
```

### 3.4.3 Atualizar a idade de uma pessoa

```
1 MATCH (p:Pessoa {nome: 'Bob'})
2 SET p.idade = 31
3 RETURN p;
```

### 3.4 Deletar Dados

**Obs:** Deletar dados no Neo4j pode ser irreversível.

```
1 MATCH (p:Pessoa {nome: 'Davi'})
2 DETACH DELETE p;
```

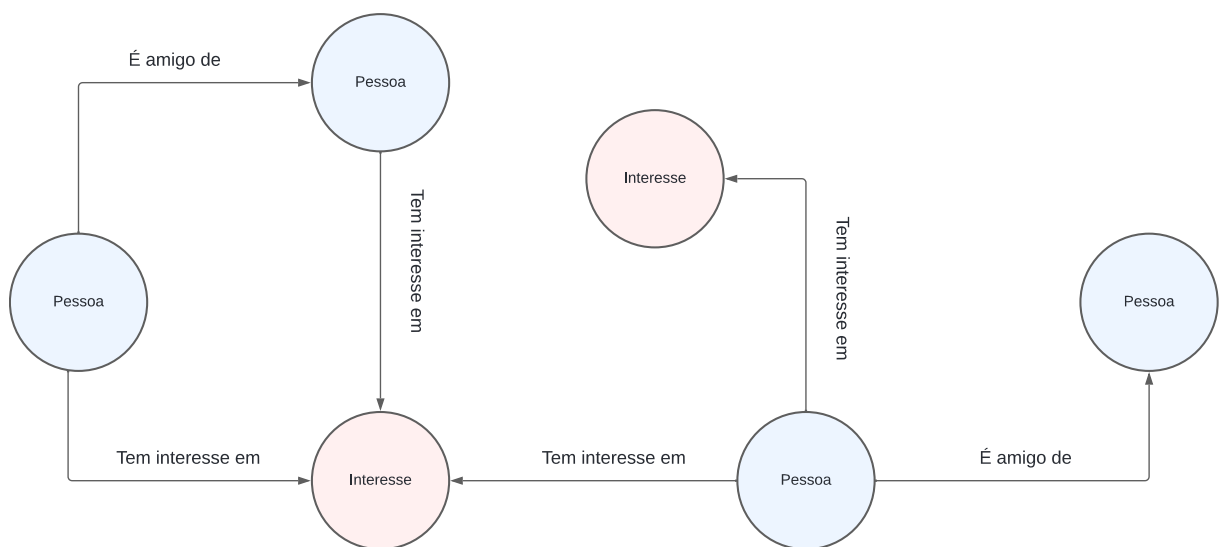
### 3.5 Criando Constraints (Restrições)

```
1 CREATE CONSTRAINT ON (p:Pessoa) ASSERT p.nome IS UNIQUE;
```

### 3.6 Criando Path (Caminho)

```
1 MATCH p = (Alice)-[:AMIGO_DE]->(Bob)-[:AMIGO_DE]->(Carol) RETURN p;
```

### 3.7 Diagrama



*Figura 3: Diagrama de nós e relacionamentos*

## **4 CONCLUSÃO**

Com base nos princípios expostos, os bancos de dados NoSQL proporcionam uma estratégia não relacional para o armazenamento de dados. Eles se sobressaem pela sua adaptabilidade, escalabilidade e desempenho elevado, particularmente em aplicações que manipulam grandes quantidades de dados e conexões complexas, como redes sociais e sistemas de recomendação.

A decisão entre um banco de dados SQL ou NoSQL é determinada pelas exigências particulares do projeto. Embora os bancos relacionais sejam perfeitos para consultas complexas e altamente estruturadas, os bancos NoSQL, como Neo4j, MongoDB e Cassandra, se mostram mais apropriados para situações que requerem rapidez, armazenamento de dados semiestruturados ou a representação de relações complexas, como em grafos.

O Neo4j, como banco de dados gráfico, exemplifica a eficácia do NoSQL em aplicações que priorizam conexões, demonstrando o potencial desse modelo para resolver problemas modernos de armazenamento e análise de dados.



## BIBLIOGRAFIA

ALVES, M. **NEO4J — Primeiros passos.** Disponível em: <<https://medium.com/neo4j-primeiros-passos/neo4j-primeiros-passos-ee4e319e33bb>>. Acesso em 02 de dezembro de 2024.

**Conheça os principais bancos de dados NoSQL (não-relacionais).** Disponível em: <<https://blog.saphir.com.br/conheca-os-principais-bancos-de-dados-nosql-nao-relacionais/>>. Acesso em 02 de dezembro de 2024.

EDPRICE-MSFT. **Dados não relacionais e NoSQL - Azure Architecture Center.** Disponível em: <<https://learn.microsoft.com/pt-br/azure/architecture/data-guide/big-data/non-relational-data>>. Acesso em 02 de dezembro de 2024.

**O que são bancos de dados NoSQL? | IBM.** Disponível em: <<https://www.ibm.com/br-pt/topics/nosql-databases>>. Acesso em 03 de dezembro de 2024.

**Por Que os Desenvolvedores Preferem Bancos de Dados NoSQL?** Disponível em: <<https://www.oracle.com/br/database/nosql/what-is-nosql/>>. Acesso em 03 de dezembro de 2024.

