


Capítulo 6 – Neo4j

Dia 1 – Grafos, Cypher e CRUD

O primeiro passo com o Neo4j é entender como criar e manipular **nós** e **relacionamentos** usando a linguagem **Cypher**. As operações seguem o conceito de CRUD (Create, Read, Update, Delete), mas adaptadas ao modelo de grafos.

Criando Nós

```
CREATE (:Pessoa {nome: "Douglas", idade: 30})
CREATE (:Pessoa {nome: "Ana", idade: 28})
```

 Cada `(:Label {propriedades})` cria um nó com rótulo (`Pessoa`) e propriedades.

Criando Relacionamentos

```
MATCH (p1:Pessoa {nome: "Douglas"}), (p2:Pessoa {nome: "Ana"})
CREATE (p1)-[:AMIGO_DE {desde: 2020}]->(p2)
```

- `MATCH` encontra os nós existentes.
- `CREATE` define um relacionamento `[:AMIGO_DE]` com propriedade `desde`.

Lendo Dados

- Buscar todos os nós:

```
MATCH (n) RETURN n
```

- Buscar pessoas e suas amizades:


```
MATCH (p:Pessoa)-[r:AMIGO_DE]->(a:Pessoa)
RETURN p.nome, a.nome, r.desde
```

- Filtrar com `WHERE`:

```
MATCH (p:Pessoa)-[:AMIGO_DE]->(a:Pessoa)
WHERE p.nome = "Douglas"
RETURN a.nome
```

Atualizando Dados

```
MATCH (p:Pessoa {nome: "Douglas"})
SET p.idade = 31
```

 **SET** adiciona ou atualiza propriedades.

Deletando

- Deletar apenas propriedades:

```
MATCH (p:Pessoa {nome: "Douglas"})
REMOVE p.idade
```

- Deletar relacionamento:

```
MATCH (p1:Pessoa {nome: "Douglas"})-[:AMIGO_DE]->(p2:Pessoa {nome:
"Ana"})
DELETE r
```

- Deletar nó (se não tiver relacionamentos):

```
MATCH (p:Pessoa {nome: "Douglas"})
DELETE p
```


- Forçar exclusão (nó + relacionamentos):

```
MATCH (p:Pessoa {nome: "Douglas"})
DETACH DELETE p
```

Resumo do Dia 1

- **CREATE** para nós e relacionamentos.
- **MATCH** para buscar padrões.

- **SET/REMOVE** para atualizar dados.
- **DELETE/DETACH DELETE** para excluir nós e relacionamentos.

 No próximo passo: **Dia 2 – REST, Indexes e Algoritmos**, onde exploraremos como expor consultas via API REST, criar índices e usar algoritmos de grafos no Neo4j.