

## Capítulo 6 – Neo4j


---

### Introdução – Neo4j é “Quadro Branco Amigável”

O **Neo4j** é o banco de dados mais conhecido do gênero **grafos**. Diferente de bancos relacionais ou de documentos, ele foi projetado para lidar **nativamente com relacionamentos complexos**.

#### Filosofia

- Enquanto um banco relacional usa tabelas e chaves estrangeiras para modelar relacionamentos, o Neo4j os trata como **entidades de primeira classe**.
- Os dados são organizados como **nós** (entidades) e **arestas** (relacionamentos).
- Isso permite consultas extremamente rápidas em cenários com muitas conexões (ex.: redes sociais, recomendações, mapas, genealogias).

 O apelido “whiteboard friendly” (amigável ao quadro branco) vem do fato de que podemos desenhar um modelo de grafo no quadro e representá-lo **quase 1:1** no Neo4j.

---

#### Estrutura básica de dados

##### 1. Nós (nodes)

2. Representam entidades (ex.: uma pessoa, um produto, um local).
3. Podem ter **labels** (rótulos) e **propriedades** (chaves/valores).

Exemplo:

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

##### 1. Arestas (relationships)

2. Conectam dois nós.
3. Sempre têm uma direção (→), mas podem ser consultadas em ambos os sentidos.
4. Também podem ter **propriedades**.

Exemplo:

```
(:Pessoa {nome: "Douglas"})-[:AMIGO_DE {desde: 2020}]->(:Pessoa {nome: "Ana"})
```

##### 1. Propriedades


2. Tanto nós quanto relacionamentos podem ter atributos adicionais.
-

## Linguagem de consulta: Cypher

O Neo4j usa a linguagem **Cypher**, projetada para ser declarativa e intuitiva.

- Sintaxe lembra diagramas de grafos: usamos **parênteses** para nós e **colchetes** para relacionamentos.
- Exemplo simples:

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

 Consulta todos os amigos da pessoa chamada *Douglas*.

---

## Casos de uso comuns

- **Redes sociais** (amigos de amigos, conexões em comum).
  - **Recomendações** (usuários que compraram X também compraram Y).
  - **Sistemas de rota e logística** (caminho mais curto em um grafo de cidades).
  - **Fraude financeira** (detecção de padrões em transações conectadas).
  - **Ontologias e conhecimento** (representar conceitos e suas relações).
- 

## Conclusão da Introdução

O Neo4j traz uma forma diferente de pensar dados: **não em tabelas, mas em conexões**.

Isso o torna poderoso para qualquer domínio em que os relacionamentos importem tanto quanto as entidades.

 No próximo passo: **Dia 1 - Grafos, Cypher e CRUD**, onde criaremos os primeiros nós, relacionamentos e consultas.