

Trabalho 2: Modelagem e Análise de Dados Legislativos com Neo4j

Banco de Dados para Ciência de Dados

Guilherme César Athayde - 748175
`guilherme.athayde@estudante.ufscar.br`

Nome do Segundo Autor - RA
`email@estudante.ufscar.br`

23 de outubro de 2025

Sumário

1	Introdução	3
1.1	Dataset Escolhido	3
1.2	Características Importantes do Dataset	3
2	Coleta de Dados	3
2.1	Script de Coleta	4
3	Modelagem dos Dados	4
3.1	Entidades (Nós)	4
3.2	Relacionamentos (Arestas)	6
4	Importação dos Dados	6
4.1	Script de Importação	6
5	Consulta de Verificação	7
5.1	Script da Consulta	7
5.2	Resultado da Consulta	8
6	Consultas Analíticas e Insights	8
6.1	Consulta 1: Presença Partidária em Votações	8
6.2	Consulta 2: Impacto de Proposições Formais	9
6.3	Consulta 3: Análise Semântica de Descrições	9
6.4	Consulta 4: Volatilidade de Resultados Partidários	10
6.5	Consulta 5: Deputados Mais Influentes	11
7	Conclusão	11
7.1	Aprendizados	11
7.2	Desafios Encontrados	12
7.3	Repositório do Projeto	12

1 Introdução

Este trabalho apresenta a modelagem, importação e análise de dados legislativos da Câmara dos Deputados do Brasil utilizando o banco de dados de grafos Neo4j. O objetivo é explorar as relações entre deputados, partidos, proposições e votações para gerar insights sobre o funcionamento do processo legislativo brasileiro.

1.1 Dataset Escolhido

O dataset utilizado foi coletado da API de Dados Abertos da Câmara dos Deputados (<https://dadosabertos.camara.leg.br>), contendo informações sobre:

- **Deputados:** 513 registros contendo informações sobre parlamentares da 57^a Legislatura (2023-2027), incluindo nome, partido, UF, email e foto.
- **Partidos:** 28 partidos políticos com representação na Câmara, contendo sigla, nome completo e URI de referência.
- **Proposições:** 15 proposições legislativas (PLs, PECs, etc.) com ementa, tipo, número e ano.
- **Votações:** 100 registros de votações realizadas em diferentes órgãos (Plenário, comissões) com descrição, resultado (aprovado/rejeitado) e data.
- **Órgãos:** 8 órgãos legislativos incluindo o Plenário e diversas comissões temáticas.
- **Frentes Parlamentares:** 15 frentes parlamentares temáticas da legislatura atual.

Os dados referem-se principalmente ao período de 2023-2025, com foco na legislatura atual e votações recentes (outubro de 2025).

1.2 Características Importantes do Dataset

- Os dados são estruturados em formato JSON seguindo o padrão da API REST da Câmara.
- Existem relacionamentos naturais entre entidades: deputados pertencem a partidos, votações são realizadas por órgãos, proposições podem estar vinculadas a votações.
- As descrições de votações frequentemente mencionam nomes de deputados (relatores, autores), permitindo análises textuais.
- O campo `aprovacao` nas votações indica o resultado (1=aprovado, 0=rejeitado, null=outros).

2 Coleta de Dados

Os dados foram coletados através de requisições HTTP à API de Dados Abertos da Câmara dos Deputados. O script de coleta utiliza `curl` para fazer as requisições e `jq` para formatar o JSON de saída.

2.1 Script de Coleta

Listing 1: Script de coleta de dados

```
1 cd datasets
2 curl -s -H "Content-Type: application/json" \
3     https://dadosabertos.camara.leg.br/api/v2/deputados | jq '.' >
4     deputados.json
5 curl -s -H "Content-Type: application/json" \
6     https://dadosabertos.camara.leg.br/api/v2/proposicoes | jq '.' >
7     proposicoes.json
8 curl -s -H "Content-Type: application/json" -H "Accept: application/json" \
9     https://dadosabertos.camara.leg.br/api/v2/votacoes | jq '.' > votacoes
10     .json
11 curl -s -H "Content-Type: application/json" \
12     https://dadosabertos.camara.leg.br/api/v2/orgaos | jq '.' > orgaos.
13     json
14 curl -s -H "Content-Type: application/json" \
15     https://dadosabertos.camara.leg.br/api/v2/partidos | jq '.' > partidos
16     .json
17 curl -s -H "Content-Type: application/json" \
18     https://dadosabertos.camara.leg.br/api/v2/frentes | jq '.' > frentes.
19     json
```

3 Modelagem dos Dados

A modelagem dos dados em grafo foi projetada para representar as relações naturais entre as entidades do processo legislativo. O diagrama abaixo ilustra o modelo de dados implementado no Neo4j:

3.1 Entidades (Nós)

- **Partido:** Partidos políticos com sigla, nome e URI.
- **Deputado:** Parlamentares com id, nome, partido, UF, legislatura, email e foto.
- **Legislatura:** Períodos legislativos identificados por ID.
- **Frente:** Frentes parlamentares temáticas.
- **Orgao:** Órgãos legislativos (Plenário, comissões).
- **Proposicao:** Proposições legislativas com tipo, número, ano e ementa.
- **Votacao:** Eventos de votação com data, descrição e resultado.

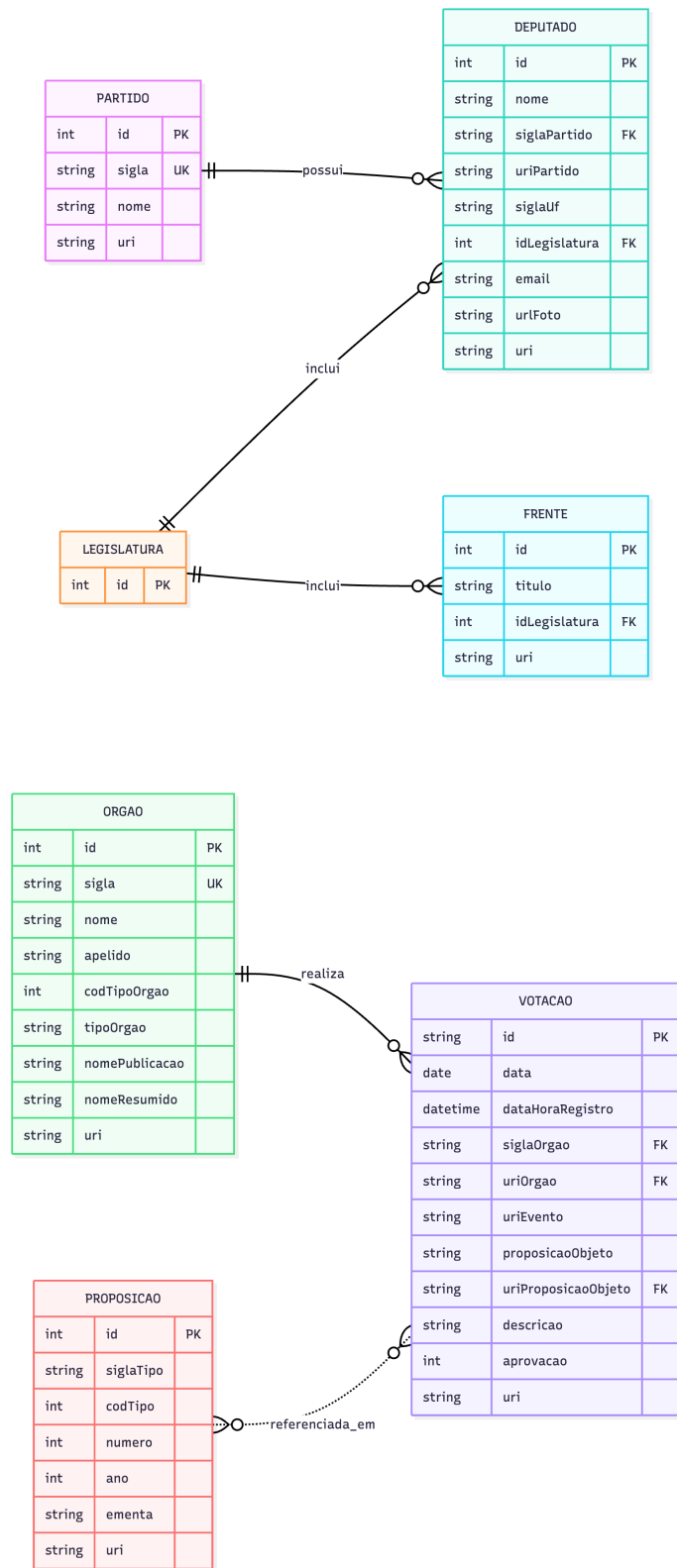


Figura 1: Diagrama do modelo de dados em grafo

3.2 Relacionamentos (Arestas)

- PERTENCE_A: Deputado → Partido
- ATUA_NA_LEGISLATURA: Deputado → Legislatura
- DA_LEGISLATURA: Frente → Legislatura
- REALIZA: Orgao → Votacao
- REFERENCIADA_EM: Proposicao → Votacao

4 Importação dos Dados

A importação dos dados JSON para o Neo4j foi realizada através de um script Python que utiliza o driver oficial do Neo4j. O script implementa as seguintes funcionalidades:

- Leitura dos arquivos JSON do diretório `datasets/`
- Criação de constraints de unicidade para cada entidade
- Merge (criação ou atualização) dos nós
- Criação dos relacionamentos entre nós
- Tratamento de valores nulos para evitar poluição do grafo

4.1 Script de Importação

Listing 2: Trecho principal do script de importação (`import_neo4j.py`)

```
1 def merge_deputados(driver: Driver, database: str,
2                     deputados: List[Dict[str, Any]]):
3     query = """
4     UNWIND $rows AS row
5     MERGE (d:Deputado {id: row.id})
6         ON CREATE SET d.nome = row.nome,
7                       d.siglaPartido = row.siglaPartido,
8                       d.siglaUf = row.siglaUf,
9                       d.idLegislatura = row.idLegislatura,
10                      d.email = row.email,
11                      d.urlFoto = row.urlFoto,
12                      d.uri = row.uri
13     ON MATCH SET d.nome = row.nome,
14                  d.siglaPartido = row.siglaPartido,
15                  d.siglaUf = row.siglaUf,
16                  d.idLegislatura = row.idLegislatura,
17                  d.email = row.email,
18                  d.urlFoto = row.urlFoto,
19                  d.uri = row.uri
20     WITH d, row
21     MATCH (p:Partido {sigla: row.siglaPartido})
22     MERGE (p)<-[:PERTENCE_A]-(d)
23     WITH d, row
24     MATCH (l:Legislatura {id: row.idLegislatura})
```

```

25     MERGE (d)-[:ATUA_NA_LEGISLATURA]->(l)
26     """
27     driver.execute_query(query, rows=deputados, database_=database)
28
29 # Processo principal de importacao
30 print("Conectado ao Neo4j. Criando constraints...")
31 create_constraints(driver, args.database)
32 print("Importando Partidos...")
33 merge_partidos(driver, args.database, partidos)
34 print("Importando Legislaturas...")
35 merge_legislaturas(driver, args.database, legislatura_ids)
36 print("Importando Deputados...")
37 merge_deputados(driver, args.database, deputados)
38 print("Importando Frentes...")
39 merge_frentes(driver, args.database, frentes)
40 print("Importando Orgaos...")
41 merge_orgaos(driver, args.database, orgaos)
42 print("Importando Proposicoes...")
43 merge_proposicoes(driver, args.database, proposicoes)
44 print("Importando Votacoes...")
45 merge_votacoes(driver, args.database, votacoes)
46 print("Importacao concluida.")

```

5 Consulta de Verificação

Após a importação, foi realizada uma consulta para verificar todos os dados importados no banco de dados Neo4j.

5.1 Script da Consulta

Listing 3: Consulta de verificação em main.ipynb

```

1 from neo4j import GraphDatabase
2
3 URI = "neo4j+s://ae951e8f.databases.neo4j.io"
4 USERNAME = "neo4j"
5 PASSWORD = "E_shPCUTJVDIZmj-iWC-p2skPNyrPBcT148nZyh9BgA"
6
7 driver = GraphDatabase.driver(URI, auth=(USERNAME, PASSWORD))
8 driver.verify_connectivity()
9
10 query = """
11 OPTIONAL MATCH (n)-[r]->(m)
12 RETURN n, r, m
13 """
14
15 records, summary, keys = driver.execute_query(query, database_="neo4j")
16
17 for i, record in enumerate(records):
18     node_n = record["n"]
19     relationship_r = record["r"]
20     node_m = record["m"]
21
22     print(f"\nRegistro {i+1}:")
23     if node_n:

```

```

24     print(f"    No de Origem (n): Labels={list(node_n.labels)}, "
25           f"Propriedades={dict(node_n)}")
26     if relationship_r:
27         print(f"    Relacao (r): Tipo='{relationship_r.type}', "
28               f"Propriedades={dict(relationship_r)}")
29     if node_m:
30         print(f"    No de Destino (m): Labels={list(node_m.labels)}, "
31               f"Propriedades={dict(node_m)}")
32
33 driver.close()

```

5.2 Resultado da Consulta

A consulta retornou 476 registros de relacionamentos entre nós, confirmando a importação bem-sucedida de:

- 513 deputados vinculados a partidos e legislaturas
- 28 partidos
- 1 legislatura (ID 57)
- 15 frentes parlamentares
- 8 órgãos
- 15 proposições
- 100 votações

O resultado completo encontra-se no arquivo `main.txt`, mostrando todos os nós, relacionamentos e suas propriedades.

6 Consultas Analíticas e Insights

Foram desenvolvidas 5 consultas Cypher avançadas para extrair insights sobre o funcionamento do processo legislativo. As consultas exploram diferentes dimensões do grafo, incluindo análises partidárias, comportamento de votações e identificação de atores influentes.

6.1 Consulta 1: Presença Partidária em Votações

Pergunta: Quais partidos aparecem mais frequentemente citados nas descrições de votações (indicando atuação de seus deputados como relatores ou autores) e essa presença está associada a maior taxa de aprovação?

Listing 4: Consulta Cypher 1

```

1 MATCH (p:Partido) <-[:PERTENCE_A]-(d:Deputado)
2 MATCH (o:Orgao)-[:REALIZA]->(v:Votacao)
3 WHERE v.descricao CONTAINS d.nome
4 WITH p, o, v
5 WITH p.sigla AS partido, o.sigla AS orgao,

```



```

6      collect(DISTINCT v.id) AS votacoes_ids,
7      count(DISTINCT v) AS qtd_votacoes,
8      avg(CASE WHEN v.aprovacao = 1 THEN 1.0 ELSE 0.0 END) AS taxa_aprov
9 RETURN partido, orgao, qtd_votacoes,
10      round(100.0 * taxa_aprov,2) AS taxaAprovacaoPercent,
11      votacoes_ids
12 ORDER BY qtd_votacoes DESC, taxaAprovacaoPercent DESC

```

Análise: Esta consulta aproxima a “visibilidade” de um partido nas votações via menções diretas a nomes de deputados em descrições oficiais (frequentemente indicando relatoria ou protagonismo). Partidos com alto número de menções em conjunto com elevada taxa de aprovação sugerem capacidade de pautar e conduzir matérias a resultados positivos, refletindo coordenação interna ou posicionamento estratégico em comissões e plenário. Já partidos muito citados com baixa taxa de aprovação podem estar assumindo temas controversos, atuando em obstrução ou em fases intermediárias (ex.: emendas rejeitadas). O insight auxilia na identificação de quais siglas convertem presença em efetividade decisória.

6.2 Consulta 2: Impacto de Proposições Formais

Pergunta: Votações associadas explicitamente a uma proposição têm maior taxa de aprovação em cada órgão do que votações sem essa referência?

Listing 5: Consulta Cypher 2

```

1 MATCH (o:Orgao)-[:REALIZA]->(v:Votacao)
2 OPTIONAL MATCH (p:Proposicao)-[:REFERENCIADA_EM]->(v)
3 WITH o, v, CASE WHEN p IS NULL THEN 0 ELSE 1 END AS temProposicao
4 WITH o.sigla AS orgao,
5      count(v) AS total,
6      sum(temProposicao) AS comProp,
7      avg(CASE WHEN v.aprovacao = 1 THEN 1.0 ELSE 0.0 END) AS taxa_geral,
8      avg(CASE WHEN temProposicao = 1 AND v.aprovacao = 1
9           THEN 1.0 ELSE 0.0 END) AS taxa_com_prop,
10     avg(CASE WHEN temProposicao = 0 AND v.aprovacao = 1
11          THEN 1.0 ELSE 0.0 END) AS taxa_sem_prop
12 RETURN orgao, total,
13        comProp AS votacoesComProposicao,
14        round(100.0 * comProp / total,2) AS percComProposicao,
15        round(100.0 * taxa_geral,2) AS taxaAprovGeral,
16        round(100.0 * taxa_com_prop,2) AS taxaAprovComProposicao,
17        round(100.0 * taxa_sem_prop,2) AS taxaAprovSemProposicao
18 ORDER BY total DESC

```

Análise: A diferença entre taxa de aprovação com e sem proposição vinculada revela o papel do rito formal na previsibilidade do resultado. Órgãos em que a aprovação depende fortemente da presença de uma proposição podem funcionar como estágios de consolidação técnica/jurídica antes de avançar para debates mais abertos. Um gap grande sugere que votações “avulsas” (ex.: requerimentos administrativos) são mais sujeitas a rejeição ou controvérsia.

6.3 Consulta 3: Análise Semântica de Descrições

Pergunta: Certos termos (“Redação Final”, “Parecer”, “Substitutivo”, “Requerimento”, “Emenda”) nas descrições estão ligados a taxas de aprovação distintas?

Listing 6: Consulta Cypher 3

```

1 MATCH (o:Orgao)-[:REALIZA]->(v:Votacao)
2 WITH o, v,
3     [t IN ['Redacao Final', 'Parecer', 'Substitutivo',
4           'Requerimento', 'Emenda']]
5     WHERE v.descricao CONTAINS t] AS termos,
6     (v.aprovacao = 1) AS aprovado
7 UNWIND termos AS termo
8 WITH o.sigla AS orgao, termo, aprovado
9 RETURN orgao, termo,
10     count(*) AS ocorrencias,
11     sum(CASE WHEN aprovado THEN 1 ELSE 0 END) AS aprovadas,
12     round(100.0 * sum(CASE WHEN aprovado THEN 1 ELSE 0 END)
13           /count(*),2) AS taxaAprovacaoTermo
14 ORDER BY orgao, ocorrencias DESC

```

Análise: Palavras como “Redação Final” e “Parecer” tendem a representar fases conclusivas do processo legislativo, frequentemente após negociações e ajustes; altas taxas de aprovação nesses casos refletem maturidade e consenso já construído. “Emenda” ou “Substitutivo” podem carregar maior volatilidade: rejeições de emendas indicam disputas sobre mérito ou estratégia. Comparar órgãos mostra onde o texto técnico (Parecer) tem mais peso (ex.: comissões temáticas) e onde expressões processuais (Requerimento) mantêm alta aprovação.

6.4 Consulta 4: Volatilidade de Resultados Partidários

Pergunta: A atuação textual (menções) de partidos em votações apresenta padrões de estabilidade (resultados quase sempre aprovados) ou alta variância?

Listing 7: Consulta Cypher 4

```

1 MATCH (p:Partido)<-[:PERTENCE_A]-(d:Deputado)
2 MATCH (o:Orgao)-[:REALIZA]->(v:Votacao)
3 WHERE v.descricao CONTAINS d.nome
4 WITH p, v
5 WITH p.sigla AS partido,
6     collect(v.aprovacao) AS resultados,
7     avg(CASE WHEN v.aprovacao = 1 THEN 1.0 ELSE 0.0 END) AS taxa,
8     stDev(toFloat(CASE WHEN v.aprovacao IS NULL THEN 0
9                   ELSE v.aprovacao END)) AS desvio
10 RETURN partido,
11     size(resultados) AS qtdCitacoes,
12     round(100.0 * taxa,2) AS taxaAprovacaoPercent,
13     round(desvio,3) AS desvioPadraoAprovacao,
14     resultados
15 ORDER BY qtdCitacoes DESC

```

Análise: O desvio-padrão quantifica volatilidade dos resultados em votações onde o partido está textual/semanticamente presente. Partidos com baixa variância e alta taxa de aprovação sugerem foco em matérias de consenso ou eficiência em negociação. Alta variância pode revelar estratégia de presença em pautas polarizadas (aceitando risco), tentativa de moldar agenda ou atuação defensiva.

6.5 Consulta 5: Deputados Mais Influentes

Pergunta: Quais deputados aparecem com maior frequência nas descrições de votações e essa visibilidade está correlacionada a maior taxa de aprovação?

Listing 8: Consulta Cypher 5

```
1 MATCH (d:Deputado)
2 MATCH (o:Orgao)-[:REALIZA]->(v:Votacao)
3 WHERE v.descricao CONTAINS d.nome
4 WITH d, v
5 WITH d.nome AS deputado,
6      d.siglaPartido AS partido,
7      d.siglaUf AS uf,
8      count(DISTINCT v) AS citacoes,
9      avg(CASE WHEN v.aprovacao = 1 THEN 1.0 ELSE 0.0 END) AS taxa
10 RETURN deputado, partido, uf, citacoes,
11        round(100.0 * taxa, 2) AS taxaAprovacaoPercent
12 ORDER BY citacoes DESC
13 LIMIT 25
```

Análise: Deputados frequentemente citados podem ocupar funções de relatoria, liderança de bancada ou protagonismo em áreas temáticas. Uma alta taxa de aprovação nas votações onde aparecem sugere capacidade de construir acordos e navegar ritos procedimentais; já citações numerosas com taxa reduzida podem refletir exposição em pautas contestadas ou papel de opositor. O ranking auxilia na identificação de “influenciadores processuais”.

7 Conclusão

7.1 Aprendizados

Este trabalho proporcionou diversos aprendizados importantes sobre modelagem e análise de dados em grafos:

- **Modelagem Natural:** O Neo4j permitiu uma representação natural e intuitiva das relações entre entidades legislativas, capturando a complexidade das interações políticas de forma mais expressiva que modelos relacionais tradicionais.
- **Poder do Cypher:** A linguagem Cypher demonstrou-se muito expressiva para consultas complexas, permitindo traversals de múltiplos níveis, agregações sofisticadas e análises estatísticas (como desvio-padrão) de forma declarativa.
- **Análise Textual:** A capacidade de combinar análise estrutural (grafos) com busca textual (CONTAINS) permitiu inferir relacionamentos implícitos (relatoria, autoria) a partir de descrições não estruturadas.
- **Visualização de Relações:** O modelo de grafo facilitou a identificação de padrões de influência e redes de colaboração que seriam difíceis de detectar em modelos tabulares.

7.2 Desafios Encontrados

Durante o desenvolvimento do projeto, foram enfrentados os seguintes desafios:

- **Qualidade dos Dados:** A API retorna dados limitados (primeiras páginas), exigindo adaptação do escopo da análise. Alguns campos contêm valores nulos ou inconsistentes.
- **Relacionamentos Implícitos:** A ausência de dados explícitos de autoria e votação nominal limitou a análise, sendo necessário recorrer a heurísticas textuais (busca de nomes em descrições), o que pode gerar falsos positivos.
- **Desempenho:** Consultas com CONTAINS em grandes volumes de texto podem ser lentas. A criação de índices full-text seria necessária para escalabilidade.
- **Normalização de Nomes:** Variações de grafia, abreviações e homônimos nos nomes de deputados podem afetar a precisão das análises textuais.

7.3 Repositório do Projeto

O código-fonte completo, datasets e documentação estão disponíveis no repositório GitHub:

<https://github.com/guiathayde/neo4j-camara-deputados>

Referências

- Câmara dos Deputados. *API de Dados Abertos*. Disponível em: <https://dadosabertos.camara.leg.br/>. Acesso em: 16 out. 2025.
- Neo4j. *Cypher Manual*. Disponível em: <https://neo4j.com/docs/cypher-manual/>. Acesso em: 16 out. 2025.
- Robinson, I., Webber, J., & Eifrem, E. (2015). *Graph Databases: New Opportunities for Connected Data*. O'Reilly Media.