



Análise de Dados de Redes Sociais com Grafos

Para projetar um protótipo funcional de análise de dados de redes sociais baseado em grafos, vou escolher o **Twitter (X)** como plataforma de exemplo. Ele possui interações claras que se traduzem perfeitamente em um modelo de grafo.

O banco de dados de grafos (Neo4j, por exemplo) será usado para modelar, armazenar e consultar essas interações.

1. Modelagem do Grafo Base (Twitter/X)

O modelo fundamental do grafo deve representar as principais entidades (Nós) e as interações (Arestas) na plataforma.

Nós (Labels e Propriedades)

Label (Entidade)	Propriedades Chave	Descrição
(:User)	userId, username, location	O usuário da plataforma.
(:Tweet)	tweetId, text, timestamp, sentiment	A publicação de conteúdo.
(:Topic)	name	Tópicos ou hashtags (#).

Arestas (Relacionamentos) e Propriedades

Tipo de Aresta	Conexão (De → Para)	Propriedades Chave	Descrição
[FOLLOWS]	(:User) \$to\$ (:User)	since (data do follow)	Define o relacionamento social.
[POSTS]	(:User) \$to\$ (:Tweet)	(Nenhuma)	Cria a autoria do conteúdo.
[MENTIONS]	(:Tweet) \$to\$ (:User)	(Nenhuma)	O tweet faz referência a outro usuário.
[REPLIES_TO]	(:Tweet) \$to\$ (:Tweet)	(Nenhuma)	Um tweet é uma resposta a outro (conversa).
[RETWEETS]	(:User) \$to\$ (:Tweet)	timestamp	Re-publicação de conteúdo (viralidade).
[LIKES]	(:User) \$to\$ (:Tweet)	timestamp	Engajamento positivo no conteúdo.
[HAS_TOPIC]	(:Tweet) \$to\$ (:Topic)	(Nenhuma)	O tweet utiliza uma hashtag ou tópico.

2. Base para Conexões no Grafo (Cypher)

Para construir a base, precisamos de comandos Cypher para criar os nós e as arestas de exemplo que representam a rede.

2.1. Criação de Usuários e Conteúdo Base

Vamos criar três usuários e um tweet inicial:

Cypher

// 1. Criar Usuários

```
CREATE (uA:User {userId: 'A1', username: 'analistaG', location: 'SP'})
```

```
CREATE (uB:User {userId: 'B2', username: 'dataNinja', location: 'RJ'})
```

```
CREATE (uC:User {userId: 'C3', username: 'trendHunter', location: 'MG'})
```

// 2. Criar Tópicos (Hashtags)

```
MERGE (tG:Topic {name: 'Grafo'})
```

```
MERGE (tC:Topic {name: 'Cypher'})
```

// 3. Usuário A posta o Tweet #1

```
CREATE (tw1:Tweet {tweetId: 'T1', text: 'Grafos são o futuro da análise de dados!', timestamp: datetime()})
```

```
CREATE (uA)-[:POSTS]->(tw1)
```

```
MERGE (tw1)-[:HAS_TOPIC]->(tG)
```

2.2. Criação de Interações (Arestas de Conexão)

Agora, vamos adicionar interações complexas entre os usuários e o tweet, formando as conexões que serão analisadas:

Cypher

```
MATCH (uA:User {userId: 'A1'}), (uB:User {userId: 'B2'}), (uC:User {userId: 'C3'}), (tw1:Tweet {tweetId: 'T1'})
```

// 1. Conexões Sociais (FOLLOWS)

```
MERGE (uA)-[:FOLLOWS {since: date('2024-01-15')}]>(uB)
```

```
MERGE (uB)-[:FOLLOWS {since: date('2024-02-20')}]>(uA) // Conexão bidirecional
```

```
MERGE (uC)-[:FOLLOWS {since: date('2024-03-01')}]>(uA)
```

```
// 2. Engajamento (LIKES e RETWEETS)
```

```
MERGE (uB)-[:LIKES {timestamp: datetime()}]->(tw1)
```

```
MERGE (uC)-[:RETWEETS {timestamp: datetime()}]->(tw1) // Sinal de viralidade
```

```
// 3. Conversa (Resposta)
```

```
CREATE (tw2:Tweet {tweetId: 'T2', text: 'Concordo, e o Cypher torna tudo mais fácil! #Cypher',  
timestamp: datetime()})
```

```
MERGE (uB)-[:POSTS]->(tw2)
```

```
MERGE (tw2)-[:REPLIES_TO]->(tw1)
```

```
MERGE (tw2)-[:HAS_TOPIC]->(tC)
```

```
RETURN 'Base de Conexões Criada!'
```

3. Exemplo de Análise com Consulta Cypher

Com a base criada, a startup pode responder a perguntas complexas, como:

Pergunta: Quais são os usuários mais influentes em um tópico específico (#Grafo), medidos pela popularidade de seu conteúdo e conexões?

Cypher

```
MATCH (topic:Topic {name: 'Grafo'})<-[:HAS_TOPIC]-(tw:Tweet)<-[:POSTS]-(author:User)
```

```
// Conta o engajamento direto (LIKES + RETWEETS) no tweet sobre o tópico
```

```
OPTIONAL MATCH (engager)-[:LIKES|RETWEETS]->(tw)
```

```
WITH author, count(DISTINCT engager) AS direct_engagement
```

```
// Conta a influência da rede (usuários que seguem o autor)
```

```
MATCH (follower:User)-[:FOLLOWS]->(author)
```

```
WHERE follower.location = 'SP' // Filtragem por comunidade/localização
```

```
WITH author, direct_engagement, count(follower) AS local_followers
```

```
// Classifica o usuário pelo engajamento e número de seguidores locais

RETURN author.username, direct_engagement, local_followers

ORDER BY direct_engagement DESC, local_followers DESC

LIMIT 5
```

Essa consulta demonstra como o Cypher pode combinar a **popularidade do conteúdo** (`direct_engagement`) com a **força da rede social** (`local_followers`) para determinar a influência.

Vamos criar uma consulta Cypher para identificar comunidades de interesse.

O conceito-chave para encontrar comunidades é o **Triângulo Fechado** e o **Conteúdo Comum**. A comunidade é formada por usuários que se **seguem mutuamente** e demonstram **alto engajamento** (curtidas ou retweets) nos mesmos tópicos.



Consulta Cypher: Identificação de Comunidades de Interesse

Esta consulta foca em encontrar usuários que interagem com o mesmo tópico e que têm laços sociais fortes (seguimento mútuo).

1. Definir o Tópico de Interesse e Laços Fortes (Triângulo Fechado)

Vamos focar no tópico **Grafo** e encontrar pares de usuários que se seguem:

Cypher

```
MATCH (u1:User)-[:FOLLOWS]->(u2:User)-[:FOLLOWS]->(u1)
```

```
// Garantir que a conexão é bidirecional (mútua)
```

```
WHERE u1.userId < u2.userId // Previne duplicação do par (u1, u2) e (u2, u1)
```

```
// 2. Encontrar Conteúdo Comum relacionado ao Tópico
```

```
MATCH (u1)-[:LIKES|RETWEETS]->(tw:Tweet)-[:HAS_TOPIC]->(t:Topic {name: 'Grafo'})
```

```
MATCH (u2)-[:LIKES|RETWEETS]->(tw)
```

```
// Amigos mútuos (u1 e u2) engajando no mesmo Tweet (tw) sobre o mesmo Tópico (t)
```

3. Agregação e Pontuação da Comunidade

Agrupamos pelo par de usuários e contamos o número de interações (tweets) em comum no tópico, que será nosso **Score de Coesão da Comunidade**.

Cypher

```
MATCH (u1:User)-[:FOLLOWS]->(u2:User)-[:FOLLOWS]->(u1)
```

```
WHERE u1.userId < u2.userId
```

```
// Arestas de Engajamento: LIKES ou RETWEETS
```

```
MATCH (u1)-[:LIKES|RETWEETS]->(tw:Tweet)-[:HAS_TOPIC]->(t:Topic {name: 'Grafo'})
```

```
MATCH (u2)-[:LIKES|RETWEETS]->(tw)
```

```
WITH u1.username AS UserA,
```

```
u2.username AS UserB,
```

```
count(tw) AS CohesionScore,
```

```
collect(tw.tweetId) AS SharedTweets
```

```
// Filtra pares com alta coesão
```

```
WHERE CohesionScore >= 2
```

```
RETURN 'Comunidade: #Grafo' AS Community,
```

```
UserA,
```

```
UserB,
```

```
CohesionScore,
```

```
SharedTweets
```

```
ORDER BY CohesionScore DESC
```

```
LIMIT 10
```



Análise da Consulta

Esta consulta define uma comunidade (o par `UserA` e `UserB`) com base em dois critérios fortes:

1. **Laços Sociais Fortes:** Eles se **seguem mutuamente** (`[[:FOLLOWS]->(u2)-[:FOLLOWS]->(u1))`).
2. **Interesse Comum:** Eles **engajaram** (curtiram ou retuitaram) o **mesmo conteúdo** (`tw:Tweet`) relacionado ao tópico de interesse (`:Topic {name: 'Grafo'}`).

O resultado são os pares de usuários mais conectados e engajados em um determinado assunto, o que é um *insight* valioso para a startup sobre a formação de comunidades na plataforma.