ANO
2024

UNINTER

CADERNO DE RESPOSTAS DA ATIVIDADE PRÁTICA DE:

NoSQL

ALUNO: JOSELINA DE ALMEIDA MARRA, RU 4688889

Caderno de Resposta Elaborado por:
Prof. MSc. Guilherme Ditzel Patriota

Atividade Prática – NEO4J

Questão 01 – IMPORTAÇÃO DOS ARQUIVOS JSON E CRIAÇÃO DE NÓS E ARESTAS COM BASE NOS DADOS DOS ARQUIVOS (Usar quantas páginas forem necessárias)

O QUE FAZER: Após configurar seu banco de dados em branco (novo DBMS em versão 4.*) e colocar no mínimo 3 e máximo 10 arquivos sequenciais do trabalho (dentre os 500 arquivos JSON disponibilizados) na pasta Import, crie um comando na linguagem do banco de dados Neo4j (Cypher), usando a biblioteca de importação de dados APOC, que leia os arquivos JSON desta pasta e crie os nós e relacionamentos com base nos dados neles. Faça a separação dos nós de mensagem em Tweet (mensagens originais), Retweet (mensagens repostadas), Quoted (mensagens que citam outras mensagens), Replied_to (mensagens de resposta à outras mensagens) com uso do campo data[x].ref_tweet.type (CUIDADO! Este campo só aparece no JSON de mensagens não originais e o uso de UNWIND para acessá-la pode impedir a criação dos nós de mensagens originais). Apenas com esta separação será possível resolver a questão 02 do trabalho.

Observação Importante: Seu banco de dados precisará conter todas as informações para resolver as questões 02 e 03. Leia elas antes, para entender o que você deseja fazer em cada uma e quais os nós, relacionamentos e atributos você irá importar dos arquivos JSON para conseguir resolver todo o trabalho sem a necessidade de recriar todo o seu banco de dados apenas para uma questão.

I. **Apresentação dos comandos (apenas query Cypher) usados (não esquecer do identificador pessoal/seu RU como parte do seu código, como um nome de atributo dos nós ou dado de um atributo de nós):**

```
1  CALL apoc.load.json("file:///tweets_coletados_154.json")
   YIELD value UNWIND value.data AS tweet MERGE (:Tweet {id:
   tweet.id, ru: '4688889'});
2
3  CALL apoc.load.json("file:///tweets_coletados_155.json")
   YIELD value UNWIND value.data AS tweet MERGE (:Tweet {id:
   tweet.id, ru: '4688889'});
4
5  CALL apoc.load.json("file:///tweets_coletados_156.json")
   YIELD value UNWIND value.data AS tweet MERGE (:Tweet {id:
   tweet.id, ru: '4688889'});
6
```

Figura 1: Importação e criação dos nós Tweet a partir dos arquivos 154, 155 e 156 usando APOC. O comando CALL apoc.load.json foi utilizado para carregar os dados de cada arquivo JSON, seguido de UNWIND para acessar a lista de dados e MERGE para criar ou atualizar os nós do tipo Tweet. O atributo ru: '4688889' garante que o RU esteja corretamente identificado em cada nó.

neo4j$

```
1  MATCH (t:Tweet) WHERE t.tipo_ref IS NULL SET t:Original;
2
3  MATCH (t:Tweet) WHERE t.tipo_ref = 'retweeted' REMOVE
   t:Tweet SET t:Retweet;
4
5  MATCH (t:Tweet) WHERE t.tipo_ref = 'quoted' REMOVE t:Tweet
   SET t:Quoted;
6
7  MATCH (t:Tweet) WHERE t.tipo_ref = 'replied_to' REMOVE
   t:Tweet SET t:RepliedTo;
8
```

Figura 2: Comando Cypher para identificar e rotular nós de mensagem: seleciona os nós Tweet usando MATCH e, conforme o tipo (NULL, retweeted, quoted, replied_to), atribui ou remove rótulos apropriados (Original, Retweet, Quoted, RepliedTo) usando SET e REMOVE. Isso organiza os dados.

```
1  MATCH (t:Original), (o:Tweet {id: t.id})
2  MERGE (t)-[:ORIGINAL]→(o);
```

Figura 3: Relacionamento para mensagens originais: conecta o nó Tweet identificado como original ao nó Original usando o relacionamento [:ORIGINAL] via MERGE. Esse passo garante que as mensagens originais estejam claramente ligadas no grafo.

neo4j$

```
1  MATCH (t:Retweet), (o:Tweet {id: t.id})
2  MERGE (t)-[:RETWEETED]→(o);
```

Figura 4: Relacionamento para mensagens Retweet: conecta os nós Tweet classificados como retweet ao relacionamento [:RETWEETED] usando MERGE. Isso facilita a análise de padrões e conexões específicas de retweets no banco de dados.

neo4j$

```
1  MATCH (t:Quoted), (o:Tweet {id: t.id})
2  MERGE (t)-[:QUOTED]→(o);
3
```

Figura 5: Relacionamento para mensagens Quoted: conecta os nós Tweet classificados como quoted (citação) ao relacionamento [:QUOTED] com MERGE. Assim, conseguimos mapear quais tweets foram citados em outras postagens.

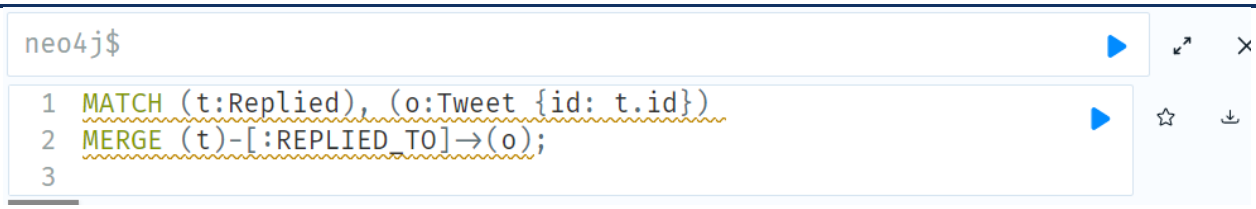

Figura 6: Relacionamento para mensagens Replied: conecta os nós Tweet classificados como replied (resposta) ao relacionamento [:REPLIED_TO] usando MERGE. Esse relacionamento é essencial para entender as interações diretas entre os usuários.

II. Apresentação dos prints do resultado (não esquecer do identificador, seu RU). Estes prints devem ser de cada tela do Neo4j Browser após a execução bem-sucedida de cada comando (não mostrar nenhum grafo nesta parte, apenas as telas de execução dos comandos da parte I. O uso de RETURN zerará a nota desta parte):

Figura 7: Importação e criação de nós Tweet a partir dos arquivos 154, 155 e 156 usando APOC, com identificação do meu RU 4688889.

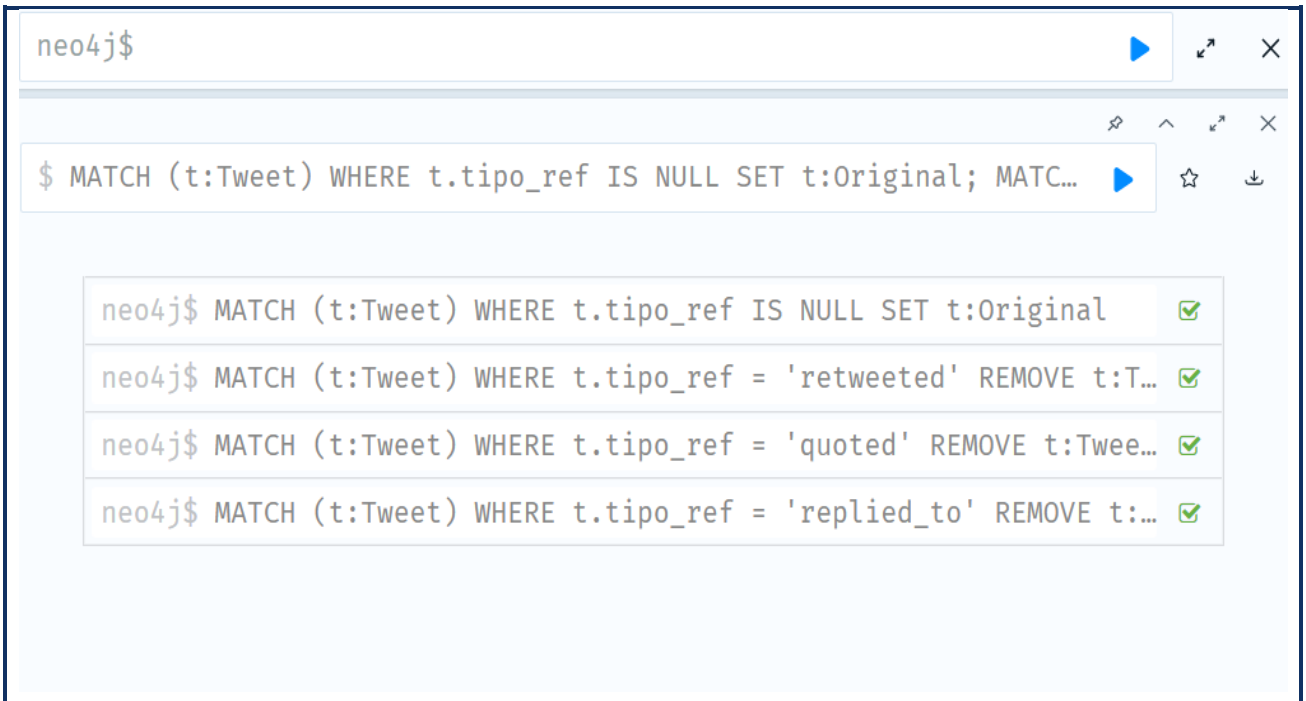

Figura 8: Comando Cypher para indentificar e rotular nós de mensagem: Tweet, Retweet, Quoted e Replied

Figura 9: Relacionamento para mensagens originais

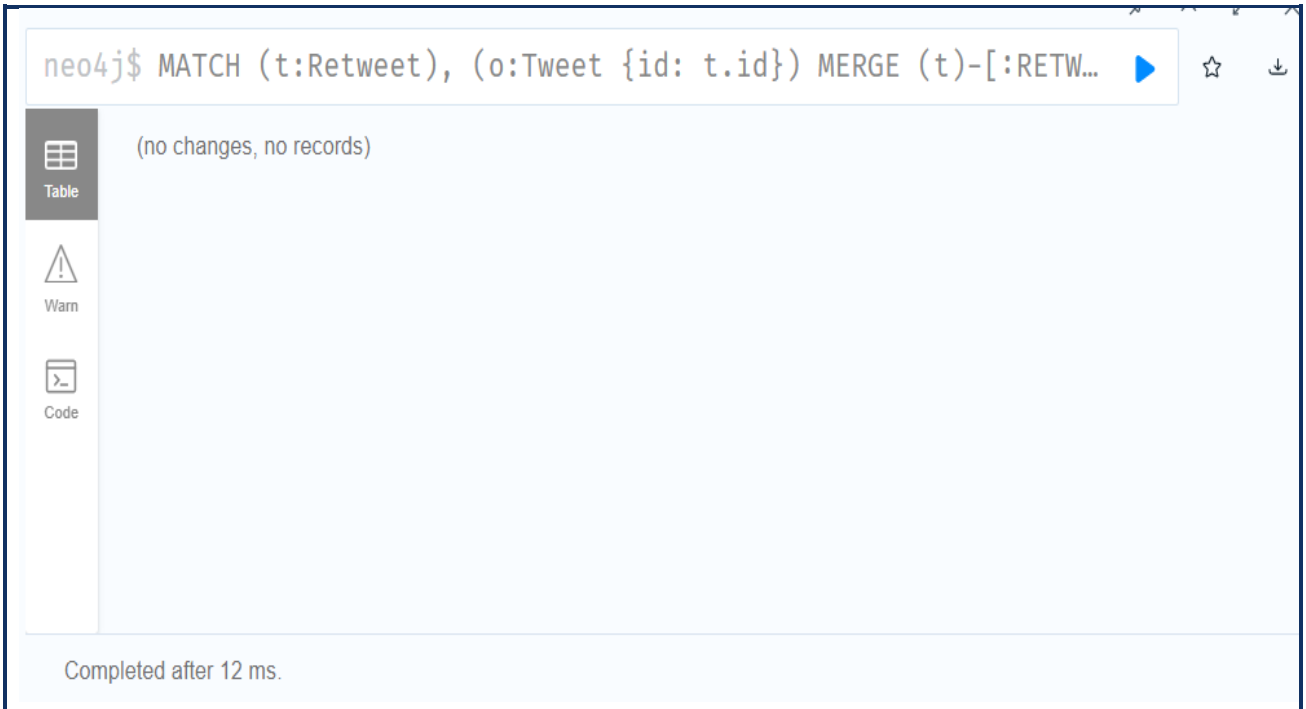

Figura 10: Relacionamento para mensagens Retweet

Figura 11: Relacionamento para mensagens Quoted

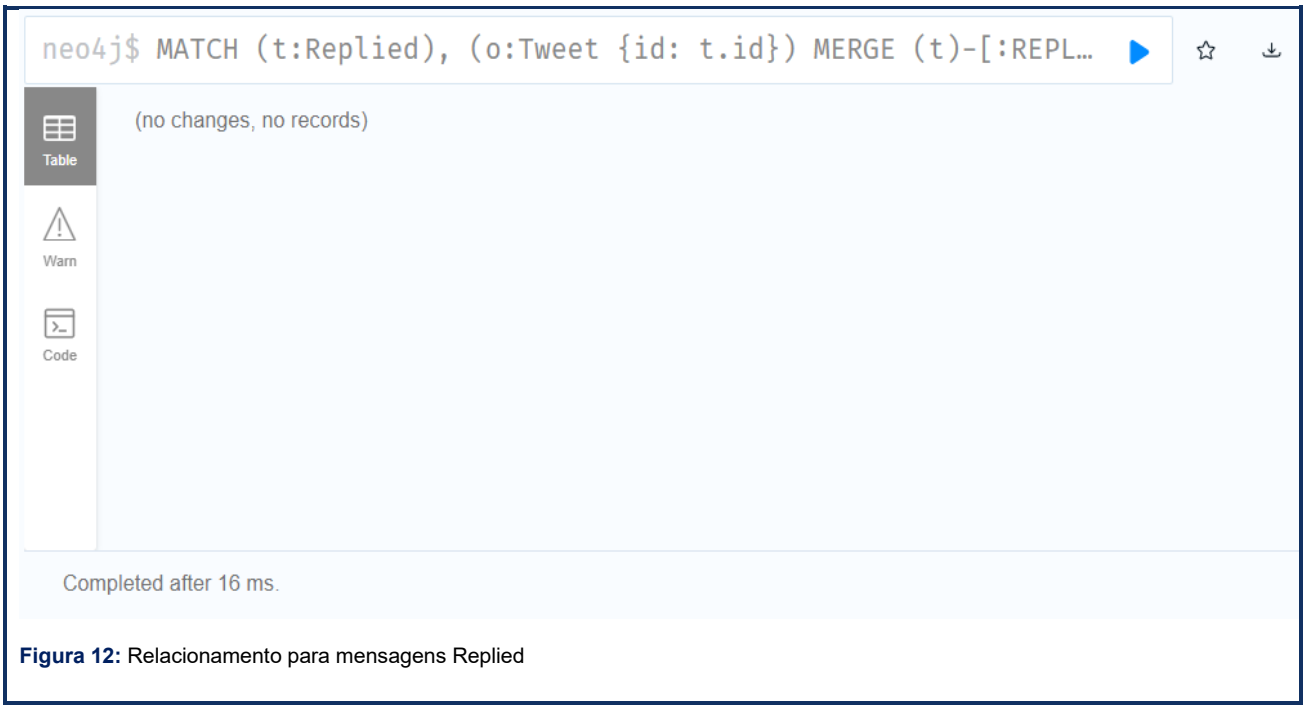

Figura 12: Relacionamento para mensagens Replied

Atividade Prática – NEO4J

Questão 02 – DESCOBERTA DA HASHTAG PRINCIPAL (Usar quantas páginas forem necessárias)

ENUNCIADO: Você deve criar e executar um comando Cypher em seu banco de dados para descobrir qual hashtag está presente em todas as mensagens originais (excluindo mensagens de retweet, citação e resposta). Este comando não deve fazer uso da biblioteca APOC. Caso seu comando tente retornar mais de 300 nós, a configuração padrão do Neo4j Browser impedirá a exibição de mais de 300 nós. Seu comando não pode ser MATCH (n) RETURN n;.

I. **Apresentação do comando (apenas query Cypher) usado (não esquecer do identificador pessoal no comando, seu RU).**

```
neo4j$
```

```
1 MATCH (t:Original)
2 UNWIND t.hashtags AS hashtag
3 RETURN hashtag, count(*) AS freq
4 ORDER BY freq DESC
5 LIMIT 1;
```

Figura 13: Figura 14: Comando para exibir a hashtag mais usada. Neste comando, usamos MATCH para localizar os nós do tipo Original, UNWIND para "abrir" as listas de hashtags, e COUNT(*) para contar quantas vezes cada hashtag aparece. Em seguida, usamos ORDER BY para ordenar pela frequência e LIMIT 1 para retornar apenas a mais frequente.

```
neo4j$
```

```
1 MATCH (h:Hashtag {name: 'depheliolopes'})←
  [:USADA_EM]-(o:Original)
2 RETURN h, o
3 LIMIT 10;
```

Figura 14: Comando para mostrar o grafo com os nós do tipo Original que estão conectados à hashtag principal 'depheliolopes'. Usamos MATCH para buscar a hashtag específica, relacionando-a a mensagens originais, e LIMIT para reduzir o número de resultados exibidos no grafo.

II. **Apresentação do grafo gerado contendo apenas 1 nó de hashtag ao centro (a sua resposta) e ao menos mais 10 nós de mensagens relacionadas a este nó de hashtag. O print deve conter o grafo sem zoom e a legenda de tipos de nós e cores geradas pelo neo4j, incluindo a informação da quantidade de nós de Hashtag mostrados (não esquecer do identificador pessoal, seu RU):**

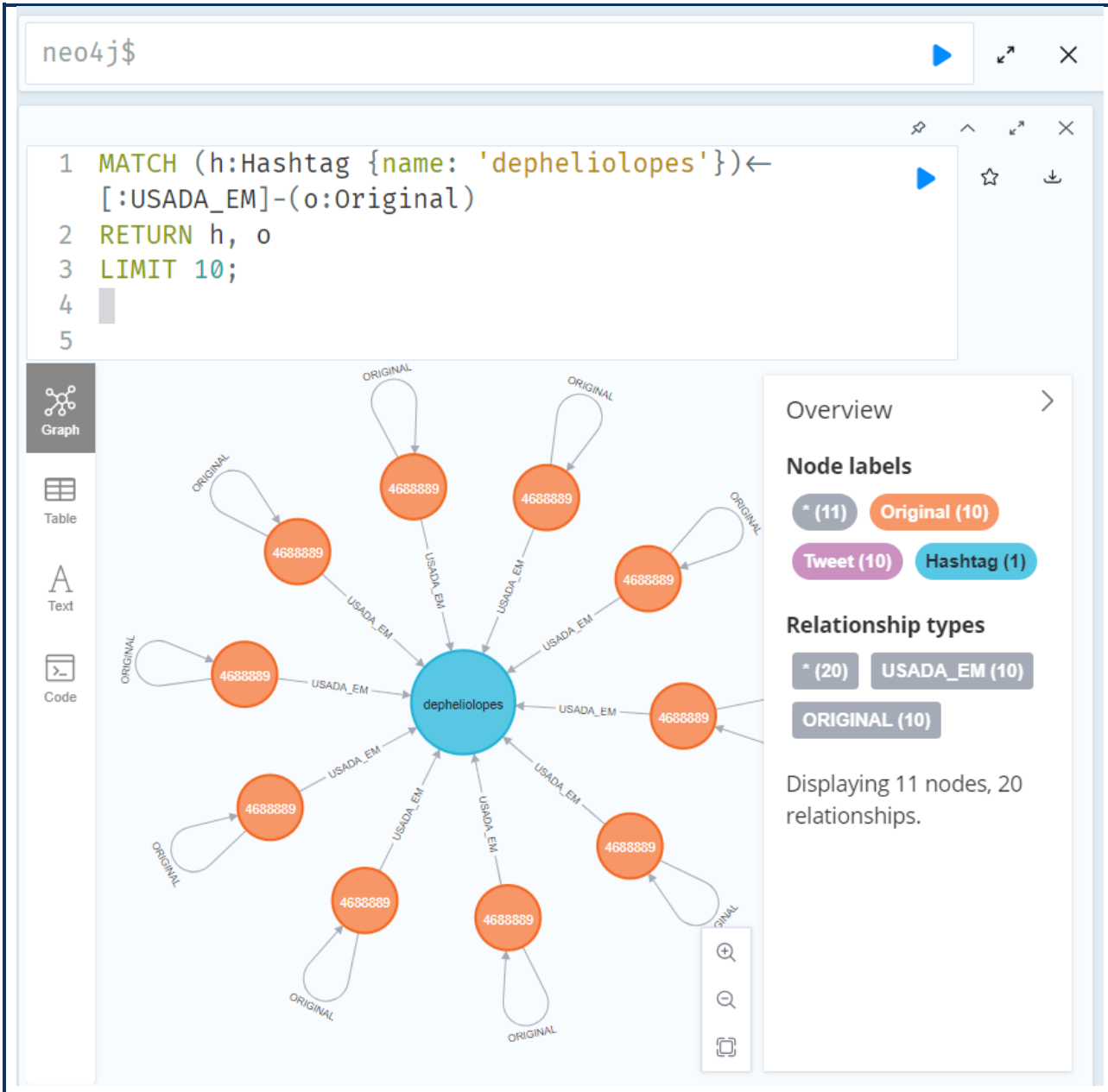

Figura 15: Grafo gerado no Neo4j mostrando 10 nós do tipo Original conectados à hashtag mais frequente "depheliolopes". O nó central representa a hashtag principal, enquanto os nós ao redor representam as mensagens originais que utilizaram essa hashtag. As arestas indicam a relação USADA_EM, mostrando como a hashtag conecta diferentes mensagens. Todos os nós estão identificados com o RU 4688889, garantindo rastreabilidade no conjunto analisado.

III. Responda à pergunta: Qual foi a hashtag usada como filtro para coleta dos dados analisados? (Esta hashtag só estará presente em nós do tipo Tweet e não em Retweet. Sua resposta deve conter apenas 1 palavra com o texto da hashtag) ?

Resposta: depheliolopes

Atividade Prática – NEO4J

Questão 03 – ANÁLISE DOS DADOS SEGUNDO VIÉS A SUA ESCOLHA (Usar quantas páginas forem necessárias)

ENUNCIADO: Usando o mesmo banco de dados já criado na questão 01 e usado na questão 02, busque alguma informação que você julgue relevante nos dados (seu comando não pode ser MATCH (n) RETURN n; nem pode conter a biblioteca APOC). Sua tarefa aqui é analisar os dados do banco e definir qual informação você gostaria de obter e que tenha potencial de gerar um grafo com 10 ou mais nós interligados entre si. Sua análise deve responder à uma pergunta clara, como por exemplo:

- Qual o dispositivo mais usado para tuitar? (depende de seu banco de dados já possuir os nós de equipamentos usados, criados na questão 01).

- Qual a Hashtag que menos foi usada?

- Qual o usuário mais movimentou a rede? (depende de você ter criado nós de usuário na questão 01)

- Quais os usuários mais citados? (depende dos seus nós de mensagem possuírem esta informação ou de relacionamentos terem sido criados para este fim).

I. Apresentação dos comandos (apenas queries Cypher) usados por você para realizar sua análise (não esquecer do identificador pessoal, seu RU) :

```
neo4j$
```

```
1  MATCH (t1:Original), (t2:Original)
2  WHERE t1 <> t2 AND
3    ANY(tag IN t1.hashtags WHERE tag IN
   t2.hashtags)
4  WITH t1, t2
5  LIMIT 20
6  MERGE (t1)-[r:COMPARTILHA_HASHTAG {ru:
   '4688889'}]→(t2)
7  RETURN t1, t2, r;
8
```

Figura 16: Comando Cypher para identificar e conectar nós de mensagens originais que compartilham hashtags em comum. O comando usa MATCH para buscar pares de nós Original, verifica com WHERE se os nós são diferentes (t1 <> t2) e se compartilham ao menos uma hashtag em comum usando ANY. O LIMIT 20 restringe o número de pares retornados, e o MERGE cria (ou confirma) o relacionamento [:COMPARTILHA_HASHTAG] entre os nós, adicionando também o RU para identificação .

II. Apresentação do print do resultado, podendo ser uma tabela ou um grafo. O print deve conter o resultado e o comando executado. (não esquecer do identificador, seu RU) :

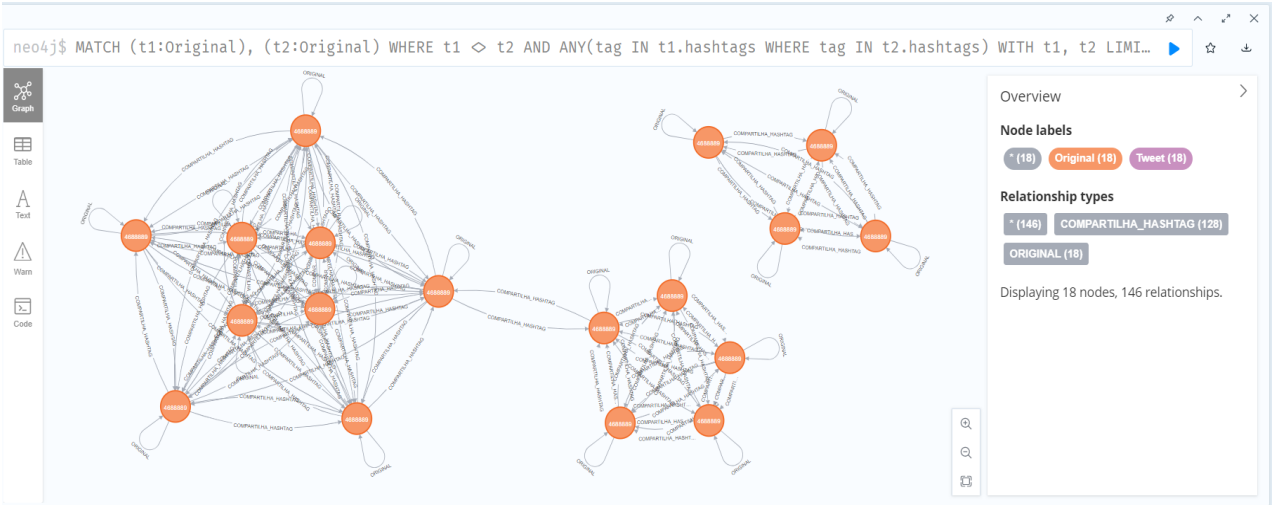

Figura 17: Grafo mostrando 18 nós do tipo Original conectados por compartilhamento de hashtag. Cada linha (aresta) representa que os tweets identificados compartilham pelo menos uma hashtag em comum, formando grupos densamente conectados. Esse resultado destaca os principais agrupamentos temáticos no conjunto de mensagens originais analisadas, permitindo identificar áreas de maior interação sem depender de retweets ou replies..

III. Explique qual foi a análise realizada, incluindo sua linha de raciocínio para criação da query Cypher e qual era sua expectativa de resultado antes da análise, comparando-a com o resultado realmente obtido após execução do comando.

Resposta: A análise buscou entender como os tweets originais se conectam quando compartilham hashtags comuns. Esperava-se identificar agrupamentos entre mensagens que tratam dos mesmos temas. O resultado confirmou que há forte interconexão entre certos tweets, formando um cluster denso no grafo. Isso monstra que mesmo sem retweets ou replies, há coesão temática nas postagens analisadas.