ANO

2025

UNINTER

CADERNO DE RESPOSTAS DA ATIVIDADE PRÁTICA DE:

NoSQL

ALUNO: ALISSON DE SOUZA RODRIGUES - 4381452

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):

```
MERGE (a:Aluno {nome: "Alisson de Souza Rodrigues"}) -[:RU]→ (r:numeroRU {RU: 4381452});
// importando os arquivos json com os tweets
CALL apoc.load.directory('*.json') YIELD value
WITH value AS arquivos
ORDER BY arquivos DESC
CALL apoc.load.json(arquivos) YIELD value
UNWIND value.data AS tweet

// criando os nós de tweet e adicionando atributos
MERGE (t:Tweet {tweet_id: tweet.id})
ON CREATE SET t += {
    texto: tweet.text,
    criado_em: tweet.created_at,
    lingua: tweet.lang,
    curtidas: tweet.public_metrics.like_count,
    retweets: tweet.public_metrics.retweet_count,
    respostas: tweet.public_metrics.reply_count,
    citacoes: tweet.public_metrics.quote_count,
    autor: tweet.author_id,
    geolocalizacao: tweet.geo.place_id
}

// criando os nós de hashtag e o relacionamento com os tweets
FOREACH (hashtag IN tweet.entities.hashtags |
    MERGE (h:Hashtag {hashtag: apoc.text.replace(apoc.text.clean(hashtag.tag), '[^a-zA-Z0-9]', '')})
    MERGE (h)←[:POSSUI]-(t)
)

```

Figura 1: Criação dos nós de Aluno e RU e do relacionamento entre eles, Importação dos arquivos json de tweets, criação dos nós de tweet e definição dos atributos do nós com base nos metadados do tweet contidos em data, criação dos nós das hashtags utilizadas nos tweets padronizando a formatação das hashtags de mesmo assunto e criando os relacionamento entre tweets e as hashtags utilizadas por eles.

```
// identificando o tipo de tweet, apagando e criando a etiqueta correta
FOREACH (ref_tweet IN tweet.referenced_tweets |
    SET t.tipo_ref = coalesce(t.tipo_ref, []) + [ref_tweet.type],
        t.id_ref = coalesce(t.id_ref, []) + [ref_tweet.id]
);

MATCH (t) WHERE "retweeted" in t.tipo_ref
REMOVE t:Tweet SET t:Retweet;

MATCH (t) WHERE "replied_to" in t.tipo_ref
REMOVE t:Tweet SET t:Resposta;

MATCH (t) WHERE "quoted" in t.tipo_ref
REMOVE t:Tweet SET t:Citacao;

MERGE (a:Aluno {nome: "Alisson de Souza Rodrigues"}) -[:RU]→ (r:numeroRU {RU: 4381452});
```

Figura 2: Identificação do tipo de tweet (retweeted, replied_to e quoted) com base no valor de referenced_tweets.type, que só existe nos tweets que não são os originais, para diferenciar tweets originais dos outros tipos de tweet, apagando a etiqueta Tweet e substituindo pela etiqueta correspondente e tentativa de criar novamente outro nó e relacionamento Aluno RU que resultará em nenhuma mudança porque MERGE só cria nós e relacionamentos que não existem.

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):**

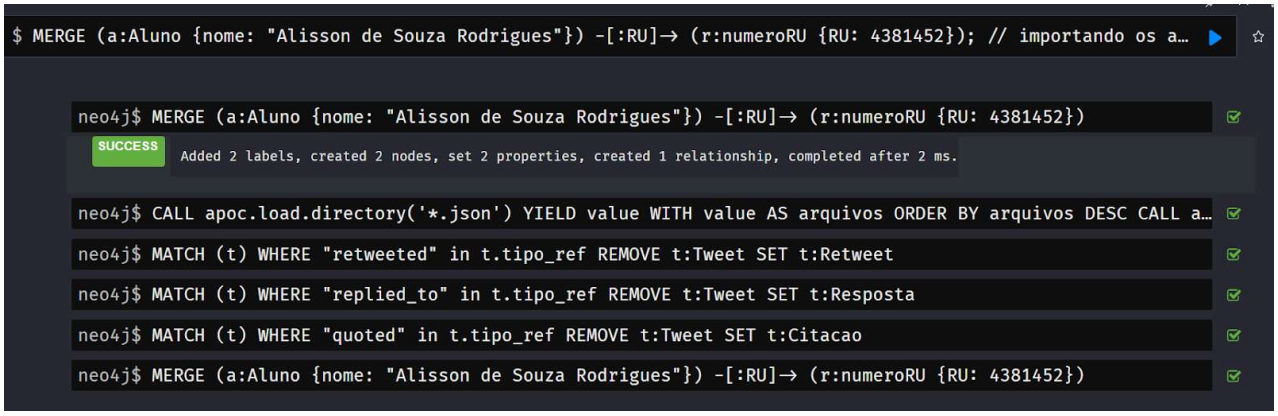


Figura 3: São criados os nós de Aluno e RU e o relacionamento entre eles.

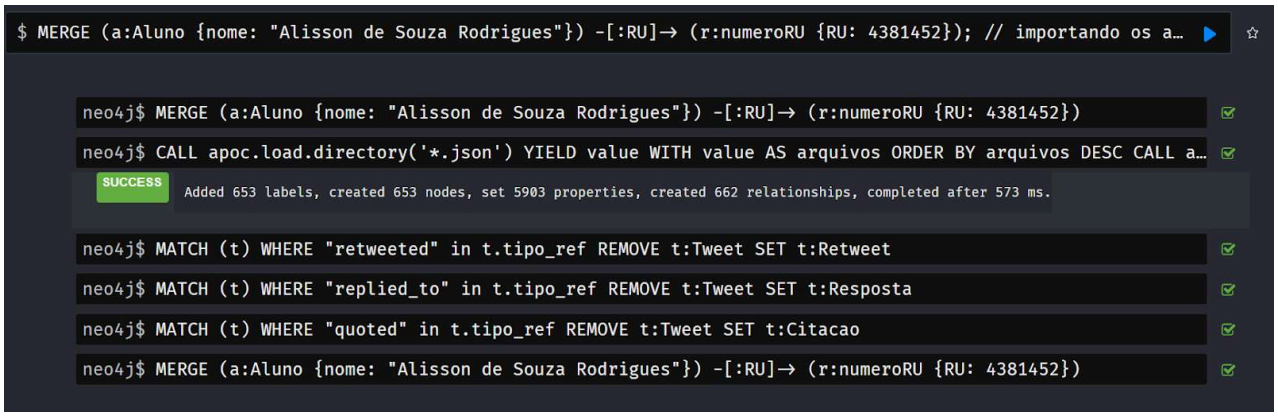
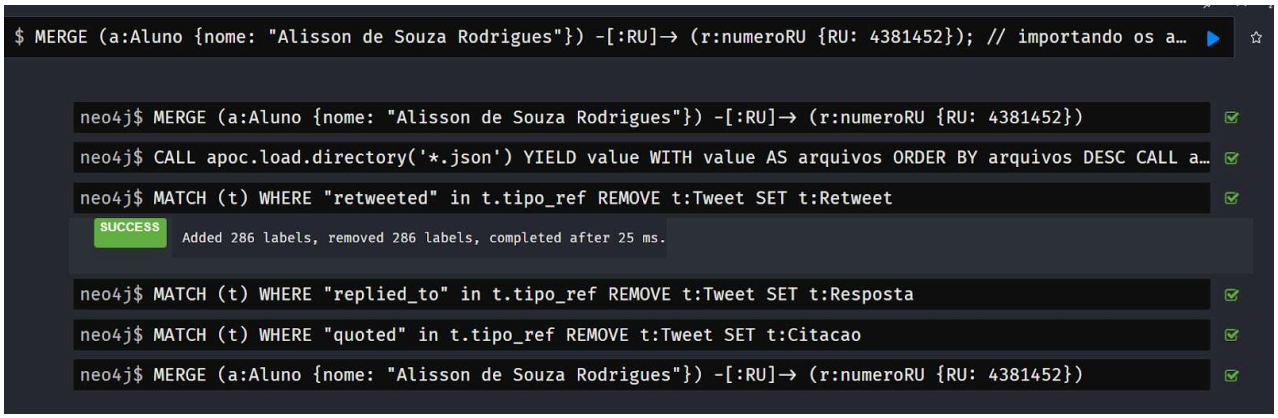
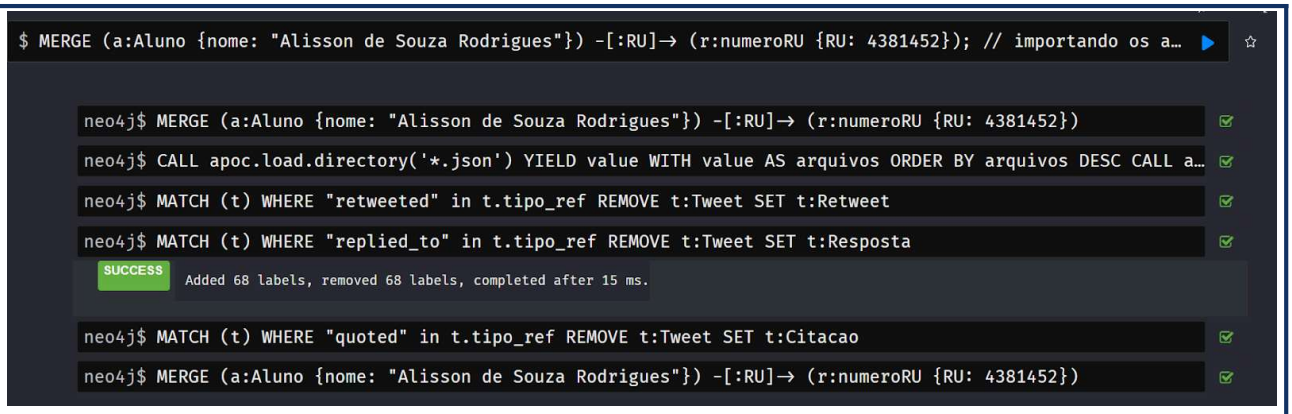
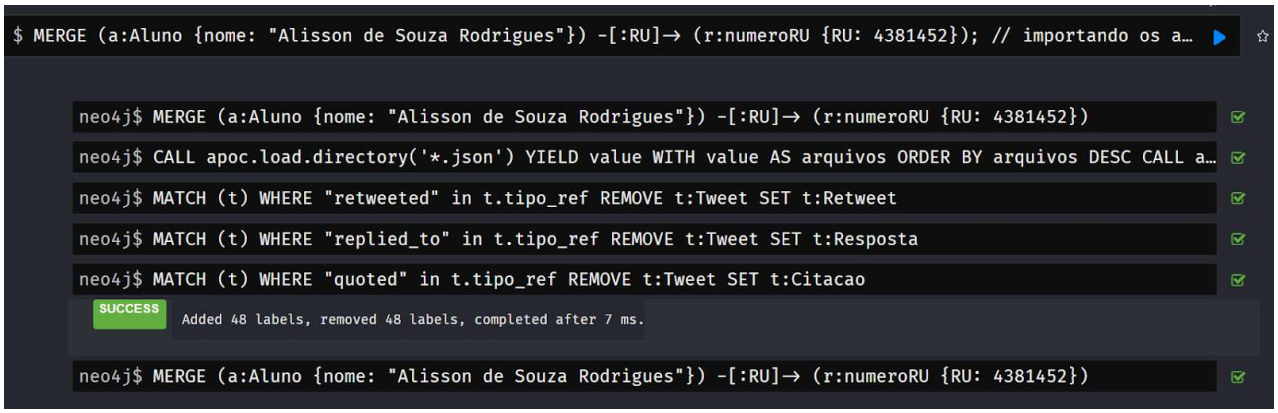


Figura 4: Os arquivos json com tweets são importados corretamente, são criados os nós de tweets com seus metadados e os nós de hashtags, são criados os relacionamentos entre os tweets e as hashtags, são adicionados os atributos de tipo de tweet aos nós de tweet.







Figuras 5, 6 e 7: São alteradas as etiquetas dos tweets não originais (retweeted, replied_to e quoted), apagando a etiqueta Tweet contida neles e substituindo pela etiqueta correta que identifique o tipo de tweet não original correspondente, diferenciando os tweets originais dos outros tipos de tweet.

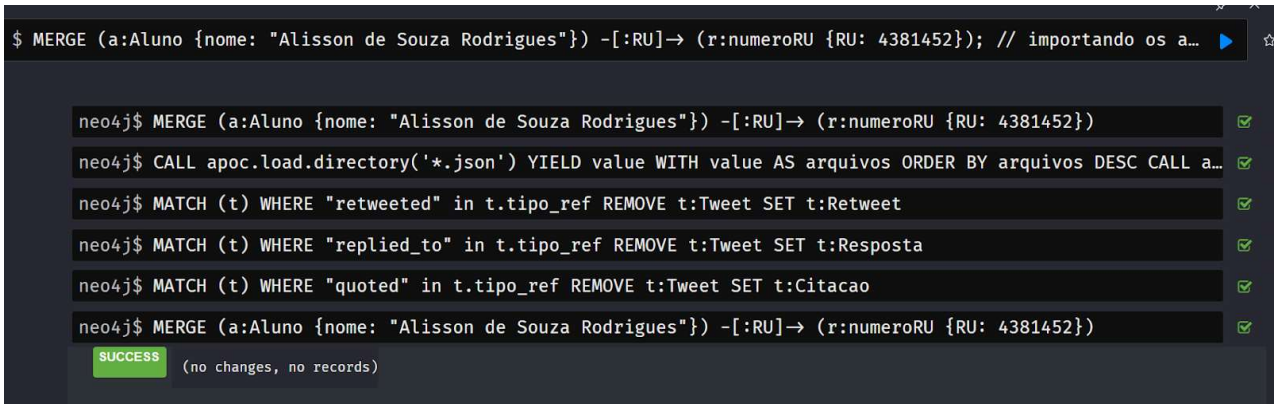


Figura 8: O comando MERGE tenta criar outros nós e relacionamento de identificação de Aluno e RU, mas nenhuma alteração é feita porque já existem esses nós e relacionamento no banco de dados.

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).**

```
1  // encontrando a hashtag mais utilizada
2  MATCH (h:Hashtag)<-[:POSSUI]-(t:Tweet)
3  WITH h, COUNT(t) AS quantidade
4  ORDER BY quantidade DESC
5  LIMIT 1
6
7  // limitando a quantidade de relacionamentos na exibição do grafo
8  MATCH (t:Tweet)-[r:POSSUI]->(h)
9  MATCH (a:Aluno) -[:RU]-> (ru4381452:numeroRU)
10 RETURN t, h, r, a, ru4381452
11 LIMIT 10
```

Figura 9: Nas linhas 2 ao 5 é realizada uma busca por todos os relacionamentos informando uma hashtag e todos os tweets que usam essa hashtag, é utilizado um contador para determinar a quantidade de tweets relacionados com cada hashtag e o resultado é organizado pela hashtag com a maior quantidade de tweets para a hashtag de menor quantidade, por fim, ao determinar LIMIT 1, somente a hashtag com mais uso, a principal, é retornada.
Nas linhas 8 ao 11 é realizada uma busca pelos relacionamentos entre a hashtag principal obtida nas linhas anteriores representada por h e todos os tweets que usam essa hashtag, também é realizada uma busca pelo relacionamento Aluno e RU e, por fim, é exibido a hashtag principal e os seus tweets, mas ao utilizar LIMIT 10, somente 10 tweets são exibidos. O relacionamento entre os nós de Aluno e RU é exibido também.

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):**

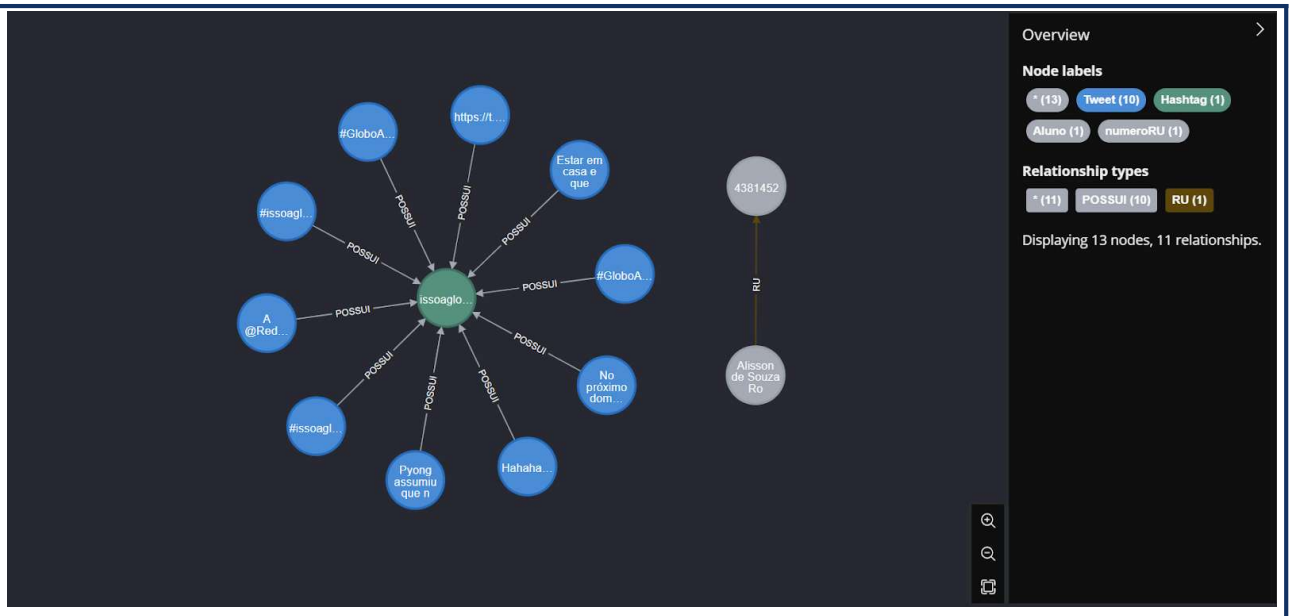


Figura 10: Apresentação da hashtag principal "issoaglobonaomostra" e seu relacionamento com 10 tweets que utilizam essa hashtag, além da apresentação dos nós de Aluno e RU e o relacionamento entre eles.

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 (apenas 1 palavra): issoaglobonaomostra

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):**

```
// encontrando as 3 hashtags mais usadas
MATCH (h:Hashtag)<-[:POSSUI]-(t:Tweet)
WITH h, COUNT(t) AS quantidade
ORDER BY quantidade DESC
LIMIT 3
WITH COLLECT(h) AS hashMaisUsadas

// buscando 5 tweets de cada hashtag
UNWIND hashMaisUsadas AS h
CALL {
  WITH h
  MATCH (t:Tweet)-[r:POSSUI]->(h)
  RETURN t, r
  LIMIT 5
}

MATCH (a:Aluno) -[:RU]-> (ru4381452:numeroRU)

RETURN DISTINCT t, h, r, a, ru4381452
```

Figura 11: Busca as 3 hashtags mais utilizadas e exibe 5 tweets para cada hashtag com o relacionamento indicando o seu uso, além de exibir os nós de identificação de Aluno e RU e o relacionamento entre eles.

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):

```
// encontrando as 3 hashtags mais usadas
MATCH (h:Hashtag)<-[:POSSUI]-(t:Tweet)
WITH h, COUNT(t) AS quantidade ORDER BY quantidade DESC LIMIT 3
WITH COLLECT(h) AS hashMaisUsadas

// buscando 5 tweets de cada hashtag
UNWIND hashMaisUsadas AS h
CALL {WITH h MATCH (t:Tweet)-[r:POSSUI]->(h) RETURN t, r LIMIT 5}
MATCH (a:Aluno) -[:RU]-> (ru4381452:numeroRU)
RETURN DISTINCT t, h, r, a, ru4381452
```

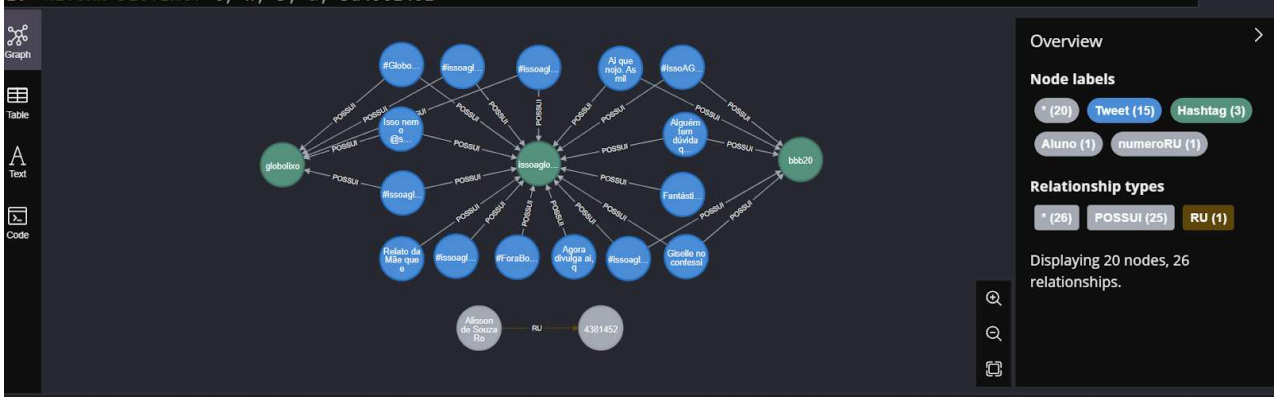


Figura 12: Apresentação do grafo com as 3 hashtags mais utilizadas e o seu relacionamento com pelo menos 5 tweets que utilizam a hashtag, além da exibição dos comandos utilizados para obter esse resultado. O código apresentado é o mesmo da figura 11, mas teve sua escrita de código comprimida para a apresentação junto ao grafo no mesmo print.

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: O objetivo da análise foi obter as 3 hashtags mais utilizadas pelos usuários do X (antigo Twitter) na base de dados escolhida para a realização da análise. Antes da execução do código, imaginei que o grafo gerado teria 3 nós de hashtag, as mais utilizadas, e 5 nós de tweet para cada hashtag, sendo que o relacionamento entre os tweets para cada hashtag seria único, isto é, cada tweet só teria um único relacionamento com uma das três hashtags mais utilizadas. Porém, o resultado obtido foi outro, pois para cada hashtag realmente foi apresentado pelo menos 5 relacionamentos com tweets, mas como os tweets exibidos apresentam mais de uma hashtag, alguns deles também apresentaram mais de uma hashtag entre as três mais utilizadas. O resultado obtido exibiu tweets com mais de um relacionamento entre 3 hashtags principais.

A query cypher utilizada é dividida em duas partes. Na primeira parte é realizada a busca por todos os relacionamentos entre as hashtags e os tweet que utilizam essa hashtag, depois é associado cada hashtag com a quantidade de tweets relacionados com elas, organiza-se os resultados por ordem da maior quantidade para a menor quantidade, o resultado é limitado para obter a três hashtags mais utilizadas e, por fim, é criado um lista para compor as três hashtags mais utilizadas.

Na segunda parte, para cada hashtag da lista obtida anteriormente, realizamos uma subconsulta para obter os relacionamentos entre a hashtag e os tweets que utilizam a hashtags, limitamos o resultado em 5 relacionamentos. Finalmente, buscamos o relacionamento que identifica o Aluno e seu RU e exibimos tudo em um grafo.