

TRABALHO PRÁTICO DE TÓPICOS AVANÇADOS DE BASE DE DADOS

2023 / 2024

1190384 André Teixeira

1231925 João Henriques

1190818 Luís Pinto

Docente: Rosa

Índice

1	<i>Introdução.....</i>	<i>1</i>
2	<i>Introdução Teórica – Neo4j.....</i>	<i>2</i>
3	<i>Construção da Base de Dados.....</i>	<i>3</i>
3.1	<i>Import do Ficheiro Cypher</i>	<i>3</i>
3.2	<i>Criação de Nós</i>	<i>7</i>
3.3	<i>Identificação do Problema</i>	<i>15</i>
4	<i>Queries</i>	<i>16</i>
5	<i>Solução do Problema.....</i>	<i>18</i>
	<i>Conclusão</i>	<i>23</i>
	<i>Referências</i>	<i>24</i>

Índice de Figuras

Figura 1 - Excerto de Código Cypher	3
Figura 2 - Schema da Base de Dados	3
Figura 3 - Relação FOLLOWS	4
Figura 4 - Relação ACTED_IN	4
Figura 5 - Relação DIRECTED.....	5
Figura 6 - Relação PRODUCED.....	5
Figura 7 - Relação WROTE	6
Figura 8 - Relação REVIEWED.....	6
Figura 9 - Criação do Nó Movie	7
Figura 10 – Visualização do Nó Movie	7
Figura 11 - Criação da Tagline	7
Figura 12 - Visualização do Nó Movie (1)	8
Figura 13 - Criação de um Nó Pessoa (Ator)	8
Figura 14 - Criação da Relação ACTED_IN	8
Figura 15 - Criação de um Nó Pessoa (Director)	8
Figura 16 - Criação da Relação DIRECTED	8
Figura 17 - Visualização do Nó Filme e Nós Pessoa associados	9
Figura 18 - Atribuição de roles	9
Figura 19 - Visualização dos roles criados	9
Figura 20 - Criação dos Nós Pessoa	10
Figura 21 - Criação da relação REVIEWED (John Wick)	10
Figura 22 - Criação da relação REVIEWED (Forrest Gump)	10
Figura 23 - Criação da relação REVIEWED (The Godfather)	11
Figura 24 - Visualização das diferentes relações para o Nó “John Wick”	11

Figura 25 - Visualização das diferentes relações para o Nó "Forrest Gump"	12
Figura 26 - Visualização das diferentes relações para o Nó "The Godfather"	12
Figura 27 - Atribuição de rating e summary às relações REVIEWED do Nó "The Godfather" .	13
Figura 28 - Atribuição de rating e summary às relações REVIEWED do Nó "John Wick"	13
Figura 29 - Atribuição de rating e summary às relações REVIEWED do Nó "Forrest Gump" ..	13
Figura 30 - Visualização das relações REVIEWED e respectivos atributos para o Nó "The Godfather"	14
Figura 31 - Visualização das relações REVIEWED e respectivos atributos para o Nó "Forrest Gump"	14
Figura 32 - Visualização das relações REVIEWED e respectivos atributos para o Nó "John Wick"	14
Figura 33 - Visualização das relações criadas	15
Figura 34 - Visualização da lista de filmes onde um dado actor tenha participado	16
Figura 35 - Visualização do cast de atores e produtores de um dado filme	16
Figura 36 - Visualização do número de reviews e respetivos reviewers de um dado filme ...	17
Figura 37 - Visualização do filme, data de lançamento, cast e rating dado por um certo reviewer	17
Figura 38 - Visualização das melhores notas de rating atribuídas pelos reviewers (top 5)	17
Figura 39 - Visualização das novas relações criadas	18
Figura 40 - Visualização do número de reviews e lista de filmes reviewed por um dado reviewer	19
Figura 41 - Visualização do Top 5 de Filmes mais bem avaliados tendo em conta todas as reviews feitas	19
Figura 42 - Visualização das relações entre Nós Person e Movie	20
Figura 43 - Query de recomendação de filmes com base em utilizadores semelhantes (filmes em comum ≥ 2) e boas avaliações (≥ 70).	21
Figura 44 - Query de recomendação de filmes otimizada, contendo apenas os dados necessários	22

1 Introdução

O seguinte relatório tem como objetivo a exploração prática de uma ferramenta de base de dados ainda não estudada durante a unidade curricular de Tópicos Avançados de Base Dados, do Mestrado em Engenharia Informática – Engenharia de Dados do Instituto Superior de Engenharia do Porto.

Como grupo, decidiu-se adotar uma base de dados denominada Neo4j. Esta ferramenta é orientada a grafos. Os grafos são constituídos por nós e arestas, que representam entidades e as relações entre as mesmas, respetivamente.

Resumidamente, o processo consistiu em criar a base de dados, os respetivos nós e arestas e finalizar com a construção de queries.

Este relatório explicará o raciocínio e o desenvolvimento de todo o processo de forma a esclarecer o método e as decisões tomadas.

2 Introdução Teórica – Neo4j

O Neo4j é um sistema avançado de gestão de base de dados de grafo (Graph Database Management System - GDBMS), notável pela sua habilidade em armazenar e consultar dados por meio de grafos. Este tipo de base de dados é projetado para representar e manipular relacionamentos entre entidades de maneira eficiente, sendo particularmente indicado para situações em que as relações entre os dados possuem tanta importância quanto os próprios dados.

No Neo4j, os dados são representados na forma de grafos, compostos por nós (nodes) e relacionamentos (relationships). Os nós representam as diferentes entidades, enquanto os relacionamentos delineiam as conexões entre essas entidades. Cada nó e relacionamento podem conter ainda propriedades específicas que armazenam informações relevantes.

A sua eficiência é notável nas consultas que envolvem relacionamentos, sendo especialmente vantajoso em cenários nos quais a natureza dessas conexões é crucial. Além disso, a sua estrutura de grafo possibilita uma fácil adaptação a mudanças nos requisitos de dados, além de ser escalável para lidar com grandes volumes de dados e tráfego.

Destaca-se, portanto, como uma ferramenta poderosa para a modelagem e consulta de dados complexos que envolvem relacionamentos. A estrutura de grafo proporciona uma abordagem intuitiva e eficiente para representar e explorar as conexões inerentes aos dados e torna-se uma escolha valiosa em ambientes nos quais a topologia das relações desempenha um papel central na compreensão e análise dos dados.

Com a observação do Schema da base de dados é possível retirar informações acerca dos diferentes tipos de relações entre os nós:

- **FOLLOWS**: Representa a relação entre o nó “Person” com ele mesmo e traduz-se na ligação que existe quando uma pessoa que segue outra pessoa;

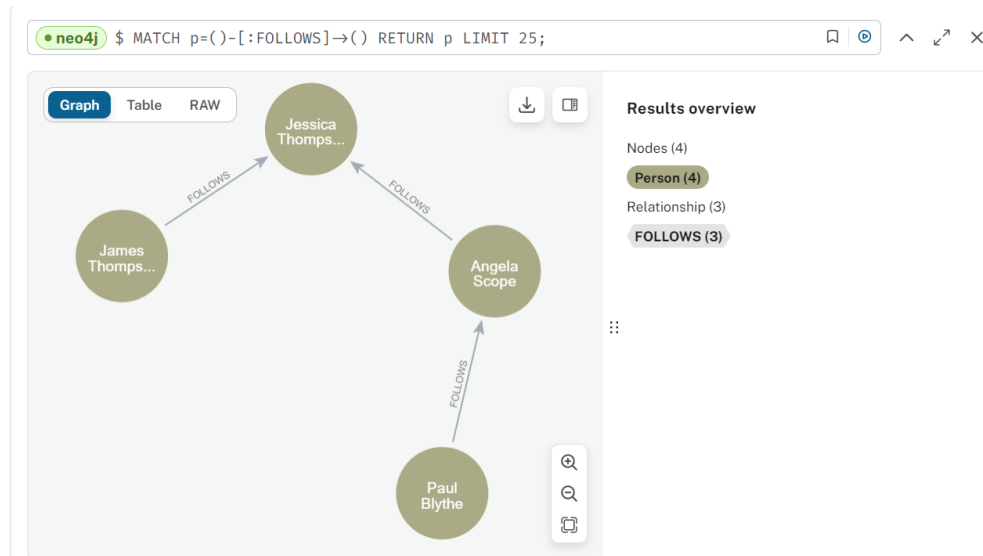


Figura 3 - Relação FOLLOWS

- **ACTED_IN**: Representa uma relação entre o nó “Person” e o nó “Movie” e traduz-se na ligação que existe quando uma pessoa atuou num certo filme;

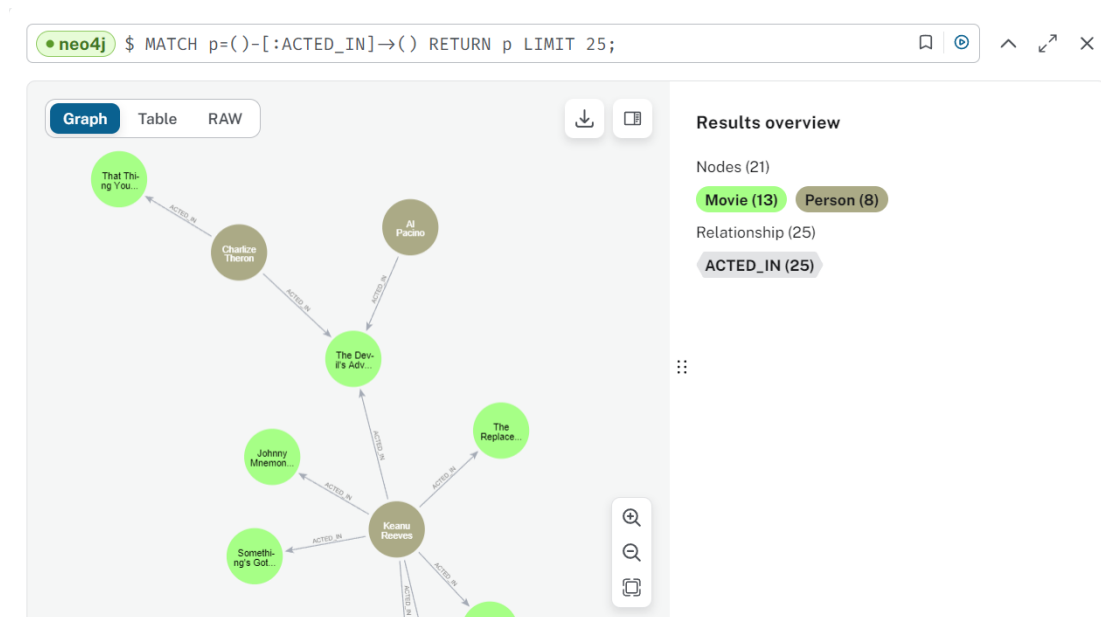


Figura 4 - Relação ACTED_IN

- DIRECTED: Representa uma relação entre o nó “Person” e o nó “Movie” e traduz-se na ligação que existe quando uma pessoa dirigiu um certo filme;

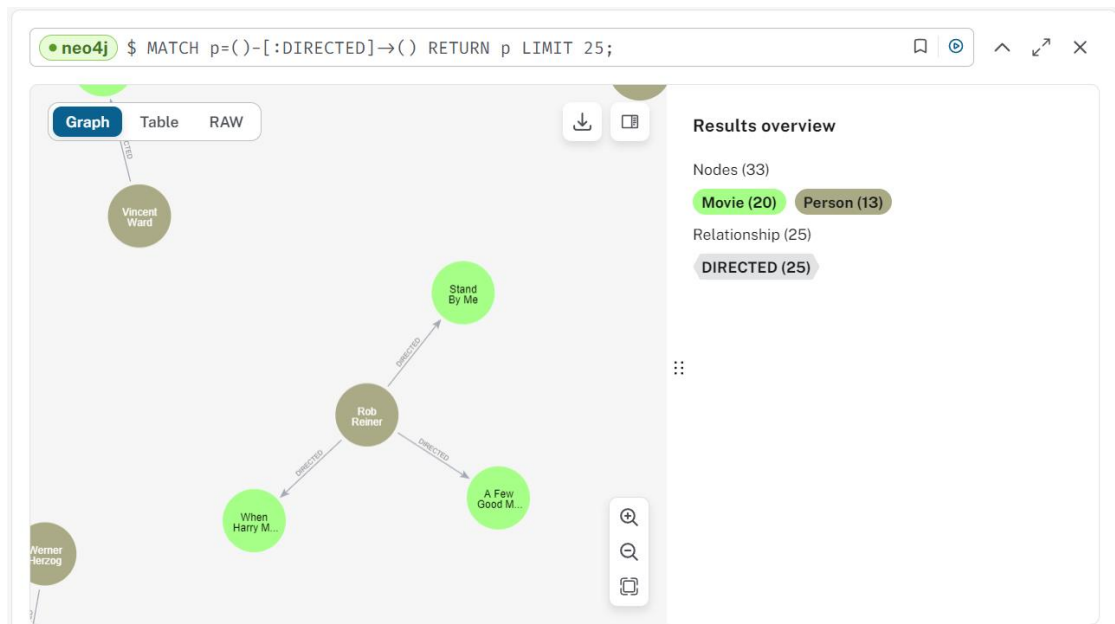


Figura 5 - Relação DIRECTED

- PRODUCED: Representa uma relação entre o nó “Person” e o nó “Movie” e traduz-se na ligação que existe quando uma pessoa produziu um certo filme;

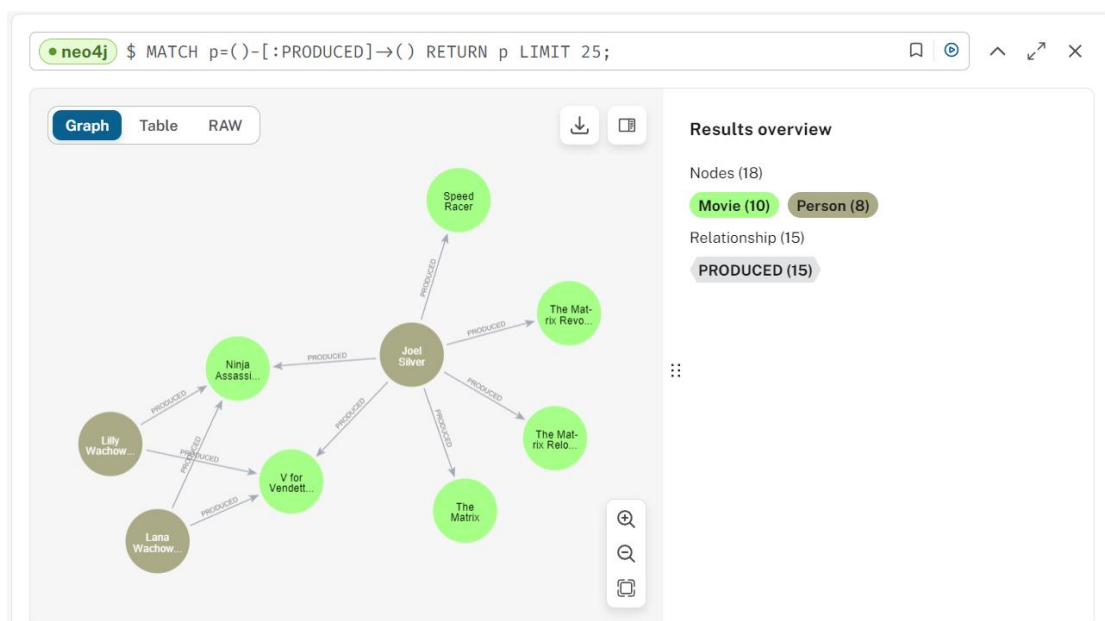


Figura 6 - Relação PRODUCED

- **WROTE**: Representa uma relação entre o nó “Person” e o nó “Movie” e traduz-se na ligação que existe quando uma pessoa escreveu um certo filme;



Figura 7 - Relação WROTE

- **REVIEWED**: Representa uma relação entre o nó “Person” e o nó “Movie” e traduz-se na ligação que existe quando uma pessoa escreveu uma crítica sobre um certo filme;

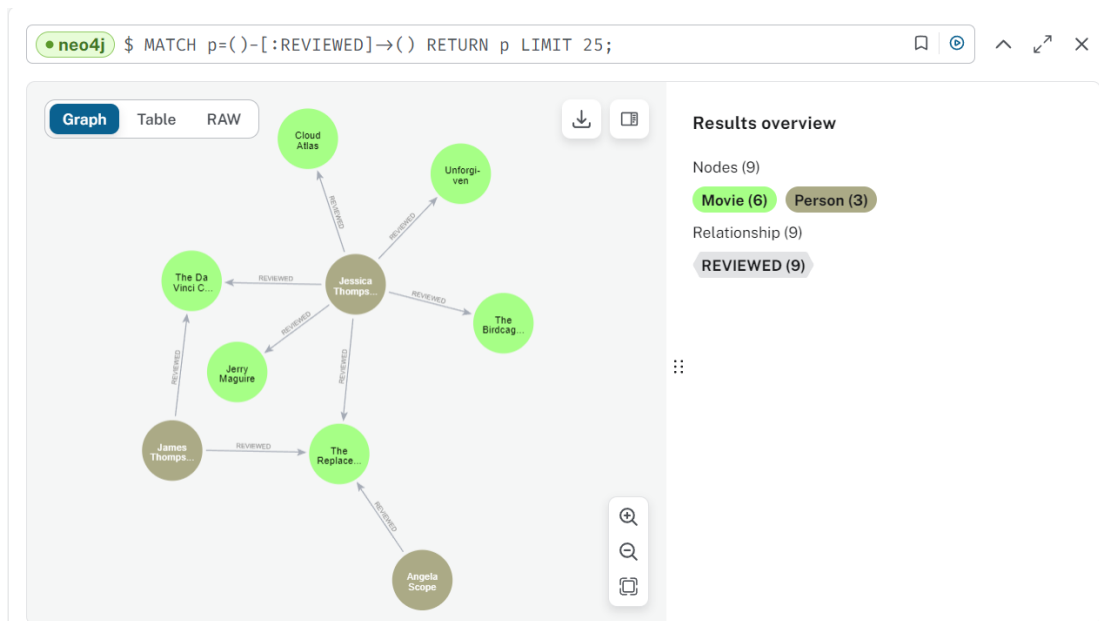


Figura 8 - Relação REVIEWED

3.2 Criação de Nós

Para tornar a base de dados “personalizada” decidiu-se que seriam adicionados 3 filmes novos com um elenco mínimo na base de dados de 2 atores e 1 diretor, e 3 pessoas novas que fizessem reviews a esses mesmos filmes. Os filmes selecionados foram: “Forrest Gump”, “John Wick” e “Godfather”.

3.2.1 Criação de Filmes

Para a criação dos nós “Movie” foi necessário um conjunto de passos de forma a modelar a nossa base de dados. Seguidamente, pode-se observar um exemplo de como foi criado um destes nós, sendo que os restantes teriam uma implementação semelhante:

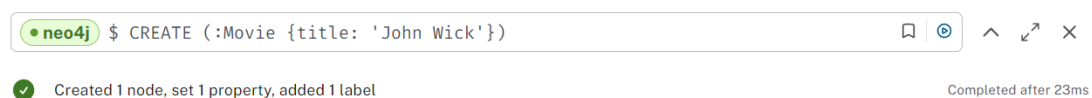


Figura 9 - Criação do Nó Movie

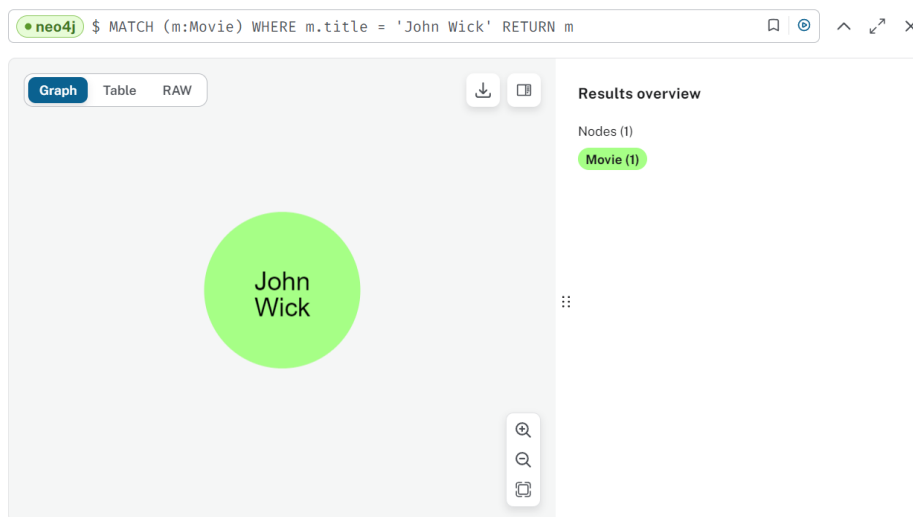


Figura 10 – Visualização do Nó Movie

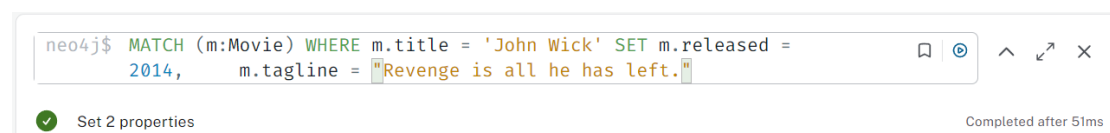


Figura 11 - Criação da Tagline

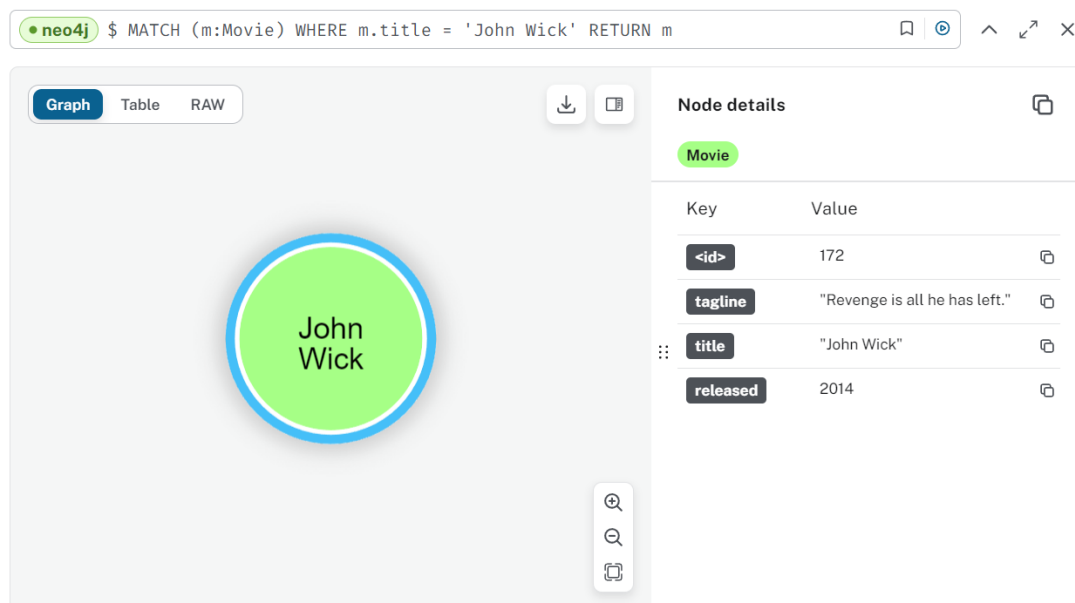


Figura 12 - Visualização do Nó Movie (1)

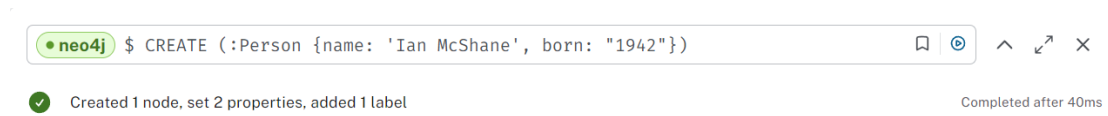


Figura 13 - Criação de um Nó Pessoa (Ator)

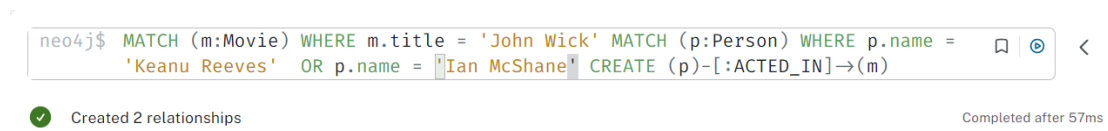


Figura 14 - Criação da Relação ACTED_IN

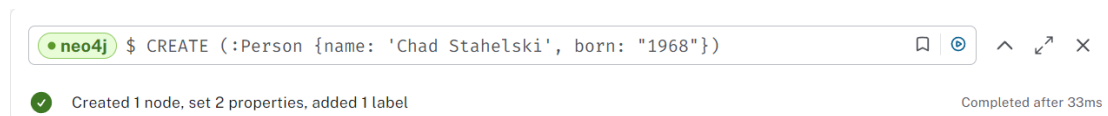


Figura 15 - Criação de um Nó Pessoa (Director)

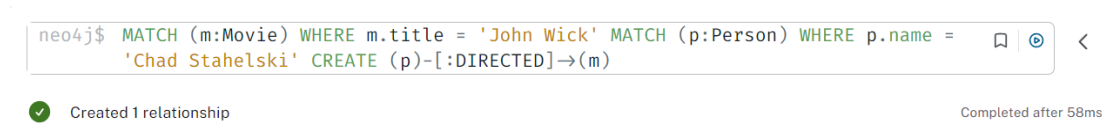


Figura 16 - Criação da Relação DIRECTED

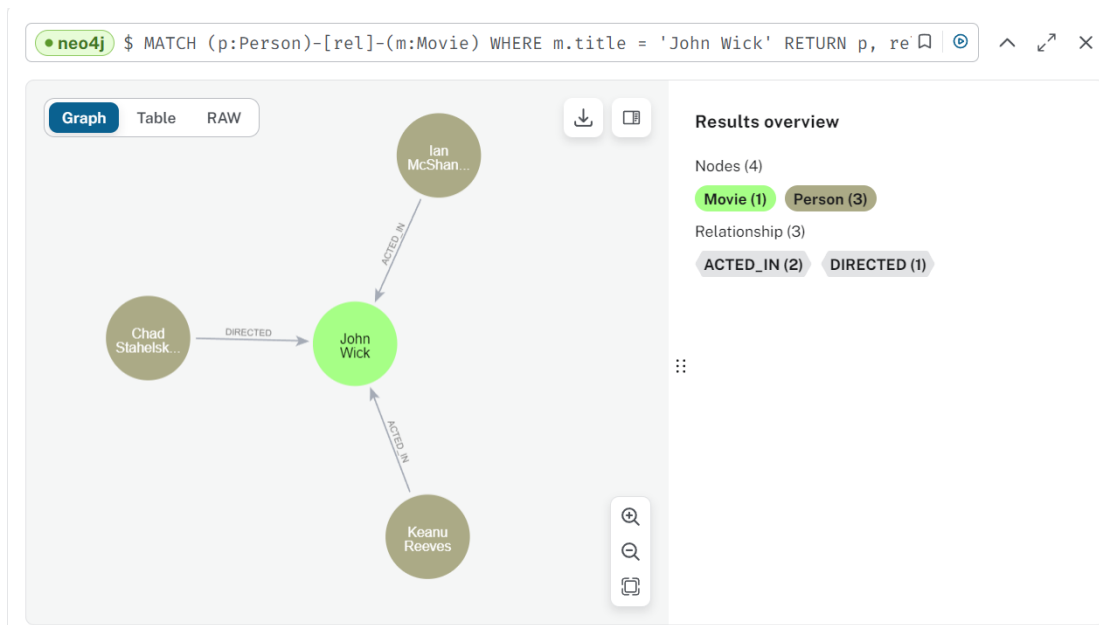


Figura 17 - Visualização do Nó Filme e Nós Pessoa associados

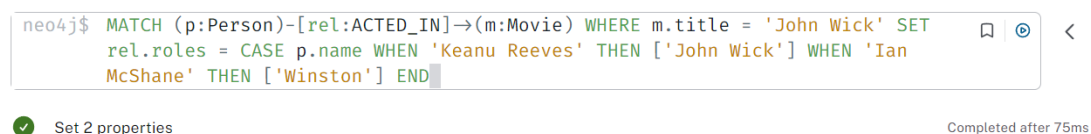


Figura 18 - Atribuição de roles



Figura 19 - Visualização dos roles criados

3.2.2 Criação de Pessoas

Para a criação dos nós “Person” com o intuito de acrescentar dados quanto às reviews dos filmes anteriormente criados, foi necessário a implementação dos seguintes passos:



Figura 20 - Criação dos Nós Pessoa

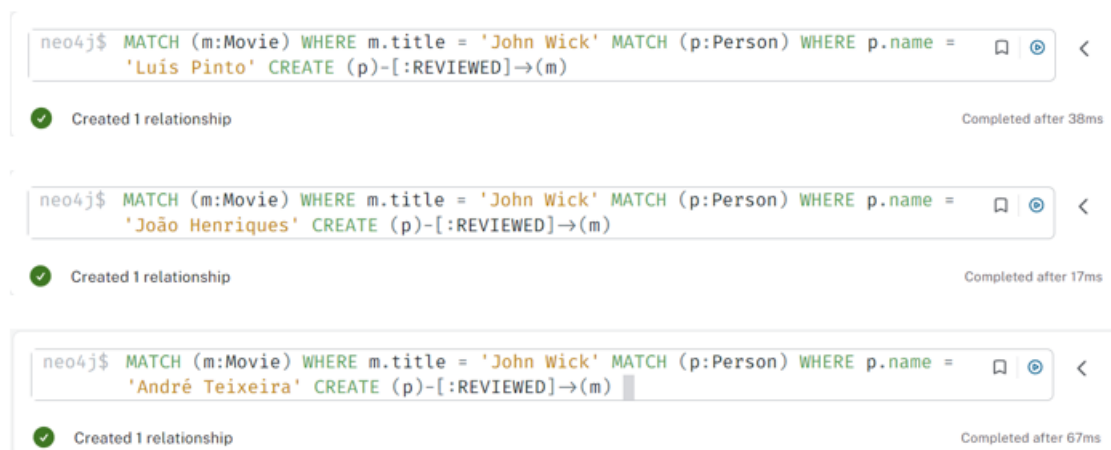


Figura 21 - Criação da relação REVIEWED (John Wick)

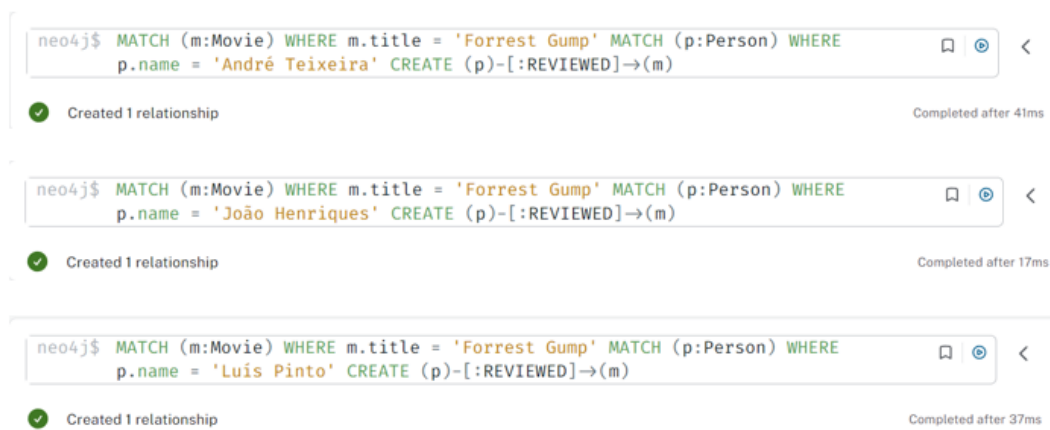


Figura 22 - Criação da relação REVIEWED (Forrest Gump)

```
neo4j$ MATCH (m:Movie) WHERE m.title = 'The Godfather' MATCH (p:Person) WHERE
p.name = 'Luís Pinto' CREATE (p)-[:REVIEWED]-(m)
```

Created 1 relationship Completed after 19ms

```
neo4j$ MATCH (m:Movie) WHERE m.title = 'The Godfather' MATCH (p:Person) WHERE
p.name = 'João Henriques' CREATE (p)-[:REVIEWED]-(m)
```

Created 1 relationship Completed after 49ms

```
neo4j$ MATCH (m:Movie) WHERE m.title = 'The Godfather' MATCH (p:Person) WHERE
p.name = 'André Teixeira' CREATE (p)-[:REVIEWED]-(m)
```

Created 1 relationship Completed after 19ms

Figura 23 - Criação da relação REVIEWED (The Godfather)

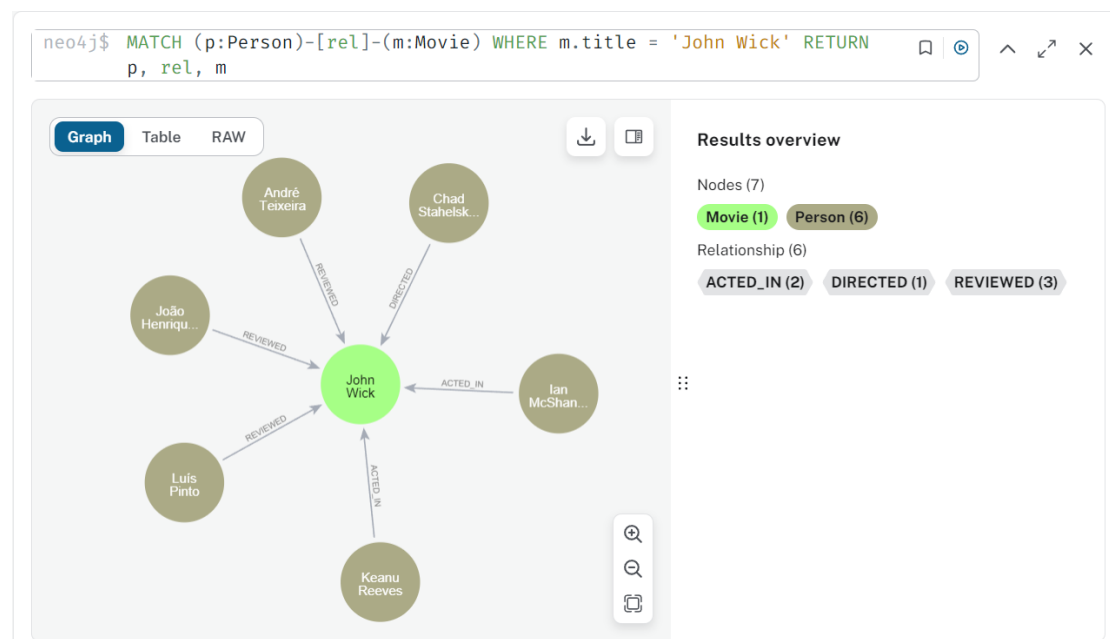


Figura 24 - Visualização das diferentes relações para o Nó "John Wick"

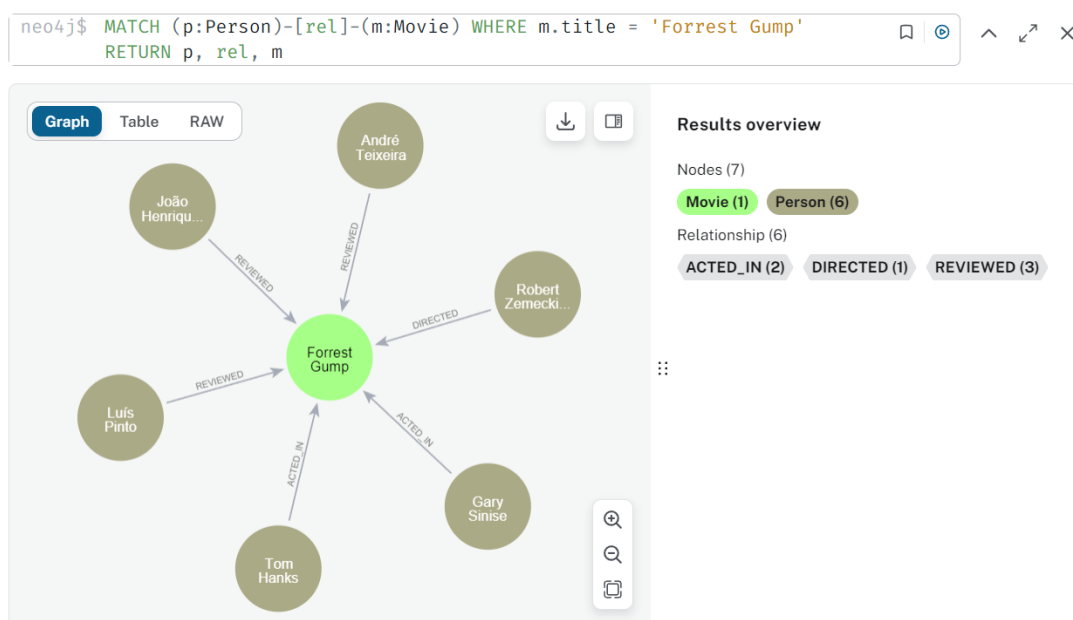


Figura 25 - Visualização das diferentes relações para o Nó “Forrest Gump”

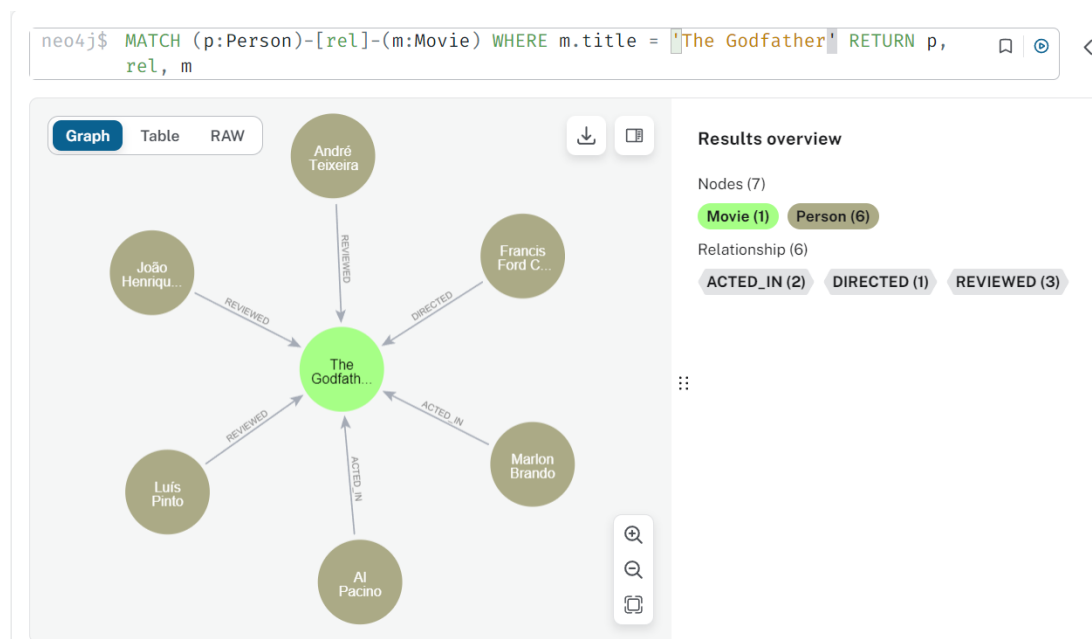


Figura 26 - Visualização das diferentes relações para o Nó “The Godfather”

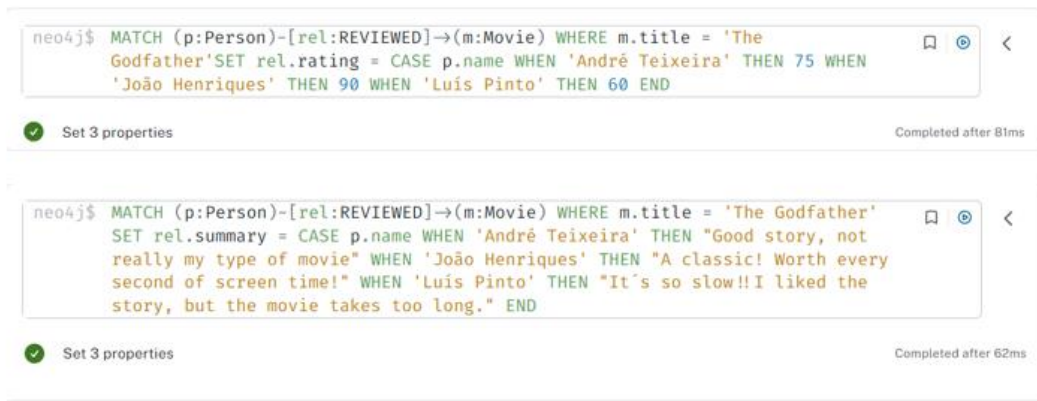


Figura 27 - Atribuição de rating e summary às relações REVIEWED do Nó "The Godfather"

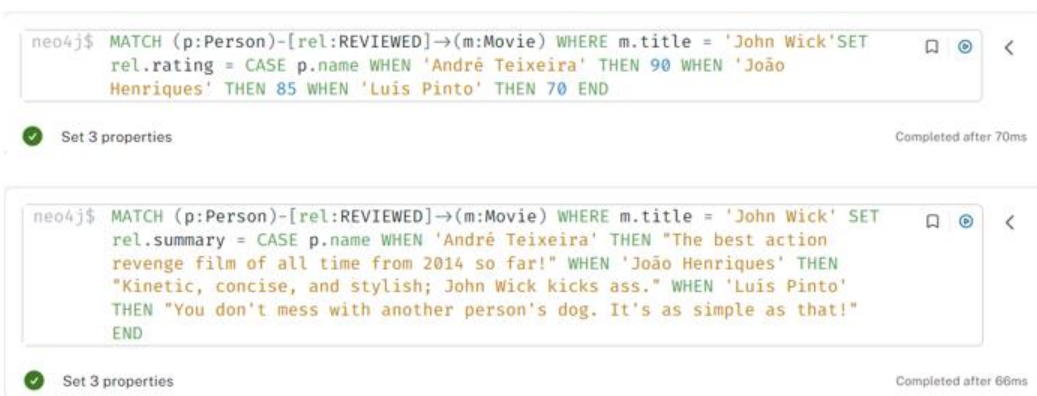


Figura 28 - Atribuição de rating e summary às relações REVIEWED do Nó "John Wick"

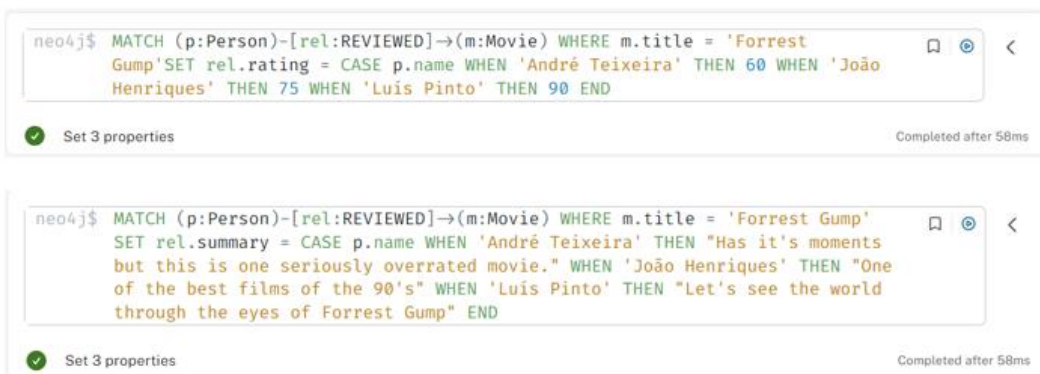


Figura 29 - Atribuição de rating e summary às relações REVIEWED do Nó "Forrest Gump"

neo4j\$ MATCH (p:Person)-[rel:REVIEWED]→(m:Movie) WHERE m.title = 'The Godfather'
RETURN p.name, rel.rating , rel.summary

Table RAW

	p.name	rel.rating	rel.summary
1	"André Teixeira"	75	"Good story, not really my type of movie"
2	"João Henriques"	90	"A classic! Worth every second of screen time!"
3	"Luís Pinto"	60	"It's so slow!!I liked the story, but the movie takes too long."

Figura 30 - Visualização das relações REVIEWED e respectivos atributos para o Nó "The Godfather"

neo4j\$ MATCH (p:Person)-[rel:REVIEWED]→(m:Movie) WHERE m.title = 'Forrest Gump'
RETURN p.name, rel.rating , rel.summary

Table RAW

	p.name	rel.rating	rel.summary
1	"André Teixeira"	60	"Has it's moments but this is one seriously overrated movie."
2	"João Henriques"	75	"One of the best films of the 90's"
3	"Luís Pinto"	90	"Let's see the world through the eyes of Forrest Gump"

Figura 31 - Visualização das relações REVIEWED e respectivos atributos para o Nó "Forrest Gump"

neo4j\$ MATCH (p:Person)-[rel:REVIEWED]→(m:Movie) WHERE m.title = 'John Wick'
RETURN p.name, rel.rating , rel.summary

Table RAW

	p.name	rel.rating	rel.summary
1	"André Teixeira"	90	"The best action revenge film of all time from 2014 so far!"
2	"João Henriques"	85	"Kinetic, concise, and stylish; John Wick kicks ass."
3	"Luís Pinto"	70	"You don't mess with another person's dog. It's as simple as that!"

Figura 32 - Visualização das relações REVIEWED e respectivos atributos para o Nó "John Wick"

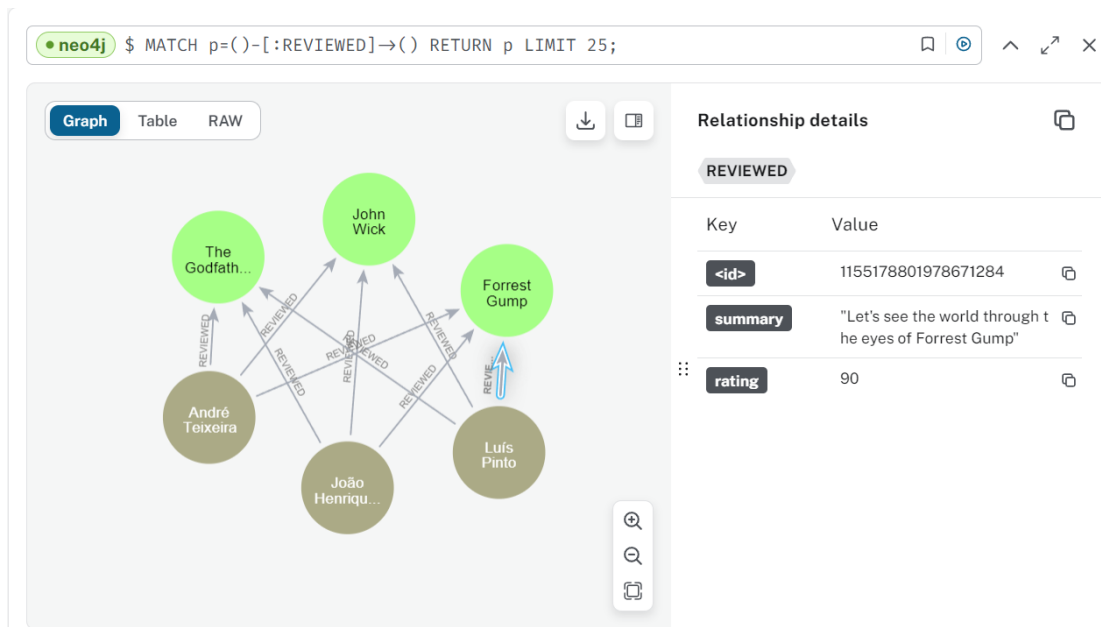


Figura 33 - Visualização das relações criadas

3.3 Identificação do Problema

Com a análise da base de dados surgem várias questões pertinentes, das quais se cogitou a recomendação de filmes. Considerando-a como um bom desafio, a implementação desta recomendação tem como base 2 fatores: Utilizadores com gostos semelhantes e filmes muito bem avaliados.

Para que a solução para este problema seja bem implementada foram criadas queries de análise à base de dados de forma a entender melhor alguns aspetos relativos à mesma.

4 Queries

neo4j

\$ MATCH (p:Person)-[:ACTED_IN]-(m:Movie) RETURN p.name as actor, collect(m.title) AS `movie list`

Table

RAW

	actor	movie list
1	"Keanu Reeves"	["The Matrix", "The Matrix Reloaded", "The Matrix Revolutions", "The Devil's Advocate", "The Replacements", "Johnny Mnemonic", "Something's Gotta Give", "John Wick"]
2	"Carrie-Anne Moss"	["The Matrix", "The Matrix Reloaded", "The Matrix Revolutions"]
3	"Laurence Fishburne"	["The Matrix", "The Matrix Reloaded", "The Matrix Revolutions"]
4	"Hugo Weaving"	["The Matrix", "The Matrix Reloaded", "The Matrix Revolutions", "Cloud Atlas", "V for Vendetta"]
5	"Emil Eifrem"	["The Matrix"]
6	"Charlize Theron"	["The Devil's Advocate", "That Thing You Do"]
7	"Al Pacino"	["The Devil's Advocate", "The Godfather"]
8	"Tom Cruise"	["A Few Good Men", "Top Gun", "Jerry Maguire"]
9	"Jack Nicholson"	["A Few Good Men", "As Good as It Gets", "Hoffa", "One Flew Over the Cuckoo's Nest", "Something's Gotta Give"]
10	"Demi Moore"	["A Few Good Men"]

Figura 34 - Visualização da lista de filmes onde um dado actor tenha participado

neo4j

\$ MATCH (a:Person)-[:ACTED_IN]-(m:Movie), (m)-[:PRODUCED]-(p:Person) WITH m, collect(DISTINCT a.name) AS cast, collect(DISTINCT p.name) AS producers RETURN DISTINCT m.title, cast, producers ORDER BY size(cast)

Table

RAW

Q

m.title	cast	producers
1 "Something's Gotta Give"	["Keanu Reeves", "Jack Nicholson", "Diane Keaton"]	["Nancy Meyers"]
2 "Ninja Assassin"	["Rick Yune", "Ben Miles", "Rain", "Naomie Harris"]	["Lilly Wachowski", "Lana Wachowski", "Joel Silver"]
3 "The Matrix Reloaded"	["Keanu Reeves", "Carrie-Anne Moss", "Laurence Fishburne", "Hugo Weaving"]	["Joel Silver"]
4 "The Matrix Revolutions"	["Keanu Reeves", "Carrie-Anne Moss", "Laurence Fishburne", "Hugo Weaving"]	["Joel Silver"]
5 "When Harry Met Sally"	["Meg Ryan", "Billy Crystal", "Carrie Fisher", "Bruno Kirby"]	["Rob Reiner", "Nora Ephron"]
6 "Cloud Atlas"	["Hugo Weaving", "Tom Hanks", "Halle Berry", "Jim Broadbent"]	["Stefan Arndt"]
7 "V for Vendetta"	["Hugo Weaving", "Natalie Portman", "Stephen Rea", "John Hurt", "Ben Miles"]	["Lilly Wachowski", "Lana Wachowski", "Joel Silver"]
8 "The Matrix"	["Keanu Reeves", "Carrie-Anne Moss", "Laurence Fishburne", "Hugo Weaving", "Emil Eifrem"]	["Joel Silver"]
9 "Speed Racer"	["Ben Miles", "Emile Hirsch", "John Goodman", "Susan"]	["Joel Silver"]

Figura 35 - Visualização do cast de atores e produtores de um dado filme

```
neo4j $ MATCH (p:Person)-[:REVIEWED]->(m:Movie) RETURN m.title as movie, count(p) as numReviews, collect(p.name) as reviewers
```

	movie	numReviews	reviewers
1	"The Replacements"	3	["Angela Scope", "Jessica Thompson", "James Thompson"]
2	"Jerry Maguire"	1	["Jessica Thompson"]
3	"The Birdcage"	1	["Jessica Thompson"]
4	"Unforgiven"	1	["Jessica Thompson"]
5	"Cloud Atlas"	1	["Jessica Thompson"]
6	"The Da Vinci Code"	2	["Jessica Thompson", "James Thompson"]
7	"Forrest Gump"	3	["André Teixeira", "João Henriques", "Luís Pinto"]
8	"John Wick"	3	["André Teixeira", "João Henriques", "Luís Pinto"]
9	"The Godfather"	3	["André Teixeira", "João Henriques", "Luís Pinto"]

Figura 36 - Visualização do número de reviews e respetivos reviewers de um dado filme

```
1 MATCH (r:Person)-[rel:REVIEWED]->(m:Movie)-[:ACTED_IN]-(a:Person)
2 RETURN DISTINCT r.name AS Reviewer, m.title AS Movie, m.released AS Released, collect(a.name) AS Cast, rel.rating AS Rating
```

Reviewer	Movie	Released	Cast	Rating
1 "Jessica Thompson"	"Jerry Maguire"	2000	["Tom Cruise", "Cuba Gooding Jr.", "Renee Zellweger", "Kelly Preston", "Jerry O'Connell", "Jay Mohr", "Bonnie Hunt", "Regina King", "Jonathan Lipnicki"]	92
2 "Angela Scope"	"The Replacements"	2000	["Keanu Reeves", "Brooke Langton", "Gene Hackman", "Orlando Jones"]	62
3 "Jessica Thompson"	"The Replacements"	2000	["Keanu Reeves", "Brooke Langton", "Gene Hackman", "Orlando Jones"]	65
4 "James Thompson"	"The Replacements"	2000	["Keanu Reeves", "Brooke Langton", "Gene Hackman", "Orlando Jones"]	100
5 "Jessica Thompson"	"The Birdcage"	1996	["Robin Williams", "Nathan Lane", "Gene Hackman"]	45
6 "Jessica Thompson"	"Unforgiven"	1992	["Gene Hackman", "Richard Harris", "Clint Eastwood"]	85

Started streaming 18 records after 127ms

Figura 37 - Visualização do filme, data de lançamento, cast e rating dado por um certo reviewer

```
neo4j $ MATCH (:Person)-[r:REVIEWED]->(m:Movie) RETURN m.title AS movie, r.rating AS rating ORDER BY r.rating DESC LIMIT 5
```

movie	rating
1 "The Replacements"	100
2 "Cloud Atlas"	95
3 "Jerry Maguire"	92
4 "Forrest Gump"	90
5 "John Wick"	90

Figura 38 - Visualização das melhores notas de rating atribuídas pelos reviewers (top 5)

5 Solução do Problema

Com a observação dos resultados anteriormente obtidos pelas queries feitas à base de dados foi possível identificar a necessidade de um volume mais elevado de dados de forma a conseguir verificar a implementação do mecanismo de recomendação de filmes de forma objetiva, nomeadamente pela falta de “reviews” por parte dos utilizadores. Com isto em mente foram criados mais 4 nós “Person” que se traduzem como utilizadores ou “reviewers”, totalizando 10 e criadas novas “reviews” até atingir um número de 5 “reviews” por utilizador.

Nas seguintes figuras podem-se observar os resultados após a criação dos nós “Person” e “reviews” associadas:

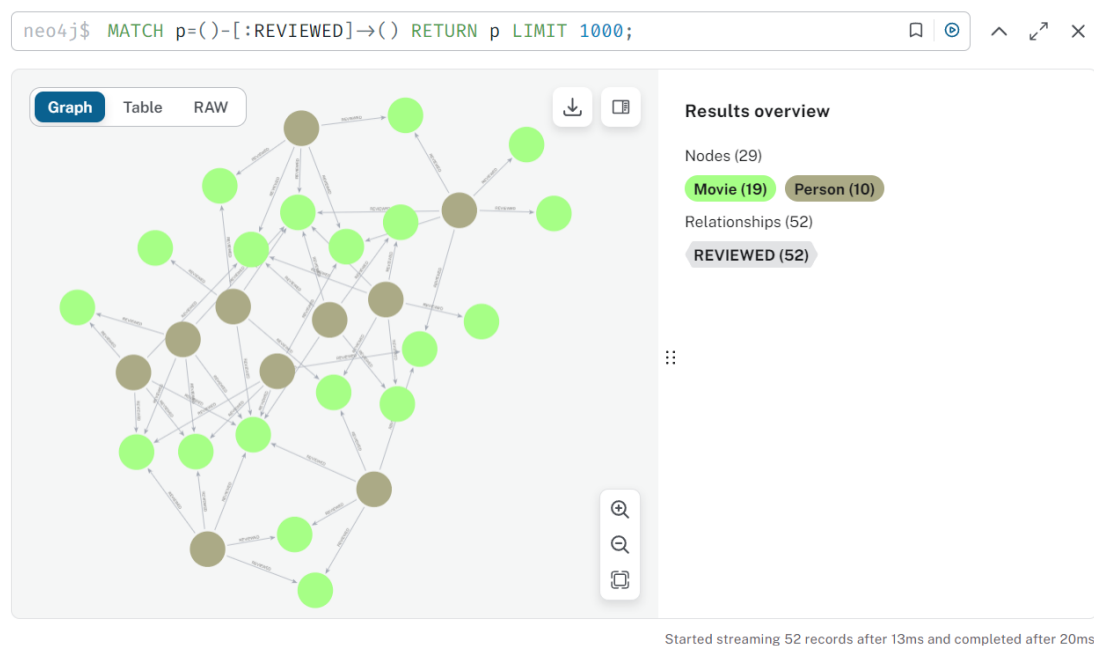


Figura 39 - Visualização das novas relações criadas

neo4j\$ MATCH (p:Person)-[:REVIEWED]→(m:Movie) RETURN p.name as Reviewers, count(p) as numReviews, collect(m.title) as Movies

	Reviewers	numReviews	Movies
1	"Angela Scope"	6	["The Matrix", "The Matrix Revolutions", "When Harry Met Sally", "The Replacements", "Speed Racer", "Ninja Assassin"]
2	"Inês Cardoso"	5	["The Matrix", "That Thing You Do", "The Replacements", "Hoffa", "The Godfather"]
3	"Diogo Pina"	5	["The Matrix", "A Few Good Men", "Top Gun", "Jerry Maguire", "The Godfather"]
4	"Diogo Sequeira"	5	["The Matrix Revolutions", "The Replacements", "Speed Racer", "Ninja Assassin", "The Godfather"]
5	"André Teixeira"	5	["A Few Good Men", "Top Gun", "Forrest Gump", "John Wick", "The Godfather"]

Started streaming 10 records after 79ms and completed after 82ms.

Figura 40 - Visualização do número de reviews e lista de filmes reviewed por um dado reviewer

Após um aumento significativo no número de “reviews” foi possível criar uma query que corresponde ao objetivo em mente, tendo em conta os filmes mais bem avaliados (top 5) e filmes que o utilizador não tenha visto e que obtenham uma avaliação superior ou igual a 70 por parte de utilizadores que tenham pelo menos 2 filmes em comum com o utilizador com avaliação superior ou igual a 70.

Desconstruindo inicialmente este objetivo podemos começar pela primeira parte, top 5 filmes mais bem avaliados:

neo4j\$ MATCH (:Person)-[r:REVIEWED]→(m:Movie) RETURN m.title AS movie, avg(r.rating) AS rating ORDER BY rating DESC LIMIT 5

	movie	rating
1	"The Matrix Revolutions"	92.5
2	"John Wick"	90.0
3	"Hoffa"	90.0
4	"The Matrix"	88.33333333333333
5	"Unforgiven"	85.0

Figura 41 - Visualização do Top 5 de Filmes mais bem avaliados tendo em conta todas as reviews feitas

Podem-se observar o top 5 de filmes mais bem avaliados em ordem decrescente após a execução desta simples query.

A segunda parte do mecanismo de recomendação de filmes sugerido requer filmes bem avaliados (≥ 70) de utilizadores que tenham em comum pelo menos 2 filmes.

Para verificar a eficácia da query vão-se utilizar os seguintes utilizadores que têm pelo menos 2 filmes em comum: “André Teixeira” e “Diogo Pina”.

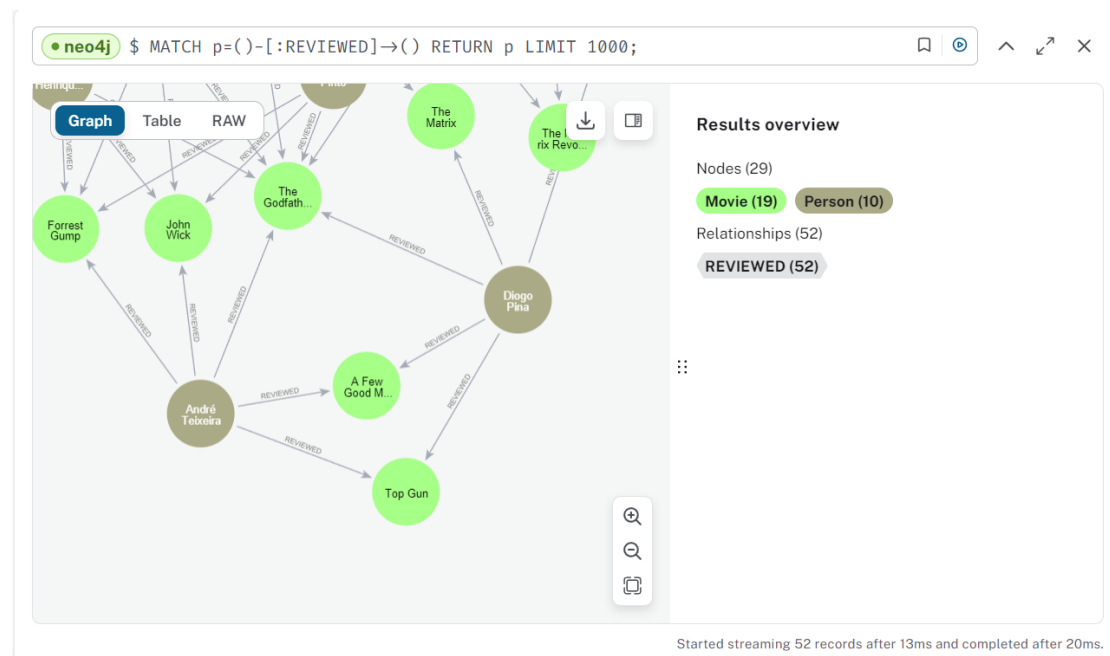


Figura 42 - Visualização das relações entre Nós Person e Movie

Utilizando o nó “Diogo Pina” como utilizador alvo pode-se observar que este e o utilizador “André Teixeira” contam com 3 filmes em comum, que não devem ser incluídos na lista de filmes recomendados pois já foram visualizados e, portanto, restam apenas 2 possíveis filmes para recomendação por parte do nó “André Teixeira”. O resultado esperado será uma lista contendo o filme “John Wick”, pois o outro filme em questão, “Forrest Gump”, conta com uma avaliação por parte do nó “André Teixeira” de 65, sendo inferior à desejada (≥ 70).

```

1 MATCH (p:Person)-[:REVIEWED]→(movie:Movie)
2 WHERE p.name = 'Diogo Pina'
3 WITH p, COLLECT(movie) AS reviewedMovies
4
5 MATCH (pp:Person)-[:REVIEWED]→(commonMovie:Movie)
6 WHERE pp < p AND commonMovie IN reviewedMovies
7 WITH p, pp, COLLECT(commonMovie) AS commonMovies
8 WHERE SIZE(commonMovies) ≥ 2
9
10 MATCH (pp:Person)-[rel:REVIEWED]→(m:Movie)
11 WHERE pp.name < 'Diogo Pina' AND rel.rating ≥ 70 AND NOT m IN commonMovies
12 RETURN pp.name AS SimilarUser, m.title AS Movie, rel.rating AS Rating

```

Table RAW

	SimilarUser	Movie	Rating
1	"Inês Cardoso"	"That Thing You Do"	70
2	"Inês Cardoso"	"Hoffa"	90
3	"André Teixeira"	"John Wick"	80
4	"Luís Pinto"	"The Da Vinci Code"	85
5	"Luís Pinto"	"Forrest Gump"	80
6	"Luís Pinto"	"John Wick"	90

Figura 43 - Query de recomendação de filmes com base em utilizadores semelhantes (filmes em comum ≥ 2) e boas avaliações (≥ 70).

Como se pode verificar, apenas o filme “John Wick” do utilizador “André Teixeira” entrou para a lista e todos os filmes sugeridos têm um “rating” maior ou igual a 70. Pode-se verificar ainda que o utilizador “Diogo Pina” tem 3 utilizadores distintos dos quais tem em comum pelo menos 2 filmes.

De forma a apresentar os resultados de uma forma mais sugestiva, a query pode ser reformulada da seguinte forma, apresentando apenas os filmes sugeridos bem como a sua avaliação que é o que é importante para o utilizador:

```

1 MATCH (p:Person)-[:REVIEWED]→(movie:Movie)
2 WHERE p.name = 'Diogo Pina'
3 WITH p, COLLECT(movie) AS reviewedMovies
4 MATCH (pp:Person)-[:REVIEWED]→(commonMovie:Movie)
5 WHERE pp < p AND commonMovie IN reviewedMovies
6 WITH p, pp, COLLECT(commonMovie) AS commonMovies
7 WHERE SIZE(commonMovies) ≥ 2
8
9 MATCH (pp:Person)-[rel:REVIEWED]→(m:Movie)
10 WHERE pp.name < 'Diogo Pina' AND rel.rating ≥ 70 AND NOT m IN commonMovies
11 MATCH (:Person)-[r:REVIEWED]→(m)
12 RETURN distinct m.title AS Movie, avg (r.rating) AS Rating

```

Table RAW

	Movie	Rating
1	"That Thing You Do"	50.0
2	"Hoffa"	90.0
3	"John Wick"	90.0
4	"The Da Vinci Code"	83.33333333333333
5	"Forrest Gump"	65.0

Figura 44 - Query de recomendação de filmes otimizada, contendo apenas os dados necessários

Conclusão

Com a realização deste projeto foi possível aprofundar conhecimentos na base de dados “Neo4j”, bem como criar um mecanismo de recomendação de filmes simples, mas consistente.

Foi possível identificar que os dados poderiam ser mais e melhores e que com mais atributos, nomeadamente no que toca aos nós “Movie”, como tipo de filme, duração e idioma poderia ser criado um mecanismo de recomendação mais sofisticado tendo em conta estes atributos.

No geral, as queries feitas à base de dados proporcionaram boas informações sobre a mesma e foi mais fácil a idealização do mecanismo de recomendação com base nestas informações. A parte mais visual da base de dados “Neo4j” demonstrou-se uma grande ajuda no decorrer deste projeto, tanto na perceção de relações como na identificação de dados em falta.

Em suma, este projeto contribui para um não só para um aprofundamento de conhecimentos, mas também para uma pequena demonstração das grandes capacidades desta base de dados que é o “Neo4j” que possibilitou solucionar o problema proposto com um mecanismo de recomendação de filmes com base em apenas 2 queries.

Referências

neo4j-graph-examples. (n.d.). NEO4J-graph-examples/movies: Built-in movie graph example.

GitHub. <https://github.com/neo4j-graph-examples/movies>

(2023). Neo4j.com. <https://guides.neo4j.com/intro-neo4j-exercises/>

Tutorial: Build a Cypher Recommendation Engine - Developer Guides. (n.d.). Neo4j Graph Data Platform. Retrieved December 30, 2023, from <https://neo4j.com/developer/cypher/guide-build-a-recommendation-engine/>