



Ingeniería y Ciencia de Datos

Ingeniería en Sistemas Computacionales

Minería de Grafos

Mini Proyecto 1

Equipo 3

ICD745653 - López Arnaud, Luis Francisco

IS727272 - Marco Ricardo Cordero Hernández

Jal., 28 de septiembre de 2023

Índice

Introducción.....	1
Desarrollo.....	3
Carga de datos.....	3
Análisis básico.....	3
Análisis de centralidad.....	9
Observaciones.....	17
Comparativa entre resultados y contexto de la temporada.....	18
Conclusiones.....	20
Bibliografía.....	21

Introducción

En el devenir del curso y sobre el contexto actual sobre el cual se comienzan a realizar análisis sobre grafos, el proyecto actual tiene como objetivo el desarrollo de un marco de trabajo sistemático el cual puede ser aplicado sobre cualquier tipo de grafo sin previo conocimiento del mismo para obtener su información relevante y posterior disección a través de distintos métodos y algoritmos pertinentes para este tipo de estructuras.

Como ejercicio práctico y aplicado al mundo real, se hará uso de un conjunto de datos relacionado a una popular serie televisiva del género drama/fantasía medieval, mejor conocida como “Game of Thrones” (Benioff y Weiss, 2011). Específicamente, se hará un análisis inicial exploratorio y de centralidad para su tercera temporada.

La manera en que estos datos son cargados y manipulados es a través de archivos de valores separados por comas, contando con *nodos* y *relaciones*. Hasta este momento, se ha hecho mención de tres aspectos que tienen que ser manejados adecuadamente: carga de datos, y componentes de un grafo. La principal tecnología para hacer frente a estos requerimientos será la herramienta *Neo4j* (s.f.), una de las bases de datos orientada a grafos más popular de los últimos años, a la cual se le puede dotar con diversas extensiones para proporcionar funcionalidades adicionales a las que ya cuenta el software base, las cuales por sí solas ya son sumamente poderosas; Hablando de estas funciones extendidas, se hará uso del plugin *Graph Data Science* [GDS] (s.f) para implementar *algoritmos de centralidad* (discutidos más adelante) y mostrar sus resultados en cada componente del grafo. También, aunque va más allá del alcance del presente, se hará uso de una herramienta más para presentar los hallazgos a obtener, siendo esta la interfaz de construcción de tableros de información *NeoDash* (de Jong et al, s.f.).

Como bien se mencionaba, el objetivo de la actividad presente consta en proporcionar evidencia de dos procesos: el análisis básico de un grafo completamente ajeno al desarrollador, y un posterior análisis de centralidad.

La relevancia con la que cuenta el analizar cualquier cosa sin que se tenga información previa es algo evidente y casi obvio, desgraciadamente, la ignorancia y orgullo son de los peores males que azotan día con día a la humanidad, la cual, aún cuando ya se encuentra atiborrada de información sin sentido, parece extremo detenerse un momento a pensar para comprender algo nuevo. Con esta cuasi reflexión, y aplicándolo al campo de los grafos, cuando se proporciona uno nuevo, no sé conoce nada del mismo, y por ende, no se podrían elaborar grandes conclusiones acerca de este más allá de lo que unas cuantas inferencias podrían sugerir. De igual manera, como se ha hecho mención anteriormente, Neo4j cuenta con diversos métodos nativos y sencillos de implementar dentro de sus posibilidades de construcción de consultas. Un poco de esta sintaxis puede ser encontrado en uno de los

materiales de referencia utilizados para el desarrollo de este documento, el cual también ha servido para brindar un panorama sucinto pero útil en el breviario teórico actual (Frame y Blumenfeld, 2022).

Ahora bien, una vez realizado el análisis básico y contando con la información necesaria para dictaminar la estructura y comportamiento del grafo, múltiples análisis de centralidad a través de algoritmos pertinentes a este rubro serán aplicados sobre el mismo, tales como algoritmos de grado, cercanía, *PageRank*, entre otros. En este caso, la relevancia depende totalmente del contexto, como lo podrían ser aeropuertos, sistemas de comunicación masiva, o cualquier tipo de red compleja con nodos de alta importancia (Ugurlu, 2022); en aras de la demostración actual, la aplicación resulta útil tanto para determinar el contexto de los personajes encontrados en el conjunto de datos original y sus relaciones.

Para una mejor comprensión de los algoritmos a implementar, se ha hecho uso del quinto capítulo del libro “*Graph Algorithms*” (Needham y Hodler, 2021), en donde se detallan de manera impecable la definición, uso, utilidad e implementación de cada uno de ellos. Contar con este tipo de recursos resulta de gran utilidad, puesto que es conveniente comprender no sólo el cómo implementar un algoritmo, sino su funcionamiento y su teoría detrás.

Finalmente, se hará una interpretación y comparativa de los resultados obtenidos para intentar enlazar los sucesos ocurridos en los contenidos de la temporada real de la serie en cuestión. Como lo indican RafeeFard et al. (2022), los sucesos de la vida real y su cotidianidad son estructuras que muestran fluctuaciones y cambios a través del tiempo, por lo que se les puede modelar como grafos, y, con una correcta interpretación, el análisis a través de los mismos y la muestra de sus resultados con el uso de frases más acercadas a la comparativa de la serie per se más que a las tecnicidades de este tipo de trabajos, hacen de la minería de grafos algo contextual pero útil y legible para todo tipo de persona que se encuentre al menos un poco inmersa en la cultura popular contemporánea.

Sin más preámbulo, las siguientes secciones muestran consultas escritas a través de Cypher, el lenguaje de consultas de Neo4j (s.f), sus resultados e interpretación, con el fin de demostrar el uso de la misma herramienta y del razonamiento lógico-práctico.

Desarrollo

A continuación se presentan una serie de preguntas que atienden tanto al análisis básico de un grafo, como a su análisis de centralidad. El grafo en cuestión contiene información de personajes y sus relaciones pertenecientes a la serie previamente mencionada “Game of Thrones”, específicamente para su *tercera temporada*. La fuente de estos datos puede ser consultada [aquí](#).

Carga de datos

Previo al trabajo de análisis requerido para comprender el grafo a analizar, evidentemente, se debe contar con el grafo. Para generarlo, se hará uso de la fuente de datos especificada y la capacidad de Neo4j de importar este tipo de datos contenidos en archivo de formato *.csv*. El código para hacer esto es el siguiente:

```
// Importar nodos
LOAD CSV WITH HEADERS FROM "file:///got-s3-nodes.csv" AS row
CREATE (n:Person)
SET n = row;

// Crear índice para los nodos
CREATE INDEX index_person FOR (n:Person) ON (n.id);

// Crear índice de búsqueda (útil para bloom) => Label
CREATE INDEX index_Person_First_Name FOR (n:Person) ON (n.Label);

// Cargar relaciones entre nodos
LOAD CSV FROM "file:///got-s3-edges.csv" AS row
MATCH (n:Person), (m:Person)
WHERE n.Id = row[0] and m.Id = row[1]
MERGE (n)-[:INTERACTS_WITH {Weight: toInteger(row[2])}]->(m);
```

Análisis básico

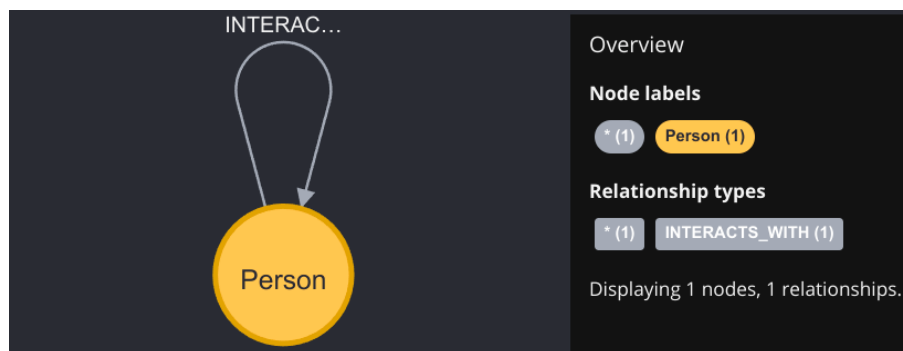
Para comprender un grafo desconocido, existen múltiples técnicas para descubrir sus contenidos, elementos, relaciones, propiedades, contexto, significado, impacto, etc. Sin embargo, como se comentaba en la introducción, la disposición y utilización de la herramienta *Neo4j* y su lenguaje *Cypher*, al ser totalmente enfocados a bases de grafos, cuenta con varios métodos que proveen información relevante y, en primera instancia, desconocida para el usuario.

La relevancia encontrada en un análisis exploratorio principal se encuentra en el valor que su información retornará, puesto que, en el contexto de un conjunto de datos sin previo conocimiento de sus contenidos, no se podría trabajar sobre él, al menos no de manera puntual o a un nivel de detalle profundo

A continuación, se presenta una propuesta de preguntas que atenderían al análisis inicial del grafo sobre el cual se realiza este desarrollo, con el propósito de brindar información relevante de sus contenidos.

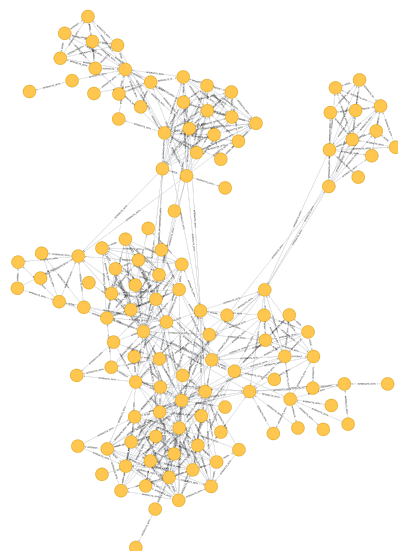
1. ¿Cuál es el esquema del grafo?

```
CALL db.schema.visualization;
```



2. ¿Cómo se ve el grafo entero?

```
MATCH p=()-[]-() RETURN p;
```



3. ¿Cuáles nodos hay?

```
CALL db.labels() YIELD label RETURN COLLECT(label) AS NODOS;
```

NODOS
["Person"]

4. ¿Cuántos nodos hay?

```
MATCH (n) RETURN COUNT(DISTINCT n) AS totalNodos;
```

totalNodos
125

5. ¿Cuáles atributos tienen los nodos?

```
CALL db.schema.nodeTypeProperties()  
YIELD nodeType AS Nodo, propertyName AS Propiedad, propertyTypes AS Tipo;
```

Nodo	Propiedad	Tipo
":Person"	"Id"	["String"]
":Person"	"Label"	["String"]

6. ¿Cuáles son las relaciones?

```
MATCH ()-[r]->() RETURN DISTINCT TYPE(r) AS RELACIONES;
```

RELACIONES
"INTERACTS_WITH"

7. ¿Cuántas son las relaciones?

```
MATCH ()-[r]->() RETURN COUNT(r) as totalRelaciones;
```

totalRelaciones
508

8. ¿Cuáles atributos tienen las relaciones?

```
CALL db.schema.relTypeProperties()
```

```
YIELD relType AS Relacion, propertyName AS Propiedad, propertyTypes AS Tipo;
```

Relacion	Propiedad	Tipo
":INTERACTS_WITH"	"Weight"	["Long"]

9. ¿Cuál es el diámetro del grafo?

```
MATCH (start), (end)
WHERE start <> end
WITH shortestPath((start)-[*]-(end)) AS path
WITH length(path) AS distance
WHERE distance IS NOT NULL
RETURN distance AS Diámetro
ORDER BY distance DESC
LIMIT 1;
```

Diámetro
7

10. ¿Cuál es el diámetro del grafo? (con dirección)

```
MATCH (start), (end)
WHERE start <> end
WITH shortestPath((start)-[*]->(end)) AS path
WITH length(path) AS distance
WHERE distance IS NOT NULL
RETURN distance AS DiámetroDirigido
ORDER BY distance DESC
LIMIT 1;
```

DiámetroDirigido
6

11. Comprobar si el grafo es completamente conexo

```
MATCH (n), (m)
WHERE id(n) <> id(m)
OPTIONAL MATCH p = shortestPath((n)-[*]-(m))
WITH n, m, p
WHERE p IS NULL
RETURN DISTINCT FALSE AS totallyConnected
LIMIT 1;
// Si el resultado es nulo, el grafo es conexo
```


(no changes, no records)

12. ¿Cuántas personas se relacionan con Jon (Snow)?

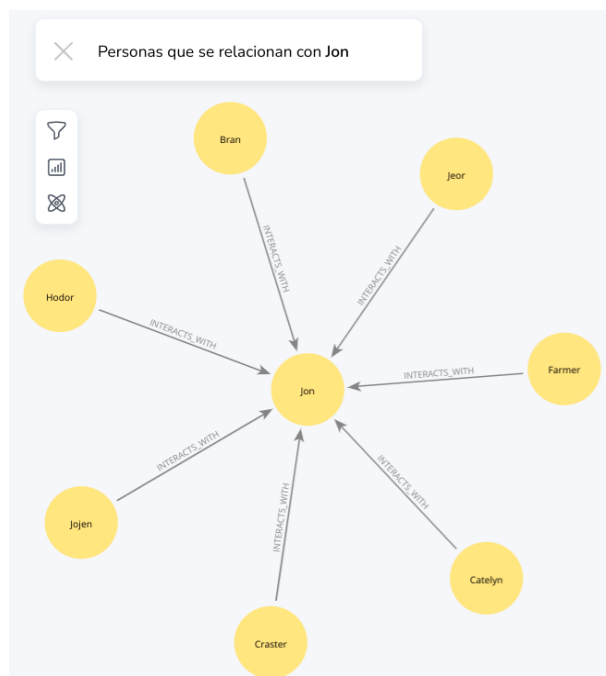
```
MATCH (m)-[:INTERACTS_WITH]->(n)
WHERE n.Label = 'Jon'
RETURN COUNT(m) AS totalRelaciones;
```

// Frase de búsqueda en Bloom: Personas que se relacionan con \$PERSONAJE

```
MATCH (m)-[r:INTERACTS_WITH]->(n)
WHERE n.Label = $PERSONAJE
RETURN m, r, n;
```

totalRelaciones

7



13. ¿Qué relación tiene más peso?

```
MATCH (n)-[r:INTERACTS_WITH]->(m)
WHERE n <> m
RETURN n.Label, m.Label, r.Weight AS Peso
ORDER BY r.Weight DESC
LIMIT 1;
```

n.Label	m.Label	Peso
"Jon"	"Ygritte"	159

14. ¿Qué relación tiene menos peso?

```
MATCH (n)-[r:INTERACTS_WITH]->(m)
WHERE n <> m
WITH n, m, r
RETURN n, m, r
ORDER BY r.Weight ASC
LIMIT 1;
```

n.Label	m.Label	Peso
"Aegon"	"Shireen"	2

15. ¿El grafo es homogéneo o heterogéneo?

```
MATCH (n)
WITH COLLECT(DISTINCT LABELS(n)) AS NodeTypes
RETURN NodeTypes, COUNT(NodeTypes) AS TOTAL;

MATCH ()-[r]->()
WITH COLLECT(DISTINCT TYPE(r)) AS RelationshipTypes
RETURN RelationshipTypes, COUNT(RelationshipTypes) AS TOTAL;
// Sí es homogéneo, ya que todos los nodos y relaciones tienen las
mismas propiedades
```

NodeTypes	TOTAL	RelationshipTypes	TOTAL
["Person"]	1	["INTERACTS_WITH"]	1

Análisis de centralidad

Una vez que se ha hecho la exploración relevante y existe una familiaridad considerable con los contenidos del grafo, es posible comenzar a realizar un tipo distinto de análisis: el de centralidad.

Para hacer posible esto, los algoritmos previamente descritos serán utilizados a través de la misma herramienta de bases de grafos, sin embargo, para llevar a cabo esto, también se hará uso de un complemento que extiende la funcionalidad nativa del software inicial: la librería de Neo4j Graph Data Science (*GDS*). Esta extensión provee múltiples algoritmos que operan sobre grafos de distintas características y tamaños, con posibilidad de personalización y especificidad de parámetros.

Existen múltiples algoritmos disponibles que pueden ser aplicados a distintos tipos de grafos con distintas características individuales, sin embargo, para el grafo actual y el contexto del momento actual del curso, se hará uso de *algoritmos de centralidad*, siendo estos los siguientes:

- *Degree centrality*: algoritmo para encontrar la cantidad de nodos que se relacionan con un nodo raíz, ya sea analizando la cantidad de nodos que lo apuntan, los nodos a los que apunta, o una combinación de ambas métricas.
- *PageRank*: algoritmo para calcular la *importancia* de los nodos dentro de un grafo, calculando la popularidad de cada uno de sus vecinos y asignándoles el valor conjunto obtenido de estos mismos.
- *Betweenness centrality*: algoritmo para determinar la capacidad de un nodo para conectar distintas secciones de un grafo, de manera que este estaría fungiendo como puente.
- *Closeness centrality*: algoritmo para detectar nodos que son capaces de distribuir información de manera eficiente a través del grafo y hacia otros nodos.

La relevancia encontrada en los algoritmos de centralidad es la de determinar la importancia de los nodos dentro un grafo en base a distintas métricas y mediciones. En el contexto actual, los nodos se transforman en personajes y las relaciones en interacciones y acciones determinadas dentro de la línea histórica de la serie. La relevancia de un nodo en base a sus mediciones puede compararse al rol que el personaje correspondiente ha desempeñado en la temporada que se analiza, así como el impacto que ha dejado sobre otros. La comparación del grafo con el mundo real o el caso de uso que se analiza resulta importante para analizar cualquier conjunto de datos desconocidos y darle una importancia adicional, más allá de sus contenidos digitales.

Las siguientes consultas demuestran la funcionalidad y utilidad de los algoritmos previamente descritos, brindando información relevante dentro del contexto sobre el cual se analiza el grafo.

Nota: El proceso para obtener la información deseada ha sido omitido, puesto que el manejo de las funciones de GDS es un tanto tedioso y extenso, no obstante, todo esto ha sido incluido

como anexo dentro de los archivos adjuntos a este reporte, donde se detalla paso por paso lo requerido para calcular los datos relevantes.

1. ¿Qué personaje tiene el mayor número de interacciones con otras personas?

```
MATCH (n)
RETURN n.Label AS Personaje, toInteger(n.DEGREE) AS DEGREE
ORDER BY n.DEGREE DESC
LIMIT 1;

// Alternativa
MATCH ()-[r]-(n)
RETURN n.Label AS PERSONAJE, COUNT(r) AS DEGREE
ORDER BY DEGREE DESC
LIMIT 1;
```

Personaje

"Robb"

2. ¿Qué personaje tiene el mayor número de personas que se relacionan con el mismo?

```
MATCH (n)
RETURN n.Label AS Personaje, toInteger(n.IN_DEGREE) AS IN_DEGREE
ORDER BY n.IN_DEGREE DESC
LIMIT 1;

// Alternativa
MATCH ()-[r]->(n)
RETURN n.Label AS Personaje, COUNT(r) AS IN_DEGREE
ORDER BY IN_DEGREE DESC
LIMIT 1;
```

Personaje

"Tywin"

3. ¿Qué personaje tiene el *menor* número de relaciones con otras personas?

```
MATCH (n)
RETURN n.Label AS Personaje, toInteger(n.OUT_DEGREE) AS OUT_DEGREE
ORDER BY n.OUT_DEGREE
```

```
LIMIT 1;
```

Personaje
"Drogo"

4. Top 5 de personajes con más interacciones

```
MATCH (n)
RETURN n.Label AS Personaje, toInteger(n.DEGREE) AS DEGREE
ORDER BY n.DEGREE DESC
LIMIT 5;
```

```
// Alternativa
```

```
MATCH ()-[r]-(n)
RETURN n.Label AS Personaje, COUNT(r) AS DEGREE
ORDER BY DEGREE DESC
LIMIT 5;
```

Personaje
"Robb"
"Tywin"
"Tyrion"
"Catelyn"
"Joffrey"

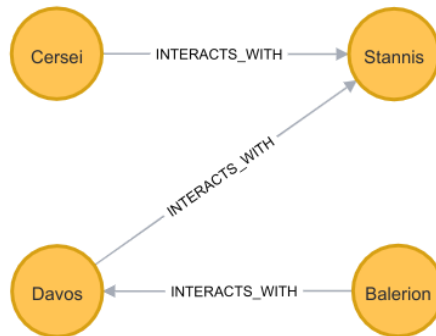
5. La línea de relación entre el personaje con la mayor cantidad de relaciones hacia otras personas y el personaje con menos relaciones

```
CALL {
  MATCH (n) RETURN n.Label AS Personaje
  ORDER BY n.OUT_DEGREE DESC LIMIT 1
} WITH Personaje AS PMR
CALL {
  MATCH (n) RETURN n.Label AS Personaje
```

```

ORDER BY n.DEGREE LIMIT 1
} WITH Personaje AS PLR, PMR AS PMR
MATCH (m), (n)
WHERE m.Label = PMR
AND n.Label = PLR
RETURN shortestPath((m)-[*]-(n));

```



6. ¿Qué personaje tiene más influencia/popularidad en esta temporada?

```

MATCH (n)
RETURN n.Label AS Personaje, ROUND(n.PageRank) AS PageRank
ORDER BY PageRank DESC
LIMIT 1;

```

Personaje

"Tywin"

7. ¿Cuáles son los 5 Starks más influyentes/populares?

```

MATCH (m)
WHERE m.Label IN ['Sansa', 'Robb', 'Jon', 'Arya', 'Ned', 'Catelyn', 'Rickon']
RETURN m.Label AS Personaje, ROUND(m.PageRank) AS PageRank
ORDER BY PageRank DESC
LIMIT 5;

```

	Personaje
1	"Catelyn"
2	"Jon"
3	"Ned"

4	"Rickon"
5	"Arya"

8. ¿Qué personajes resultan más influyentes en el grafo al aplicar un damping factor de 0.6 y 30 iteraciones?

```
MATCH (n)
RETURN n.Label AS Personaje, ROUND(n.PageRank_MOD) AS PageRank_MOD
ORDER BY PageRank_MOD DESC
LIMIT 5;
```

Personaje
"Tywin"
"Walder"
"Tyrion"
"Sam"
"Varys"

9. ¿Qué personajes se conectan entre varios grupos de personas?

```
MATCH (n)
RETURN n.Label AS Nombre, ROUND(n.Betweenness, 4) AS Score
ORDER BY Score DESC
LIMIT 5;
```

Nombre	Score
"Robb"	374.4399
"Jon"	232.4
"Catelyn"	207.1838
"Rickon"	188.3238
"Ned"	184.0399

10. ¿Qué personaje no tendría una eficacia óptima para hacer pactos entre otros grupos de personas?

```

MATCH (n)
WHERE n.Betweenness > 0
RETURN n.Label AS Nombre, ROUND(n.Betweenness, 4) AS Score
ORDER BY Score
LIMIT 5;
// Se revisan personajes que sí tienen al menos una baja posibilidad

```

Nombre	Score
"Mero"	0.3333
"Lothar"	0.5
"Grey Worm"	0.5
"Joyeuse"	0.5
"Talisa"	0.75

11. ¿Qué personaje podría mandar mensajes de mejor manera?

```

MATCH (n)
RETURN n.Label AS Personaje
ORDER BY n.Closeness DESC
LIMIT 1;

```

Personaje

"Arya"

12. ¿Cuál es la línea de relación entre el personaje con menor posibilidad de comunicar mensajes y Arya (Stark)?

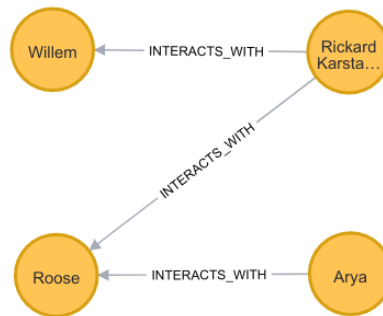
```

CALL {
  MATCH (n)
  WHERE n.Closeness <> 0
  RETURN n.Label AS Personaje
  ORDER BY n.Closeness
  LIMIT 1
}
WITH Personaje AS LME
MATCH (n), (m)
WHERE n.Label = LME
AND m.Label = 'Arya'

```



```
RETURN shortestPath((n)-[*]-(m));
```



13. Tomando en cuenta su habilidad de relacionar grupos y comunicar mensajes ¿Cuáles son los 5 personajes más relevantes?

```
MATCH (n)
RETURN n.Label AS Personaje,
       ROUND(n.Betweenness, 4) AS Betweenness,
       ROUND(n.Closeness, 4) AS Closeness
ORDER BY Betweenness DESC, Closeness DESC
LIMIT 5;
```

Personaje	Betweenness	Closeness
"Robb"	374.4399	0.5368
"Jon"	232.4	0.5862
"Catelyn"	207.1838	0.875
"Rickon"	188.3238	0.5333
"Ned"	184.0399	0.4923

14. Tomando en cuenta su influencia y sus relaciones ¿Cuáles son los 5 personajes más relevantes?

```
MATCH (n)
RETURN n.Label AS Personaje,
       ROUND(n.PageRank, 4) AS PageRank,
       ROUND(n.DEGREE, 4) AS DEGREE
ORDER BY PageRank DESC, DEGREE DESC
LIMIT 5;
```

Personaje	PageRank	DEGREE
"Walder"	3.4195	15.0
"Tywin"	3.3013	29.0
"Varys"	2.7299	14.0
"Tyrion"	1.561	25.0
"Sam"	1.4951	17.0

15. Tomando en cuenta todas las estadísticas anteriores ¿Cuáles son los 5 personajes más relevantes?

```
MATCH (n)
RETURN n.Label AS Personaje,
       ROUND(n.DEGREE, 4) AS DEGREE,
       ROUND(n.PageRank, 4) AS PageRank,
       ROUND(n.Betweenness, 4) AS Betweenness,
       ROUND(n.Closeness, 4) AS Closeness
ORDER BY DEGREE DESC, PageRank DESC,
         Betweenness DESC, Closeness DESC
LIMIT 5;
```

Personaje	DEGREE	PageRank	Betweenness	Closeness
"Robb"	31.0	0.8674	374.4399	0.5368
"Tywin"	29.0	3.3013	50.1184	0.4706
"Tyrion"	25.0	1.561	31.6203	0.4792
"Catelyn"	22.0	0.2921	207.1838	0.875
"Joffrey"	21.0	0.3093	129.6565	0.4651

Observaciones

Como ya se mencionaba en la introducción, el uso de frases comprensibles para el público general se ha empleado como método de comunicación dentro del análisis de centralidad. Las preguntas planteadas originalmente lucían como lo siguiente.

#	Pregunta original	Pregunta mostrada
1	¿Qué nodo tiene el mayor número de relaciones con otros nodos?	¿Qué personaje tiene el mayor número de interacciones con otras personas?
2	¿Qué nodo tiene el mayor número de relaciones entrantes desde otros nodos?	¿Qué personaje tiene el mayor número de personas que se relacionan con el mismo?
3	¿Qué nodo tiene el <i>menor</i> número de relaciones con otros nodos?	¿Qué personaje tiene el <i>menor</i> número de relaciones con otras personas?
4	Los 5 nodos con mayor cantidad de relaciones indistintas	Top 5 de personajes con más interacciones
5	El camino más corto entre el nodo con la mayor cantidad de relaciones hacia otros nodos y el nodo con la menor cantidad de relaciones indistintas	La línea de relación entre el personaje con la mayor cantidad de relaciones hacia otras personas y el personaje con menos relaciones
6	¿Qué nodo tiene mayor relevancia dentro del grafo?	¿Qué personaje tiene más influencia/popularidad en esta temporada?
7	¿Cuáles son los 5 nodos de la familia Stark con mayor relevancia dentro del grafo?	¿Cuáles son los 5 Starks más influyentes/populares?
8		
9	¿Qué nodos tienen mayor posibilidad de formar puentes entre varios grupos de otros nodos?	¿Qué personajes se conectan entre varios grupos de personas?
10	¿Qué nodos no tendrían una cualidad óptima para realizar relaciones entre grupos de otros nodos?	¿Qué personaje no tendría una eficacia óptima para hacer pactos entre otros grupos de personas?
11	¿Cuál nodo sería el más óptimo para distribuir información a través del grafo?	¿Qué personaje podría mandar mensajes de mejor manera?
12	¿Cuál es el camino más corto entre el nodo menos óptimo para distribuir información a través del grafo y el nodo con etiqueta "Arya"?	¿Cuál es la línea de relación entre el personaje con menor posibilidad de comunicar mensajes y Arya?

13	Tomando en cuenta sus cualidades de conexión de distintas partes del grafo y distribución de información ¿Cuáles son los 5 nodos más relevantes?	Tomando en cuenta su habilidad de relacionar grupos y comunicar mensajes ¿Cuáles son los 5 personajes más relevantes?
14	Tomando en cuenta su relevancia dentro del grafo y sus relaciones con dirección indistinta ¿Cuáles son los 5 nodos más relevantes?	Tomando en cuenta su influencia y sus relaciones ¿Cuáles son los 5 personajes más relevantes?
15		

Comparativa entre resultados y contexto de la temporada

La mejor manera de darle sentido a los resultados obtenidos sin duda alguna es viendo toda la serie hasta este punto, es decir, hasta su tercera temporada; y claro, continuar por el contenido del resto de sus 5 temporadas posteriores si así se desea. Sin embargo, con una duración de poco más de 9 horas tan solo para la tercera temporada, puede resultar agobiante la visualización de este material, especialmente si el género fantasioso que lo caracteriza no resulta del particular agrado de algún individuo.

Para el interés del desarrollo actual, se propone el siguiente resumen de la tercera instancia de este programa televisivo, presentando los personajes y sucesos más relevantes de la temporada.

Contexto

El tema principal de la serie se centra alrededor de la búsqueda del control de un “trono de hierro” para reinar a siete reinos a lo largo de “Westeros”, territorio donde se desarrolla la mayoría de los conflictos pivotaes y espacio que alberga dichos reinos en distintas regiones.

En la tercera temporada, la lucha por el trono se intensifica en comparación de las pasadas, encontrando puntos conflictivos como la “guerra de los cinco reyes” o la “boda roja”, en donde distintos reinos entran en conflicto y algunas alianzas se ven comprometidas a través de la traición.

También, existe una región de “forasteros” conocida como “la pared”, una división conformada por hielo que divide a los habitantes de Westeros y demás entidades detrás de esta barrera. Los habitantes que residen al otro lado de la pared son considerados como amenazas, ya que entre estos se encuentran los “caminantes blancos”, entidades sobrenaturales con capacidades de nigromancia.

Personajes y comparativa

- Robb Stark: el rey del norte (asesinado en la boda roja). A pesar de tener el grado más alto del grafo y de ser sin duda uno de los personajes principales de la serie, tiene uno

de los PageRank más bajos de la familia Stark (incluso debajo de Ned Stark, que lleva 2 temporadas muerto); esto indica que los personajes con los que interactúa en esta temporada no resultan demasiado relevantes.

- Sansa: hermana mayor de la familia Stark. A pesar de estar presente tanto en la temporada actual como en el set de datos, no tiene demasiada relevancia, lo cual puede ser corroborado con el análisis realizado.
- Tyrion Lannister: la mano del rey Joffrey y agente de la paz en la capital del reino “King’s Landing”, pero a las espaldas es menospreciado por su familia por su enanismo. A pesar de esto, los algoritmos de centralidad reflejan su elevada importancia en esta temporada.
- Joffrey Baratheon: el rey de los siete reinos, un puesto de supuesto prestigio, sin embargo, los análisis de centralidad muestran que tanto Tywin Lannister (activo en la boda roja) como Tyrion Lannister aparecen arriba de él en estadísticas de centralidad. Estas comparaciones y sobrepasos reflejan el contenido de la temporada en cuestión de apariciones.
- Daenerys Targaryen: una princesa exiliada de Westeros que busca recuperar el trono (también llamada “madre de dragones”). En la serie tiene muchos minutos en pantalla, pero no aparece en ningún punto del análisis, demostrando su relación aislada comparada con el resto de personajes, como su muestra en la serie.
- Jon Snow: un infiltrado dentro del grupo de forasteros conocidos como “gente libre” que habitan más allá de la pared, con la misión de aprender sus hábitos para detener posibles amenazas. La situación que refleja en la serie es la de vincular a 2 grupos, por lo cual figura entre los primeros resultados para el algoritmo de betweenness.
- Ygritte: una guerrera/arquera forastera parte de la gente libre, por la cual Jon Snow desarrolla un interés amoroso (aún cuando este no sabe nada). Es con ella que el mismo personaje comienza a forjar alianzas, demostrando la formación de “puentes” entre varias partes del grafo.

Evidentemente, estos son solo algunos de la gran variedad de personajes con los que cuenta la temporada actual y la serie en general, no obstante, las comparaciones demostradas son útiles para demostrar cómo la construcción de un fenómeno, ya sea ficticio o del mundo real, puede ser representado de distintas formas y a través de múltiples estructuras de datos con el propósito de aplicar técnicas analíticas que usualmente no podrían ser implementadas en el estado puro del conjunto de datos.

Conclusiones

Llegada la parte final de este desarrollo, el aprendizaje obtenido es invaluable y la forma tan práctica de ponerlo a prueba es sin duda alguna una de las mejores herramientas pedagógicas, puesto que la necesidad del análisis, así como su propuesta y respuesta, han sido planteados por el mismo equipo que ha llevado a cabo y ha hecho posible la realización de este documento.

Pudiera parecer tedioso continuar con la misma dinámica de la exploración inicial de grafos, como la que se tuvo la oportunidad de realizar en esta ocasión, sin embargo, no hay palabras suficientes para indicar la relevancia de este paso tan importante al tratar con un grafo desconocido. Este proceso puede ser abstraído al grado en que todo aspecto, objeto, tópico o cualquier elemento desconocido con el cual se desee trabajar, debe ser conocido y comprendido al grado de familiaridad y suficiente comodidad al manipularle.

Para el caso actual, ambos análisis conducidos fueron realizados a través de la misma herramienta, pero, no hay que perder de vista que esto no ha sido sino una forma más de abordar y dar solución a la misma necesidad.

Por la parte de la significancia interpretada de ambos análisis, es de suma importancia mencionar la múltiple utilidad encontrada al realizar estos proyectos, porque no solo ha servido como método de asentamiento de las bases necesarias para navegar por el mundo de la minería de grafos, puesto que también se ha dado una interpretación desde el punto de vista de la cultura y referencia popular, en donde los conceptos que podrían verse aparentemente difusos para una gran parte de las personas, en realidad es una forma solo un poco más formal de un fenómeno tan grande como lo fue la serie analizada.

Ahora, por otra parte, y haciendo referencia a lo último mencionado, el conjunto de datos sobre el cual se ha trabajado es lo suficientemente bueno para presentar la aplicación de los algoritmos de centralidad aprendidos en el devenir de la materia, no obstante, tanto los nodos facilitados como las relaciones entre ellos no contaban con más que un atributo para un solo par de componentes dentro del grafo, es decir, un solo tipo de nodo y de relación. Para el alcance propuesto esto es más que suficiente, pero posiblemente ha sido una tarea compleja darle sentido al significado que almacenan los nodos en relación con la serie que intentan representar. De igual manera, posiblemente si se desea abordar desde otro punto de vista o para análisis de algún otro tipo, esto sería poco interesante o quizás completamente imposible; a pesar de esto, para reafirmar lo mencionado, los datos satisfacen las necesidades previas a la realización del desarrollo y con ello basta por el momento.

Finalmente, se hace hincapié en la relevancia de este tipo de proyectos y sobre todo en su extensión, puesto que, a pesar de no contar con todo el material visitado en el curso, es sin duda algo útil para consolidar el que se ha visto hasta el momento.

Bibliografía

Benioff, D., Weiss, D. B. (Productores Ejecutivos). (2011-2019) *Game of Thrones* [Serie de televisión]. Home Box Office; HBO Entertainment.

Neo4j. (s.f.). *Neo4j Graph Database*. Recuperado de <https://neo4j.com/product/neo4j-graph-database/>.

de Jong, N., Conjeaud, M., Agudelo Ramirez, H., et al. (s.f.). *NeoDash - Dashboard Builder for Neo4j*. Recuperado de <https://neo4j.com/labs/neodash/>.

Neo4j. (s.f.). *The Neo4j Graph Data Science Library Manual v2.4*. Recuperado de <https://neo4j.com/docs/graph-data-science/current/>.

Frame, A., Blumenfeld, Z. (2022). *Graph Data Science For Dummies*. John Wiley & Sons, Inc..

Ugurlu, O. (2022). Comparative analysis of centrality measures for identifying critical nodes in complex networks. *Journal of Computational Science*, 62, 101738. <https://doi.org/10.1016/j.jocs.2022.101738>.

Needham, M., & Hodler, A. E. (2021). *Graph algorithms: Practical examples in Apache Spark and neo4j*. O'Reilly.

J. RafieeFard, B. Teimourpour, M. Shaghrouzi and M. Jami, "A Comparison of Centrality Measures for Community Evolution Discovery," *2022 8th International Conference on Web Research (ICWR)*, Tehran, Iran, Islamic Republic of, 2022, pp. 183-188, doi: 10.1109/ICWR54782.2022.9786228.

Neo4j. (s.f.). *Cypher Query Language*. Recuperado de <https://neo4j.com/developer/cypher/>.