Ingeniería en Sistemas Computacionales

Minería de Grafos

Mtro. Víctor Hugo Ortega Guzmán

Práctica 1: Analizando la información contenida en una base de datos basada en grafos

Marco Ricardo Cordero Hernández

Tlaquepaque, Jal., 11 de septiembre de 2023

Índice

Introducción

A lo largo de la historia, las matemáticas, enraizadas en la filosofía, han traído consigo una inmensa cantidad de aplicaciones que a día de hoy se sigan utilizando, incluso haciendo posible la creación y visualización de este documento. Un campo de estudio con particular interés multidisciplinario es aquel de la teoría de grafos, encontrando su nacimiento en el deseo de resolver un acertijo, en donde se deseaba saber si era posible atravesar 7 puentes encontrados a través de una isla, con la restricción de pasar una sola ocasión por cada uno de ellos. El legendario matemático Leonhard Euler demostraría que esto es imposible, e, inadvertidamente, probaría el primer teorema de la teoría de grafos, y consigo, su nacimiento (Carlson, 2023).

En la actualidad, la aplicación de la teoría de grafos dista de contar con la sola posibilidad de demostrar cosas triviales, sino que ahora le da soporte a cosas tan elementales para el estilo de vida modernista como lo son las redes (The Network Pages, s.f.), incluso haciendo posible la reducción de latencia entre componentes de red al optimizar estas redes (Nakao et al., 2021).

A pesar de lo fascinante que puedan parecer estas aplicaciones, no es posible pasar de la teoría a la práctica sin conocer al menos un poco de los conceptos fundamentales que dan soporte a lo que ha llegado a convertirse este campo de estudio. Un elemento crítico, usualmente ignorado por la enseñanza tradicional de esta línea del conocimiento, es el *diámetro de un grafo*, concepto definido como la longitud del camino más corto entre el par de nodos con mayor distancia entre ellos (The Geography of Transport Systems, 2023); en una definición complementaria, se le puede ver como la longitud del camino más largo y corto entre dos vértices (Weisstein, s.f.). La relevancia de este dato puede no parecer aparente en primera instancia, sin embargo, la utilización de este elemento, por ejemplo, en redes de distribución de contenido (CDN), es algo sumamente crítico para respuestas rápidas desde clientes web con fines indistintos, pero iguales de relevantes.

Pero ¿De qué sirve el diámetro de un grafo si poco se conoce de sus componentes? El propósito de esta práctica es el de realizar un análisis inicial y sistemático sobre un grafo previamente construido, de manera que, sin saber ninguno de sus datos iniciales, finalmente se proporcionen estadísticas y datos relevantes para el mismo conjunto. La manera en que se pretende lograr esto es a través del software de bases de datos *orientadas a grafos* Neo4j (s.f.), una herramienta que además de proporcionar a sus usuarios con herramientas para definición y manipulación de datos como alternativas a las bases de datos relacionales, es idónea para conducir tareas relacionadas a la ciencia de datos orientada a grafos.

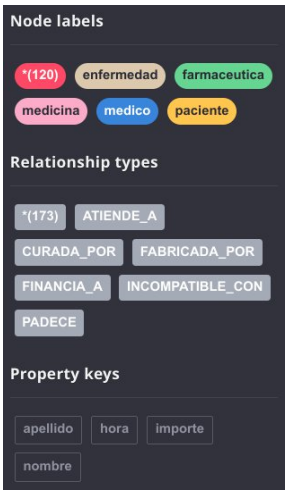
Al finalizar esta práctica, se pretende que se haya logrado el objetivo de contar con el conocimiento y métodos necesarios para extraer información preliminar de un grafo sin previa conciencia del mismo. Este será el punto de partida para comenzar a desarrollar habilidades fundamentales para futura referencia en análisis más profundos sobre grafos mucho más extensos.

Desarrollo

Dado un archivo contenedor del modelo del grafo (*Hospital.cql*), la siguiente serie de consultas son resueltas de forma que al final se contará con un análisis fuertemente fundamentado, capaz de ofrecer un elevado nivel de detalle correspondiente al contexto actual.

El resultado de importar los datos directamente desde el navegador de Neo4j es el siguiente:

```
Added 120 labels, created 120 nodes, set 255 properties, created 173 relationships, completed after 1033 ms.
```

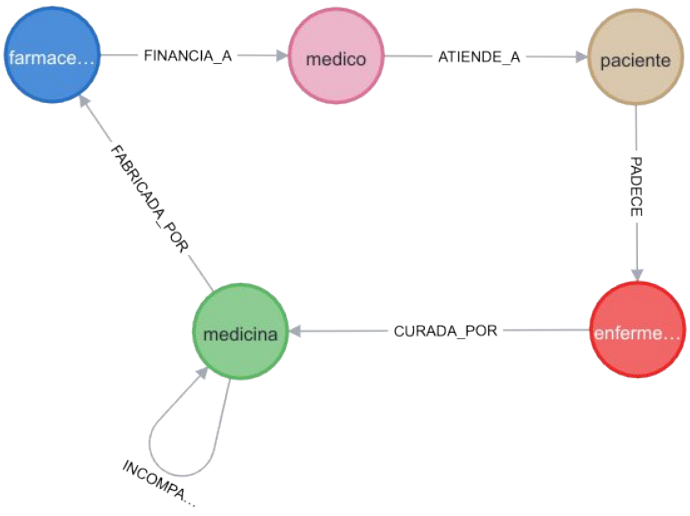


Esto puede verse como un punto de referencia para corroborar datos iniciales, sin embargo, dista de ser un análisis completo, ni siquiera suficiente.

Las siguientes consultas pretenden otorgar un análisis del esquema recientemente importado.

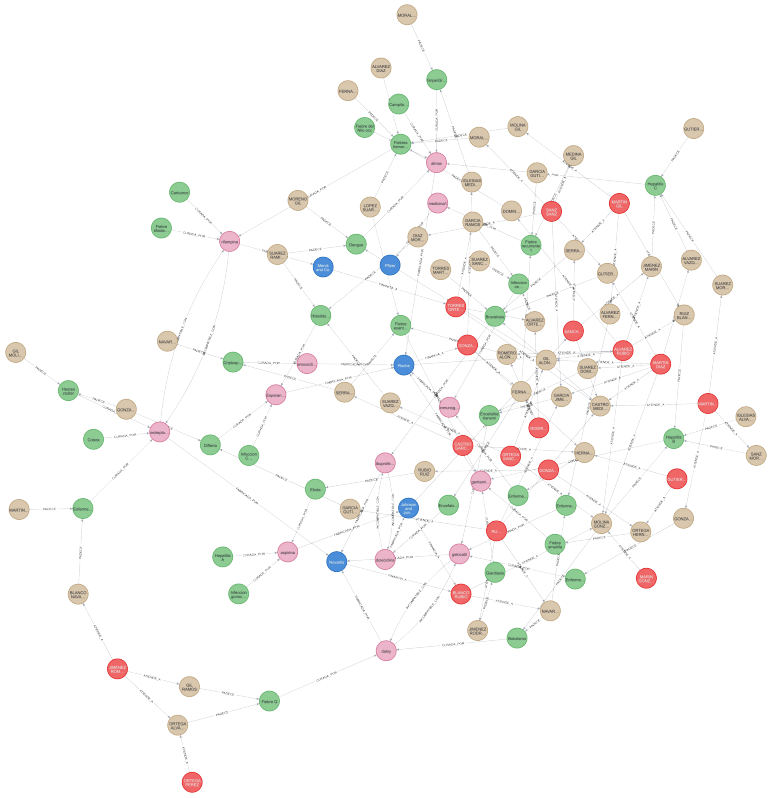
4.1. Una imagen del esquema general del grafo.

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



4.2. Una imagen del grafo completo.

```
MATCH (n)-[r]->(m) RETURN n, r, m;
// Alternativa: MATCH p = ()-->() RETURN p
```



4.3. ¿Cuáles son los tipos de nodos y sus atributos?

```
CALL db.labels(); // Cuáles nodos
MATCH (n) RETURN COUNT(n) as totalNodos; // Cuántos nodos
CALL db.schema.nodeTypeProperties() // Atributos y tipos
   YIELD nodeType AS Nodo, propertyName AS Propiedad, propertyTypes AS Tipo;
```

NODOS
"medico"
"paciente"
"enfermedad"
"farmaceutica"
"medicina"

totalNodos
120

Nodo	Propiedad	Tipo
":`paciente`"	"apellido"	["String"]
":`paciente`"	"nombre"	["String"]
":`enfermedad`"	"nombre"	["String"]
":`farmaceutica`"	"nombre"	["String"]
":`medicina`"	"nombre"	["String"]
":`medico`"	"apellido"	["String"]
":`medico`"	"nombre"	["String"]

4.4. ¿Cuáles son los tipos de relaciones y sus atributos?

```
CALL db.schema.relTypeProperties() // Cuáles relaciones y tipos
YIELD relType AS Relacion, propertyName AS Propiedad, propertyTypes AS Tipo;
MATCH ()-[r]->() RETURN COUNT(r) as totalRelaciones; // Cuántas relaciones
```

Relacion	Propiedad	Tipo
":`ATIENDE_A`"	"hora"	["String"]
":`PADECE`"	*null*	*null*
":`CURADA_POR`"	*null*	*null*
":`FABRICADA_POR`"	*null*	*null*
":`INCOMPATIBLE_CON`"	*null*	*null*
":`FINANCIA_A`"	"importe"	["Long"]

totalRelaciones
173

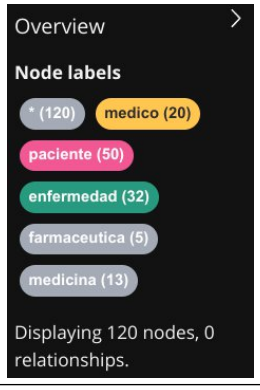
4.5. Una consulta en cypher que me indique cuantos nodos hay de cada tipo.

```
MATCH (n) RETURN DISTINCT LABELS(n) AS Nodo, COUNT(n) AS Total;
```

Nodo	Total
["medico"]	20
["paciente"]	50
["enfermedad"]	32
["farmaceutica"]	5
["medicina"]	13

4.6. ¿Cómo demuestras que es correcto el resultado de la consulta anterior?

```
MATCH (n) RETURN n;
```



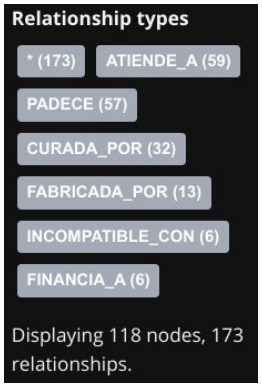
4.7. Una o varias consultas en cypher que me indique cuantas relaciones hay de cada tipo.

```
MATCH ()-[r]->() RETURN TYPE(r) as Relacion, COUNT(r) AS Total;
```

Relacion	Total
"ATIENDE_A"	59
"PADECE"	57
"CURADA_POR"	32
"FABRICADA_POR"	13
"INCOMPATIBLE_CON"	6
"FINANCIA_A"	6

4.8. ¿Cómo demuestras que es correcto el resultado de la consulta anterior?

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



4.9. ¿Cuál es el diámetro del grafo?

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

// No dirigido
```

```
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 DiametroNoDirigido
ORDER BY distance DESC
LIMIT 1;
```

DiametroDirigido
17

DiametroNoDirigido
11

4.10. Explica con tus palabras para que te sirve conocer el diámetro del grafo

La obtención e interpretación de este dato es igual de relevante que conocer cualquier dato primitivo de un grafo. Su conocimiento es vital para analizar la eficiencia del mismo, y, cuando se traducen hacía aplicaciones reales como lo podrían ser redes de comunicaciones o redes neuronales, es importante su obtención para determinar la cantidad de recursos necesaria para llegar desde un nodo céntrico hacía uno de extremo, lo cual, al realizarle un análisis independiente, podría determinar estadísticas como latencia, tiempo consumido, etc.

4.11. Un listado de los médicos, el número de pacientes que ha atendido, como lista el nombre de los pacientes y como lista el apellido de los pacientes. El resultado debe estar ordenado en forma descendente con base en el número de pacientes que ha atendido que ha atendido cada médico. ***(Resultados divididos para mejor visualización)***

```
MATCH (m:medico)-[r:ATIENDE_A]->(p:paciente)
RETURN m.nombre + ' ' + m.apellido AS NombreMedico, COUNT(p) AS TotalAtendidos,
COLLECT(p.nombre) AS NombrePaciente, COLLECT(p.apellido) AS ApellidoPaciente;
```

NombreMedico	TotalAtendidos	NombrePaciente	ApellidoPaciente
"Roberto JIMENEZ ROMERO"	3	["Lucio", "Federico", "Roberta"]	["ORTEGA ALVAREZ", "BLANCO NAVARRO", "GIL RAMOS"]
"Bienvenido GARRIDO RUBIO"	5	["Juan", "Juana", "Roberta", "Joaquina", "Victoria"]	["CASTRO MEDINA", "JIMENEZ RODRIGUEZ", "GARCIA JIMENEZ", "GARCIA GUTIERREZ", "NAVARRO MARIN"]
"Bienvenida ORTEGA SANCHEZ"	3	["Manuel", "Juana", "Pedro"]	["SUAREZ VAZQUEZ", "GIL ALONSO", "MOLINA GONZALEZ"]
"Joaquina RODRIGUEZ MARTIN"	3	["Roberto", "Pedro", "Roberta"]	["ROMERO ALONSO", "ALVAREZ ORTEGA", "FERNANDEZ ORTEGA"]
"Roberto BLANCO RUBIO"	2	["Pedro", "Victoria"]	["MOLINA GONZALEZ", "NAVARRO MARIN"]
"Lucio CASTRO GARCIA"	2	["Lucio", "Manuel"]	["HERNANDEZ BLANCO", "SERRANO SANCHEZ"]
"Roberta MARTINEZ GARRIDO"	4	["Juan", "Bienvenida", "Manuela", "Victoria"]	["CASTRO MEDINA", "SUAREZ MORALES", "GONZALEZ JIMENEZ", "SANZ MORENO"]
"Bienvenida GONZALEZ ALONSO"	2	["Juana", "Manuela"]	["RUBIO RUIZ", "ORTEGA HERNANDEZ"]
"Roberto ALVAREZ RUBIO"	6	["Roberto", "Roberto", "Juana", "Roberto", "Pedro", "Joaquina"]	["ROMERO ALONSO", "RUIZ BLANCO", "GIL ALONSO", "DOMINGUEZ RUIZ", "MOLINA GONZALEZ", "JIMENEZ MARIN"]
"Federica SANZ SANZ"	5	["Roberta", "Bienvenido", "Roberta", "Roberto", "Roberta"]	["GARCIA JIMENEZ", "DIAZ MORALES", "GUTIERREZ IGLESIAS", "MEDINA GIL", "MORALES HERNANDEZ"]

"Victoria MARIN GONZALEZ"	1	["Pedro"]	["MOLINA GONZALEZ"]
"Victoria MARTIN GIL"	4	["Federica", "Pedro", "Roberto", "Juana"]	["MOLINA GIL", "ALVAREZ FERNANDEZ", "RUIZ BLANCO", "SERRANO ALVAREZ"]
"Pedro GONZALEZ IGLESIAS"	3	["Lucio", "Joaquina", "Roberta"]	["HERNANDEZ BLANCO", "IGLESIAS MEDINA", "FERNANDEZ ORTEGA"]
"Juana MARTIN DIAZ"	7	["Juana", "Juan", "Roberta", "Roberta", "Roberto", "Bienvenida", "Pedro"]	["GIL ALONSO", "CASTRO MEDINA", "GARCIA JIMENEZ", "GUTIERREZ IGLESIAS", "ALVAREZ VAZQUEZ", "SUAREZ DOMINGUEZ", "MOLINA GONZALEZ"]
"Joaquina SANCHEZ RAMOS"	3	["Juan", "Joaquina", "Juana"]	["CASTRO MEDINA", "JIMENEZ MARIN", "SERRANO ALVAREZ"]
"Roberta TORRES ORTEGA"	3	["Pedro", "Roberto", "Juan"]	["ALVAREZ ORTEGA", "GARCIA RAMOS", "CASTRO MEDINA"]
"Federica GUTIERREZ MARTINEZ"	2	["Lucio", "Manuela"]	["HERNANDEZ BLANCO", "ORTEGA HERNANDEZ"]
"Victoria ORTEGA PEREZ"	1	["Lucio"]	["ORTEGA ALVAREZ"]

4.12. Una consulta que reciba el nombre del médico y muestre una tabla con el médico, el número de pacientes que ha atendido, la lista de pacientes y el número de enfermedades. Utiliza al médico que más pacientes ha atendido como valor del parámetro para probar tu consulta.

```
// Médico con mayor atendidos
MATCH (m:medico)-[:ATIENDE_A]->(p:paciente)
RETURN m.nombre AS Nombre, m.apellido AS Apellido, COUNT(p) AS Atendidos
ORDER BY Atendidos DESC
LIMIT 1;


// Consulta parametrizada
MATCH (m:medico)-[a:ATIENDE_A]->(p:paciente)-[pd:PADECE]->(e:enfermedad)
WITH m, p, pd, e, p.nombre + ' ' + p.apellido AS nombrePaciente
WHERE toUpper(m.nombre) = toUpper(:nombre)
AND m.apellido = toUpper(:apellido)
RETURN m.nombre + ' ' + m.apellido AS NombreMedico, COUNT(DISTINCT p) AS TotalAtendidos,
COLLECT(DISTINCT nombrePaciente) AS Pacientes, COUNT(DISTINCT e) AS TotalEnfermedades;
```

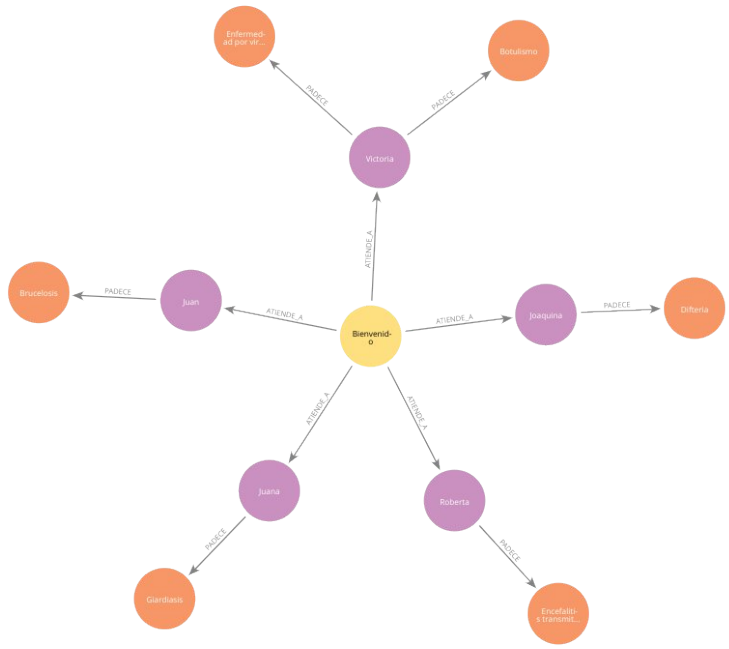
Nombre	Apellido	Atendidos
"Juana"	"MARTIN DIAZ"	7

NombreMedico	TotalAtendidos
"Juana MARTIN DIAZ"	7

Pacientes	TotalEnfermedades
["Juana GIL ALONSO", "Juan CASTRO MEDINA", "Roberta GARCIA JIMENEZ", "Roberta GUTIERREZ IGLESIAS", "Roberto ALVAREZ VAZQUEZ", "Bienvenida SUAREZ DOMINGUEZ", "Pedro MOLINA GONZALEZ"]	6

4.13. Realiza una frase de búsqueda en Bloom que reciba el nombre del médico como parámetro y te regrese, el grafo que incluya la relación con los pacientes, los pacientes, la relación con las enfermedades y las enfermedades. Prueba con un médico de tu elección como parámetro; Muestra el grafo resultante como par ordenado.

```
MATCH (m:medico)-[a:ATIENDE_A]->(p:paciente)-[pd:PADECE]->(e:enfermedad)
WHERE toUpper(m.nombre) = toUpper($NOMBRE)
AND m.apellido = toUpper($APELLIDO)
RETURN m, a, p, pd, e;
// Medico elegido: Bienvenido Garrido Rubio
```



(Bienvenido, Victoria)

(Bienvenido, Joaquina)

(Bienvenido, Roberta)

(Bienvenido, Juana)

(Bienvenido, Juan)

(Victoria, Enfermedad por virus Chikungunya)

(Victoria, Botulismo)

(Joaquina, Difteria)

(Roberta, Encefalitis transmitida por garrapatas)

(Juana, Giardiasis)

(Juan, Brucelosis)

4.14. Una consulta que muestre el IN_DEGREE de los nodos tipo medicina. El resultado debe estar ordenado de forma descendente por IN_DEGREE.

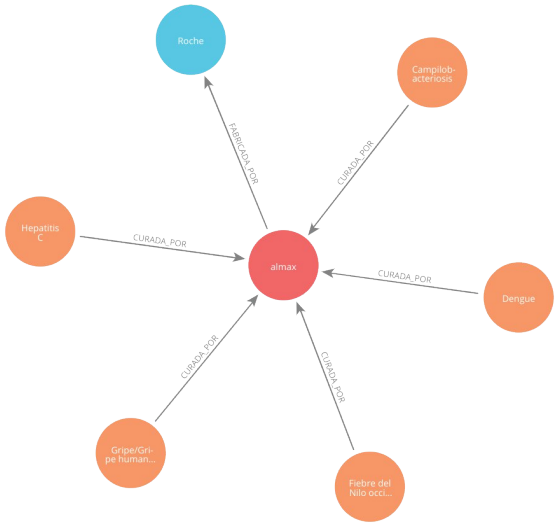
```
MATCH ()-[r:]->(:medicina) RETURN COUNT(r) AS IN_DEGREE;
```

IN_DEGREE
38

4.15. Realiza una frase de búsqueda en Bloom que reciba el nombre de una medicina como parámetro y regrese el grafo con el nombre de la medicina, la relación con la enfermedad, el nombre de las enfermedades, la relación con la farmacéutica y el nombre de la farmacéutica que la produce. Prueba con una medicina de tu elección como parámetro; Muestra el grafo resultante como una matriz.

```
MATCH r=(e:enfermedad)-[c:CURADA_POR]->(m:medicina)-[fb:FABRICADA_POR]->(f:farmaceutica)
WHERE m.nombre = $MEDICINA
RETURN r;
```

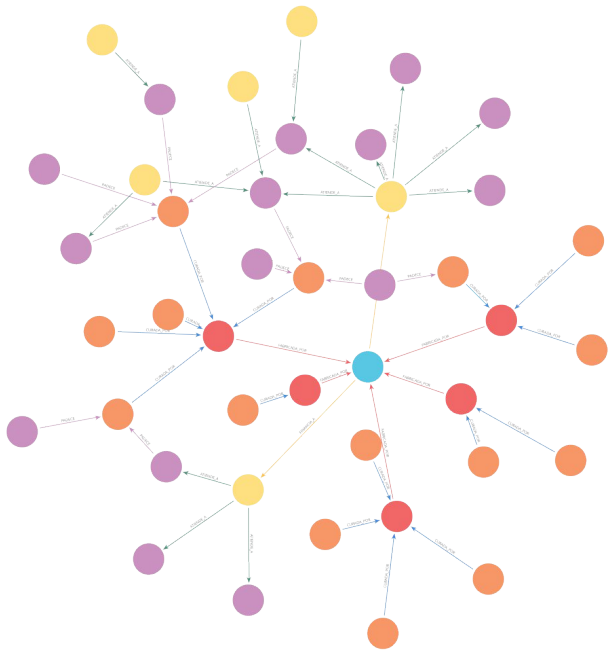
```
// Medicina elegida: Almax
```



	Almax	Roche	Campilobacteriosis	Dengue	Fiebre del Nilo occidental	Gripe/Gripe humana por un nuevo subtipo de virus	Hepatitis C
Almax	0	1	0	0	0	0	0
Roche	0	0	0	0	0	0	0
Campilobacteriosis	1	0	0	0	0	0	0
Dengue	1	0	0	0	0	0	0
Fiebre del Nilo occidental	1	0	0	0	0	0	0
Gripe/Gripe humana por un nuevo subtipo de virus	1	0	0	0	0	0	0
Hepatitis C	1	0	0	0	0	0	0

4.16. Realiza una frase de búsqueda en Bloom que reciba el nombre de una medicina y muestre el grafo, sin importar la dirección, que conecta el nodo inicio, las relaciones y los nodos destino a 3 saltos. Prueba con almax como entrada y muestra el grafo resultante.

```
MATCH (m:medicina)
MATCH r=(m)-[*1..3]-(n)
WHERE m.nombre = $MEDICINA
RETURN r;
```

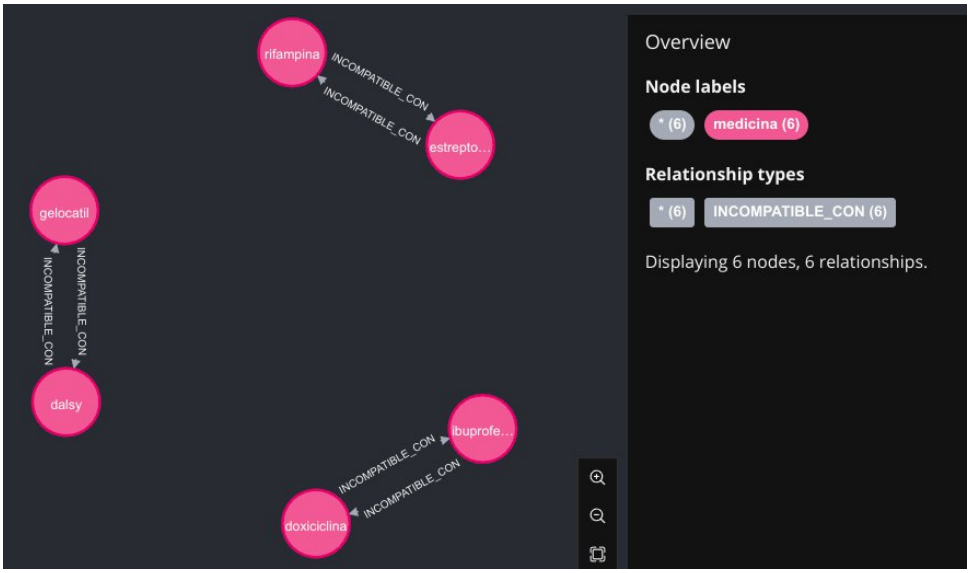


4.17. Realiza una consulta que muestre cuantos patrones (:Medicina) –[:INCOMPATIBLE]->(:Medicina) se presentan en la base de datos. ¿Cómo compruebas que tu resultado es correcto?

```
MATCH (:medicina)-[i:INCOMPATIBLE_CON]->(:medicina)
RETURN COUNT(i) AS Total;

// Comprobación
MATCH (m1:medicina)-[r]->(m2:medicina)
RETURN m1, r, m2;
```

Total
6



4.18. Una consulta que muestre el top 3 de nodos con mayor DEGREE.

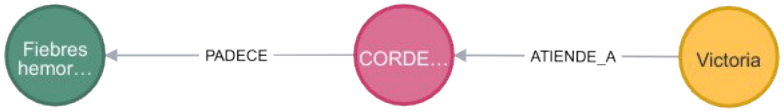
```
MATCH (n)-[r]-()
RETURN LABELS(n) AS Tipo, n.nombre AS nombre, n.apellido AS apellido, COUNT(r) AS DEGREE
ORDER BY DEGREE DESC
LIMIT 3;
```

Tipo	nombre	apellido	DEGREE
["paciente"]	"Pedro"	"MOLINA GONZALEZ"	7
["medico"]	"Roberto"	"ALVAREZ RUBIO"	7
["medico"]	"Juana"	"MARTIN DIAZ"	7

4.19. Crea los nodos y relaciones necesarias para que tú seas un paciente, te atienda el médico con menor OUT_DEGREE a causa de la enfermedad con mayor IN_DEGREE.

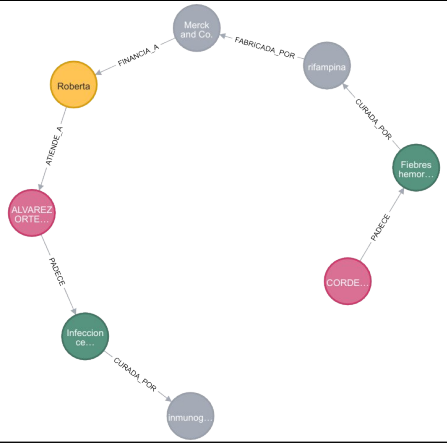
```
CALL {
    MATCH (m:medico)-[r]->()
    RETURN m.nombre AS Nombre, m.apellido AS Apellido, COUNT(r) AS OUT_DEGREE
    ORDER BY OUT_DEGREE
    LIMIT 1
}
WITH Nombre AS NombreMedico, Apellido as ApellidoMedico
CALL {
    MATCH ()-[r]->(e:enfermedad)
    RETURN e.nombre AS Nombre, COUNT(r) AS IN_DEGREE
    ORDER BY IN_DEGREE DESC
    LIMIT 1
}
WITH Nombre AS NombreEnfermedad, NombreMedico, ApellidoMedico
MATCH (m:medico {nombre: NombreMedico, apellido: ApellidoMedico}), (e:enfermedad {nombre: NombreEnfermedad})
MERGE (p:paciente {nombre: 'Marco', apellido: 'CORDERO HERNÁNDEZ'})
MERGE (m)-[:ATIENDE_A {hora: toString(time())}]->(p)-[:PADECE]->(e);
```

Added 1 label, created 1 node, set 3 properties, created 2 relationships, completed after 235 ms.



4.20. Muestra la ruta más corta entre un paciente y la medicina de tu elección.

```
MATCH (p:paciente), (m:medicina)
WHERE p.apellido = 'CORDERO HERNÁNDEZ'
AND m.nombre = 'inmunoglobulina antitoxina'
RETURN shortestPath((p)-[*]->(m));
```



Conclusiones y recomendaciones

Como ya se advertía al inicio de este desarrollo, el conjunto de datos encontrados en el grafo analizado es relativamente pequeño. La relatividad de esto puede demostrarse de dos maneras: comparando el tamaño de los datos en contra de una base que de soporte a una red social, en donde los tipos de nodos por lo menos se triplicarían, y la cantidad de datos quedaría muy por encima de los trabajados; otra manera de abordar este tópico es la comparativa desde el modelado de aeropuertos, en donde no se modela la interacción de personas (al menos no explícitamente), pero sí se ve la interconexión aérea entre ciudades, algo que también estaría dejando muy por atrás el análisis actual.

Lo ya mencionado no tiene el objetivo de demeritar o despreciar el ejercicio llevado a cabo, al contrario, no es sino para demostrar que tanta información puede obtenerse de un grafo diminuto, que aún con su tamaño, supone una extensa serie de recopilaciones de datos para llegar a comprenderlo en menor o mayor medida.

Resulta complejo abordar el análisis general de un grafo sin un objetivo claro más que el de conocer su estructura, sin embargo, a pesar de su magnitud, los datos vistos podrían ser sujeto de análisis más complejos, como la disparidad en cantidad de médicos de algún sexo, la afectación de enfermedades y los posibles monopolios farmacéuticos. Quizás eso sería sujeto de estudio de científicos de datos, pero incluso estos profesionistas tendrían que pasar por un análisis preliminar antes de conjeturar sobre los datos.

Ahora, hablando personalmente, pasar por esta serie de ejercicios resulta muy interesante, puesto que no solo se ha tenido la oportunidad de descubrir otro uso de la herramienta aplicada, sino que se ha podido visualizar otra perspectiva de abordaje hacía el análisis de un grafo, mucho más allá de las posibles simplezas que su exclusiva importación pudiera brindar. Jamás hay que dar por hechas las bases epistemológicas de un concepto, ya que, incluso demostrar una suma conlleva un largo proceso y razonamiento crítico (Whitehead, Rusell, 1925).

Bibliografía

Carlos, S. C. (2023). *Graph theory*. Recuperado el 08 de septiembre de 2023 de https://www.britannica.com/topic/graph-theory.

The Network Pages. (s.f.). *Graph Theory*. Recupera el 08 de septiembre del 2023 de https://www.networkpages.nl/graph-theory/.

Nakao, M., Sakai, M., Hanada, Y. et al. (2021). *Graph optimization algorithm for low-latency interconnection networks*. Recuperado el 08 de septiembre del 2023 de https://www.sciencedirect.com/science/article/pii/S0167819121000569.

The Geography of Transport Systems. (s.f.). *Diameter of a Graph*. Recuperado el 08 de septiembre del 2023 de https://transportgeography.org/contents/methods/graph-theory-measures-indices/diameter-graph/.

Weisstein, E. W. (s.f.). *Graph Diameter*. Recuperado el 08 de septiembre de https://mathworld.wolfram.com/GraphDiameter.html.

Neo4j. (s.f.). Neo4j Graph Database. Recuperado el 08 de septiembre del 2023 de https://neo4j.com/product/neo4j-graph-database/.

Whitehead, A. N., Russell, B. (1925-1927). *Principia Mathematica*. Cambridge University Press.