



Ingeniería en Sistemas Computacionales

Minería de Grafos

Mtro. Víctor Hugo Ortega Guzmán

Práctica 2: Aplicando algoritmos de centralidad y comunidades para analizar un grafo

Marco Ricardo Cordero Hernández

Tlaquepaque, Jal., 23 de octubre de 2023

Índice

Introducción.....	1
Desarrollo.....	2
Importe y creación de los datos	2
Análisis básico	3
Preguntas guía.....	7
Centralidad.....	8
Comunidad.....	15
Modificación de la estructura del grafo	21
Proceso base de ejecución de los algoritmos.....	21
Centralidad.....	22
Comunidad.....	25
Análisis estructural.....	27
Tablero general.....	28
Tablero de centralidad.....	28
Tablero de comunidad.....	29
Conclusiones y recomendaciones	30
Bibliografía.....	31

Introducción

Siguiendo la línea de desarrollo del conocimiento con la cual se ha venido trabajando para descubrir una capa adicional de la abstracción perteneciente a la información representada en forma de grafos, el documento presente tiene la intención de demostrar la adquisición de nuevos aprendizajes pertinentes a algoritmos de análisis adicionales a los vistos previamente, siendo estos en esta ocasión *algoritmos de comunidad*.

Según Needham y Hodler (2021), estos algoritmos tienen el propósito de agrupar nodos en distintos segmentos o *comunidades* con el fin de realizar un análisis más puntual, puesto que las relaciones formadas entre los grupos generados serían más significativas, de forma que su disección probablemente dará una mejor contextualización del comportamiento del grafo general en comparación del análisis global.

A día de hoy, la librería de Neo4j (s.f.) *Graph Data Science* [GDS] (s.f.) cuenta con 15 algoritmos aplicables a diversos tipos de grafos, sin embargo, todos estos pueden ser catalogados dentro de 4 grandes áreas:

- Densidad de relaciones
- Búsqueda y localización de grupos interconectados
- Categorización de grupos a través de sus tipos
- Calidad de agrupamiento y jerarquías

La correcta aplicación de estos algoritmos brindarán un mejor entendimiento de los grafos sobre los cuales sean aplicados, y, hablando de aplicaciones, también pueden ser útiles para propósitos como detección de grupos susceptibles a enfermedades epidémicas (grafos de pacientes), identificación de posibles grupos delictivos a través del análisis de posibles actividades terroristas (grafos de cuentas virtuales), segmentación de mercados a través de ofertas de mercadotecnia altamente personalizada (cliente de un producto como grafo), y muchas otras aplicaciones que trascienden más allá de la simple vista abstracta entre nodos y relaciones (Karataş y Şahin, 2018).

Nuevamente, y como se ha venido realizando desde un tiempo atrás, se hará uso de las ya mencionadas herramientas *Neo4j* como principal base de datos orientada a grafos para contener la información a analizar, y, *GDS* para facilitar la aplicación de una serie de algoritmos que no se limita a comunidades, sino que también se traen de vuelta aquellos referentes a la centralidad de un grafo, esto con el propósito de proporcionar un análisis sumamente completo (más no perfecto) que finalmente podría servir para elaborar reportes con un posible significado administrativo o comercial.

Adicional a lo anterior, como se podrá ver, los datos proporcionados para el grafo exceden las capacidades de Neo4j en determinado momento, por lo que se recurrirá a una herramienta adicional: Gephi (2022). Este instrumento altamente especializado en grafos para su visualización y análisis hace uso de un motor gráfico en tercera dimensión para mostrarlos en tiempo real y con velocidades mayores a las que ofrece una herramienta de alto nivel como lo puede ser la base con la que se trabajará.

Desarrollo

Importe y creación de los datos

En esta ocasión, se proveen dos archivos como punto de entrada para la creación de la base: *nodos.csv* y *enlaces.csv*. Mediante técnicas de importación de datos propias de Neo4j, se puede hacer uso de estos formatos para generar la estructura entera de la práctica (teniendo en cuenta la presencia de estos archivos dentro del directorio *import* de la base). El código para lograr esto es el siguiente:

```
// 1. Importar datos
LOAD CSV WITH HEADERS FROM 'file:///nodos.csv' AS row
MERGE (c:Country {name: row.country})
CREATE (a:Airport)
SET a.id = toInteger(row.Id),
    a.name = row.Label,
    a.latitude = toFloat(row.latitude),
    a.longitude = toFloat(row.longitude)
MERGE (a)-[:LOCATED_IN]->(c);

// 2. Índices
// Índice de identificador
CREATE INDEX index_airId FOR (a:Airport) ON (a.id);

// Índice de búsqueda
CREATE INDEX index_airName FOR (a:Airport) ON (a.name);

// 3. Relaciones
LOAD CSV WITH HEADERS FROM 'file:///enlaces.csv' AS row
MATCH (a1:Airport), (a2:Airport)
WHERE a1.id = toInteger(row.Source)
AND a2.id = toInteger(row.Target)
MERGE (a1)-[:CONNECTS_WITH]->(a2);
```

Una vez que las sentencias anteriores han sido ejecutadas, los datos necesarios dentro de la base estarán disponibles para ser analizados. El esquema general del grafo se encuentra en la primera parte de la siguiente sección.

Nota: El código anterior contempla un nivel adicional del modelado de datos, de forma que los países encontrados en *nodos.csv* son extraídos hacia nodos independientes. Esto no es necesario para obtener los resultados deseados, sin embargo, es una buena práctica de normalización.

Análisis básico

Al igual que en desarrollos pasados, el primer paso que interactúa con los datos directamente es el del análisis básico de los mismos. Este paso siempre será relevante para cualquier tipo de desarrollo relacionado con análisis de grafos o cualquier otra exploración de datos, y por tanto, siempre estará presente como parte inicial de un mayor análisis.

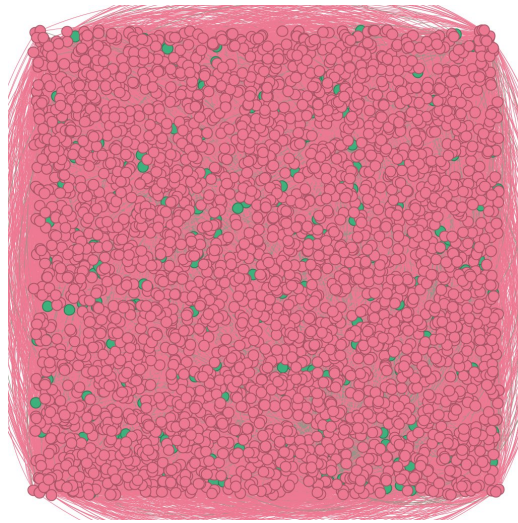
2.1. Una imagen del esquema general del grafo

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



2.2. Una imagen del grafo completo

Vista generada a través de Gephi (Neo4j solo soporta 1000 nodos dentro del browser y Bloom no despliega todos)



Esquema de colores: Rosa – Aeropuertos; Verde - Países

2.3. Número de nodos y atributos

```
MATCH (n) RETURN COUNT(n) AS totalNodos; // Número de nodos
// Atributos de nodos
CALL db.schema.nodeTypeProperties()
YIELD nodeType AS Nodo, propertyName AS Propiedad, propertyTypes AS Tipo;
```

<div>totalNodos</div> <div>3439</div>	Nodo	Propiedad	Tipo
	": Airport"	"name"	["String"]
	": Airport"	"id"	["Long"]
	": Airport"	"latitude"	["Double"]
	": Airport"	"longitude"	["Double"]
	": Country"	"name"	["String"]

2.4. Número de relaciones y atributos

```
MATCH ()-[r]->() RETURN COUNT(r) AS totalRelaciones; // Número de relaciones
// Atributos de relaciones
CALL db.schema.relTypeProperties()
YIELD relType AS Relacion, propertyName AS Propiedad, propertyTypes AS Tipo;
```

<div>totalRelaciones</div> <div>40121</div>	Relacion	Propiedad	Tipo
	": CONNECTS_WITH"	null	null
	": LOCATED_IN"	null	null

2.5. Diámetro del grafo e interpretación

Dado que en esta ocasión el grafo cuenta con dimensiones significativamente mayores a las vistas anteriormente, como ya se mencionaba en la introducción, se hace uso de Gephi para obtener el dato pertinente a este punto, dando como resultado un valor de **13** para una interpretación dirigida y **8** para una no dirigida en relación con el diámetro del grafo (The Geography of Transport Systems). Como se ha analizado en ocasiones anteriores, en el contexto presente, estos datos sirven para determinar cuántos aeropuertos tendrían que visitarse para estar en posible contacto con *todo el mundo*.

Diameter: 13
Radius: 0
Average Path length: 4.003937634352336

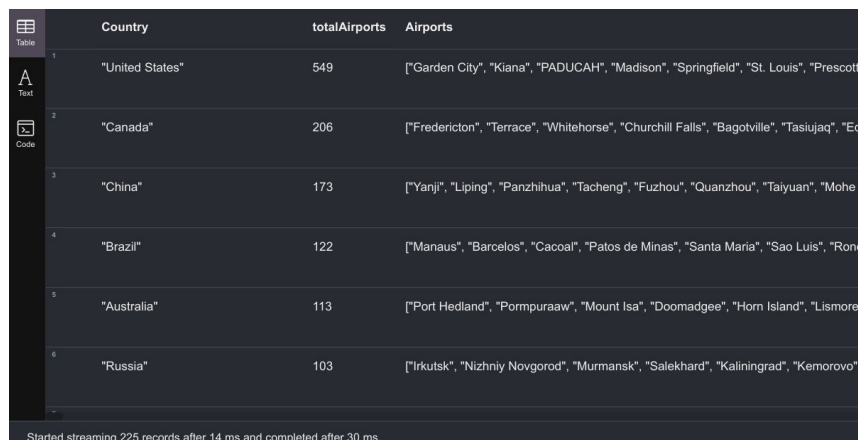
Diameter: 8
Radius: 5
Average Path length: 3.805772035209851

2.6. Densidad del grafo e interpretación

Si se considera a la densidad de un grafo como la medida para determinar *qué tan conectado está* (Datta, 2022), al aplicar este análisis nuevamente a través de Gephi, se obtienen dos valores para su interpretación dirigida y no dirigida: 0.003 y 0.004, respectivamente. Tomando en cuenta el rango de valores entre 0 y 1, estas densidades representan tan solo el 0.3% y 0.4%, lo cual, al interpretarse en conjunto con el diámetro previamente obtenido, en realidad tiene sentido, ya que, al haber tanta información contenida en el grafo con relaciones relativamente vagas, se puede decir que no cuenta con un buen nivel de conectividad. A pesar de esto, los algoritmos que se verán más adelante serán útiles para demostrar que en realidad sí existen conexiones que no se ven a simple vista a través de métricas como esta.

2.7. Una consulta con cypher que muestre el número aeropuertos por cada país y una la lista de ellos

```
MATCH (a:Airport)-[:LOCATED_IN]->(c:Country)
RETURN c.name AS Country, COUNT(a) AS totalAirports, COLLECT(a.name) AS Airports
ORDER BY totalAirports DESC;
```



	Country	totalAirports	Airports
1	"United States"	549	["Garden City", "Kiana", "PADUCAH", "Madison", "Springfield", "St. Louis", "Prescott"]
2	"Canada"	206	["Fredericton", "Terrace", "Whitehorse", "Churchill Falls", "Bagotville", "Tasiujaq", "Ed"]
3	"China"	173	["Yanji", "Liping", "Panzhihua", "Tacheng", "Fuzhou", "Quanzhou", "Taiyuan", "Mohe"]
4	"Brazil"	122	["Manaus", "Barcelos", "Cacoal", "Patos de Minas", "Santa Maria", "Sao Luis", "Rond"]
5	"Australia"	113	["Port Hedland", "Pormpuraaw", "Mount Isa", "Doomadgee", "Horn Island", "Lismore"]
6	"Russia"	103	["Irkutsk", "Nizhniy Novgorod", "Murmansk", "Salekhard", "Kaliningrad", "Kemorovo"]

Los resultados han sido acortados por su magnitud, sin embargo, se puede observar que existen 225 lugares que cuentan con cantidades variables de aeropuertos dentro de los mismos.

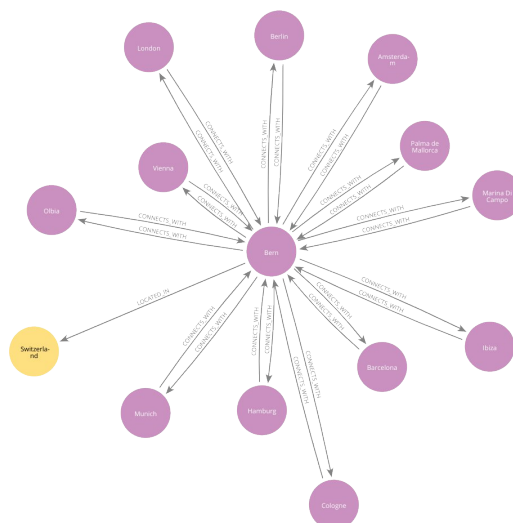
2.8. Una consulta con cypher que muestre el número de conexiones que cada país tiene con otros países

```
MATCH (c1:Country)<-[:LOCATED_IN]-(a:Airport)-[:CONNECTS_WITH]->(b:Airport)-
[:LOCATED_IN]->(c2:Country)
WHERE c1.name <> c2.name
RETURN c1.name AS Origin, COUNT(DISTINCT c2) AS totalLinks
ORDER BY totalLinks DESC, Origin;
```

Origin	totalLinks
"France"	115
"United Kingdom"	99
"Germany"	97
"United States"	93
"Turkey"	91
"United Arab Emirates"	91

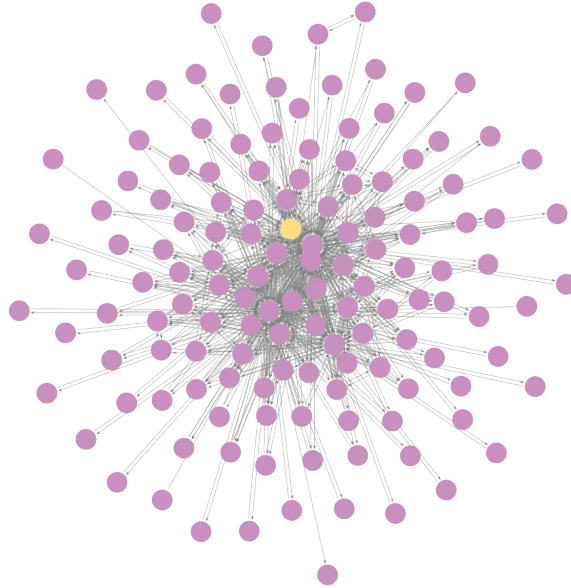
2.9. Una frase de búsqueda en Bloom que reciba el nombre del aeropuerto como parámetro y te regrese su grafo a un salto; Prueba con un aeropuerto de tu elección de Europa

```
// Frase: Graph of airport $AIRPORT at one hop
// Parámetro de entrada: AIRPORT := Nombre del aeropuerto
// Aeropuerto seleccionado: Bern (Suiza)
MATCH (a:Airport {name: $AIRPORT})
MATCH p=(a)-[*1..1]-(n)
RETURN p;
```



2.10. Una frase de búsqueda en Bloom que reciba el nombre del país como parámetro y te regrese su grafo a dos saltos; Prueba con México

```
// Frase: Graph of country $COUNTRY at two hops
// Parámetro de entrada: COUNTRY := Nombre del país
MATCH (c:Country {name: $COUNTRY})
MATCH p=(c)-[*1..2]-(n)
RETURN p;
```



Preguntas guía

Una vez que se ha explorado el grafo al menos de manera poco más que superficial, se propone una serie de preguntas que atienden a dos propósitos relacionados: aplicar algoritmos de centralidad y comunidad, y brindar información relevante a negocios. Esto último hace referencia al caso *real* en el que un grafo es una forma de representar datos extraídos de las interacciones y operación de una empresa, en donde se les desea aplicar un análisis adicional que revele información oculta a simple vista. Las siguientes preguntas están redactadas de tal forma que un usuario común de aeropuertos o un agente administrativo de estos puedan comprender lo que se pretende realizar sin necesidad de que tengan un amplio conocimiento técnico más allá de las representaciones puntuales que los contenidos y elementos del grafo representan por si mismos, esto con el fin de ser capaces de brindar un informe con un trasfondo fundamentado por ciencia de datos aplicada, pero en un formato amigable para cualquier persona.

Centralidad

Pregunta técnica	Pregunta de negocio												
<p>¿Qué nodos tienen la mayor cantidad de relaciones con otros nodos?</p> <p>Se pretende que se responda a través de Degree Centrality, algoritmo que obtiene la suma de las relaciones entrantes y salientes desde un nodo.</p>	<p>¿Cuáles aeropuertos tienen la mayor conexión hacia otros aeropuertos?</p> <p>Los aeropuertos con alto Degree Centrality son útiles para pasajeros que viajan frecuentemente desde casa, de forma que un aeropuerto con múltiples conexiones es idóneo para ir y venir de manera conveniente.</p>												
<pre>MATCH (a:Airport) RETURN a.name AS Airport, a.Degree AS Conexiones ORDER BY Conexiones DESC LIMIT 5;</pre>													
	<table> <thead> <tr> <th>Airport</th><th>Conexiones</th></tr> </thead> <tbody> <tr> <td>"Frankfurt"</td><td>477.0</td></tr> <tr> <td>"Paris"</td><td>470.0</td></tr> <tr> <td>"Amsterdam"</td><td>463.0</td></tr> <tr> <td>"Istanbul"</td><td>451.0</td></tr> <tr> <td>"Atlanta"</td><td>433.0</td></tr> </tbody> </table>	Airport	Conexiones	"Frankfurt"	477.0	"Paris"	470.0	"Amsterdam"	463.0	"Istanbul"	451.0	"Atlanta"	433.0
Airport	Conexiones												
"Frankfurt"	477.0												
"Paris"	470.0												
"Amsterdam"	463.0												
"Istanbul"	451.0												
"Atlanta"	433.0												

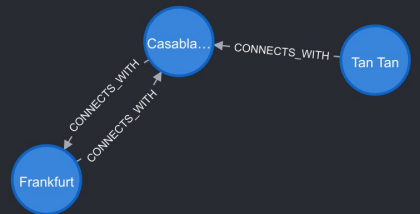
Pregunta técnica	Pregunta de negocio
<p>¿Qué nodos tienen la mayor cantidad de relaciones hacia otros nodos?</p> <p>Se pretende que se responda a través de Out Degree Centrality, algoritmo que obtiene la cantidad de relaciones salientes hacia otros nodos.</p>	<p>¿Qué aeropuertos cuentan con la mayor cantidad de destinos?</p> <p>Los aeropuertos con mayor Out Degree son ideales para pasajeros que se trasladan hacia muchos lugares lejanos a partir de un origen, de forma que estos aeropuertos cuentan con muchas salidas hacia muchos lugares sin la necesidad de corroborar el regreso.</p>
<pre>MATCH (a:Airport) RETURN a.name AS Airport, a.Out_Degree AS Salidas ORDER BY Salidas DESC LIMIT 5;</pre>	

Airport	Salidas
"Frankfurt"	239.0
"Paris"	237.0
"Amsterdam"	232.0
"Istanbul"	224.0
"Atlanta"	217.0

Pregunta técnica	Pregunta de negocio												
<p>¿Qué nodos tienen la <i>menor</i> cantidad de relaciones desde otros nodos?</p> <p>Se pretende que se responda a través de In Degree Centrality, algoritmo que obtiene la cantidad de relaciones entrantes desde otros nodos.</p>	<p>¿Cuáles aeropuertos tienen difícil acceso?</p> <p>Los aeropuertos más remotos pueden ser listados a través de In Degree, de forma que podrían verse como puntos conflictivos al realizar estrategias en rutas de entregas internacionales.</p>												
<pre> MATCH (a:Airport) RETURN a.name AS Airport, a.In_Degree AS Entradas ORDER BY Entradas LIMIT 5; </pre>													
	<table> <tr> <th>Airport</th><th>Entradas</th></tr> <tr> <td>"Cape Lisburne"</td><td>0.0</td></tr> <tr> <td>"Tatalina"</td><td>0.0</td></tr> <tr> <td>"Tan Tan"</td><td>0.0</td></tr> <tr> <td>"Ovda"</td><td>0.0</td></tr> <tr> <td>"Port Moller"</td><td>0.0</td></tr> </table>	Airport	Entradas	"Cape Lisburne"	0.0	"Tatalina"	0.0	"Tan Tan"	0.0	"Ovda"	0.0	"Port Moller"	0.0
Airport	Entradas												
"Cape Lisburne"	0.0												
"Tatalina"	0.0												
"Tan Tan"	0.0												
"Ovda"	0.0												
"Port Moller"	0.0												

Pregunta técnica	Pregunta de negocio
<p>¿Cuál es el tercer nodo con la mayor cantidad de relaciones?</p> <p>Nuevamente se hace uso de Degree Centrality.</p>	<p>¿Cuál es el tercer aeropuerto con mayor posibilidad de contar con conexiones a todo el mundo?</p> <p>Se extrae este dato a través de Degree</p>

	Centrality para determinar la posición en un ranking mundial de aeropuertos con mayores conexiones.				
<pre>CALL { MATCH (a:Airport) RETURN a.name AS Airport, a.Degree AS Conexiones ORDER BY Conexiones DESC LIMIT 3 } WITH Airport, Conexiones RETURN Airport, Conexiones ORDER BY Conexiones LIMIT 1;</pre>					
	<table><tr><th>Airport</th><th>Conexiones</th></tr><tr><td>"Amsterdam"</td><td>463.0</td></tr></table>	Airport	Conexiones	"Amsterdam"	463.0
Airport	Conexiones				
"Amsterdam"	463.0				

Pregunta técnica	Pregunta de negocio
<p>El camino más corto entre el nodo con mayor cantidad de relaciones salientes y el nodo con la menor cantidad de relaciones entrantes.</p> <p>Se pretende la combinación de algoritmos básicos de centralidad para el análisis de relaciones de un nodo y caminos más cortos entre nodos.</p>	<p>¿Qué aeropuertos tendrían que visitarse para ir desde el aeropuerto con mayor cantidad de destinos hacia el aeropuerto con destinación menos popular a nivel mundial?</p> <p>Un viajero podría usar esta información para planificar su travesía ahorrando tiempo y dinero al construir su ruta con la menor cantidad de transbordos.</p>
<pre>CALL { MATCH (n:Airport) RETURN n.name AS Airport ORDER BY n.Out_Degree DESC LIMIT 1 } WITH Airport AS a1 CALL { MATCH (n:Airport) RETURN n.name AS Airport ORDER BY n.In_Degree LIMIT 1 } WITH Airport AS a2, a1 AS a1 MATCH (a:Airport), (b:Airport) WHERE a.name = a1 AND b.name = a2 RETURN shortestPath((a)-[*]-(b));</pre>	
	

Pregunta técnica	Pregunta de negocio				
<p>¿Cuál nodo es el más popular dentro del grafo?</p> <p>Se pretende ver la implementación de Page Rank con parámetros normales para determinar este dato.</p>	<p>¿Qué aeropuerto tiene las mejores conexiones hacia otros aeropuertos?</p> <p>El uso de Page Rank en este contexto se traduce directamente a aeropuertos con la mejor calidad de conexiones. Esto puede verse como aeropuertos con destinos populares o de alto impacto comercial.</p>				
<pre>CALL { MATCH (a:Airport) RETURN a.PageRank AS PR ORDER BY PR DESC LIMIT 1 } WITH toFloat(PR) AS PR MATCH (a:Airport) WITH a, toFloat(a.PageRank) AS pr, PR RETURN a.name AS Airport, toString(ROUND(pr/PR, 2)*100) + '%' AS Popularidad ORDER BY a.PageRank DESC LIMIT 1;</pre>					
	<table> <thead> <tr> <th>Airport</th><th>Popularidad</th></tr> </thead> <tbody> <tr> <td>"Atlanta"</td><td>"100.0%"</td></tr> </tbody> </table>	Airport	Popularidad	"Atlanta"	"100.0%"
Airport	Popularidad				
"Atlanta"	"100.0%"				

Pregunta técnica	Pregunta de negocio
<p>¿Cuáles nodos no resultan tan relevantes dentro del grafo?</p> <p>Con la misma aplicación de Page Rank, se pueden determinar rankings a partir de esta medición para obtener al menos 5 de estos nodos poco populares.</p>	<p>¿Cuáles aeropuertos resultan complejos al momento de conectarlos con otros de mayor afluencia?</p> <p>Encontrar aeropuertos con baja relevancia resulta importante en casos como análisis de proximidad desde aeropuertos con mayor tamaños hacia anexos que posiblemente pudieran ser demolidos para implantar otro tipo de comercios o darles propósitos alternos.</p>
<pre>CALL { MATCH (a:Airport) RETURN a.PageRank AS PR ORDER BY PR DESC LIMIT 1</pre>	

```

} WITH toFloat(PR) AS PR
MATCH (a:Airport)
WITH a, toFloat(a.PageRank) AS pr, PR
RETURN
    a.name AS Airport,
    toString(ROUND(pr/PR, 2)) + '%' AS Popularidad
ORDER BY a.PageRank
LIMIT 5;

```

	Airport	Popularidad	
	"Tan Tan"	"0.01%"	
	"Ovda"	"0.01%"	
	"Portland"	"0.01%"	
	"Igiugig"	"0.01%"	
	"Port Heiden"	"0.01%"	

Pregunta técnica	Pregunta de negocio
<p>¿Qué nodos conectan a distintas partes del grafo?</p> <p>Se pretende la utilización de Betweenness para determinar cuáles nodos fungen como puentes dentro del grafo.</p>	<p>¿Qué aeropuertos podrían ser importantes para el transbordo de múltiples destinos en locaciones alejadas?</p> <p>El algoritmo de Betweenness aplicado al contexto aeroportuario sirve para analizar destinos que pudieran verse como pasajeros, de forma que conectarían un origen hacia un destino de forma intermedia.</p>

```

CALL {
    MATCH (a:Airport)
    RETURN a.BetweennessCentrality AS BT
    ORDER BY BT DESC LIMIT 1
} WITH toFloat(BT) AS BT
MATCH (a:Airport)
WITH a, toFloat(a.BetweennessCentrality) AS bt, BT
RETURN
    a.name AS Airport,
    toString(ROUND(bt/BT, 2)*100) + '%' AS Conecta
ORDER BY a.BetweennessCentrality DESC
LIMIT 5;

```

Airport	Conecta
"Paris"	"100.0%"
"Los Angeles"	"96.0%"
"Dubai"	"92.0%"
"Anchorage"	"86.0%"
"Frankfurt"	"82.0%"

Pregunta técnica	Pregunta de negocio
<p>¿Qué nodos tienen la mejor capacidad para distribuir información a través del grafo?</p> <p>Complementario a los demás algoritmos de centralidad, se pretende el uso de Closeness para determinar al menos 5 nodos relevantes a este propósito.</p>	<p>¿Qué aeropuertos resultan críticos en la distribución de bienes comerciales?</p> <p>Closeness, al determinar la calidad distributiva, puede aplicarse en este contexto para ver cuáles aeropuertos son importantes arterías comerciales en operaciones de importación, exportación, o simple transporte de productos.</p>
<pre>CALL { MATCH (a:Airport) RETURN a.ClosenessCentrality AS CT ORDER BY CT DESC LIMIT 1 } WITH toFloat(CT) AS CT MATCH (a:Airport) WITH a, toFloat(a.ClosenessCentrality) AS ct, CT RETURN a.name AS Airport, toString(ROUND(ct/CT, 2)*100) + '%' AS Distribuye ORDER BY a.ClosenessCentrality DESC LIMIT 5;</pre>	
Airport	Distribuye
"Noumea"	"100.0%"
"Melbourne"	"100.0%"
"Itoqortoormiit"	"100.0%"
"Sao Felix do Araguaia"	"100.0%"
"Unalaska"	"100.0%"

Pregunta técnica	Pregunta de negocio																		
Si se combinaran las mediciones de Betweenness y Closeness ¿Cuáles nodos resultan ser mejores para distribuir información a través de distintas partes del grafo?	<p>¿Cuáles aeropuertos son vitales tanto para pasajeros como para comerciantes?</p> <p>Al combinar Betweenness con Closeness, es posible la determinación de los aeropuertos que mejor sirven como escalas entre destinos y como venas comerciales en donde se pueden hacer entregas rápidas hacia clientes.</p>																		
<pre>CALL { MATCH (a:Airport) RETURN a.BetweennessCentrality AS BT ORDER BY BT DESC LIMIT 1 } WITH toFloat(BT) AS BT CALL { MATCH (a:Airport) RETURN a.ClosenessCentrality AS CT ORDER BY CT DESC LIMIT 1 } WITH toFloat(CT) AS CT, BT MATCH (a:Airport) WITH a, toFloat(a.BetweennessCentrality) AS bt, toFloat(a.ClosenessCentrality) AS ct, BT, CT RETURN a.name AS Airport, toString(ROUND(bt/BT, 2)*100) + '%' AS Conecta, toString(ROUND(ct/CT, 2)*100) + '%' AS Distribuye ORDER BY a.BetweennessCentrality DESC, a.ClosenessCentrality DESC, Airport LIMIT 5;</pre>																			
	<table><tr><th>Airport</th><th>Conecta</th><th>Distribuye</th></tr><tr><td>"Paris"</td><td>"100.0%"</td><td>"41.0%"</td></tr><tr><td>"Los Angeles"</td><td>"96.0%"</td><td>"39.0%"</td></tr><tr><td>"Dubai"</td><td>"92.0%"</td><td>"40.0%"</td></tr><tr><td>"Anchorage"</td><td>"86.0%"</td><td>"30.0%"</td></tr><tr><td>"Frankfurt"</td><td>"82.0%"</td><td>"41.0%"</td></tr></table>	Airport	Conecta	Distribuye	"Paris"	"100.0%"	"41.0%"	"Los Angeles"	"96.0%"	"39.0%"	"Dubai"	"92.0%"	"40.0%"	"Anchorage"	"86.0%"	"30.0%"	"Frankfurt"	"82.0%"	"41.0%"
Airport	Conecta	Distribuye																	
"Paris"	"100.0%"	"41.0%"																	
"Los Angeles"	"96.0%"	"39.0%"																	
"Dubai"	"92.0%"	"40.0%"																	
"Anchorage"	"86.0%"	"30.0%"																	
"Frankfurt"	"82.0%"	"41.0%"																	

Comunidad

Teniendo la primera parte de estas preguntas analíticas revisada y atendida, es momento de ver el siguiente objetivo y enfoque primordial de la práctica presente: algoritmos de comunidad. Como se mencionaba en la introducción, estos algoritmos serán aplicados para agrupar distintos componentes del grafo base; en el contexto de la implementación, la utilidad de estas herramientas recae en el análisis de inmediaciones pertenecientes a zonas geográficas relacionadas o elementos de una colección/entidad administrativa como lo puede ser el Grupo Aeroportuario del Pacífico en México (s.f). De hecho, el análisis de comunidades en este contexto resulta más útil para fines administrativos. Los algoritmos (Needham y Hodler, 2021) (Neo4j, s.f.) a utilizar son los siguientes:

- Louvain: detección de comunidades
- Triangle count: cohesión entre comunidades y estabilidad de grafos
- Label propagation: categorización de nodos para detección inicial de comunidades
- Weakly connected components: análisis de la estructura inicial de un grafo y auxiliar para comprobar conexión general
- Strongly connected components: detección de dependencias circulares y relaciones de alta calidad
- K-1 Coloring: caracterización de nodos y asignación óptima de recursos

Pregunta técnica	Pregunta de negocio
<p>¿Cuáles comunidades de nodos relacionados existen dentro del grafo?</p> <p>Múltiples algoritmos de comunidad pueden ser aplicados para atender este punto, sin embargo, se busca la implementación de Louvain para mostrar comunidades fuertemente conectadas.</p>	<p>¿Cuáles conjuntos de aeropuertos existen a nivel mundial?</p> <p>La forma más sencilla de ver la aplicación de Louvain en un contexto real es al realizar conjuntos de aeropuertos pertenecientes a un país. Para este grafo, la presencia de esta procedencia resulta útil para su comparación.</p>

```
MATCH (a:Airport)
RETURN
    a.louvain AS grupoAeroportuario,
    COLLECT(a.name) AS Aeropuertos
ORDER BY grupoAeroportuario;
```

grupoAeroportuario	Aeropuertos
319	["Andoya", "Alta", "Bodo", "Batsfjord", "Harstad/Narvik", "Hasvik", "Kirkenes", "Mosjoen", "Lakselv", "Tromsø", "Tromsø Lufthavn, Skjerve"]
722	["Eilat", "Haifa", "Tel-aviv"]
760	["Keflavik", "Bejaia", "Algier", "Djanet", "Tamanrasset", "Annaba", "Constantine", "Tebessa", "Tiremp"]
863	["Bocas Del Toro", "Changuinola", "David", "Panama", "Jaqué", "Sinop", "Puerto Obaldia"]
937	["Kone", "Koumac", "Lifou", "Noumea", "Mare", "Touho", "Ouvea", "Tiga", "Wasla", "Île des Pins"]
1027	["Cotonou", "Ouagadougou", "Bobo-dioulasso", "Accra", "Tamale", "Sunyani", "Takoradi", "Abidjan", "Lomé"]

Pregunta técnica	Pregunta de negocio												
<p>¿Cuáles nodos demuestran conexiones estrechas entre sí?</p> <p>La mejor opción para responder esto es a través de Triangle Count, de forma que se pueden detectar pequeñas comunidades de nodos con relaciones de componentes importantes.</p>	<p>¿Existen conjuntos de aeropuertos con alta frecuencia de viajes entre ellos?</p> <p>Triangle Count puede detectar viajes nacionales comunes dentro del grafo, de forma que se podría visualizar un comportamiento de viaje en donde el origen es una región o estado secundario, y el destino se encuentra en la capital del país.</p>												
<pre>MATCH (a:Airport) RETURN a.name AS Aeropuerto, a.triangleCount AS conjuntosDeAeropuertos ORDER BY conjuntosDeAeropuertos DESC LIMIT 5;</pre>													
	<table> <thead> <tr> <th>Aeropuerto</th><th>conjuntosDeAeropuertos</th></tr> </thead> <tbody> <tr> <td>"Amsterdam"</td><td>4543</td></tr> <tr> <td>"Frankfurt"</td><td>4357</td></tr> <tr> <td>"Paris"</td><td>4136</td></tr> <tr> <td>"Munich"</td><td>3658</td></tr> <tr> <td>"London"</td><td>3154</td></tr> </tbody> </table>	Aeropuerto	conjuntosDeAeropuertos	"Amsterdam"	4543	"Frankfurt"	4357	"Paris"	4136	"Munich"	3658	"London"	3154
Aeropuerto	conjuntosDeAeropuertos												
"Amsterdam"	4543												
"Frankfurt"	4357												
"Paris"	4136												
"Munich"	3658												
"London"	3154												

Pregunta técnica	Pregunta de negocio				
<p>¿Cuántos triángulos conforman al nodo con mayor cantidad de triadas?</p> <p>Para obtener la solución a este reactivo, basta con aplicar Triangle Count y obtener la propiedad con mayor valor asignado.</p>	<p>A nivel mundial, ¿Cuál aeropuerto cuenta con el mayor nivel de relaciones en una escala local?</p> <p>Al aplicar Triangle Count sobre todos los aeropuertos, se puede determinar fácilmente aquel que cuente con la mayor cantidad de conexiones a escala local para establecerlo como elemento importante dentro del transporte aéreo.</p>				
<pre>MATCH (a:Airport) RETURN a.name AS Aeropuerto, a.triangleCount AS conjuntosDeAeropuertos ORDER BY conjuntosDeAeropuertos DESC LIMIT 1;</pre>					
	<table> <thead> <tr> <th>Aeropuerto</th><th>conjuntosDeAeropuertos</th></tr> </thead> <tbody> <tr> <td>"Amsterdam"</td><td>4543</td></tr> </tbody> </table>	Aeropuerto	conjuntosDeAeropuertos	"Amsterdam"	4543
Aeropuerto	conjuntosDeAeropuertos				
"Amsterdam"	4543				

Pregunta técnica	Pregunta de negocio														
<p>¿Cuántas comunidades se encuentran dentro del grafo al etiquetar todos sus nodos?</p> <p>Al hablar de etiquetas, evidentemente se trata de Label Propagation, una de las soluciones más rápidas cuando se trata de agrupar distintas secciones de un grafo.</p>	<p>Después de un consenso global de los aeropuertos del mundo, ¿Cuántas agrupaciones distintas de estos pueden encontrarse?</p> <p>Para obtener una rápida categorización de todos los aeropuertos presentes en las colecciones actuales, Label Propagation puede ser usado para cumplir este objetivo de manera eficaz y eficiente.</p>														
<pre>MATCH (a:Airport) RETURN a.labelPropagation AS Agrupación, COLLECT(a.name) AS Aeropuertos ORDER BY Agrupación;</pre>															
<table> <thead> <tr> <th>Agrupación</th><th>Aeropuertos</th></tr> </thead> <tbody> <tr> <td>226</td><td>["Goroka", "Madang", "Mount Hagen", "Nadzab", "Port Moresby", "Tambora", "Tidore", "Tidore-Teluk", "Tidore-Tidore", "Tidore-Tidore", "Tidore-Tidore"]</td></tr> <tr> <td>232</td><td>["Narssarssuaq", "Godthaab", "Sondrestrom", "Thule", "Flin Flon", "Narsarsuaq", "Sondrestrom", "Thule", "Flin Flon"]</td></tr> <tr> <td>239</td><td>["Akureyri", "Egilsstadir", "Isafjordur", "Reykjavik"]</td></tr> <tr> <td>251</td><td>["Cambridge Bay", "Coppermine", "Holman Island", "Gjoa Haven", "Cape Wankarem", "Cape Wankarem", "Cape Wankarem", "Cape Wankarem"]</td></tr> <tr> <td>259</td><td>["Clyde River", "Iqaluit", "Pond Inlet", "Resolute", "Cape Dorset", "Cape Dorset", "Cape Dorset", "Cape Dorset"]</td></tr> <tr> <td>268</td><td>["Inuvik", "Tuktoyaktuk", "Fort McPherson", "Paulatuk"]</td></tr> </tbody> </table>		Agrupación	Aeropuertos	226	["Goroka", "Madang", "Mount Hagen", "Nadzab", "Port Moresby", "Tambora", "Tidore", "Tidore-Teluk", "Tidore-Tidore", "Tidore-Tidore", "Tidore-Tidore"]	232	["Narssarssuaq", "Godthaab", "Sondrestrom", "Thule", "Flin Flon", "Narsarsuaq", "Sondrestrom", "Thule", "Flin Flon"]	239	["Akureyri", "Egilsstadir", "Isafjordur", "Reykjavik"]	251	["Cambridge Bay", "Coppermine", "Holman Island", "Gjoa Haven", "Cape Wankarem", "Cape Wankarem", "Cape Wankarem", "Cape Wankarem"]	259	["Clyde River", "Iqaluit", "Pond Inlet", "Resolute", "Cape Dorset", "Cape Dorset", "Cape Dorset", "Cape Dorset"]	268	["Inuvik", "Tuktoyaktuk", "Fort McPherson", "Paulatuk"]
Agrupación	Aeropuertos														
226	["Goroka", "Madang", "Mount Hagen", "Nadzab", "Port Moresby", "Tambora", "Tidore", "Tidore-Teluk", "Tidore-Tidore", "Tidore-Tidore", "Tidore-Tidore"]														
232	["Narssarssuaq", "Godthaab", "Sondrestrom", "Thule", "Flin Flon", "Narsarsuaq", "Sondrestrom", "Thule", "Flin Flon"]														
239	["Akureyri", "Egilsstadir", "Isafjordur", "Reykjavik"]														
251	["Cambridge Bay", "Coppermine", "Holman Island", "Gjoa Haven", "Cape Wankarem", "Cape Wankarem", "Cape Wankarem", "Cape Wankarem"]														
259	["Clyde River", "Iqaluit", "Pond Inlet", "Resolute", "Cape Dorset", "Cape Dorset", "Cape Dorset", "Cape Dorset"]														
268	["Inuvik", "Tuktoyaktuk", "Fort McPherson", "Paulatuk"]														

Pregunta técnica	Pregunta de negocio
<p>¿Cuál sería una forma alterna de revisar los nodos fuertemente conectados dentro del grafo después de aplicar los algoritmos anteriores?</p> <p>Label Propagation, al brindar los posibles mismos resultados que otros algoritmos de comunidad, puede ser usado como método alternativo para comprobar las mediciones generadas.</p>	<p>Después de las estadísticas obtenidas previamente, se pide una manera alterna de comprobar los informes presentados en relación a las conexiones existentes entre aeropuertos ¿Cuál sería una propuesta adecuada para esto?</p> <p>En este contexto, Label Propagation puede demostrar los conjuntos de aeropuertos unidos por múltiples viajes.</p>
<pre>MATCH (a:Airport) RETURN a.labelPropagation AS Agrupación,</pre>	

<pre>SUM(a.triangleCount) AS conjuntosDeAeropuertos, COLLECT(a.name) AS Aeropuertos ORDER BY conjuntosDeAeropuertos DESC, Agrupación;</pre>			
	Agrupación	conjuntosDeAeropuertos	Aeropuertos
	273	299947	["Keflavik", "Sault Sainte Marie", "St. Anthony"]
	226	1091	["Goroka", "Madang", "Mount Hagen", "Nadza"]
	1357	160	["Parana", "Rosario", "Santa Fe", "Buenos Aires"]
	1784	109	["Ujung Pandang", "Biak", "Nabire", "Timika", "Tanjung"]
	1149	107	["Rurutu", "Tubuai", "Tikehau", "Fakarava", "Makemo"]
	2086	94	["Tatalina", "Iliamna", "Port Moller", "Dillingham"]

Pregunta técnica	Pregunta de negocio														
<p>¿Cuáles nodos cuentan con relaciones débiles indistintas?</p> <p>La palabra <i>indistintas</i> muestra que las relaciones pueden ser dirigidas o no. Para ello, el mejor algoritmo posible es Weak Components, de forma que se demostrarán las colecciones de todos los nodos que cuenten con caminos hacia otros nodos.</p>	<p>Sin importar si los aeropuertos reciben vuelos desde otro o si son el punto de partida ¿Cómo se agruparían?</p> <p>Al no importar si se analizan vuelos entrantes o salientes, Weak Components asigna categorías a estas ubicaciones para agruparlos de forma que se pueden analizar sus conexiones entrantes y salientes simultáneamente.</p>														
<pre>MATCH (a:Airport) RETURN a.weaklyConnected AS Agrupación, COLLECT(a.name) AS Aeropuertos ORDER BY Agrupación;</pre>															
	<table> <tr> <th>Agrupación</th><th>Aeropuertos</th></tr> <tr> <td>0</td><td>["Goroka", "Madang", "Mount Hagen", "Nadzab", "Port Moresby", "Wewak", "Tanjung"]</td></tr> <tr> <td>931</td><td>["Kone", "Koumac", "Lifou", "Noumea", "Mare", "Touho", "Ouvea", "Tiga", "Wa"]</td></tr> <tr> <td>1837</td><td>["Seattle", "Port Angeles", "Friday Harbor", "Eastsound"]</td></tr> <tr> <td>1903</td><td>["Unalaska", "Akutan", "Atka", "Nikolski"]</td></tr> <tr> <td>2072</td><td>["Victoria", "Vancouver"]</td></tr> <tr> <td>2259</td><td>["Neerlerit Inaat", "Ittoqqortoormiit"]</td></tr> </table>	Agrupación	Aeropuertos	0	["Goroka", "Madang", "Mount Hagen", "Nadzab", "Port Moresby", "Wewak", "Tanjung"]	931	["Kone", "Koumac", "Lifou", "Noumea", "Mare", "Touho", "Ouvea", "Tiga", "Wa"]	1837	["Seattle", "Port Angeles", "Friday Harbor", "Eastsound"]	1903	["Unalaska", "Akutan", "Atka", "Nikolski"]	2072	["Victoria", "Vancouver"]	2259	["Neerlerit Inaat", "Ittoqqortoormiit"]
Agrupación	Aeropuertos														
0	["Goroka", "Madang", "Mount Hagen", "Nadzab", "Port Moresby", "Wewak", "Tanjung"]														
931	["Kone", "Koumac", "Lifou", "Noumea", "Mare", "Touho", "Ouvea", "Tiga", "Wa"]														
1837	["Seattle", "Port Angeles", "Friday Harbor", "Eastsound"]														
1903	["Unalaska", "Akutan", "Atka", "Nikolski"]														
2072	["Victoria", "Vancouver"]														
2259	["Neerlerit Inaat", "Ittoqqortoormiit"]														

Pregunta técnica	Pregunta de negocio		
Suponiendo relaciones con baja relevancia ¿Cuántas comunidades pueden ser encontradas en todo el grafo?	¿Cuántos grupos de aeropuertos existen en donde existen viajes poco frecuentes entre ellos?		
Después de la ejecución inicial de Weak Components, basta realizar un conteo general a través del grafo para obtener el total de comunidades generadas.	La sola aplicación del algoritmo Weak Componentes puede determinar el número de conjuntos de aeropuertos con viajes de baja frecuencia entre los mismos.		
<pre>MATCH (a:Airport) RETURN COUNT(DISTINCT a.weaklyConnected) AS Agrupaciones;</pre>			
	<table><tr><th>Agrupaciones</th></tr><tr><td>7</td></tr></table>	Agrupaciones	7
Agrupaciones			
7			

Pregunta técnica	Pregunta de negocio														
<p>¿Cuáles nodos cuentan con caminos bidireccionales?</p> <p>La contraparte del algoritmo anterior, Strongly Connected Components, trabaja exclusivamente con nodos que cuenten con relaciones hacia otros en donde los segundos también cuenten con relaciones hacia el nodo original.</p>	<p>¿Qué aeropuertos pueden ofrecer viajes redondos?</p> <p>Al requerir caminos mutuos, Strongly Connected Components es capaz de determinar aquellos aeropuertos con capacidad de brindar un servicio de ida y vuelta entre destinos.</p>														
<pre>MATCH (a:Airport) RETURN a.stronglyConnected AS Agrupación, COLLECT(a.name) AS Aeropuertos ORDER BY Agrupación;</pre>															
	<table> <tr> <th>Agrupación</th><th>Aeropuertos</th></tr> <tr> <td>0</td><td>["Goroka", "Madang", "Mount Hagen", "Nadzab", "Port Moresby", "Wewak"]</td></tr> <tr> <td>931</td><td>["Kone", "Kourmac", "Lifou", "Noumea", "Mare", "Touho", "Ouvea", "Tiga", "Touho"]</td></tr> <tr> <td>1837</td><td>["Seattle", "Port Angeles", "Friday Harbor", "Eastsound"]</td></tr> <tr> <td>1903</td><td>["Unalaska", "Akutan", "Atka", "Nikolski"]</td></tr> <tr> <td>2072</td><td>["Victoria", "Vancouver"]</td></tr> <tr> <td>2259</td><td>["Neerlerit Inaat", "Iltoqqortoormiit"]</td></tr> </table>	Agrupación	Aeropuertos	0	["Goroka", "Madang", "Mount Hagen", "Nadzab", "Port Moresby", "Wewak"]	931	["Kone", "Kourmac", "Lifou", "Noumea", "Mare", "Touho", "Ouvea", "Tiga", "Touho"]	1837	["Seattle", "Port Angeles", "Friday Harbor", "Eastsound"]	1903	["Unalaska", "Akutan", "Atka", "Nikolski"]	2072	["Victoria", "Vancouver"]	2259	["Neerlerit Inaat", "Iltoqqortoormiit"]
Agrupación	Aeropuertos														
0	["Goroka", "Madang", "Mount Hagen", "Nadzab", "Port Moresby", "Wewak"]														
931	["Kone", "Kourmac", "Lifou", "Noumea", "Mare", "Touho", "Ouvea", "Tiga", "Touho"]														
1837	["Seattle", "Port Angeles", "Friday Harbor", "Eastsound"]														
1903	["Unalaska", "Akutan", "Atka", "Nikolski"]														
2072	["Victoria", "Vancouver"]														
2259	["Neerlerit Inaat", "Iltoqqortoormiit"]														

Pregunta técnica	Pregunta de negocio								
<p>¿Cuáles nodos cuentan con las mejores relaciones entre ellos?</p> <p>Usando Strongly Connected Components, es posible determinar la mejor calidad de las relaciones en todo el grafo.</p>	<p>¿Qué aeropuertos indican buenos pactos o conexiones entre ellos?</p> <p>Strongly Connected Components puede ser aplicado en este contexto para ver cómo se catalogan los grupos de aeropuertos que demuestren viajes de buena calidad entre ellos, ya sea por zona geográfica, turismo, duración de los viajes, etc.</p>								
<pre>MATCH (a:Airport) RETURN a.stronglyConnected AS Agrupación, COLLECT(a.name) AS Aeropuertos ORDER BY Agrupación, SIZE(Aeropuertos) LIMIT 3;</pre>									
	<table><tr><th>Agrupación</th><th>Aeropuertos</th></tr><tr><td>0</td><td>["Goroka", "Madang", "Mount Hagen", "Nadzab", "Port Moresby", "Wewak"]</td></tr><tr><td>931</td><td>["Kone", "Koumac", "Lifou", "Noumea", "Mare", "Touho", "Ouvea", "Tiga", "Uvea"]</td></tr><tr><td>1837</td><td>["Seattle", "Port Angeles", "Friday Harbor", "Eastsound"]</td></tr></table>	Agrupación	Aeropuertos	0	["Goroka", "Madang", "Mount Hagen", "Nadzab", "Port Moresby", "Wewak"]	931	["Kone", "Koumac", "Lifou", "Noumea", "Mare", "Touho", "Ouvea", "Tiga", "Uvea"]	1837	["Seattle", "Port Angeles", "Friday Harbor", "Eastsound"]
Agrupación	Aeropuertos								
0	["Goroka", "Madang", "Mount Hagen", "Nadzab", "Port Moresby", "Wewak"]								
931	["Kone", "Koumac", "Lifou", "Noumea", "Mare", "Touho", "Ouvea", "Tiga", "Uvea"]								
1837	["Seattle", "Port Angeles", "Friday Harbor", "Eastsound"]								

Pregunta técnica	Pregunta de negocio				
<p>¿Cuántos colores serían necesarios para colorear el grafo entero con la menor cantidad posible de colores?</p> <p>Se pretende que se haga uso del algoritmo de K-1 Coloring (se infiera su convergencia).</p>	<p>Por temas de marketing, las entidades administrativas de aeropuertos desean asignar una temática característica a cada uno de ellos</p> <p>¿Cuántos colores serían necesarios para asignar los temas a cada instalación con la menor cantidad posible de recursos?</p> <p>K-1 Coloring tiene la bondad de ser el paso preliminar en temas visuales, de forma que, al contar con una colección finita de identificadores, estos pueden asignarse a una gama de colores del mismo tamaño para luego mostrar a los elementos como componentes visuales con una temática predefinida pero con un sustento encontrado en la algoritmia.</p>				
<pre>MATCH (a:Airport) RETURN COUNT(DISTINCT a.color) AS totalColores;</pre>					
<table> <thead> <tr> <th></th><th>totalColores</th></tr> </thead> <tbody> <tr> <td></td><td>35</td></tr> </tbody> </table>			totalColores		35
	totalColores				
	35				

Modificación de la estructura del grafo

Como se pudo apreciar en la subsección previa a la actual, múltiples algoritmos de comunidad fueron revisados y se proporcionaron respuestas a las cuestiones planteadas. A pesar de que esta porción del documento se encuentra previa al capítulo presente, la realidad es que los códigos que a continuación son presentados han sido ejecutados de forma anticipada a la resolución previa, esto con motivo de la repetición de reactivos con el mismo objetivo de la práctica. Otra forma de ver esto es que al realizar los requerimientos que esta sección exige, se puede responder a lo anterior; de igual forma, al responder lo anterior, automáticamente estarían resueltos los contenidos a presentar.

Bajo el acuerdo anterior, los siguientes puntos modificarían la estructura del grafo original para dotar a cada uno de los nodos (al menos del tipo aeropuerto) con nuevos atributos correspondientes a todos y cada uno de los algoritmos descritos anteriormente; esto será logrado a través de la librería previamente mencionada, *GDS*.

Nota: el proceso de escritura de nuevos atributos a través de la librería en cuestión son repetitivos, por ende, solo se muestra el proceso base a continuación y es omitido en los incisos subsecuentes (a menos que se requiera mostrar un paso adicional). Las sentencias de ejecución completas se encuentran como anexos a este documento o en este repositorio.

Proceso base de ejecución de los algoritmos

```
// Creación de subgrafo/proyecto
CALL gds.graph.project(
  'nombre_del_proyecto',
  'TIPO_DE_NODO',
  {
    NOMBRE_DE_RELACION: {
      orientation: 'ORIENTACION_DE_LAS_RELACIONES [NATURAL | REVERSE | UNDIRECTED]',
      properties: 'PROPIEDAD(ES)_DE_LAS_RELACIONES'
    }
  }
);

// Cálculo de memoria requerida (Posible únicamente para algoritmos con calidad de producción)
CALL gds.ALGORITMO_A_APLICAR.write.estimate(
  'nombre_del_proyecto',
  {writeProperty: 'PROPIEDAD_A_CALCULAR'}
)
YIELD nodeCount, relationshipCount, bytesMin, bytesMax, requiredMemory;

// Ejecución del algoritmo
CALL gds.[CALIDAD_DEL_ALGORITMO.]ALGORITMO_A_APLICAR.stream('nombre_del_proyecto')
YIELD nodeId, score
RETURN gds.util.asNode(nodeId).PROPIEDAD_DEL_NODO As Name, score
ORDER BY score DESC, Name ASC;
```

```
// Creación de atributo con resultado individual
CALL gds.[CALIDAD_DEL_ALGORITMO.]ALGORITMO_A_APLICAR.write(
  'nombre_del_proyecto',
  {writeProperty: 'PROPIEDAD_A_ESCRIBIR'}
)
YIELD centralityDistribution, nodePropertiesWritten
RETURN
  centralityDistribution.min AS MinScore,
  centralityDistribution.max AS MaxScore,
  centralityDistribution.mean AS MeanScore,
  nodePropertiesWritten;

// Resultado
MATCH (n:TIPO_DE_NODO)
RETURN n.PROPIEDAD_DEL_NODO AS NODO_X, PROPIEDAD_A_ESCRIBIR AS ALGORITMO_A_APLICAR
ORDER BY n.ALGORITMO_A_APLICAR DESC
LIMIT X;
```

Centralidad

4.1. Añade un atributo Degree con su valor correspondiente a cada nodo Airport

```
MATCH (a:Airport)
RETURN a.name AS Airport, toInteger(a.Degree) AS Degree
ORDER BY Degree DESC
LIMIT 5;
```

bytesMin	bytesMax	requiredMemory
56	56	"56 Bytes"

MinScore	MaxScore	MeanScore	nodePropertiesWritten
1.0	477.00194549560547	22.96646265339688	3214

Airport	Degree
"Frankfurt"	477
"Paris"	470
"Amsterdam"	463
"Istanbul"	451
"Atlanta"	433

4.2. Añade un atributo In_Degree con su valor correspondiente a cada nodo Airport

```
MATCH (a:Airport)
RETURN a.name AS Airport, toInteger(a.In_Degree) AS In_Degree
ORDER BY In_Degree DESC
```



```
LIMIT 5;
```

bytesMin	bytesMax	requiredMemory
56	56	"56 Bytes"

MinScore	MaxScore	MeanScore	nodePropertiesWritten
0.0	238.00096893310547	11.483230567082309	3214

Airport	In_Degree
"Frankfurt"	238
"Paris"	233
"Amsterdam"	231
"Istanbul"	227
"Atlanta"	216

4.3. Añade un atributo Out_Degree con su valor correspondiente a cada nodo Airport

```
MATCH (a:Airport)
RETURN a.name AS Airport, toInteger(a.Out_Degree) AS Out_Degree
ORDER BY Out_Degree DESC
LIMIT 5;
```

bytesMin	bytesMax	requiredMemory
56	56	"56 Bytes"

MinScore	MaxScore	MeanScore	nodePropertiesWritten
0.0	239.00096893310547	11.483230662034325	3214

Airport	Out_Degree
"Frankfurt"	239
"Paris"	237
"Amsterdam"	232
"Istanbul"	224
"Atlanta"	217

4.4. Añade un atributo PageRank con su valor correspondiente a cada nodo Airport

```
MATCH (a:Airport)
RETURN a.name AS Airport, ROUND(a.PageRank) AS PageRank
ORDER BY PageRank DESC
LIMIT 5;
```

bytesMin	bytesMax	requiredMemory
78368	78368	"76 KiB"

ranIterations
20

Airport	PageRank
"Atlanta"	15.0
"Chicago"	14.0
"Istanbul"	14.0
"Denver"	14.0
"Paris"	13.0

4.5. Añade un atributo Betweenness Centrality con su valor correspondiente a cada nodo Airport

```
MATCH (a:Airport)
RETURN a.name AS Airport, ROUND(a.BetweennessCentrality, 4) AS BetweennessCentrality
ORDER BY BetweennessCentrality DESC
LIMIT 5;
```

bytesMin	bytesMax	requiredMemory
1930056	1930056	"1884 KiB"

MinScore	MaxScore	MeanScore	nodePropertiesWritten
0.0	654231.9999998808	9318.372833723894	3214

Airport	BetweennessCentrality
"Paris"	654230.3294
"Los Angeles"	630259.6845
"Dubai"	599730.909
"Anchorage"	562916.3994
"Frankfurt"	539673.4077

4.6. Añade un atributo Closeness Centrality con su valor correspondiente a cada nodo Airport

```
MATCH (a:Airport)
RETURN a.name AS Airport, ROUND(a.ClosenessCentrality, 4) AS ClosenessCentrality
ORDER BY ClosenessCentrality DESC
LIMIT 5;
```

MinScore	MaxScore	MeanScore	nodePropertiesWritten
0.0	1.000007152557373	0.26066112043550466	3214

Airport	ClosenessCentrality
"Victoria"	1.0
"Unalaska"	1.0
"Melbourne"	1.0
"Noumea"	1.0
"Neerlerit Inaat"	1.0

Comunidad

4.7. Añade un atributo louvain con su valor correspondiente a cada nodo Airport

```
MATCH (a:Airport)
RETURN a.louvain AS Group, COLLECT(a.name) AS Airport;
```

bytesMin	bytesMax	requiredMemory
211393	2212520	"[206 KiB ... 2160 KiB]"

Group	Airport
2523	["Goroka", "Madang", "Mount Hagen", "Nadzab", ...]
1989	["Narssarssuaq", "Godthaab", "Sondrestrom", "Tasiilaq", ...]
760	["Keflavik", "Bejaja", "Algier", "Djanet", "Tamanrasset", ...]
1340	["Sault Sainte Marie", "Campbell River", "Brandon", "Winnipeg", ...]
2322	["St. Anthony", "Baie Comeau", "Bagotville", "Baie d'Urfelt", ...]
2323	["Dryden", "Kenora", "Sioux Lookout", "Summer", ...]

Started streaming 23 records after 13 ms and completed after 34 ms.

communityCount	modularity	modularities
23	0.6573226257318456	[0.6121475254244145, 0.6541997935024616, 0.6570943195018571, 0.6573226257318456]

4.8. Añade un atributo triangle count con su valor correspondiente a cada nodo Airport

```
MATCH (a:Airport)
RETURN a.name AS Airport, a.triangleCount as triangleCount
ORDER BY triangleCount DESC
LIMIT 5;
```

bytesMin	bytesMax	requiredMemory
25872	25872	"25 KiB"

globalTriangleCount	nodeCount
100657	3214

Airport	triangleCount
"Amsterdam"	4543
"Frankfurt"	4357
"Paris"	4136
"Munich"	3658
"London"	3154

4.9. Añade un atributo label propagation con su valor correspondiente a cada nodo Airport

```
MATCH (a:Airport)
RETURN a.labelPropagation AS Group, COLLECT (a.name) AS Airports;
```

bytesMin	bytesMax	requiredMemory
27728	551504	"[27 KiB ... 538 KiB]"

communityCount	ranIterations	didConverge
35	9	true

Group	Airports
226	["Goroka", "Madang", "Mount Hagen", "Nadzab", "Port Moresby", "Wewak", "N...
232	["Narssarsuaq", "Godthaab", "Sondrestrom", "Thule", "Flin Flon", "Fort Simp...
239	["Akureyri", "Egilsstadir", "Isafjordur", "Reykjavik"]
273	["Keflavik", "Saut Sainte Marie", "St. Anthony", "Baie Comeau", "Bagotville", "C...
304	["Baker Lake", "Coral Harbour", "Eskimo Point", "Rankin Inlet", "Repulse Bay", "...
251	["Cambridge Bay", "Coppermine", "Holman Island", "Gjoa Haven", "Sachs Har...

Record fields have been truncated. Started streaming 35 records after 13 ms and completed after 26 ms.

4.10. Añade un atributo weakly connected components con su valor correspondiente a cada nodo Airport

```
MATCH (a:Airport)
RETURN a.weaklyConnected AS Group, COLLECT(a.name) AS Airports;
```

bytesMin	bytesMax	requiredMemory
25808	25808	"25 KiB"

nodePropertiesWritten	componentCount
3214	7

Group	Airports
0	["Goroka", "Madang", "Mount Hagen", "Nadzab", "Port Moresby", "Wewak", "N...
931	["Kone", "Koumac", "Lifou", "Noumea", "Mare", "Touho", "Ouvea", "Tiga", "Wa...
1837	["Seattle", "Port Angeles", "Friday Harbor", "Eastsound"]
1903	["Uhalaska", "Akutan", "Aika", "Nikolski"]
2072	["Victoria", "Vancouver"]
2259	["Neerlerit Inaat", "Iltoqqortoormiit"]

Record fields have been truncated. Started streaming 7 records after 12 ms and completed after 29 ms.

4.11. Añade un atributo strongly connected components con su valor correspondiente a cada nodo Airport

```
MATCH (a:Airport)
RETURN a.stronglyConnected AS Group, COLLECT(a.name) AS Airports;
```

setCount	maxSetSize	minSetSize
7	3188	2

Group	Airports
0	["Goroka", "Madang", "Mount Hagen", "Nadzab", "Port Moresby", "Wewak", "Na...
931	["Kone", "Koumac", "Lifou", "Noumea", "Mare", "Touho", "Ouvea", "Tiga", "Waal...
1837	["Seattle", "Port Angeles", "Friday Harbor", "Eastsound"]
1903	["Unalaska", "Akutan", "Atka", "Nikolski"]
2072	["Victoria", "Vancouver"]
2259	["Neerlerit Inaat", "Illoqqortoormiit"]

Record fields have been truncated. Started streaming 7 records after 17 ms and completed after 42 ms.

4.12. Añade un atributo coloreado con su valor correspondiente a cada nodo Airport

```
MATCH (a:Airport)
RETURN a.color AS Color, COLLECT(a.name) AS Airports
ORDER BY Color;
```

nodeCount	colorCount	ranIterations	didConverge
3214	35	2	true

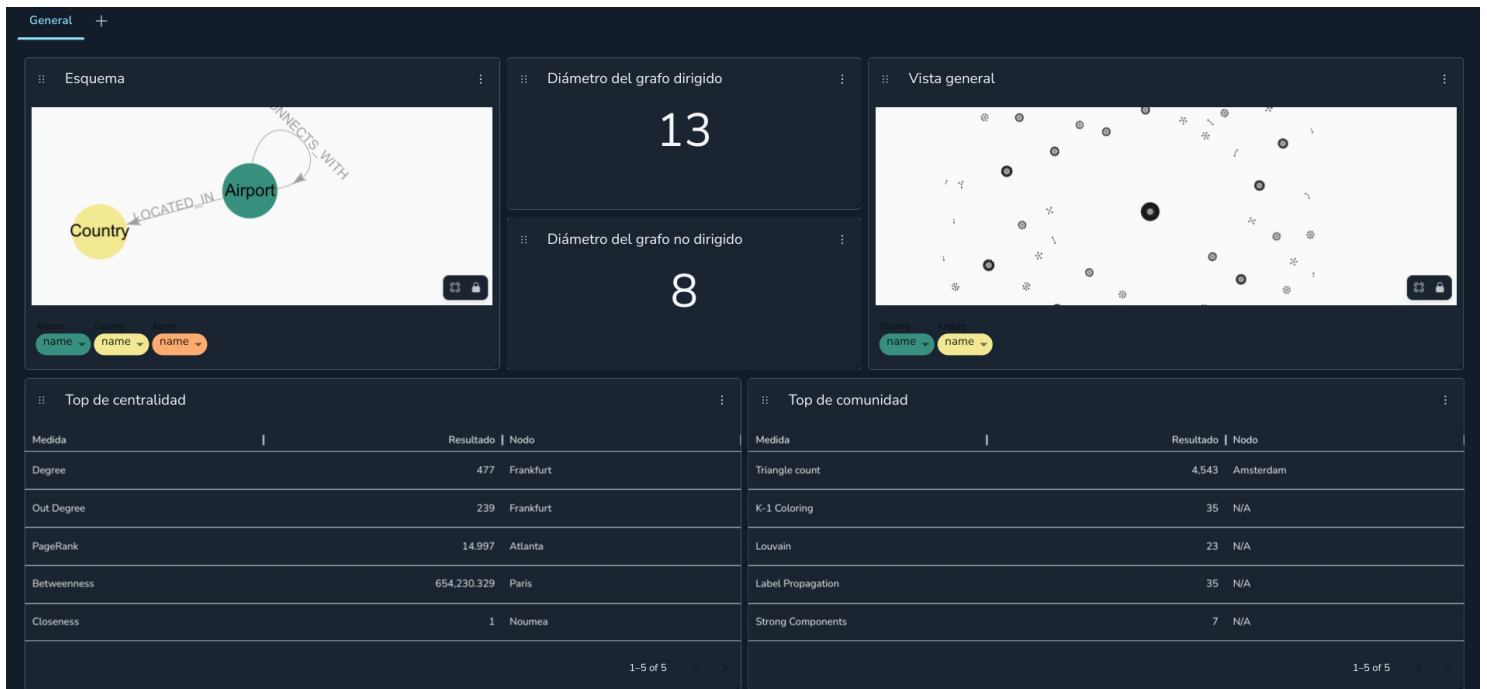
Color	Airports
0	["Goroka", "Wewak", "Narsarssuaq", "Thule", "Akureyri", "Egilsstadir", "Isafjordur",
1	["Madang", "Godthaab", "Coppermine", "Edmonton", "Iqaluit", "Iles De La Madelein...
2	["Mount Hagen", "Sondrestrom", "Reykjavik", "Holman Island", "Ottawa", "Comox",
3	["Nadzab", "Quebec", "Moncton", "Timmins", "Vancouver", "Churchill", "St. John's",
4	["Port Moresby", "Toronto", "Winnipeg", "Sept-iles", "Ouargla", "Lagos", "Leipzig",
5	["Montreal", "Lome", "Aberdeen", "Glasgow", "Rotterdam", "Trondheim", "Lusaka",

Record fields have been truncated. Started streaming 35 records after 11 ms and completed after 21 ms.

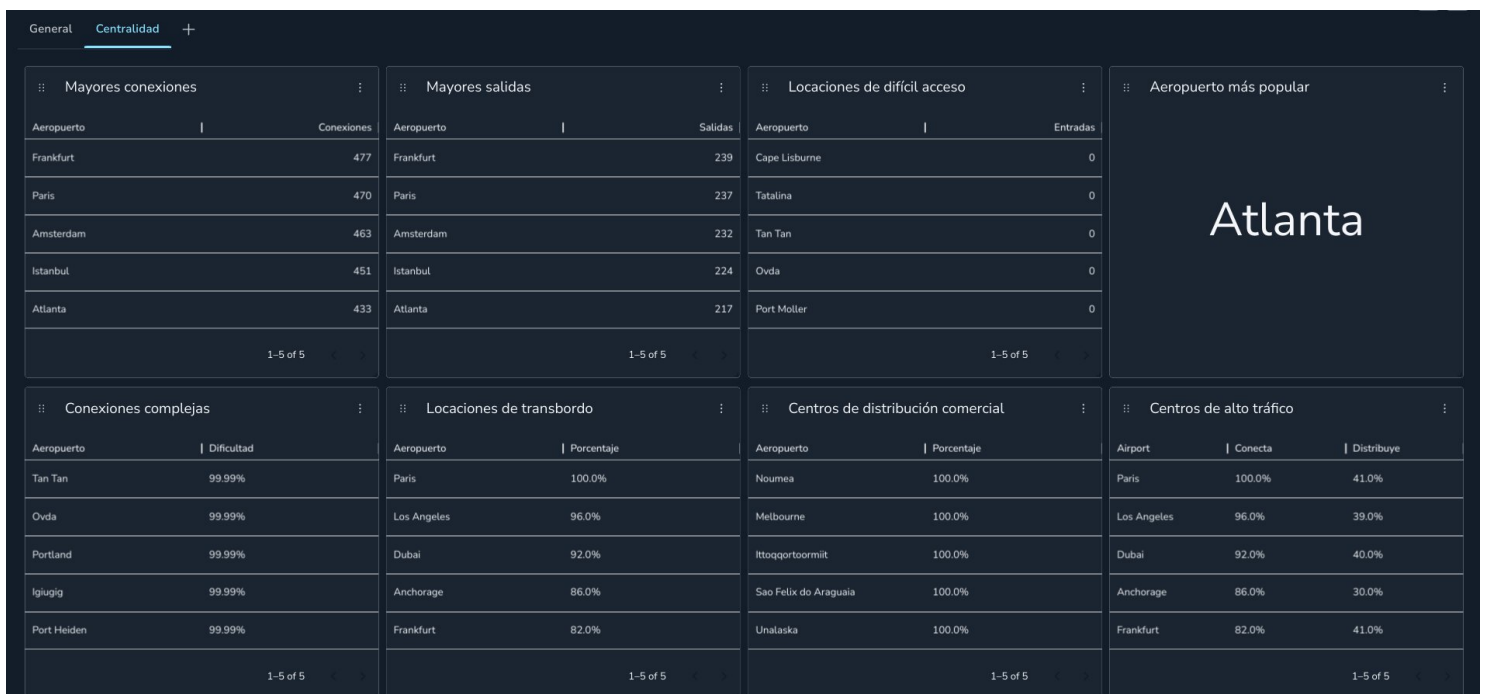
Análisis estructural

Una vez que se tiene toda la información recabada y concentrada dentro de los nodos, existen múltiples métodos de presentación de los resultados obtenidos. En la implementación presente se ha optado por un enfoque modernista en donde se hace uso de tableros de información a través de una extensión del mismo Neo4j: *NeoDash* (de Jong et al., s.f.). Este complemento permite crear presentaciones visuales de manera rápida e integra sin la necesidad de transformar los datos contenidos en el grafo hacia otra estructura para adaptarlos al formato de otro lenguaje.

Tablero *general*



Tablero de centralidad



Tablero de comunidad

[illegible]

Conclusiones y recomendaciones

Comenzando esta última sección de la práctica presente, se ha de mencionar el significado y relevancia con la que cuenta la realización de este ejercicio. En ocasiones pasadas, se han empleado ejemplos que posiblemente no serían encontrados tan sencillamente en contextos reales, no obstante, la utilización de un grafo que representa inmediaciones de suma relevancia (sin demeritar a entidades previas) como lo han sido aeropuertos, resulta más llamativo tanto en el aspecto teórico como en el práctico. Aún con tan poca información proporcionada para la construcción de la base, se ha logrado construir cuestiones que atienden al carácter administrativo con el que cuentan estos centros de transporte y distribución, así como el análisis superficial de los elementos del ocio encontrado en viajes, o asuntos comerciales como distribución nacional, importación, o sencillamente la necesidad del desplazamiento de manera más eficiente.

El desarrollo actual no tenía un propósito establecido en cuanto a resolver una problemática se refiere, más bien ha sido una práctica en toda la extensión de la palabra en donde se ha tenido la oportunidad de implementar nuevos algoritmos aprendidos en sesiones pasadas, y el reforzamiento de los conocimientos previamente adquiridos con el propósito de continuar con los aportes hacia la línea del conocimiento que se pretende se siga. Retomando el punto del objetivo de la práctica y sin dejar pasar la oportunidad de realizar al menos una recomendación, sería interesante analizar el comportamiento del mismo grafo y su interpretación en el contexto supuesto al momento de agregar peso a las relaciones; este ejercicio debería ser analizado cautelosamente, ya que sería bastante sencillo dotar a las mismas conexiones con valores aleatorios y repetir los algoritmos, sin embargo, una conducta inesperada posiblemente tendría lugar, ya que un aeropuerto remoto súbitamente estaría conectado a otro de alto nivel, lo cual en realidad sí es posible, pero poco probable, es por ello que se requeriría una curación puntual de estos nuevos datos.

Finalmente, para hablar del significado más acercado a la teoría, realizar y concluir la práctica revisada ha tenido un gran impacto tanto a nivel del contexto de la materia como a nivel del conocimiento, ya que hacía ambos campos del entorno presente se han hecho aportes significativos. Puede que parezcan repetitivos los procesos de análisis básico del grafo construido, la misma construcción tal cual, y ahora los análisis de centralidad, sin embargo, como se ha podido observar, los grafos cuentan con diversas formas de ser interpretados, y más aún cuando se les otorga el basto componente del contexto para comprender qué es lo que finalmente estarían representado estos grafos y la información que contienen.

Así como muchos otros aspectos en donde se ha tenido la oportunidad de revisarlos, los contenidos desarrollados a través del sustento proporcionado por el material consultado en las sesiones de aprendizaje, concluir la práctica presente es un paso más hacia la conclusión mayor e integra de los objetivos preestablecidos para el curso, y al ser así, trabajar con las nuevas herramientas analíticas en conjunto de las ya adoptadas no es sino una alegoría al proceso más puro de implementación del conocimiento general en la vida cotidiana; en este contexto del análisis de grafos, la práctica actual funge como el camino hacia el posible desarrollo de un proyecto de mayor escala con fundamentos revisitados pero adquiridos en este espacio de desarrollo.

Bibliografía

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

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

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

Karataş, A., Şahin, S. (2018). *Application Areas of Community Detection: A Review*. Recuperado de https://openaccess.iyte.edu.tr/bitstream/11147/9418/1/Application_Areas.pdf.

Gephi. (2022). *About*. Recuperado de <https://gephi.org/about/>.

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/>.

Datta, S. (2022). *Graph Density*. Recuperado de <https://www.baeldung.com/cs/graph-density>.

Grupo Aeroportuario del Pacífico. (s.f.). *¿Quiénes somos?*. Recuperado de <https://www.aeropuertosgap.com.mx/es/2012-03-02-17-22-49.html>.

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