INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE OCCIDENTE

Departamento de Electrónica, Sistemas e Informática

Desarrollo tecnológico y generación de riqueza sustentable

PROYECTO DE APLICACIÓN PROFESIONAL (PAP)

PAP PROGRAMA DE CIUDADES INTELIGENTES

4L05 Vida Digital

Algoritmos de grafos en Neo4j

PRESENTAN

Programas educativos y Estudiantes

Ing. Sistemas Computacionales. Darío Arias Muñoz

Profesor PAP: Mtro. Luis Eduardo Pérez Bernal

Asesor PAP: Mtro. Víctor Hugo Ortega Guzmán

Tlaquepaque, Jalisco, diciembre de 2020

ÍNDICE

Contenido

REPORTE PAP

Presentación Institucional de los Proyectos de Aplicación Profesional

Los Proyectos de Aplicación Profesional (PAP) son una modalidad educativa del ITESO en la que el estudiante aplica sus saberes y competencias socio-profesionales para el desarrollo de un proyecto que plantea soluciones a problemas de entornos reales. Su espíritu está dirigido para que el estudiante ejerza su profesión mediante una perspectiva ética y socialmente responsable.

A través de las actividades realizadas en el PAP, se acreditan el servicio social y la opción terminal. Así, en este reporte se documentan las actividades que tuvieron lugar durante el desarrollo del proyecto, sus incidencias en el entorno, y las reflexiones y aprendizajes profesionales que el estudiante desarrolló en el transcurso de su labor.

Resumen

Este proyecto tiene el propósito de generar un cuaderno de trabajo que ejemplifique casos de uso concretos y de la vida real donde te puedes apoyar de bases de datos de grafos y sus algoritmos para realizar análisis sofisticados. El proyecto describe el proceso de encontrar los datos que se necesitan para hacer el análisis utilizando dos *datasets* distintos y como se parte de ahí para realizar el análisis. El proyecto tiene un carácter educativo, realizado para que sirva como guía a estudiantes de la materia de Bases de Datos para Apoyar la Toma de Decisiones en el tema particular de minería de grafos.

1. Introducción

1.1. Objetivo

Generar material de apoyo en la forma de un cuaderno de trabajo para la clase de Bases de Datos para la Toma de Decisiones, específicamente el tema de bases de datos basadas en grafos y minería de grafos. Este cuaderno de trabajo deberá cubrir los temas esenciales sobre algoritmos de grafos, teoría de grafos, lenguaje de consultas *Cypher* y utilización básica de Neo4j.

1.1.1

Encontrar e investigar un conjunto de datos en el formato que sea, que puedan representarse con una base de datos de grafos y su análisis se de interés y complejidad suficiente para demostrar las capacidades de esta herramienta. A partir de ahora nos referiremos a ese conjunto de datos como *dataset* o *set* de datos.

1.1.2

Hacer limpieza de los datos y adaptarlos para que puedan ser cargados con facilidad a Neo4j. Cargar la información más relevante a Neo4j de manera que todos los atributos, relaciones, e información esté presente en la base de datos para ser manipulada.

1.1.3

A partir del *dataset* formular preguntas de interés y responderlas utilizando uno o varios algoritmos de minería de grafos. Mostar la utilidad de diferentes clases de algoritmos de grafos y cuáles son sus casos de uso más comunes.

1.1.3

Producir un cuaderno del trabajo que describa el proceso a seguir para realizar los pasos anteriores que sirva de guía a los alumnos para replicar el trabajo y los conceptos de este tema en particular sean más claros y significativos.

1.2. Justificación

Las bases de datos basadas en grafos se han vuelto comercialmente viables y ampliamente disponibles hasta muy recientemente, al menos en comparación con las bases de datos relaciones que han estado presentes desde hace varias décadas. Por esta razón, el contenido de ejemplos y tutoriales de acceso abierto que exploten su potencial y sirvan

para despertar interés en los estudiantes y profesionales es más limitado. Por esto, es necesaria la creación de este cuaderno de trabajo que les permita a los estudiantes conocer todo el potencial de las bases de datos de grafos. Para lograr esto primero se intentó utilizar el trabajo de investigación de Hetionet realizado por el Dr. Daniel Himmelstein. Hetionet, también conocido como proyecto de Repehetio, que surgió de la pregunta: ¿Cómo enseñar a una computadora biología? Para hacer eso, resultó ser muy adecuado utilizar una red heterogénea (*heterogenous network* "HetNet"), expuesta en una base de datos de grafos nativa. Una vez que se construyó está red (o Grafo) se utilizó para determinar posibles aplicaciones para medicamentes existentes para tratar enfermedades que no estaban contempladas ni asociadas para ser tratadas por ese medicamento inicialmente. (Himmelstein, 2016)

Creo que esté proyecto puede ser de gran interés para los estudiantes de las carreras de Ingeniería en Sistemas Computacionales e Ingeniería en Empresas de Servicios por varias razones:

- Es una aplicación científica real de las bases de datos basadas en grafos.
- Al ser un conjunto de datos de biología en general puede responder a más preguntas además de las propuestas por el autor.
- Permite a los alumnos ahondar en ciertos algoritmos de grafos ya que es un conjunto de datos muy rico en relaciones y nodos. Tanto en tipos como en cantidad.

Este set de datos tenía muchos problemas de compatibilidad con el DBMS que utilizamos y finalmente optamos por otro *dataset* de productos de Amazon, sin embargo, después se intentará este *dataset* original.

1.3 Antecedentes

A partir de que el profesor titular de la asignatura ingresa al doctorado empieza a bajar temas de bases de datos de grafos a la asignatura que se oferta a nivel licenciatura, incluyendo nuevos temas en cada periodo que se imparte. A la vez, su buscó un manejador de bases de datos basadas en grafos nativos que sea útil, accesible y esté relacionado con

los conocimientos de SQL para que la curva de aprendizaje de los alumnos sea más rápida y tengan menos dificultades.

Los alumnos de la materia Bases de Datos para la Toma de Decisiones en ocasiones tienen dificultades con el tema de Bases de Datos en Grafos y Minería de Grafos, sobre todo cuando no tienen unas bases sólidas de matemáticas discretas o algoritmia. Este cuaderno surge en un esfuerzo por aumentar el nivel de involucramiento de los alumnos y apoyo de parte de los maestros para trabajar con las bases de datos de grafos. Esto con la esperanza de que los aprendizajes obtenidos en este tema sean más significativos y duraderos que les permita a los alumnos encontrar sus propias aplicaciones y utilizarlo en proyectos diversos para que finalmente se convierta en una herramienta profesional de valor. Con el recurso del cuaderno podrá ser consultado como apoyo, guía y hasta para incitar la curiosidad del alcance de las bases de datos en grafos, sus posibles aplicaciones y su potencial.

Otro hecho importante que derivó este trabajo fue que las bases de datos basada en grafos son relativamente nuevas y hasta ahora no hay demasiadas aplicaciones de estas con un enfoque educativo. Nos concentramos en el uso del manejador Neo4j por tener una interfaz muy amigable para usuarios finales, y con una gran flexibilidad que incluso permite añadir algoritmos nuevos escritos en Java. (Neo4j, Inc., s.f.)

Por otro lado, el antecedente de la comunidad global ahora no hay muchos repositorios y datos libres listos para ser usados en las bases de datos de grafos. Para todos hay que hacer un gran trabajo de mantenimiento y adaptación para poder ser utilizados. Este proceso de limpieza reduce en gran medida el trabajo que se hace con la herramienta de grafos y canalizan el esfuerzo en otra dirección menos interesante, como obtener los datos, prepararlos, normalizarlos, etc. Por eso preparar un *dataset* interesante que esté listo para usar puede despertar interés en los conceptos clave del análisis de grafos.

Finalmente, el antecedente tecnológico que llevó a la propuesta de este proyecto es la tendencia a nivel departamento para adoptar nuevas tecnologías y generar material para mantener a sus estudiantes al día de lo que se usa en la industria y las empresas que a menudo no es igual al ambiente académico. Esta tendencia del departamento sigue de

cerca a la consistente tendencia global del desarrollo de nuevas tecnologías de software, al grado que parece que se crea una nueva todos los días.

1.4. Contexto

La materia se ofrece dos veces por año con un promedio de 60 alumnos por año. Y el tema abarca de 4 a 5 a de las 16 semanas del semestre. Cuando este tema se imparte muchos de los temas sólo se pueden cubrir superficialmente y eso hace que las probabilidades de que los alumnos aprovechen y absorban el conocimiento sean bajas. También por la curva de aprendizaje no se alcanzan a cubrir muchos temas que son importantes, y sí se reduce la curva de aprendizaje, se podrían cubrir más temas.

El contexto y motivación para el conjunto de datos a elegido surge del costo de desarrollar un nuevo fármaco terapéutico que se estima hasta en 1,400 millones de dólares y el proceso suele tardar hasta 15 años desde el compuesto principal hasta el mercado con una muy baja probabilidad de éxito. Sorprendentemente, los costos se han duplicado cada 9 años desde 1970 en una especie de Ley de Moore inversa, que está lejos de ser una estrategia óptima tanto desde la perspectiva empresarial como de la salud pública. La adaptación de fármacos, es decir, la identificación de nuevos usos para tratamientos existentes puede reducir drásticamente la duración, las tasas de fracaso y los costos de aprobación. Estos beneficios se derivan de la rica información preexistente sobre los medicamentos aprobados, incluido el perfil toxicológico extenso realizado durante el desarrollo, los modelos preclínicos, los ensayos clínicos y la vigilancia posterior a la comercialización. (Himmelstein, 2016).

Por otro lado, se utilizó el *dataset* de Amazon porque replica un caso de uso muy común dentro de las bases de datos de grafos. Este caso es apoyar a las ventas en línea con recomendaciones personalizadas y hacer anuncios con audiencias objetivo más específicas. Al ser un caso real de uso también puede resultar de gran interés y su esquema es más sencillo de entender.

2. Desarrollo

2.1. Sustento teórico y metodológico

Aprovecharé esta sección para introducir a los conceptos principales alrededor de las bases de datos basadas en grafos, sus aplicaciones clásicas, así como la investigación que se ha realizado alrededor de ellas.

Las bases de datos son un área fundacional de los sistemas computacionales, y aunque sin duda un área muy madura es a la vez un área muy vibrante en desarrollo e investigación. Particularmente en los tiempos donde la Ciencia de Datos está en todas partes, las bases de datos en muchas formas hacen que este naciente campo sea posible. Inicialmente nacieron como una forma de abstraer la capa de almacenamiento de datos de cualquier programa, para estandarizar y simplificar el desarrollo, pero desde entonces han evolucionado y crecido a pasos agigantados y ahora las tenemos de todos los sabores y colores: en columnas, en cubos, en documentos, relacionales, no relacionales y de todos los tipos imaginables. Dentro de esta gran variedad de abstraer la capa del acceso de datos, las bases de datos de grafos existen para resolver nuevos problemas y darles nuevas soluciones a problemas viejos. A continuación, algunos de los conceptos necesarios para entender este trabajo.

Grafos: Aunque los grafos nacieron como una idea abstracta matemática han probado ser una manera muy eficiente y útil de modelar y analizar datos. Hay dos objetos que forman un grafo nodos y aristas. Los nodos se conectan o relacionan a través de aristas (*edges* en inglés). (Needham, 2019)

Bases de datos: Es una colección organizada de información estructurada, o datos, típicamente almacenados electrónicamente en un sistema de computadora. Una base de datos es usualmente controlada por un sistema de gestión de base de datos (DBMS). En conjunto, los datos y el DBMS, en conjunto con otras aplicaciones se conocen como un sistema de base de datos, que a menudo se reducen a solo base de datos. (Oracle, 2015)

API: Interfaz de Programación de Aplicaciones que define un conjunto de directivas que pueden ser usadas para tener una pieza de software funcionando con algunas otras. En el cuaderno de trabajo se utilizan API nativas de Neo4j que provee nuevas funcionalidades y extiende las ya existentes. (Mozilla, 2019)

Librería: En ciencias computacionales, una biblioteca o librería es una colección de recursos persistentes utilizados por programas informáticos, a menudo para el desarrollo de software. Estos pueden incluir datos de configuración, documentación y datos de ayuda. Aunque, la mayoría de las veces se trata código prescrito, subrutinas y clases que ofrecen una funcionalidad especifica.

Funciones de agregación: Una función de agregación realiza un cálculo sobre un conjunto de valores y devuelve un solo valor. Todas las funciones agregadas son deterministas. En otras palabras, las funciones agregadas devuelven el mismo valor cada vez que se llaman, cuando se llaman con un conjunto específico de valores de entrada. A menudo estas funciones nos ayudan a conocer un valor sobre el conjunto de datos, como su median, media, suma, valor máximo, etc. (Microsoft Corporation, 2018)

Neo4j: La plataforma de gráficos Neo4j admite el procesamiento transaccional y el procesamiento analítico de datos de grafos. Incluye almacenamiento de grafos y cálculo con herramientas de análisis y gestión de datos. El conjunto de herramientas integradas se basa en un protocolo común, API, y lenguaje de consulta (Cypher) para proporcionar un acceso efectivo para diferentes usos. (Needham, 2019)

JSON: Es una sintaxis para serializar objetos, arreglos, números, cadenas, booleanos y nulos. Está basado sobre sintaxis JavaScript. (Mozilla, 2019)

Python: Python es un lenguaje de programación interpretado, de alto nivel y de propósito general. La filosofía de diseño de Python enfatiza la legibilidad del código con su notable uso de espacios en blanco significativos. Sus construcciones de lenguaje y su enfoque orientado a objetos tienen como objetivo ayudar a los programadores a escribir código claro y lógico para proyectos de pequeña y gran escala. En ciencia de datos Python se utiliza mucho porque tiene mucho soporte de la comunidad y es ideal para extracción y limpieza de datos. También nos sirve para comunicarnos con la base de dato porque existe un cliente nativo. (Kuhlman, 2012)

Algoritmos de comunidad: Los algoritmos de comunidad buscan encontrar relaciones estrechas entre nodos que permitan formar grupos y encontrar como están relacionados.

Identificar estos subconjuntos de nodos que están agrupados en comunidades nos permite conocer más respecto a la estructura global del grafo. (Needham, 2019)

Algoritmos de centralidad: Los algoritmos de centralidad nos ayudan a descubrir roles individuales de ciertos nodos que juegan papeles importantes en el grafo global. Encontrar estos nodos de importancia en la red puede ser muy útil para descubrir comportamientos de la red, o sobre que nodos se puede enviar más rápido la información. Muchas veces estos algoritmos se utilizan en análisis de redes sociales y páginas web. (Needham, 2019)

Algoritmos de semejanza: También conocidos como *Similarity algorithms*, su objetivo es darle un valor a la semejanza que pueda existir entre un par de nodos usando diferentes tipos de métricas, como distancias vectoriales, grado del nodo o intersección de conjuntos. (Needham, 2019)

OLAP y OLTP: En el procesamiento de transacciones en línea (OLTP), los sistemas de información generalmente facilitan y administran aplicaciones orientadas a transacciones. En cambio, el procesamiento analítico en línea, u OLAP, es un enfoque para responder consultas analíticas. OLAP es parte de la categoría más amplia de inteligencia empresarial, que también abarca bases de datos relacionales, redacción de informes y minería de datos. Las aplicaciones típicas de OLAP incluyen informes comerciales para ventas, marketing, etc. (Bog, 2013)

Small world: Es un tipo de grafo cuya característica es que la mayoría de los nodos no están conectados entre sí, pero los vecinos están muy fuertemente conectados. A la vez, algunos individuos de los vecinos tienen conexiones muy remotas. En una red social esto significaría que incluso dos completos extraños pudieran estar conectados por un número relativamente pequeño de aristas. (Watts & Strogatz, 1998)

APOC: La biblioteca APOC (Awesome Procedures On Cypher) es una biblioteca de utilidades estándar para procedimientos y funciones comunes. Esto permitió a los desarrolladores de todas las plataformas e industrias usar una biblioteca estándar para procedimientos comunes y solo escribir su propia funcionalidad para la lógica empresarial y las necesidades específicas de casos de uso. Como obtener fechas, cargar archivos json o csv entre otras funciones comunes. (Neo4j, s.f.)

Graph Data Science Library: En este trabajo, usaremos mucho la biblioteca de ciencia de datos de grafos de Neo4j. La biblioteca se instala como un complemento junto con la base de datos y proporciona un conjunto de procedimientos configurables por el usuario que se pueden ejecutar mediante el lenguaje de consulta Cypher. (Needham, 2019)

Proyecciones: El término de proyección lo acuñó por primera vez Ted Codd en el famoso paper que vio nacer las bases de datos y se refiere a tomar un subconjunto de datos y presentarlos, en el contexto de las bases de datos de grafos se refiere a mostrar sólo algunos nodos y sólo algunas de sus propiedades. (Codd, 1970)

Catálogo de grafos y grafos anónimos: Los algoritmos de grafos se ejecutan en un modelo de grafos que es una proyección del modelo de grafos de propiedades de Neo4j. Una proyección de grafo puede verse como una vista sobre el grafo almacenado, que contiene solo información relevante. Las proyecciones de grafos se almacenan completamente en la memoria utilizando estructuras de datos comprimidos optimizados para operaciones de búsqueda. El catálogo de grafos es un concepto dentro de la biblioteca GDS que permite administrar múltiples proyecciones de grafos por nombre. Usando su nombre, un grafo creado se puede usar muchas veces en el flujo de trabajo analítico. Los grafos con nombre se pueden crear utilizando una proyección nativa o una proyección en Cypher. Después de su uso, los grafos con nombre se pueden eliminar del catálogo para liberar memoria principal. También se pueden crear grafos cuando se ejecuta un algoritmo sin colocarlos en el catálogo. Nos referimos a estos gráficos como gráficos anónimos. (Neo4j, Inc., s.f.)

2.2. Planeación y seguimiento del proyecto

Descripción del proyecto.

El proyecto consiste en un cuaderno de trabajo con los temas necesarios para hacer más sencillo el progreso de los alumnos a través de los temas de algoritmos de grafos con ejemplos concretos y explicaciones claras. Para llegar a este entregable es importante encontrar un *dataset* que sea de interés y permita la demostración de los algoritmos, además que debe de ser posible de cargar como grafo después de alguna manipulación a través del lenguaje Python. El *dataset* luego deber hacerse disponible y público en algún formato que permita la fácil importación de los datos a una base de datos basada en grafos.

Plan de trabajo

Actividades previstas para la realización del proyecto:

Actividades	Recursos Humanos	Tipo de actividad	Fecha Inicio	Fecha Entrega	Semana
Realizar plan de trabajo	Darío Arias	Operativo	01/09/2020	01/09/2020	3
Reporte	Darío Arias	Operativo	03/11/2020	04/12/2020	3
Documentar reporte punto 1 y principios del 2	Darío Arias	Operativo	03/11/2020	07/11/2020	3 a 15
Buscar *dataset* para realización del proyecto	Darío Arias	Operativo	02/09/2020	09/09/2020	3 a 4
Busccar *dataset* para la realización del proyecto	Darío Arias	Operativo	02/09/2020	09/09/2020	3 a 4
Balancear *datasets* encontrados y escoger el mejor	Darío Arias	Operativo	07/09/2020	07/09/2020	4
Reunión de selección del *dataset*	Darío Arias y Víctor Ortega	Operativo	08/09/2020	08/09/2020	4
Limpieza y adecuación del *dataset*	Darío Arias	Técnica	09/09/2020	16/09/2020	4 a 5
Reunión de revisión de progreso interno	Darío Arias	Operativo	14/09/2020	14/09/2020	5
Crear script para carga de datos en Neo4j	Darío Arias	Técnico	15/09/2020	15/09/2020	5
Crear Script para carga de datos en Neo4j	Darío Arias	Técnico	16/09/2020	30/09/2020	5 a 7
Reunión de revisión de progreso interno	Darío Arias	Operativo	21/09/2020	21/09/2020	6
Análisis de los datos como un grafo	Darío Arias	Operativo	22/09/2020	22/09/2020	6
Reunión de revisión de progreso interno	Darío Arias y Víctor Ortega	Operativo	28/09/2020	28/09/2020	7
Generar propuesta de preguntas y análisis a realizar sobre la base de datos.	Darío Arias	Operativo	29/09/2020	29/09/2020	7
Escribir introducción y	Darío Arias	Técnico	30/09/2020	07/10/2020	7 a 8

conceptos básicos en el cuaderno de trabajo					
Escribir acerca de las plataformas de procesamiento de grafos en la guía.	Darío Arias	Técnico	30/09/2020	07/10/2020	7 a 8
Reunión de revisión de progreso interno (cuaderno de trabajo y base de datos)	Darío Arias y Víctor Ortega	Operativo	07/10/2020	07/10/2020	8
Reunión de revisión de progreso del cuaderno de trabajo	Darío Arias	Operativo	09/10/2020	09/10/2020	8
Investigar sobre librería de algoritmos de grafos, leer documentación y hacer pruebas preliminares	Darío Arias	Técnico	30/09/2020	07/10/2020	8 a 9
Escribir sobre el uso de la librería en el cuaderno	Darío Arias	Operativo	05/10/2020	05/10/2020	9
Reunión de revisión de progreso interno	Darío Arias y Víctor Ortega	Operativo	06/10/2020	06/10/2020	9
Escribir sobre el esquema de la base de datos que se va a utilizar y las preguntas que se plantean resolver	Darío Arias	Operativo	12/10/2020	12/10/2020	10
Reunión de revisión de progreso interno	Darío Arias y Víctor Ortega	Operativo	13/10/2020	13/10/2020	10
Investigar y escribir sobre algoritmos de comunidad y aplicar un algoritmo al set de datos	Darío Arias	Técnico	19/10/2020	19/10/2020	11
Reunión de revisión de progreso interno	Darío Arias	Operativo	20/10/2020	20/10/2020	11
Investigar y escribir sobre algoritmos de similitud y aplicar	Darío Arias	Técnico	26/10/2020	26/10/2020	12

un algoritmo al set de datos					
Reunión de revisión de progreso interno	Darío Arias y Víctor Ortega	Operativo	27/10/2020	27/10/2020	12
Investigar y escribir sobre algoritmos de centralidad y aplicar un algoritmo al set de datos	Darío Arias	Técnico	01/11/2020	02/11/2020	13
Reunión de revisión de progreso interno	Darío Arias y Víctor Ortega	Operativo	03/10/2020	03/10/2020	13

Tabla 1 Planeación del proyecto

Desarrollo de propuesta de mejora

Semana 3

En las semanas anteriores se plantearon las posibles propuestas de proyecto. Una vez decidido que se iba a trabajar en un cuaderno de trabajo se comenzó por buscar diferentes *datasets* que pertenecieran a alguna temática de interés social como: salud pública, educación o trabajo y desarrollo.

Las páginas donde se comenzó a buscar son las siguientes:

SNAP: Stanford Network Analysis

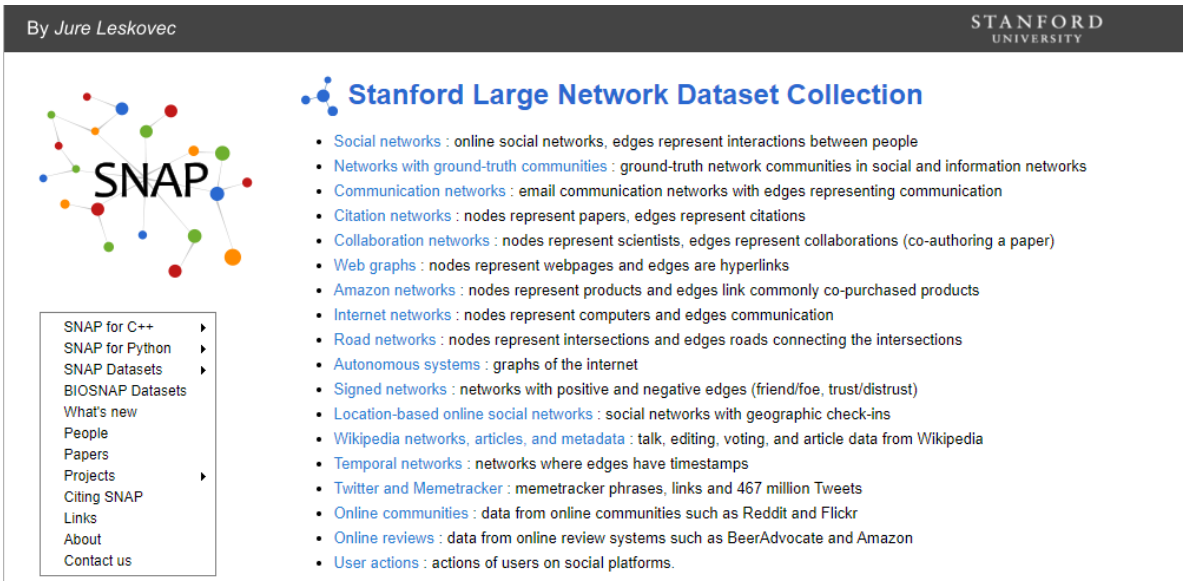

Ilustración 1. SNAP página de inicio

Esta página es un conglomerado de *datasets* adquiridio por Stanford. Incluye bases de datos nativas de grafos de stack overflow, Facebook, Wikipedia, grafos Web, carreteras entre

muchos otros. La bondad de estos datos es que ya están organizados como grafos, cono nodos y aristas y la importación a una base de datos nativa es más sencilla. Desafortunadamente, después de revisar a detalla todos los diferentes *datasets* que había, ninguno tenía información relevante en los rubros que buscamos por lo que fueron descartados.

Semana 4

Para encontrar datos más relevantes de los rubros que estamos más interesados, decidimos mejor buscar ESANUT: La encuesta de salud y en Datos Abiertos del gobierno de México. Sin embargo, ambos sitios presentaban información en forma de encuestas y muy estadística que hubiera proveído una pobre base de datos de grafos, porque no cuenta con muchas relaciones, más bien existen indicadores y medidas que no hubieran sido de demasiada utilidad. De cualquier forma, se mencionan porque puede ser de utilidad para el lector conocer estos recursos.

Ilustración 2 Página de inicio de ESANUT

Ilustración 3 Página de inicio de Datos Abiertos del Gobierno de México

Semana 5

En esa semana encontré el proyecto Rephetio, del Dr. Daniel Himmelstein y fue un proyecto muy emocionante por encontrar, porque tenía potencial para un impacto real. Fuera del

contexto de investigación en el que se concibió aún había mucho que descubrir de las relaciones presentes en este *dataset*. Los datos son completamente orientados a una red y por lo tanto pueden ser adaptados a una base de datos de grafos sin mayor problema. De hecho, lo hicieron. Los investigadores del proyecto Rephetio pusieron una base de datos de Neo4j abierta en la web para que las personas pudieran acceder a los datos y explorarla. Este explorador está disponible en: Hetionet Neo4j y tiene le siguiente esquema.

La capacidad de predecir computacionalmente si un compuesto trata una enfermedad podría mejorar la tasa de éxito de la aprobación de medicamentos. Con este propósito se puede utilzar Hetionet, una red integradora que codifica el conocimiento de millones de estudios biomédicos. Actualmente esta red consta de 47.031 nodos de 11 tipos y 2.250.197 relaciones de 24 tipos.

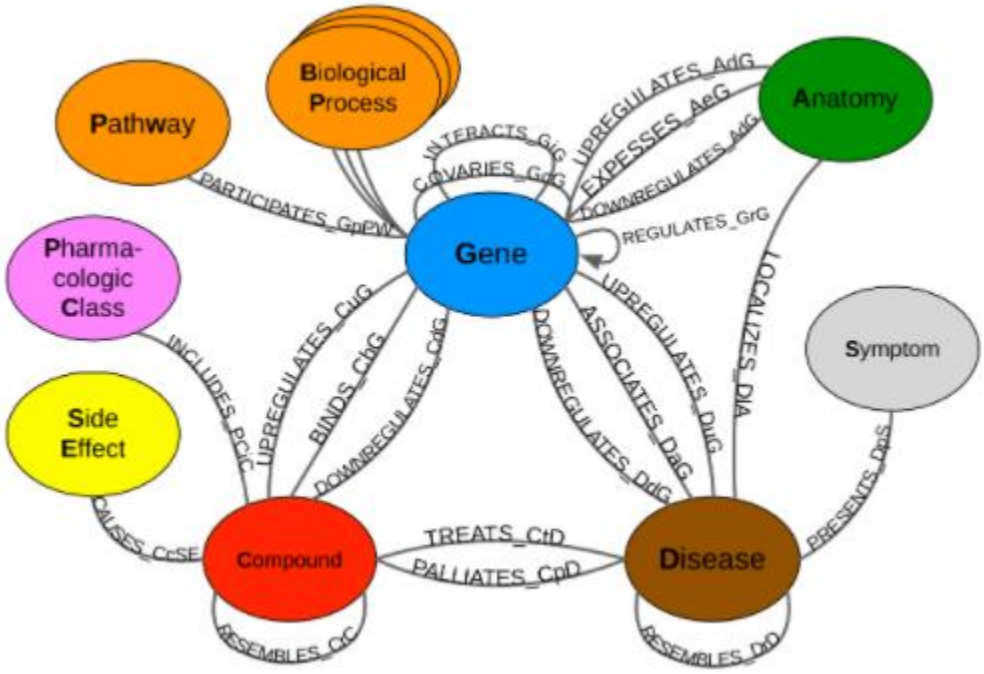

Ilustración 4 Esquema de Hetionet

Estas características la convierten en una base de datos ideal para hacer análisis, sin embargo, como sólo está disponible en línea no se pueden aplicar los algoritmos de grafos y es necesario portar la base de datos a Neo4j. Es posible hacer esto ya los desarrolladores hacen disponible un archivo JSON con todos los datos en su repositorio. Hetionet Github JSON.

Semana 6

A partir del JSON encontrado con la información es necesario cargarlo a Neo4j, entonces se descargó un cliente de Neo4j para Python para poder recorrer el archivo JSON, dar formato a los datos e introducirlos en la base de datos. Véase el anexo 1.


```
"nodes": [
  {
    "kind": "Molecular Function",
    "identifier": "GO:0031753",
    "name": "endothelial differentiation G-protein coupled receptor binding",
    "data": {
      "source": "Gene Ontology",
      "license": "CC BY 4.0",
      "url": "http://purl.obolibrary.org/obo/GO_0031753"
    }
  },
  {
    "kind": "Gene",
    "identifier": 5345,
    "name": "SERPINF2",
    "data": {
      "description": "serpin peptidase inhibitor, clade F (alpha-2 antiplasmin, pigment epithelium derived factor), member 2",
      "source": "Entrez Gene",
      "license": "CC0 1.0",
      "url": "http://identifiers.org/ncbigene/5345",
      "chromosome": "17"
    }
  }
]
```


Ilustración 5. Ejemplo de cómo se ven los nodos en el archivo JSON

Semana 7

No fue posible cargar los datos por el gran volumen y tamaño del archivo y la base de datos una vez cargada en la herramienta mostraba problemas de compatibilidad. Se decidió por un nuevo set de datos relacionado con compra de productos en Amazon.

Semana 8

Esta semana tuvo un enfoque en decidir qué categorías de algoritmos se cubrirían por el cuaderno de trabajo. Balanceando las que eran más relevantes y tuvieran mayores posibilidades de ser utilizadas en un futuro por los estudiantes. Además, se investigó sobre los algoritmos de detección de comunidades y grupos en grafos y sus posibles aplicaciones en el grafo que estamos trabajando. Me decidí por el algoritmo de cálculo del coeficiente de agrupamiento, ya que me interesaba saber si la red exhibía comportamientos de *small-world.*

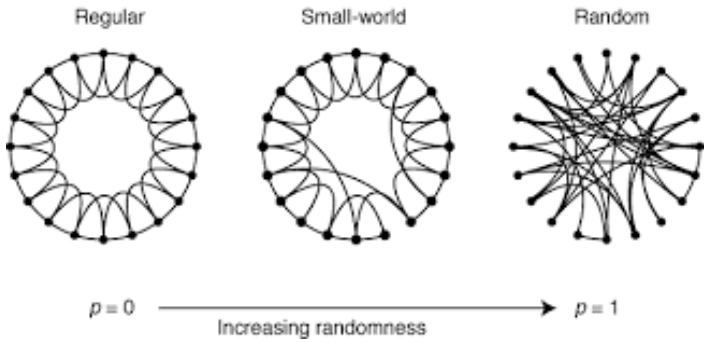

Ilustración 6 Diferentes tipos de redes clasificadas por aleatoriedad (Watts & Strogatz, 1998)

Semana 9

Esta semana se investigaron diferentes algoritmos de semejanza y se balancearon sus propiedades para encontrar el más adecuado para este trabajo. Al final me decidí por la métrica de Jaccard para todos los pares de nodos del grafo porque es más sencilla de explicar y su intuición es muy concreta de implementar. Se implementó con éxito. La métrica de Jaccard sólo es la razón de la intersección de dos conjuntos sobre su unión.

Ilustración 7 Métrica de Jaccard (Kosub, 2016)

Semana 10

Finalmente se revisaron los algoritmos de centralidad, también tratando de encontrar el más apropiado. Primero iba a tratar con el algoritmo de PageRank, pero es muy dependiente de que las relaciones entre nodos sean de distintos pesos para encontrar nodos de mayor relevancia que otros. A pesar de ser un algoritmo muy importante, decidí favorecer practicidad y realismo sobre relevancia. Por esta razón el algoritmo revisado e

implementado en el grafo fue ordenamiento por grado de los nodos, donde nodos con mayor número de relaciones entrantes y salientes tienden a ser más influyentes en una red.

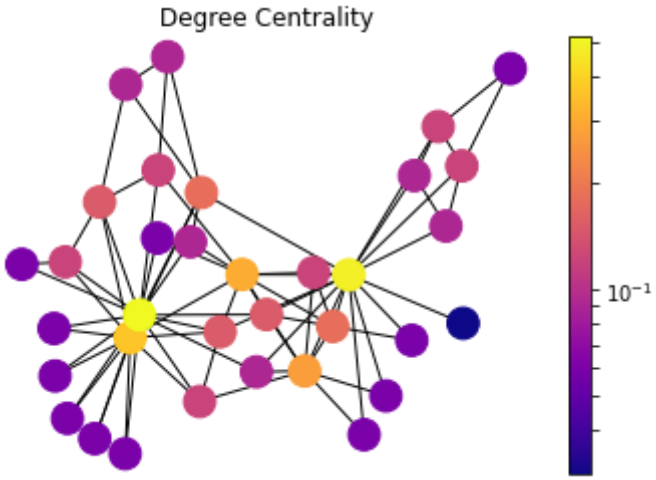

Ilustración 8 Ejemplo de centralidad de grado (Güney Aksakalli, 2017)

Semana 11

Se investigó posibles aplicaciones de combinar algoritmos de grafos para producir resultados más completos.

Semana 12

Esta semana trabajé en el desarrollo de algunas partes introductorias del cuaderno de trabajo, donde resumí diferentes fuentes teóricas y prácticas sobre el tema. Incluí investigación de otras herramientas de bases de datos nativas, y sus posibles ventajas y desventajas. Además de una muy completa introducción a realizar consultas con los algoritmos, su sintaxis, modos de ejecución y niveles de producción.

Semana 13

Esta semana fue para integrar los resultados de las semanas previas en un solo cuaderno de trabajo que incluya todos los temas previstos. Es decir, introducción al tema, a la herramienta y procesos detallados de como utilicé los respectivos algoritmos dentro de la base de datos para que los ejercicios puedan ser replicados si es necesario. Las siguientes semanas fueron dedicadas a revisiones tanto de reporte como asegurar la calidad y legibilidad del cuaderno de trabajo.

3. Resultados del trabajo profesional

1. Una base de datos adecuada a grafos lista para ser utilizada por alumnos que tomen la materia de Bases de datos para la toma de decisiones. La base de datos se ofrece en una carpeta graph.db compartida en la nube.
2. Alternativamente, se realizó un script hecho en Python que permite hacer un análisis de los datos contenidos en un JSON y los prepara para que sean agregados a la base de datos Neo4j.
3. Un cuaderno de trabajo con documentación detallada sobre la teoría de grafos y uso de algoritmos de grafos. Con su correspondencia en la librería de Grafos para Ciencia de Datos y las consultas de Cypher apropiadas.
4. Formulación de preguntas y respuestas con distintos tipos de algoritmos de grafos especializados para cada tipo de resultado esperado. Realizados con su respectiva configuración adecuado a los datos y explicación de las decisiones de configuración.

4. Reflexiones del alumno o alumnos sobre sus aprendizajes, las implicaciones éticas y los aportes sociales del proyecto

Aprendizajes profesionales

Algunas de las competencias desarrolladas gracias a este PAP fueron la planificación y la organización. Me di cuenta de lo vital que es tener objetivos bien claros y definidos desde el comienzo para que el proyecto pueda tomar un buen curso. Además de que estos objetivos estén bien definidos, una planeación realista que distribuya el trabajo correctamente durante toda la duración de un proyecto puede ser la diferencia entre un proyecto que se completa y un proyecto no satisfactorio. Otra competencia que se ejercitó mucho fue reconocer la importancia de las habilidades comunicativas, en muchas ocasiones, yo trabajé durante mucho tiempo tratando de resolver algo por mí cuenta, cuando pidiendo ayuda se podía resolver mucho más rápido.

Por otro lado, una competencia muy relacionada con mi perfil profesional que pude desarrollar fue la de buscar y analizar datos. Pasé por muchos *datasets* distintos preguntándome que información relevante podrían ofrecer o cómo sería la mejor forma de desplegar su información o si se podrían adaptar a un grafo.

Profesionalmente, también adquirí aprendizajes sobre el contexto. Lo que fue lo más importante para mí lo más importante sobre el contexto del campo profesional fue darme cuenta de que se puede aprovechar un gran cantidad de recursos para aprender sobre cualquier tema relacionado de tecnología porque en la actualidad existe mucha gente dispuesta a compartir el conocimiento y contribuir a una comunidad global, las fronteras están completamente eliminadas y cualquier día en cualquier momento puedo hacer preguntas y recibir ayuda de personas al otro lado del mundo. El set de datos con el que inicialmente estaba trabajando provenía de una investigación realizada por un sueco, que hizo disponible y accesible a quien fura que estuviera interesado en ella, mejorando así las posibilidades de que su investigación arroje nuevos resultados ahora que diferentes mentes pueden trabajar en lo mismo.

En este proyecto tuve que demostrar una buena capacidad de redacción para expresar mis ideas claramente, incluso para personas sin experiencia en este tipo de bases de datos. No obstante, mi saber más utilizado fue el de bases de datos. A pesar de que mi conocimiento

era sobre bases de datos relacionales, es fascinante darse cuenta, que todas esas tecnologías resuelven las mismas problemáticas y por lo tanto ofrecen más o menos las mismas funcionalidades, así que podía utilizar mi intuición sobre conocimientos previos y usarlos, aunque provinieran de un contexto distinto. Gracias a estos saberes, mi curva de aprendizaje en lenguaje Cypher y sus diferentes librerías no fue demasiado grande y pude adaptarme sin problemas. Además, se puso a prueba mi capacidad de tomar distintas tecnologías y combinarlas para hacer un entregable.

Gracias a este PAP mi confianza para realizar proyectos creció y siento que no tendría problemas en incorporarme a la industria gracias a esta experiencia.

Aprendizajes sociales

Gracias a este proyecto tengo una visión más social, particularmente enfocada a la educación, un tema que siempre me ha gustado mucho y dónde hay mucho por hacer aquí en México y quizá ahora que me doy cuenta de la calidad necesaria para producir un proyecto de este tipo podría participar en más.

Producir innovación en el campo de la tecnología no es sencillo; hay mucha competencia y diariamente se producen nuevas tecnologías que sobre pasan en una gran a escala a las tecnologías del día anterior. La tecnología es un gran ciclo que se retroalimenta, cada desarrollo tecnológico produce más y más desarrollo. Es por esto por lo que una innovación, aunque no sea directamente social, alguien en alguna parte del ciclo trabajando en aspectos sociales será capaz de aprovecharse de ese nuevo desarrollo. Así que de forma muy optimista digo que el desarrollo hecho en mí PAP, aunque no es directamente social, puede llegar a serlo por alguien que se beneficie de un mejor RDBMS. Ahora sé que los grandes desarrollos y cambios no suceden de la noche a la mañana, sino del esfuerzo conjunto de una gran cantidad de personas apasionadas por el cambio que puede traer la tecnología.

La intención es que este proyecto beneficié sobre todo a estudiantes y haga su proceso de aprendizaje más fluido e interesante de manera que les puede dejar conocimiento más claros y profundos.

Ahora el aprendizaje no tiene por qué limitarse a mis compañeros de universidad, creo que existe una gran demanda de educación de calidad en México y cada pequeña aportación cuenta, proyectos como este me inspiran a querer aplicar este conocimiento para producir

un bien más generalizado, quizá dar clases a comunidades de algún tema de computación sea un comienzo en la dirección correcta que acorte la brecha de oportunidades.

Este proyecto puede continuar, en primera instancia expandiendo y mejorando el contenido del cuaderno de trabajo de manera que sirva como una guía completa para hacer algoritmos de grafos.

Aprendizajes éticos

La ética personal y profesional está presente en todas las decisiones y comportamientos que tenemos constantemente. No puedo recordar un momento en particular dónde se haya puesto a prueba mi ética profesional. Sin embargo, esta experiencia me lleva a buscar mucha calidad de ingeniería y desarrollo en mi vida profesional porque me enfrenté a una calidad de trabajo y de producir software de calidad mundial. Mi profesión tiene potencial de crear sociedades más igualitarias, sin embargo, eso proviene de un esfuerzo colectivo.

Aprendizajes en lo personal

Esta experiencia me hizo darme cuenta de que aún falta mucho camino para aprender el manejo del tiempo y organización de proyectos que es posible adquirir a través de experiencia, midiendo con precisión el esfuerzo y horas requeridas para realizar una tarea.

La interacción con el profesor y otros compañeros se vio reducida por la naturaleza del proyecto, pero me hubiera gustado tener el apoyo y opinión de otros compañeros que permitiera producir un resultado más completo.

Relacionado con la manera de interactuar con mi mentor, dada la situación del COVID-19, se me otorgó la facilidad de poder colaborar de manera remota. Esto me hizo sentir muy afortunado, ya que comprendo que dentro del rubro profesional de Sistemas Computacionales es una práctica muy común por la facilidad inherente que esta área ofrece, sin embargo, muchas empresas no confían en los estudiantes siendo autónomos y cumpliendo con sus responsabilidades si no les están observando y checando constantemente. Esta situación hizo que yo mismo me comprometiera en desenvolverme de la manera más profesional y responsable posible.

Aprendí a balancear el costo de tratar de hacer un desarrollo por ti mismo y buscar soluciones ya hechas y adecuarlas. Nunca se puede estar seguro de cuando es la correcta y

hay que elegir un camino dependiendo el contexto y la situación. Anteriormente he formado parte de equipos de trabajos en empresas grandes y trasnacionales, en donde sus procesos de trabajo y organización están completamente definidos, y mis responsabilidades son muy delimitadas y específicas desde el primer día. El primer reto con el que me encontré al estar desarrollando fue lidiar con la ambigüedad y los distintos roles que tuve que fungir, como ser mi propio Administrador de proyectos, diseñador de contenido, etc. Tener que planear mis responsabilidades en lugar de solo tomar el trabajo que se me asigna fue difícil en un comienzo, pero logré adaptarme y seguir adelante con una cosa a la vez.

Personalmente, tuve muchos retos al estar trabajando este proyecto en una época tan difícil como la que vivimos durante este año. Estar colaborando y teniendo el doble de responsabilidades al estar atendiendo la escuela paralelamente, causó muchos momentos de estrés y en donde tuve que poner en práctica el control de mis emociones y técnicas de manejo del estrés

5. Conclusiones

Finalmente, y a pesar de los aspectos difíciles que tuve que sobrellevar para el desarrollo de este proyecto, puedo añadir que me gustó mucho realizarlo y finalizarlo. Desde un inicio me pareció increíble saber que estaría participando con temas de grafos, ya que esto se alinea con el enfoque que me gustaría dar a mi carrera. Considero que, como es ya bien sabido, los sistemas computacionales han revolucionado al mundo y permitido que se tenga un crecimiento exponencial durante los últimos treinta años de la historia humana. Pero aún quedan distintos sectores desatendidos que, con un pequeño impulso tecnológico, podrían ser de gran ayuda hacia los seres humanos y las bases de grafos con su versatilidad podrían ser un nuevo impulso tecnológico. Creo fuertemente que un buen sistema computacional no debería de tratarse únicamente de qué tecnologías usa y cómo estas están bien construidas, también debería de tomarse en cuenta el propósito que esta tiene, y si este beneficiará a la mayoría de sus usuarios.

En este proyecto PAP pude trabajar mi comunicación, mi creatividad y mi resolución creativa de problemas entre muchas otras habilidades que no había ejercitado tanto como me hubiera gustado en el transcurso de mi carrera. Este proyecto me dejó un crecimiento personal y profesional muy significativo que cambiará como me desenvuelvo en ambientes similares en un futuro.

Estoy muy satisfecho y orgulloso con los resultados obtenidos, fue un proyecto muy retador y creí que no iba a conseguir terminarlo porque había demasiadas cosas que no entendía, pero seguí empujándome hacia un objetivo y finalmente realicé un trabajo del que me siento feliz, porque siento que puede tener un impacto, aunque sea pequeño en los estudiantes.

6. Bibliografía

Bog, A. (2013). *Benchmarking Transaction and Analytical Processing Systems: The Creation of a Mixed Workload Benchmark and its Application.*

Codd, T. (1970). *A Relational Model of Data for Large Shared Data Banks.* San Jose: IBM.

Güney Aksakalli, C. (17 de July de 2017). *Network Centrality Measures and Their Visualization*. Obtenido de Notes from my journey: https://aksakalli.github.io/2017/07/17/network-centrality-measures-and-their-visualization.html

Himmelstein, D. (2016). *Rephetio: Repurposing drugs on a hetnet.* San Francisco: University of California.

Kosub, S. (2016). *A note on the triangle inequality for the Jaccard distance.*

Kuhlman, D. (2012). *A Python Book: Beginning Python, Advanced Python, and Python Exercises.*

Microsoft Corporation. (15 de August de 2018). *Aggregate Functions*. Obtenido de SQL Docs: https://docs.microsoft.com/en-us/sql/t-sql/functions/aggregate-functions-transact-sql?view=sql-server-ver15

Mozilla. (2019). *MDN web docs*. Obtenido de Javascript: https://developer.mozilla.org/es/docs/Web/JavaScript

Mozilla. (2019). *API*. Obtenido de MDN web docs: https://developer.mozilla.org/es/docs/Glossary/API

Mozilla. (2019). *JSON*. Obtenido de MDN web docs: https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/JSON

Needham, M. (2019). *Graph Algorithms.* Sebastopol: O'Reilly.

Neo4j. (s.f.). *Neo4j APOC Library*. Recuperado el 28 de 11 de 2020, de https://neo4j.com/developer/neo4j-apoc/

Neo4j, Inc. (s.f.). *4.2. Native projection*. (Neo4j, Inc.) Recuperado el 28 de 11 de 2020, de https://neo4j.com/docs/graph-data-science/1.1/management-ops/native-projection/

Neo4j, Inc. (s.f.). *5.4.1. Node Similarity*. (Neo4j) Recuperado el 17 de November de 2020, de 5.4.1. Node Similarity

Neo4j, Inc. (s.f.). *GDS Chapter 1. Introduction*. Recuperado el 17 de November de 2020, de https://neo4j.com/docs/graph-data-science/1.1/introduction/

Oracle. (2015). *What is a database*. Obtenido de Oracle Tutorials:

https://www.oracle.com/mx/database/what-is-database.html

Watts, D., & Strogatz, S. (1998). *Collective dynamics of 'small-world' networks.* Nature.

7. Anexos (en caso de ser necesarios)

Anexo A. Script de Python para interpretar Hetionet JSON.

```
import ijson
import json
import re
from neo4j import GraphDatabase

class Neo4jClientDB:

    def __init__(self, uri, user, password):
        self.driver = GraphDatabase.driver(uri, auth=(user, password))

    def close(self):
        self.driver.close()

    def creat_node(self, nodeKind, id, name, data):
        with self.driver.session() as session:
            log = "Not equal to any kind"
            log = session.write_transaction(self._create_and_return_node, no
deKind, id, name, data)
            print(log)

    def create_relationship(self, relationType, sourceId, targetId, data):
        with self.driver.session() as session:
            log = "Not equal to any kind"
            log = session.write_transaction(self._create_and_return_relation
ship, relationType, sourceId, targetId, data)
            print(log)


    @staticmethod
    def _create_and_return_node(tx, kind, id, name, data):
        queryStr =  'CREATE (g:{kind_}) '.format(kind_=kind.replace(" ", "_"
))
        queryStr += 'SET g.identifier = "{id_}" '.format(id_=id)
        queryStr += 'SET g.name       = "{name_}" '.format(name_=name)
        if 'chromosome' in data:
```

```
            queryStr += 'SET g.chromosome = "{cr}" '.format(cr=data['chromos
ome'])

        if 'description' in data:
            queryStr += 'SET g.description = "{desc}" '.format(desc=data['de
scription'])

        if 'source' in data:
            queryStr += 'SET g.source = "{src}" '.format(src=data['source'])

        if 'url' in data:
            queryStr += 'SET g.url = "{url_}" '.format(url_=data['url'])

        if 'inchikey' in data:
            queryStr += 'SET g.inchikey = "{inc}" '.format(inc=data['inchike
y'])

        if 'mesh_id' in data:
            queryStr += 'SET g.mesh_id = "{mesh}" '.format(mesh=data['mesh_i
d'])

        if 'class_type' in data:
            queryStr += 'SET g.class_type = "{cls}" '.format(cls=data['class
_type'])

        queryStr += 'RETURN "{kind_} " + g.name + " created succesfully "'.f
ormat(kind_=kind)

        try:
            result = tx.run(queryStr)
        except:
            print("---------
- A terrible exception was hit involving this query " + queryStr)
            return "help"

        return result.single()[0]



if __name__ == "__main__":
    #neo4jClient = Neo4jClientDB("bolt://localhost:11004", "neo4j", "1234")
    nodes = True

    if nodes:
        with open('hetionet-v1.0.json') as json_file:
```

```
            parser = ijson.parse(json_file)
            tempDict = {}
            for prefix, event, value in parser:
                if prefix == 'nodes' and event == 'end_array':
                    break

                if prefix == 'nodes.item' and event == 'start_map':
                    tempDict = {}
                    tempDict['data'] = {}

                if prefix == 'nodes.item' and event == 'end_map':
                    neo4jClient.creat_node(tempDict['kind'], tempDict['ident
ifier'], tempDict['name'], tempDict['data'])

                if re.search("nodes\.item\.[^d]", prefix):
                    x = prefix.split(".")
                    tempDict[x[2]] = value
                    # print(prefix)

                if re.search("nodes\.item\.data\.+", prefix):
                    x = prefix.split(".")
                    tempDict['data'][x[3]] = value
    else:
        with open('mini-hetionet.json') as json_file:
            parser = ijson.parse(json_file)
            tempDict = {}
            flag = False
            for prefix, event, value in parser:
                if prefix !='edges':
                    flag = True
                if flag:
                    print(f'prefix: {prefix}, event:{event}, value:{value}')

    neo4jClient.close()
    print("Program finished succesfully")
```

Anexo B. Código Cypher

```
// ------------------------------------------------------------------//
//*                                                                  //
//*              Algoritmos de similaridad.                            //
//         Node similarity                                        //
// ------------------------------------------------------------------//

Match(d1:Disease{name:"type 2 diabetes mellitus"})-[dps1:PRESENTS_DpS]-
>(s1:Symptom)
```

```
WITH d1, collect(id(s1)) AS d1Symptoms
Match(d2:Disease{name:"type 1 diabetes mellitus"})-[dps2:PRESENTS_DpS]-
>(s2:Symptom)
WITH d1, d1Symptoms, d2, collect(id(s2)) As d2Symptoms
RETURN d1.name as from,
       d2.name as to,
       gds.alpha.similarity.jaccard(d1Symptoms, d2Symptoms) as Jaccard_Simil
arity

// Estimate memory
CALL gds.nodeSimilarity.stream.estimate({
    nodeProjection: ["PRODUCT", "CUSTOMER"],
    relationshipProjection: {
        REVIEW : {
            type: "REVIEW",
            orientation: "NATURAL"
        }
    },
    topK : 1,
    topN : 200,
    degreeCutoff : 10,
    similarityCutoff : 0.5
})
YIELD nodeCount, relationshipCount, bytesMin, bytesMax, requiredMemory

// node similarity computed in stream mode
CALL gds.nodeSimilarity.stream({
    nodeProjection: ["PRODUCT", "CUSTOMER"],
    relationshipProjection: {
        REVIEW : {
            type: "REVIEW",
            orientation: "NATURAL"
        }
    },
    topK : 1,
    topN : 200,
    degreeCutoff : 10,
    similarityCutoff : 0.5
})
YIELD node1, node2, similarity
WHERE similarity < 1.0
RETURN gds.util.asNode(node1).id AS Customer1, gds.util.asNode(node2).id AS
Customer2, similarity
```

```
// ------------------------------------------------------------------//
//*                                                                  //
//*             Algoritmos de comunidad.                             //
//         Triangles   count                                        //
// ------------------------------------------------------------------//

/// give me all the triangles
CALL gds.alpha.triangle.stream({
    nodeProjection: "PRODUCT",
    relationshipProjection : {
        COPURCHASING : {
            type: "COPURCHASING", //o similar
            orientation: "UNDIRECTED"
        }
    }
})
YIELD nodeA, nodeB, nodeC
WHERE gds.util.asNode(nodeA).MINST = 20 or
      gds.util.asNode(nodeB).MINST = 20 or
      gds.util.asNode(nodeC).MINST = 20
RETURN gds.util.asNode(nodeA).titulo AS nodeA,
       gds.util.asNode(nodeB).titulo AS nodeB,
       gds.util.asNode(nodeC).titulo AS nodeC


// Computed Tomography : Fundamentals, System Technology, Image Quality, App
lications

// Number of triangles per node ordered
CALL gds.alpha.triangleCount.stream({
    nodeProjection: "PRODUCT",
    relationshipProjection : {
        COPURCHASING : {
            type: "COPURCHASING",
            orientation : "UNDIRECTED"
        }
    }
})
YIELD nodeId, triangles
RETURN gds.util.asNode(nodeId).titulo AS Título, triangles
ORDER BY triangles DESC

// Coeficient.
CALL gds.alpha.triangleCount.stream({
    nodeProjection: "PRODUCT",
```

```
    relationshipProjection : {
        COPURCHASING : {
            type: "COPURCHASING",
            orientation : "UNDIRECTED"
        }
    }
})
YIELD nodeId, triangles, coefficient
RETURN percentileCont(coefficient, 0.5) as Clustering_Coefficient_Median,
       avg(coefficient) as Clusternig_Coefficient_Mean

// ------------------------------------------------------------------//
//*                                                                  //
//*              Algoritmos de centralidad                            //
//                                                                   //
// ------------------------------------------------------------------//

// stream mode the most co purchased product
// since this are bidirectional relationships it's not necesarry to

// get the 99th percentile of useful information
CALL gds.alpha.degree.stream({
    nodeProjection: "PRODUCT",
    relationshipProjection : {
        COPURCHASING : {
        type : 'COPURCHASING',
        orientation : 'REVERSE'
        }
    }
})
YIELD nodeId, score
RETURN percentileCont(score, 0.99) as Copurchases

// specify order.
CALL gds.alpha.degree.stream({
    nodeProjection: "PRODUCT",
    relationshipProjection : {
        COPURCHASING : {
        type : 'COPURCHASING',
        orientation : 'REVERSE'
        }
    }
})
YIELD nodeId, score
WHERE score > 19.0
```

```
RETURN gds.util.asNode(nodeId).titulo as Título, score as Copurchases
ORDER BY score DESC



// Betweeness centrality
CALL gds.alpha.betweenness.stream({
    nodeProjection: "PRODUCT",
    relationshipProjection : "COPURCHASING"
})
YIELD nodeId, centrality
RETURN gds.util.asNode(nodeId).titulo as Título, centrality as Betweeness
ORDER BY centrality DESC
```