UNIVERSIDAD
DE LA REPUBLICA
URUGUAY

Análisis de Interacciones en Entornos Educativos mediante Bases de Datos Orientadas a Grafos y Herramientas de Analítica del Aprendizaje Social

Bartolomé Estol Diego Julio, Carriquiry Gorlero Juan José,

Mendes Lopes Barba Mathias Nicolás

Programa de Grado en Ingeniería en computación
Facultad de Ingeniería
Universidad de la República

Montevideo – Uruguay
Noviembre de 2020

UNIVERSIDAD
DE LA REPUBLICA
URUGUAY

Análisis de Interacciones en Entornos Educativos mediante Bases de Datos Orientadas a Grafos y Herramientas de Analítica del Aprendizaje Social

Bartolomé Estol Diego Julio, Carriquiry Gorlero Juan José,

Mendes Lopes Barba Mathias Nicolás

Informe de Proyecto de Grado presentado al Tribunal Evaluador como requisito de graduación de la carrera en Ingeniería en computación.

Tutora:

Dra. Libertad Tansini

Montevideo – Uruguay
Noviembre de 2020

Bartolomé Estol Diego Julio, Carriquiry Gorlero Juan José, Mendes Lopes Barba Mathias Nicolás

/ Montevideo: Universidad de la República, Facultad de Ingeniería, 2020.

XII, 116 p.: il.; 29, 7cm.

Tutora:

Libertad Tansini

Tesis de Grado – Universidad de la República, Programa en Ingeniería en computación, 2020.

Referencias bibliográficas: p. 91 – 97.

1. Aprendizaje Social, 2. Base de datos de grafos, 3. Learning Management System (LMS), 4. Neo4j, 5. Evimed. I. Tansini, Libertad , . II. Universidad de la República, Programa de Posgrado en Ingeniería en computación. III. Título.

INTEGRANTES DEL TRIBUNAL

MSc. Ewelina Bakala

Dra. Adriana Marotta

MSc. Sandro Moscatelli

Montevideo – Uruguay
Noviembre de 2020

Agradecimientos

Quisiéramos agradecer a nuestra tutora de proyecto de grado Libertad Tansini y a Antonio López representante de la institución educativa EviMed por habernos brindado su tiempo a lo largo del trabajo. Y por último a nuestras familias que nos han acompañado desde el inicio de nuestras carrera.

RESUMEN

El presente proyecto de grado se centra en la investigación de plataformas educativas que almacenan información valiosa proveniente de interacciones que se dan entre distintos actores que intervienen en los procesos de aprendizaje y que habitualmente son desarrolladas sobre bases de datos relacionales. En este trabajo de grado se busca brindar herramientas que permitan visualizar las interacciones de forma más natural y realizar análisis más profundos haciendo uso de bases de datos orientadas a grafos y herramientas de Aprendizaje Social.

La implementación incluye el trabajo de migrar los datos que están en la plataforma en una base de datos relacional a una de grafos. Para esto se debió primero seleccionar una base de datos orientada a grafos con una infraestructura escalable. Luego, definir un modelo y cargar los datos desde un modelo relacional hacia el nuevo modelo orientado a grafos. Para posteriormente ejecutar primitivas y algoritmos a demanda con el fin de aportar al análisis del Aprendizaje Social. Y por último, obtener información visual que agrega valor, brindando un análisis gráfico de los resultados obtenidos.

Con el objetivo de validar el trabajo y el potencial de las herramientas investigadas y construidas se ejecutan algoritmos y consultas sobre grafos con un importante volumen de datos, obteniendo tiempos de ejecución buenos y realizando un posterior análisis de los resultados obtenidos, logrando contextualizarlos en la realidad de la plataforma de la institución educativa EviMed, dando a conocer el potencial y el valor agregado en el área de Aprendizaje Social.

Palabras claves:
Aprendizaje Social, Base de datos de grafos, Learning Management System (LMS), Neo4j, Evimed.

Tabla de contenidos

Lista de figuras

Lista de tablas

Capítulo 1

Introducción

El uso de las plataformas educativas en el aprendizaje ha ido cobrando mayor importancia en la última década [2], siendo partícipes fundamentales en la educación universitaria de muchos estudiantes en diversos países del mundo. En Uruguay varias instituciones utilizan Moodle [42] como plataforma educativa, entre las que se pueden mencionar: la Universidad de la República, la Administración Nacional de Educación Pública, la Universidad ORT, la Universidad Católica, la Intendencia de Montevideo, Presidencia de la República, la Universidad del Trabajo de Uruguay, la Universidad de Montevideo, la Administración Nacional de Telecomunicaciones, entre otras [64]. A consecuencia de esto se incorpora un nuevo concepto para definir el modelo de aprendizaje, surgiendo con gran fuerza una alfabetización digital en la que se necesita de habilidades tecnológicas. Con la finalidad de ser autosuficientes, se requiere cierto nivel de conocimiento para poder navegar por variadas fuentes de información, tener la capacidad de discriminar la información recibida y en mayor medida poder contar con el dominio de la sobrecarga de la información brindada a través de la web. Sumado a esto, y tomando en consideración la velocidad con la cual avanzan las distintas tecnologías hoy en día [31], en particular las relacionadas con internet e informática, las plataformas de teleformación (LMS-Learning Management System, que podría traducirse como sistemas para la gestión de aprendizaje) [65] se tornan de gran interés.

Es de suma importancia para las organizaciones, docentes y estudiantes, dominar y comprender estas plataformas dado que cada vez son utilizadas con más frecuencia a la hora de mejorar, compartir y evaluar el aprendizaje en diversos contextos.

Las plataformas LMS almacenan información valiosa que permite identificar interacciones de los usuarios a medida que se da el proceso de aprendizaje dentro de las mismas, por ejemplo, entre estudiantes y docentes de un curso, o de los estudiantes entre sí por medio de los foros disponibles para el curso. Situaciones como la anterior, generan a nivel de modelo de datos de la plataforma, redes que se dan a partir de la interacción o relación que tienen las distintas entidades (estudiantes, docentes, cursos, foros, mensajes, etc..) presentes en el proceso de aprendizaje en la plataforma. Esta información que se genera, es importante en las distintas jerarquías del proceso educativo. Permite el desarrollo de funcionalidades que pueden aportar desde el punto de vista analítico y ofrecerlas como tal al usuario final. Por otro lado, es de interés para docentes analizar información, como pueden ser: qué impacto tienen sobre los estudiantes, qué tipos de relaciones se dan entre ellos, si son positivas o no, cómo es el comportamiento de los alumnos dentro de cada curso, entre otros posibles análisis.

El estudio y la posterior toma de decisiones puede realizarse a distintos niveles, como por ejemplo, a nivel individual, a nivel de actividades de cada curso, a nivel del curso propiamente dicho, de la institución o incluso por regiones y países.

Es aquí donde surge la necesidad de contar con herramientas que permitan aportar datos analíticos desde el punto de vista visual y gráfico de las interacciones. Este tipo de análisis logra la detección de situaciones de riesgos en el aprendizaje, como por ejemplo, el posible abandono del curso o situaciones de aislamiento. También pueden aportar información sobre estrategias con resultados exitosos para volver a aplicarlas posteriormente, tener posibilidades de ver comunidades y usuarios que son importantes e influyen en el aprendizaje del resto. Entre muchas otras conclusiones que se desprenden de este tipo de información según las necesidades de los diferentes actores.

Este trabajo de grado busca implementar una solución que permita a aquellas plataformas, que poseen información correspondiente a las interacciones de los usuarios en un modelo de datos relacional, realizar análisis de esta migrando a un modelo de grafos. Esto permite ejecutar diversos algoritmos y consultas que no serían posibles de no contar con un modelo de estas características. El problema a resolver cuenta con los siguientes requisitos:

- Migrar desde una base de datos relacional a una de grafos:

- Construir una nueva base de datos en una infraestructura escalable.
- Cargar datos desde un modelo relacional a un nuevo modelo orientado a grafo.
- Lograr ejecutar primitivas y algoritmos a demanda.
- Obtener información visual que agregue valor.
- Brindar un análisis de los resultados obtenidos.

Para llevar adelante el proyecto de grado, en primera instancia se realiza una exhaustiva investigación que abarca varios frentes de estudio con foco en entender y a su vez aclarar las necesidades reales de la institución educativa EviMed. Seguido a esto, se utiliza una infraestructura escalable para alojar cada uno de los artefactos involucrados en el desarrollo de esta solución, como lo es la base de datos orientada a grafos. Dado a la importancia de esto último se realiza un análisis de diversas bases de datos orientadas a grafos, respondiendo a la pregunta ¿cuál de ellas es mejor para este propósito?. A partir de esta elección, se crea el flujo de carga de datos de forma que sea lo más automatizado posible. Se investiga la posibilidad de visualizar los resultados mediante una interfaz independiente de la provista por la base de datos para poder generar una animación visual del grafo, que brinde una buena manipulación de una red con gran cantidad de nodos y aristas. A su vez, esta alternativa debe permitir ser integrada de forma fácil y configurable. Esto último es importante pensando en integraciones con plataformas y sitios que quieran mostrar estos grafos.

Luego, se implementa la ejecución de los algoritmos y se diseñan las consultas utilizadas, a modo de entender y explotar los datos representados en forma de grafo. Se desarrollan evaluaciones e implementan ejemplos de casos reales. Posteriormente se realiza un análisis más profundo, brindando una contextualización de los datos.

Las principales contribuciones realizadas en el marco de este proyecto son:

- La implementación de una solución que permite la extracción, normalización, carga de una base de datos de grafos en una infraestructura escalable, a partir de una base de datos relacional.
- La ejecución de algoritmos y consultas sobre grafos de forma intuitiva e integrable con otros sistemas.
- Estudio y contextualización de datos analíticos en base a experimentación de los resultados obtenidos.

- Una base importante que permite analizar los procesos de aprendizaje y permite continuar estudios en áreas de inteligencia artificial aplicada.

El informe se organiza de la siguiente manera. El capítulo 2 expone los principales fundamentos teóricos utilizados en este proyecto. En el capítulo 3 se presenta un resumen de la importancia que tienen las plataformas actuales, y las principales tecnologías que las componen. Así como también se muestran dos ejemplos de plataformas.

Luego en el capítulo 4 se describe la metodología seguida para el desarrollo del proyecto conjunto con diagramas que muestran el trabajo planificado.

En el capítulo 5 se expone la realidad de la institución educativa EviMed con respecto a su plataforma de aprendizaje redEMC, se menciona contenido y funcionamiento actual de la misma y su necesidad de contar con herramientas que permitan el estudio de las interacciones en forma visual y logren explotar mejor los datos. En el capítulo 6 se expone paso a paso el desarrollo e implementación de cada uno de los componentes necesarios para llegar al resultado deseado. Finalmente en el capítulo 7 se describen casos reales en base a los datos contenidos en la plataforma de EviMed y los resultados de la solución desarrollada en el presente proyecto. También se dan a conocer datos relacionados a los tiempos de ejecución de los componentes que componen la solución. Para terminar, en el capítulo 8 se mencionan las conclusiones finales del proyecto y el trabajo a futuro teniendo como partida la investigación y el trabajo realizado. Adicionalmente, se adjuntan dos anexos que explican dos puntos claves en el desarrollo del trabajo.

Capítulo 2

Marco teórico

En esta sección se exponen definiciones relacionadas a grafos y redes que más adelante en el documento forman parte importante para entender el desarrollo del proyecto de grado. Gran parte de estas definiciones son expuestas tal cual se describe "Detección de comunidades en redes: Algoritmos y aplicaciones"[6], ya que utiliza las definiciones que busca describir esta sección.

2.1. Grafos

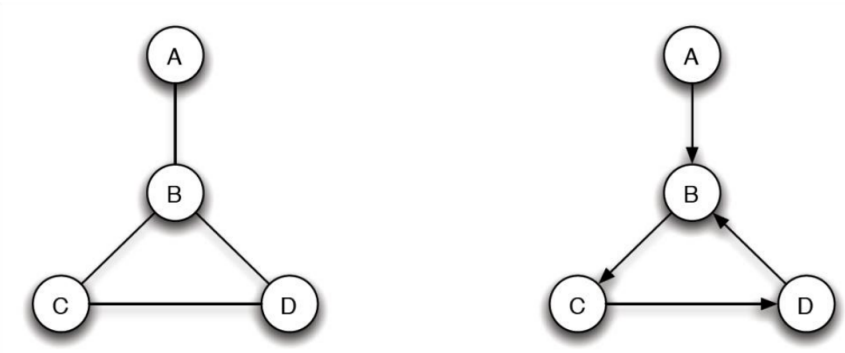
Un **grafo no dirigido** es un par, E donde es un conjunto (finito) de vértices o nodos, es un conjunto de pares no ordenados de elementos distintos de V. A estos pares se les llama **aristas**.

Existen situaciones en las cuales es necesario modelar más de una arista por cada par de nodos. Esta situación se conoce como multígrafo. Al igual que un grafo, un multígrafo consta de nodos y aristas, pero con la diferencia de que no existe restricción en la existencia de aristas múltiples entre pares de nodos.

Cuando existe una arista para conectar un nodo consigo mismo, esta arista se conoce como **lazo**, pero con un grafo simple o un multígrafo no es posible modelarlo. Para esto se utiliza un **pseudografo** [58].

Por el contrario, un **grafo dirigido** $G = (V, E)$ consiste en un conjunto no vacío V de nodos y de un conjunto E de aristas dirigidas. Esta dirección es asociada con un par ordenado de nodos. Una **arista dirigida** es asociada con un par ordenado (u, v) donde el nodo u es el inicio y v es el final de la arista. Por lo general, un grafo dirigido se representa de la siguiente manera

$\underset{G}{\rightarrow} = (V, \underset{E}{\rightarrow})$.



a) Grafo no dirigido 4 nodos. b) Grafo dirigido 4 nodos.

Figura 2.1: Ejemplo de grafo dirigido y no dirigido

2.1.1. Adyacencia

Sea un grafo $G = (U, V)$. Un nodo $u \in V$ es **adyacente** o **vecino** a un nodo $v \in V$ si y solo si hay un arista entre ellos; es decir, si y solo si $(u, v) \in E$. El conjunto de vecinos de i es

$$N(u) = v \in V | (u, v) \in E \tag{2.1}$$

Si $e = u, v$, se dice que la arista e es **incidente** en los nodos u y v. También se dice que la arista e **conecta** u y v.

2.1.2. Peso

Existen casos en los cuales las redes no sólo representan conexiones simples de tipo booleano. Algunas veces es necesario modelar un peso, fuerza o valor asociado a la conexión. Por ejemplo en un modelo de una red de comunicaciones, los nodos representan terminales y enrutadores, mientras que las aristas representan las conexiones y un valor asociado a cada arista puede representar el ancho de banda del que se dispone en esa conexión [51].

El **peso** de una arista representa un valor intrínseco a ella. Cuando en un grafo se modelan los pesos, este se denomina **grafo ponderado**.

2.1.3. Grado

El **grado** o **conectividad** de un nodo de un grafo es el número de aristas incidentes en él, cada arista contribuye con dos unidades al grado de los nodos que conecta y por lo general se denota k_i o $deg(i)$ [58].

Esta medida es de gran importancia en varios métodos de detección de comunidades, ya que permite comprender de una manera la relación entre los nodos y el número de vecinos adyacentes al nodo.

De manera formal el grado de un nodo se puede expresar como:

$$deg(i) = |N(i)| \tag{2.2}$$

El conjunto $N(i)$ es algunas veces llamado la **frontera** o **límite** de v. El límite de v junto con v es llamado el **cierre** de v.

$$cierre(v) = N(v) \cup \{v\} \tag{2.3}$$

A los nodos de grado cero,$deg(i) = 0$, se les llama **aislados**. **Si** $deg(i) = 1$, se les denomina nodos **colgantes** u **hojas**.

2.1.4. Matriz de adyacencia

La **matriz de adyacencia** permite representar, para un grafo no dirigido $G = (V, E)$ con un conjunto de nodos $V = v_1, ..., v_n$, las parejas de nodos que se encuentran interconectados. La matriz $A = [a_{i,j}]$ es de tamaño nxn, donde n es el número total de nodos [37].

La matriz de adyacencia se define como:

$$a_{i,j} = \begin{cases} 1, & si\ existe\ una\ arista\ entre\ el\ nodo\ i\ y\ el\ nodo\ j \\ 0, & en\ caso\ contrario \end{cases}$$

De la misma manera, un grafo ponderado se puede representar mediante una matriz de adyacencia [51].

$$a_{i,j} = \begin{cases} w, & si\ existe\ una\ arista\ con\ peso\ entre\ el\ nodo\ i\ y\ el\ nodo\ j \\ 0, & en\ caso\ contrario \end{cases}$$

2.1.5. Caminos

Si un par de nodos (i, j) no adyacentes pueden ser conectados por medio de una secuencia de m aristas $(i, l_1), (l_1, l_2), ..., (l_{m-1}, j)$ este conjunto de aristas describe un **camino** entre el nodo i y el nodo j, donde m es la **longitud del camino**.

Se puede decir que dos nodos se encuentran **conectados** si existe al menos un camino entre ellos.

Si todos los nodos y aristas que componen un camino entre dos nodos i y j no se repiten, el camino se conoce comúnmente como **camino simple** o **path**.

Un **bucle** se define como un camino donde el nodo inicial y final es el mismo nodo y se pasa una única vez por cada uno de los $m - 1$ nodos intermedios [14].

2.1.6. Subgrafos

Sea $G = (V, E)$ un grafo no dirigido. $G_x = (X, E_x)$ es un **subgrafo** de G (inducido por X) si y solo si $X \subseteq V$ y $E_x \leq (X * X) \cap E$; es decir, si contiene un subconjunto de nodos de G y las aristas correspondientes [9].

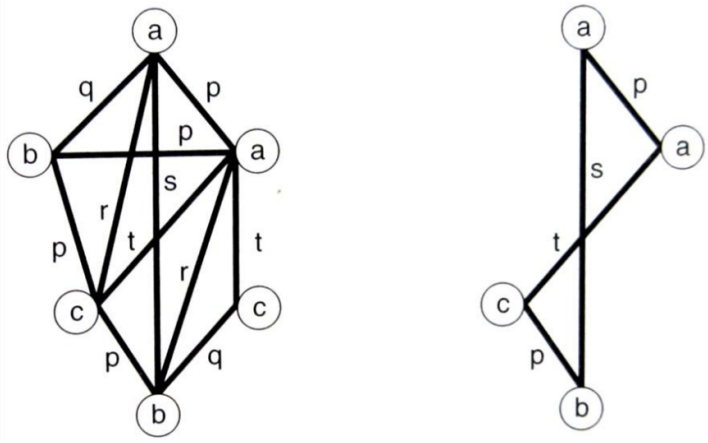


Figura 2.2: Ejemplo de subgrafo.

2.1.7. Comunidades

Uno de los problemas fundamentales de cara al desarrollo de técnicas de detección de comunidades es definir de manera cuantitativa una comunidad en un grafo. Debido a que no existe una definición universal para este problema, la

caracterización de las comunidades debe realizarse dependiendo del problema analizado. De manera intuitiva, podríamos definir una comunidad como aquel conjunto de nodos interconectados por una mayor cantidad de aristas que los demás nodos [27].

La definición del término "comunidad" no es una definición absoluta, depende desde el punto de vista desde el cual se estudia. Desde un punto de vista social, una comunidad se define como un conjunto de personas con intereses o actividades en común que formen parte de una sola red [61].

Una definición es la propuesta por Newman y Girvan: "una comunidad es la división de los nodos de una red en grupos dentro de los cuales las conexiones son densas, pero entre ellos son escasas" [52], esta definición se puede ver en la Figura 2.3.

Desde un punto de vista matemático, basado en la definición anteriormente vista, Papadopoulos define una **comunidad** C como un subgrafo de un grafo $G = (V, E)$, el cual comprende un conjunto de nodos que se encuentran interconectados más densamente que el resto de nodos, debido a que comparten un interés común.

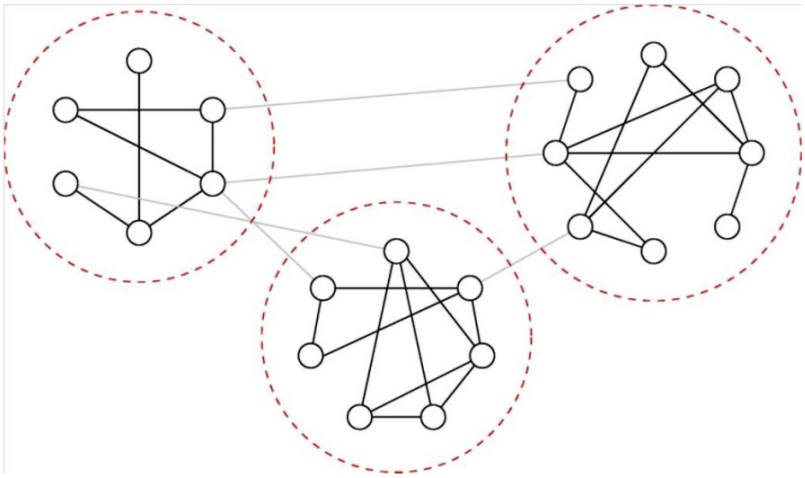


Figura 2.3: Ejemplo de comunidades en un grafo

En la **Figura 2.3** se describe gráficamente el concepto de Girvan y Newman, así como también el de Papadopoulos donde en el grafo se pueden identificar claramente tres comunidades, las cuales tienen una alta interconexión entre sus nodos y las aristas que conectan a las comunidades se encuentran en una menor cantidad. Varios de los métodos de detección de comunidades se centran en resaltar estas características.

Las comunidades **explícitas** son aquellas que se construyen a partir de la intervención humana, es el humano quien decide los enlaces entre los nodos de la red. Ejemplos de este tipo de comunidades son Facebook, Google+, LinkedIn o Flickr.

Las comunidades **implícitas** son aquellas donde se supone la existencia de una comunidad en un conjunto de datos y el sistema es el encargado de "descubrir" la existencia de esas comunidades. La investigación en el desarrollo de técnicas que permitan descubrir de manera automática las comunidades, ofrece ventajas al no requerir la intervención humana en su elaboración [61].

A partir de la idea descrita anteriormente, donde una comunidad se caracteriza por una mayor cantidad de conexiones entre sus nodos con respecto a los de fuera, varias definiciones se han ideado desde el campo de las ciencias humanas y, de manera más reciente, desde la física y las ciencias de la computación, distinguiéndose tres tipos de definición, local, global y basados en la similitud de nodos.

Definición local

Desde el punto de vista local, una comunidad es un conjunto con mayor conexión entre los nodos dentro, que entre los de fuera de la comunidad, hasta el punto que se pueden analizar de manera independiente del resto de la red. El estudio de las comunidades basadas en este concepto centran sus estudios en el subgrafo y los nodos vecinos de manera inmediata, pero dejando de lado el resto de la red.

Definición global

El análisis de redes también se puede realizar sobre todo su grafo en conjunto. Estos casos se presentan cuando las comunidades no pueden "aislarse" del resto de la red, ya que forman parte fundamental de su estructura. La idea fundamental parte de la comparación de la red analizada con el modelo de una red aleatoria. Por definición, una red aleatoria no tiene una estructura de comunidad en su interior. Si se presentan diferencias entre la red analizada y la aleatoria, se puede inferir que existe algún tipo de comunidad.

2.1.7.1. Definición basada en similitud

Es de suponer que los integrantes de una comunidad tienen características similares. Esta similitud se puede calcular mediante alguna medida ya sea que los nodos estén conectados por una arista o no. Entre las medidas más comunes

de similitud podemos nombrar la distancia euclídea, la distancia Manhattan o la similitud del coseno, entre otras muchas.

Pero las comunidades no se definen únicamente por las similitudes entre nodos. También se emplean otro tipo de métricas, como el número de caminos independientes entre dos nodos o el flujo máximo que puede transitar entre los dos nodos. Este flujo máximo se puede determinar con ayuda del teorema del flujo máximo / corte mínimo, en tiempo polinómico [27] [23].

2.2. Medidas y algoritmos en redes y comunidades

Una de las áreas importantes en el estudio de las redes y comunidades es determinar qué características se deben resaltar de los grupos a encontrar. Estas características pueden ser de centralidad, transitividad, entre otras. Muchas de estas medidas se originan en los primeros estudios sobre redes sociales, elaborados por sociólogos y demás profesiones afines a las ciencias humanas, que hoy en día se han complementado con otros enfoques como la física, las ciencias de la computación o la biología [6].

2.2.1. Distancia

En un grafo $G = (V, E)$ la **distancia** entre dos nodos i y j se define como el camino más corto existente entre ellos. Si $i = j$, entonces $d(i, j) = 0$. Si no existe un camino posible entre i y j, entonces $d(i, j) = \infty$ [14].

Si el grafo es ponderado la distancia se calcula como la suma de los pesos asociados a las aristas del camino [15]:

$$d(i, j) = \sum w_{i,j} \tag{2.4}$$

Un **camino geodésico** es el camino más corto que puede existir entre un par de nodos que se encuentren conectados por medio de un camino [15].

La **distancia media o camino medio** mide la distancia promedio que existe entre un par de nodos i y j que se comunican por el camino geodésico, de manera formal se define como [14]:

$$d_m(G) = \frac{1}{\mathrm{n(n-1)}} \sum_{i,j} d(i,j) \tag{2.5}$$

El **diámetro** en una red es la máxima distancia que puede existir entre dos nodos i y j de entre todos los nodos del grafo que estén unidos por un camino geodésico. Esta medida es de gran importancia para establecer el tamaño de la red [15].

$$Diámetro(G) = \mathrm{máx}\, d(i,j)) \forall i,j \in V$$

Para calcular su valor, se deben calcular todos los caminos geodésicos para cada par de nodos existentes en la red y, de ellos, escoger el de mayor longitud.

2.2.2. Densidad

La **densidad** $\delta(G)$ es la relación entre el número de aristas presentes en el grafo (m) y el número máximo de aristas que puede tener el grafo, cuyo valor se calcula a partir del número de nodos presentes (n) [53].

De manera formal, se expresa como:

$$\delta(G) = \frac{\mathrm{m}}{\mathrm{n(n-1)}} \tag{2.6}$$

En un grafo completo, la densidad presente es máxima, $\delta(G) = 1$.

La **densidad intra-cluster** $\delta_{int}(C)$ es la relación entre un número de aristas internas y el número aristas que podrían existir dentro de un cluster. [27].

$$\delta_{int}(C) = \frac{\#aristas\ internas\ de\ C}{\frac{n_c(n_c-1)}{2}} \tag{2.7}$$

Recíprocamente, se define la **densidad inter-cluster** $\delta_{ext}(C)$ como la relación entre el número de aristas que conectan el grupo con el resto del grafo y el número de todas las posibles aristas [27].

$$\delta_{int}(C) = \frac{\#aristas\ externas\ de\ C}{n_c(n-n_c)} \tag{2.8}$$

2.2.3. Medidas de centralidad

Estas medidas de centralidad [29] [38] se refieren a una familia de medidas estructurales calculadas para encontrar la posición o importancia de un nodo

o vértice dentro de un grafo dado. Estas medidas han sido usadas de manera exitosa para descubrir la relevancia de vértices en distintos tópicos como el Análisis de Redes Sociales [63], los Sistemas de Recuperación de Información [10] o las herramientas de Procesamiento del Lenguaje Natural [40] entre otros.

Las medidas de centralidad ayudan a evaluar los nodos en busca de aquellos que sean más importantes o céntricos en una agrupación. El concepto de la centralidad se inicia en los estudios de comportamiento humano y sus relaciones; basándose en el uso de conceptos matemáticos, estadísticos y de teoría de grafos [30].

Hay diferentes tipos de medidas y algoritmos para la centralidad, a continuación se explicarán algunos de ellos:

- **Centralidad de grado (degree centrality):** Es definida como el número de aristas incidentes a un vértice dentro de un grafo dado.
  Se denota como k_i donde i representa el nodo. De manera formal, se expresa de la siguiente forma a partir de la matriz de adyacencia del grafo:

  $$k_i = \sum_{j=1}^{n} a_{i,j}$$

  El grado medio de un grafo se determina como:

  $$c = \frac{1}{n} \sum_{j=1}^{n} k_i$$

  Existen dos subversiones de este tipo de centralidad, la centralidad de **grado de entrada (in-degree)**, la cual se refiere al número de aristas que apuntan a un vértice dado, y la centralidad de **grado de salida (out-degree)**, en la cual se mide el número de aristas que apuntan a otros vértices en el grafo desde un vértice dado. Mientras mayor sea la centralidad de grado de un nodo, mayor será el grado de influencia en la red [43].

- **Centralidad de cercanía (closeness centrality):** La centralidad de cercanía es una forma de detectar nodos que pueden difundir información de manera muy eficiente a través de un grafo, consiste en medir su distancia promedio (distancia inversa) a todos los demás nodos. Los nodos con un alto valor de cercanía tienen las distancias más cortas a los

nodos del resto del grafo. Para cada nodo, el algoritmo de centralidad de cercanía calcula la suma de sus distancias a todos los demás nodos, basándose en el cálculo de las rutas más cortas entre todos los pares de nodos. La suma resultante se invierte para determinar la puntuación de centralidad de cercanía para ese nodo.

$$C(u) = \frac{1}{\sum_{v=1}^{n-1} d(u,v)}$$

Donde u es un nodo, n es el número de nodos en el grafo y $d(u,v)$ es la distancia del camino más corto entre otro nodo v y el nodo u.
Lo más común es normalizar este valor para que represente la longitud promedio de los caminos más cortos en lugar de su suma, permitiendo con este ajuste comparaciones de centralidad de cercanía de nodos de grafos con diferentes tamaños.

$$C_{norm}(u) = \frac{n-1}{\sum_{v=1}^{n-1} d(u,v)}$$

Se aplica este algoritmo para saber qué nodos permiten difundir información más rápido en una red. Por lo que permite descubrir individuos en posiciones muy favorables para controlar y adquirir información y recursos dentro de una organización.
El uso de relaciones ponderadas puede ser especialmente útil para evaluar velocidades de acción en comunicación y análisis de comportamiento.
El algoritmo funciona mejor en grafos conectados, ya que si se utiliza la fórmula original en un grafo no conectado, es posible terminar con una distancia infinita entre dos nodos en componentes disconexas. Esto significa que el algoritmo termina con una puntuación de centralidad de cercanía infinita cuando se sumen todas las distancias desde ese nodo.
En la práctica, se utiliza la siguiente variación de la fórmula original para evitar dicho problema.

$$C_{WF}(u) = \frac{n-1}{N-1}\left(\frac{n-1}{\sum_{v=1}^{n-1} d(u,v)}\right)$$

Donde u es un nodo, N es el número total de nodos del grafo, n es el número de nodos pertenecientes a la misma componente que u y $d(u,v)$ es la distancia del camino más corto entre otro nodo v y el nodo u [43].

- **Centralidad de intermediación (betweenness centrality):** La centralidad de intermediación es una forma de detectar la cantidad de influencia que un nodo tiene sobre el flujo de información en un grafo. Por lo general se usa para encontrar nodos que sirvan como puente de una parte del grafo a otra. El algoritmo que brinda esta medida calcula las rutas más cortas entre todos los pares de nodos del grafo no ponderado, cada nodo recibe una puntuación, basada en el número de rutas más cortas que pasan a través del nodo. Los nodos que se encuentran con mayor frecuencia en los caminos más cortos entre otros nodos tendrán valores de centralidad de intermediación más altos.
  La centralidad de intermediación de un nodo se calcula sumando los resultados del siguiente fórmula para todos los caminos más cortos:

$$B(u) = \sum_{s \neq u \neq t} \frac{p(u)}{p}$$

  Donde u es un nodo, p el número total de caminos más cortos entre los nodos s y t y $p(u)$ es el número de caminos más cortos entre los nodos s y t que pasan por le nodo u [43].

- **PageRank:** El algoritmo de PageRank mide la influencia de cada nodo dentro del grafo, en función de la cantidad de relaciones entrantes y la influencia de los nodos de origen correspondientes. La suposición en términos generales es que un nodo es tan importante como los nodos que enlazan con él. Todos los otros algoritmos de centralidad que se han presentado hasta ahora miden la influencia directa de un nodo, mientras que PageRank considera la influencia de un nodo mediante sus vecinos.
  La intuición detrás de la influencia de un nodo es que las relaciones con nodos más importantes contribuyen más a la influencia del nodo en cuestión en la red, en comparación con conexiones a nodos menos importantes
  La medición de la influencia generalmente implica puntuar nodos, a menudo con relaciones ponderadas y luego actualizar los puntajes después de muchas iteraciones. Algunas veces se puntúan todos los nodos y, a veces, se utiliza una selección aleatoria como distribución representativa.

PageRank es definido de la siguiente manera [11]:

$$PR(u) = (1-d) + d\left(\frac{PR(T1)}{C(T1)} + \cdots + \frac{PR(Tn)}{C(Tn)}\right)$$

Donde se asume que la página u tiene citas de las páginas $T1$ a Tn, d es un factor de amortiguamiento que es seteado entre 0 y 1, usualmente es seteado en $0,85$. Luego se asume que $1-d$ es la probabilidad de que un nodo sea alcanzado directamente sin seguir ninguna relación y $C(Tn)$ es definida como el grado de salida del nodo T [43].

- **Centralidad de vector propio (eigenvector centrality):** Esta es una medida de la importancia o influencia de cada vértice en el grafo. La suposición de esta métrica es que cada medida de centralidad sobre un vértice es la suma de las medidas de centralidad de los vértices que están conectados a este.
  En la centralidad por grado se considera la importancia de un nodo con respecto a su entorno por el número de conexiones. En este caso, se considera la importancia de que el nodo tenga conexiones con otros nodos de igual importancia, siendo esta la base para la centralidad por vector propio (eigenvector centrality)
  Formalmente la centralidad por vector propio se define de la siguiente manera [51]:

$$c_e(i) = \frac{1}{k_1}\sum_{j=1} A_{i,j}x_j$$

  Donde Aij es la matriz de adyacencia, K_1 es una constante [51]
  Con esta medida, un nodo adquiere "importancia" por medio de dos caminos: tener muchas conexiones con otros miembros de la comunidad o tener conexiones con otros nodos "importantes", aunque sean pocos [51].
  Una de las limitantes de esta medida es el análisis sobre redes dirigidas [51].

- Excentricidad (eccentricity): La excentricidad no es una medida de centralidad, pero sirve para detectar la distancia mínima desde un vértice hacia cualquier otro vértice en un grafo siguiendo los camino más cortos

presentes en la estructura [43].

2.2.4. Pathfinding

Los algoritmos de pathfinding exploran caminos entre nodos, comenzando en uno particular y atravesando relaciones hasta que se alcanza el destino.

Estos algoritmos se utilizan para identificar rutas óptimas, incluso encontrar el camino más corto es probablemente la tarea más frecuente que se realiza con algoritmos de grafos y es un precursor de diferentes tipos de análisis.

El camino más corto es la ruta transversal con la menor cantidad de saltos o de menor peso. Si el grafo está dirigido, entonces es el camino más corto entre dos nodos según lo permitido por las direcciones de relación.

- **Breadth First Search (BFS):** Es uno de los algoritmos fundamentales de recorrida de grafos. El algoritmo parte desde un nodo elegido y explora todos sus vecinos en un salto, luego visita a todos los vecinos a dos saltos de distancia, y así sucesivamente. BFS se usa más comúnmente como base para otros algoritmos, por ejemplo, el Shortest Path, Componentes Conexas y la centralidad de proximidad [43].

- **Depth First Search (DFS):** Depth First Search (DFS) es el otro algoritmo fundamental de recorrida de grafos. El algoritmo comienza desde un nodo raíz, elige uno de sus vecinos y luego recorre el grafo tan lejos como puede a lo largo de ese camino antes de realizar backtracking [43].

- **ShortestPath:** El algoritmo ShortestPath calcula la ruta más corta (ponderada) entre un par de nodos. En esta categoría, el algoritmo de Dijkstra es el más conocido. Dijkstra no soporta pesos negativos en las relaciones entre nodos, el algoritmo supone que agregar una relación a una ruta nunca puede acortar una la misma, una invariante que se violaría con pesos negativos [43].

- **A*:** El algoritmo A* Shortest Path mejora con respecto a Dijkstra al encontrar rutas más cortas de forma más rápida. Hace esto permitiendo la inclusión de información adicional que el algoritmo puede usar, como parte de una función heurística, al determinar cuál es el camino siguiente a explorar.

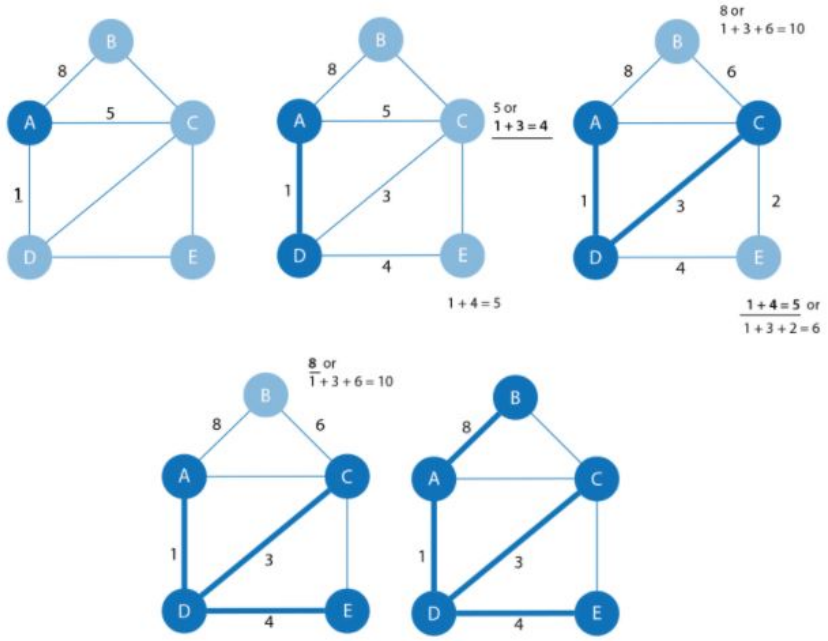


Figura 2.4: Pasos del algoritmo SSSP, figura extraída de [43] página 68

El algoritmo A* opera determinando cuál de sus caminos parciales se va a expandir en cada iteración de su ciclo principal. Lo hace en base a una estimación del costo (heurístico) que aún queda por llegar al nodo objetivo.
A* selecciona el camino que minimiza la siguiente función:

$$f(n) = g(n) + h(n)$$

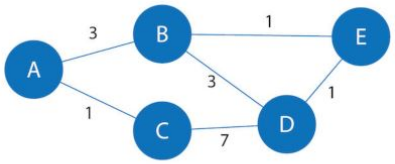
Donde $g(n)$ es el costo del camino que empieza en el nodo n y $h(n)$ es el costo estimado del camino desde el nodo n hasta el nodo final calculado por una heurística [43].

- **Single Source Shortest Path (SSSP):** El algoritmo Single Source Shortest Path calcula la ruta más corta (ponderada) desde un nodo a todos los demás nodos en el grafo.

  1. El algoritmo comienza con un nodo raíz desde el que se medirán todas las rutas, en el ejemplo se selecciona al nodo A como raíz.
  2. Se selecciona la relación con el peso más pequeño proveniente de ese nodo raíz y agregado al árbol, junto con su nodo conectado. En este caso es $d(A, D) = 1$.

3. La siguiente relación con el peso acumulado más pequeño del nodo raíz a cualquier nodo no visitado, es seleccionada y se agrega al árbol de la misma manera. Se tiene $d(A, B) = 8, d(A, C) = 5$ directamente o 4 a través de $A - D - C$, y $d(A, E) = 5$. Entonces, se elige la ruta a través de $A - D - C$ y se agrega C al árbol.
4. El proceso continúa hasta que no hay más nodos para agregar. Obteniendo así el camino más corto del nodo raíz seleccionado al resto de los nodos.

- **All Pairs Shortest Path (APSP):** El algoritmo APSP calcula la ruta más corta (ponderada) entre todos los pares de nodos del grafo. Es más eficiente que ejecutar el Single Source Shortest para cada par de nodos en el grafo. Inicialmente, el algoritmo asume una distancia infinita a todos los nodos, cuando se determina el nodo de inicio, la distancia a ese nodo se establece en 0. El cálculo procede entonces como sigue:



All nodes start with a ∞ distance and then the start node is set to a 0 distance			Each Step Keeps or Updates to the Lowest Value Calculated so Far Only steps for node A to all nodes shown				
			1st from A	2nd from A to C to Next	3rd from A to B to Next	4th from A to E to Next	5th from A to D to Next
A	∞	0	0	0	0	0	0
B	∞	∞	3	3	3	3	3
C	∞	∞	1	1	1	1	1
D	∞	∞	∞	8	6	5	5
E	∞	∞	∞	∞	4	4	4

Figura 2.5: Pasos para calcular el camino más corto desde el nodo A al resto de los nodos, figura extraída de [43] página 63

1. Desde el nodo inicial A, se evalúa el costo de mudarse a los nodos vecinos y se actualizan esos valores buscando el valor más pequeño, en el ejemplo se tiene la opción del nodo B (costo de 3) o C (costo de 1). Se selecciona C para la siguiente fase de recorrido.

2. Ahora desde el nodo C, el algoritmo actualiza las distancias acumuladas de A a nodos a los que se puede acceder directamente desde C. Los valores solo se actualizan cuando se ha encontrado un costo menor: $A = 0, B = 3, C = 1, D = 8, E = \infty$
3. Luego, se selecciona B como el siguiente nodo más cercano que aún no se ha visitado. Tiene relaciones con los nodos A, D y E. El algoritmo calcula la distancia a los nodos sumando la distancia de A a B con la distancia de B a cada uno de esos nodos.

$$d(A, A) = d(A, B) + d(B, A) = 3 + 3 = 6$$

$$d(A, D) = d(A, B) + d(B, D) = 3 + 3 = 6$$

$$d(A, E) = d(A, B) + d(B, E) = 3 + 1 = 4$$

   En este paso, la distancia del nodo A a B y de regreso a A, se muestra como $d(A, A) = 6$, es mayor que la distancia más corta ya calculada (0), por lo que su valor no es actualizado. Las distancias para los nodos $D(6)$ y $E(4)$ son menores que las calculadas previamente por lo que sus valores se actualizan.
4. E es seleccionado a continuación, el total acumulado para alcanzar $D(5)$ ahora es más bajo, y por tanto, es el único actualizado.
5. Cuando finalmente se evalúa D, no hay nuevos pesos mínimos, por lo que no hay ninguna actualización y el algoritmo termina.

2.2.5. Medidas de detección de comunidades

La conectividad es un concepto central de la teoría de grafos que permite el análisis de una red y encontrar comunidades dentro de la misma. La mayoría de las redes del mundo exhiben subgrafos más o menos independientes.

La conectividad se utiliza para encontrar comunidades y cuantificar la calidad de las agrupaciones, por lo tanto, evaluar diferentes tipos de comunidades dentro de un grafo puede descubrir estructuras, como centros, jerarquías, y tendencias de grupos para atraer o repeler a otros.

El principio general de encontrar comunidades es que sus miembros tendrán más relaciones dentro de la comunidad a la que pertenecen que con nodos externos a ella. La identificación de estos conjuntos relacionados revela grupos

de nodos, grupos aislados y la estructura del grafo. Esta información ayuda a inferir comportamientos o preferencias similares de los grupos de pares, estimar la resistencia y encontrar relaciones anidadas.

Modularidad

El índice de modularidad fue propuesto por Girvan y Newman [52], para medir la presencia de *clusters* en una red no aleatoria. Se fundamenta en determinar las densidades dentro y fuera de cada *cluster* y comparándolas con la densidad general que tendrían si la red fuera construida de manera aleatoria [51].

De manera formal se define como:

$$Q = \frac{1}{2m} \sum_{i,j} (a_{i,j} - \frac{k_i k_j}{2m}) \delta(c_i c_j) \tag{2.9}$$

Este Q es llamado modularidad, m es el número de aristas, k_i es el grado del nodo i, $a_{i,j}$ es el elemento de la matriz de adyacencia de los nodos i y j, $\delta(m, n)$ es la función delta de Kronecker[1] y c_i es el *cluster* i [51].

- **Label Propagation (LPA):** Label Propagation es un algoritmo de tiempo casi lineal para detectar estructuras comunitarias en redes a gran escala. Un algoritmo rápido para encontrar comunidades en un grafo. Funciona propagando etiquetas a través de la red y formando comunidades basadas en este proceso de propagación de etiquetas. La intuición detrás del algoritmo es que, una sola etiqueta puede convertirse rápidamente en dominante en un grupo de nodos densamente conectados, pero tendrá problemas para cruzar una región escasamente conectada, las etiquetas quedarán atrapadas dentro de un grupo de nodos densamente conectados, y aquellos nodos que terminan con la misma etiqueta cuando finalizan los algoritmos pueden considerarse parte de la misma comunidad.

[1] La función delta de Kronecker toma el valor de 1 si las dos variables son iguales y 0 si las dos variables son diferentes.

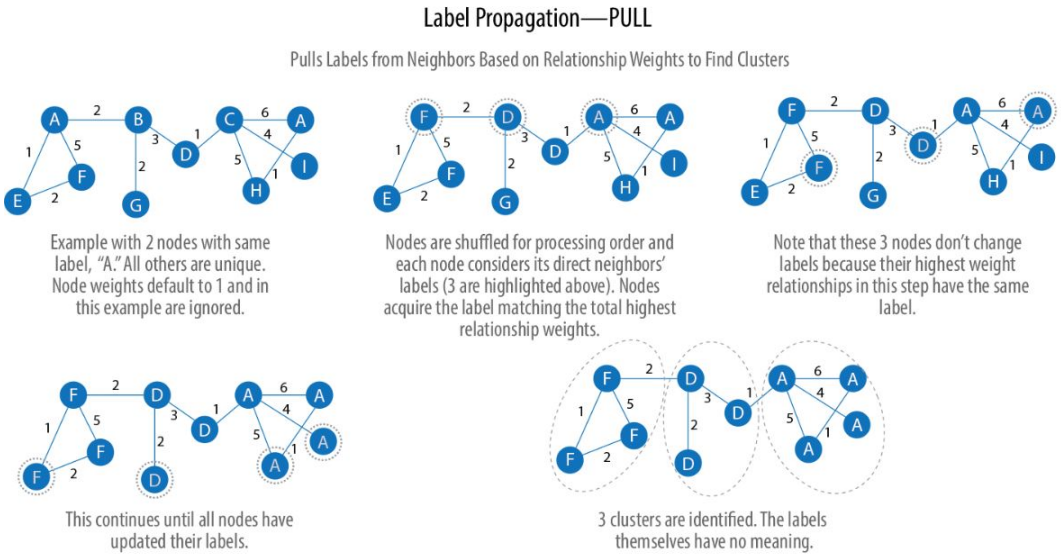


Figura 2.6: Variación PULL del algoritmo LPA, figura extraída de [43] página 136

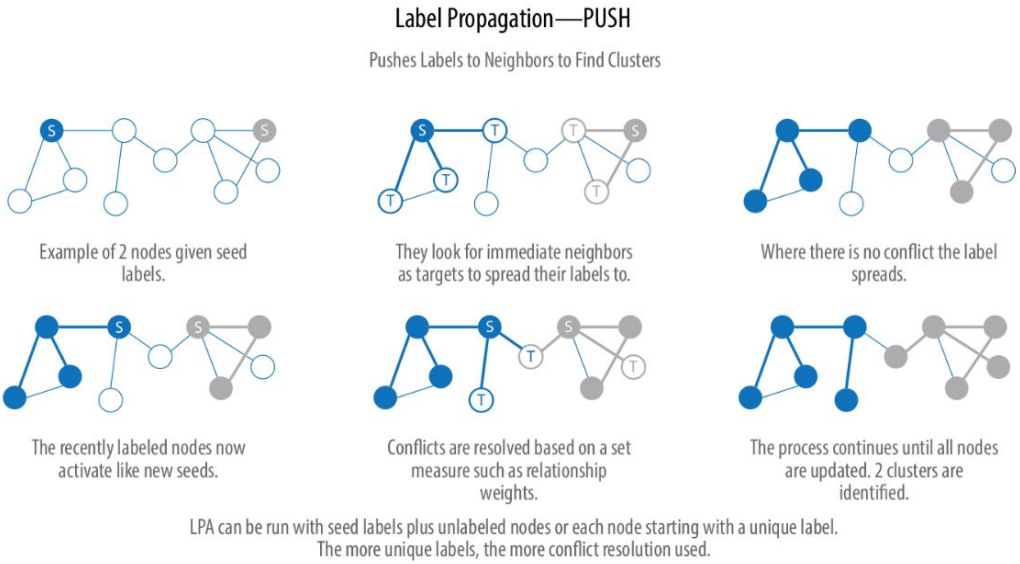


Figura 2.7: Variación PUSH del algoritmo LPA, figura extraída de [43] página 136

Las figuras representan dos variaciones del algoritmo, un método de empuje simple y el método de extracción más típico que se basa en relaciones ponderadas. El método de extracción funciona bien con la paralelización. Los pasos que se utilizan a menudo para el método de extracción de propagación de etiquetas son:

1. Cada nodo se inicializa con una etiqueta única (un identificador).
2. Estas etiquetas se propagan a través de la red.

3. En cada iteración de propagación, cada nodo actualiza su etiqueta para que coincida con la que tiene el peso máximo, que se calcula en función de los pesos de los nodos vecinos y sus relaciones. Los lazos se rompen de manera uniforme y aleatoria.
4. LPA alcanza la convergencia cuando cada nodo tiene la etiqueta mayoritaria de sus vecinos.

A medida que las etiquetas se propagan, los grupos de nodos densamente conectados alcanzan rápidamente un consenso sobre una etiqueta única. Al final de la propagación solo quedarán unas pocas etiquetas, la mayoría habrán desaparecido. Se dice que los nodos que tienen la misma etiqueta de comunidad en convergencia pertenecen a la misma comunidad.
A diferencia de otros algoritmos, Label Propagation puede devolver diferentes comunidades cuando se ejecuta varias veces en el mismo grafo. Esto debido a que el orden en que LPA evalúa los nodos de uso, puede influir en las comunidades finales que devuelve.
El rango de soluciones se reduce cuando algunos nodos reciben etiquetas preliminares (es decir, etiquetas de semillas), mientras que otros no están etiquetados. Los nodos no etiquetados tienen más probabilidades de adoptar las etiquetas preliminares.
Este uso del LPA puede considerarse como un método de aprendizaje semi-supervisado para encontrar comunidades. El aprendizaje semi-supervisado es una clase de tareas y técnicas de aprendizaje automático que operan en una pequeña cantidad de datos etiquetados, junto con una mayor cantidad de datos sin etiquetar. También podemos ejecutar el algoritmo repetidamente en grafos a medida que estos evolucionan. LPA a veces no converge en una sola solución. En esta situación, los resultados de las comunidades devueltas por el algoritmo cambiarán continuamente entre algunas comunidades notablemente similares y este nunca terminaría.Las etiquetas de semillas guían al algoritmo hacia una solución.

- **Louvain:** El método de Louvain para la detección de comunidades maximiza una puntuación de modularidad para cada comunidad, donde la modularidad cuantifica la calidad de una asignación de nodos a las comunidades. Esto significa evaluar qué tan densamente conectados están los nodos dentro de una comunidad, en comparación con lo conectados que

estarían en una red aleatoria. El algoritmo de Louvain es un algoritmo de agrupamiento jerárquico, que fusiona comunidades de forma recursiva en un solo nodo y ejecuta el agrupamiento de modularidad en los grafos condensados. Este método heurístico se basa en la optimización de la modularidad. Es extremadamente rápido, con buenos resultados y permite ser aplicado sobre redes con peso.
Los algoritmos de modularidad optimizan las comunidades localmente y luego globalmente, utilizando múltiples iteraciones para probar diferentes agrupaciones. Esta estrategia identifica las jerarquías de la comunidad y proporciona una amplia comprensión de la estructura. Sin embargo, todos los algoritmos de modularidad sufren de dos inconvenientes:

- Fusionan comunidades más pequeñas en comunidades más grandes.
- Puede producirse una meseta cuando están presentes varias opciones de partición con similares modularidad, formando máximos locales e impidiendo el progreso.

2.2.6. Redes

Una red es una forma de representar las relaciones entre los elementos de una colección. Se compone de un conjunto de objetos, llamados nodos o vértices, los cuales se encuentran conectados por medio de enlaces, llamados aristas o arcos. Si dos nodos se encuentran conectados por medio de una arista, se denominan nodos vecinos [22]. En matemáticas, el concepto más básico de red se conoce también como grafo 2.1 [58].

2.2.7. Redes sociales de aprendizaje

Las redes sociales de aprendizaje consisten de actores (estudiantes, docentes y recursos) y promueven las conexiones entre ellos. Un link (lazo) describe la relación entre los actores y puede ser de tipo fuerte ó débil dependiendo de su frecuencia, importancia o tipo. El Análisis de las Redes Sociales investiga los lazos, relaciones, roles y configuraciones de las redes. El Análisis de las Redes Sociales de Aprendizaje estudia cómo esas redes se construyen y apoyan los procesos de aprendizaje. Por su énfasis en el análisis de las relaciones entre los actores y la visión de que el uso de la tecnología forma parte del proceso de aprendizaje, este tipo de análisis ofrece la posibilidad de identificar inter-

venciones que incrementan el potencial de la red para apoyar el aprendizaje de los actores [67].

El Análisis de las Redes Sociales se puede centrar en un solo individuo o bien en toda la red. Un enfoque egocéntrico puede identificar personas que apoyan el aprendizaje de un estudiante. El análisis de la red global, provee una visión de los intereses y prácticas de un conjunto de personas, identificando los rasgos que mantienen a la red social integrada. Tiene también el potencial de ayudar a identificar grupos dentro de la red, que pueden apoyar los procesos de aprendizaje, por ejemplo comunidades y grupos de afinidad. El Análisis de las Redes Sociales es refinado en el contexto del aprendizaje social y tiene el potencial de combinarse con otros tipos de análisis semántico para permitir la detección de patrones de interacción (entre los estudiantes de un grupo entre sí, entre un estudiante y los recursos de aprendizaje y entre el docente con los estudiantes), que caractericen el perfil del usuario y revelar situaciones potenciales de riesgo con impacto negativo en el aprendizaje de los estudiantes, como por ejemplo el cyberbullying [34].

Una red social es una red de personas o de grupos de personas. Estas personas o grupos se representan como los nodos de la red y las aristas representan las conexiones entre ellos, como las amistades entre personas o los vínculos entre grupos de personas. Varias de las herramientas utilizadas para el análisis de estas redes corresponden al área de la estadística [6].

2.3. Base de datos

En esta sección se pretende describir por un lado las bases de datos relacionales y por el otro las bases de datos orientadas a grafos para luego compararlas en los aspectos más básicos. Se utilizó como antecedente la tesis "Generador del Modelo Relacional y Esquemas de Bases de Datos a partir del modelo Entidad/Relación."[1] y el trabajo "Introducción a las Bases de Datos de Grafos: Experiencias en Neo4j"[66] que exponen en gran parte las definiciones que busca describir esta sección.

2.3.1. Base de Datos Relacional

Una base de datos relacional es un tipo de base de datos que almacena y proporciona acceso a puntos de datos relacionados entre sí. Las bases de datos

relacionales se basan en el modelo relacional, una forma intuitiva y directa de representar datos en tablas. En una base de datos relacional, cada fila de la tabla es un registro con un ID único llamado clave. Las columnas de la tabla contienen atributos de los datos, y cada registro generalmente tiene un valor para cada atributo, lo que facilita el establecimiento de las relaciones entre los puntos de datos [54].

2.3.2. Sistema Gestor de Base de Datos (SGBD)

Es un conjunto de datos interrelacionados y con herramientas computacionales específicas para acceder a dichos datos. Su lugar de almacenamiento se denomina base de datos, pues es aquella que contiene información relevante de una empresa u organización. El objetivo principal de un SGBD es de almacenar y recuperar la información de la organización o empresa de una forma práctica y eficiente

2.3.3. Entidad

Una entidad es una representación de un objeto individual concreto del mundo real. Las entidades tienen atributos.

2.3.4. Atributo

Un atributo de una entidad es una característica interesante sobre ella, es decir, representa alguna propiedad que nos interesa conocer

2.3.5. Clave primaria

Se denomina clave al conjunto de atributos que identifican de forma unívoca una entidad

2.3.6. Relación

Las entidades se vinculan mediante relaciones que, en ciertas variantes de la notación, pueden también tener sus propios atributos. En principio, estas relaciones pueden ser n-arias, pero en la práctica se suele trabajar con relaciones binarias. Por ejemplo, una relación ternaria entre entidades A, B y C

puede representarse por una nueva entidad D que tenga relaciones binarias con cada una de A, B y C

2.3.7. Modelo de Datos

Es una colección de herramientas conceptuales para describir los datos, las relaciones, la semántica y las restricciones de consistencia.

El modelo de base de datos relacional se diseñó para resolver el problema de varias estructuras de datos arbitrarias.

El modelo relacional proporcionó una forma estándar de representar y consultar datos que cualquier aplicación podría utilizar.

En un Sistema de Base de Datos relacional, la información se almacena en "tablas". Cada tabla contiene un conjunto de información asociada a un grupo de similar entidad.

2.3.8. Columna

Una "columna" representa un tipo único de información acerca de la entidad (atributo) [41].

2.3.9. Fila

Una "fila" es un conjunto de tipos de información que describe una instancia de entidad [41].

2.3.10. Tabla

Generalmente, la tabla está compuesta de múltiples filas, que constituyen un conjunto de entidades similares que son descritas de acuerdo con un criterio predefinido [41].

2.3.11. Base de datos de grafos

Una Base de Datos en Grafo (GDB) es un tipo de base de datos NoSQL que usa estructuras de grafos con nodos, aristas y propiedades para representar y almacenar la información Figura 2.8.

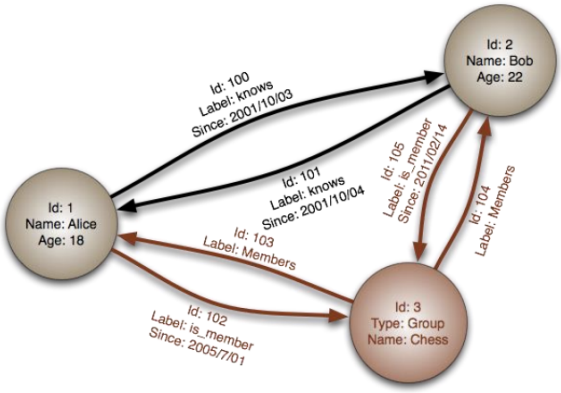


Figura 2.8: Ejemplo base de datos de grafo

Los **nodos** pueden representar entidades tales como personas, partes de un texto, conceptos, o cualquier tópico, real o no, que pueda ser útil para modelar un problema.

Las **aristas** representan relaciones entre entidades, y a diferencia de las relaciones en las RDB, en las GDB mucha de la información relevante al modelo se almacena en estas aristas. Habitualmente, cuando se decide por una implementación del modelo en un GDB es porque se requiere estudiar los patrones que emergen en las conexiones e interconexiones entre nodos, propiedades y aristas, y no tanto en la información relativa a cada uno de ellos por separado.

Las propiedades son unidades de información relevante para los nodos y las aristas. Por ejemplo, si un nodo representa una comedia, propiedades relevantes podrían ser: la fecha de edición, género, extensión, etc... el hecho de que algunas de estas propiedades pasen a convertirse en nodos independientes del grafo, adquieran entidad propia, depende del objetivo a cumplir y está relacionado con el cómo estructurar la información.

Sistema de gestión de base de datos de grafos (SGBDG)

Un SGBDG, es un sistema de base de datos específicamente diseñado para poseer las siguientes capacidades [7].

- Administrar datos de tipo grafo. Es decir, su modelo de datos lógico está basado en alguna de las variantes de la definición matemática básica de grafo, como por ejemplo, grafos dirigidos o no dirigidos, con vértices y arcos etiquetados o no etiquetados, con propiedades en nodos y arcos, hipergrafos e hipernodos [59] [21]. Consecuentemente, las operaciones

CRUD (Create-Read-Update-Delete, operaciones básicas de los SGBDs) trabajan sobre grafos y sus elementos. Sin embargo esto no significa que en el almacenamiento subyacente efectivamente se encuentren grafos. Algunos SGBDGs utilizan almacenamiento nativo es decir, sus estructuras de datos físicas están diseñadas y optimizadas para almacenar y administrar grafos. También existen gestores que mapean los grafos a otras estructuras [18], como por ejemplo tablas u objetos. En cuanto a la implementación de las conexiones entre nodos, también se presentan diferentes enfoques, algunos que eligen la adyacencia sin índices, y otros que no.

- Satisfacer los principios básicos de todo SGBD, es decir, el almacenamiento persistente, la independencia física/lógica, la integridad y la consistencia de los datos.

2.3.12. Modelo de Grafos vs Modelo Relacional

El modelo de datos relacional y los SGBDs relacionales son ampliamente conocidos y populares. Por ello, a continuación, en la tabla 2.1, se presentan algunos de los elementos básicos del enfoque relacional y sus análogos en el modelo de grafos. Cabe aclarar que dicha correspondencia no es absoluta ya que cada pareja de términos presenta particularidades significativas.

Bases de Datos Relacionales	**Bases de Datos de Grafos**
Filas	Nodos
Columnas	Propiedades
Nombre de las Tablas	Etiquetas en Nodos/Aristas
Claves Foráneas	Aristas entre Nodos

Tabla 2.1: Comparación BDR vs BDG

Capítulo 3

Revisión de antecedentes y tecnologías

En este capítulo se presenta la problemática a abordar en este proyecto de grado, así como también, conceptos teóricos que serán necesarios para comprender mejor el mismo. Por otro lado, se analizan proyectos en relación que pueden aportar a la completitud del trabajo, además de servir como guía de la investigación, se da un resumen sobre distintas plataformas educativas como por ejemplo Blackboard [8] y Moodle [42].

3.1. Plataformas educativas

En esta sección se busca introducir al lector en la importancia que tienen hoy en día las plataformas de educativas en el aprendizaje de los estudiantes. Analizando cómo han tomado relevancia en la mayoría de los centros de estudio y cómo han evolucionado en lo que hoy conocemos como plataformas de teleformación (LMS-Learning Management System, que podría traducirse como sistemas para la gestión de aprendizaje), exponiendo sus principales características y beneficios[65]. Así como también mencionando algunas de las principales plataformas, sus características y las herramientas que las componen.

En los últimos años se ha incorporado un nuevo concepto para definir el modelo de sociedad. De una alfabetización tradicional basada en las habilidades de lectoescritura, y que constituye la base de enseñanza en la escolaridad, se está pasando a una alfabetización digital en la que se necesita de habilidades

tecnológicas e informaciones.

Para ser autosuficientes, se necesita saber navegar por infinitas fuentes de información, poder discriminar la información recibida y cada vez más saber dominar la sobrecarga de información que se brinda a través de la web [55].

Las universidades han tenido que adaptarse a este cambio trabajando para brindar buena, correcta y adecuada información y compartir conocimientos a sus docentes, estudiantes, personal, etc. Con la finalidad de aprender constantemente y enriquecer el vocabulario. Las nuevas propuestas pedagógicas y académicas llevan a que se comiencen a utilizar herramientas de apoyo y acompañamiento educativo, que permiten con el uso de las nuevas tecnologías avanzar en el estilo de enseñanza aprendizaje convencional.

Cada vez son más las instituciones y los docentes que se animan a explorar y utilizar variados recursos tecnológicos para acompañar el proceso de enseñanza y aprendizaje de los estudiantes.

A diferencia de la formación tradicional y presencial a la que se estaba acostumbrado, surgen otras herramientas como son las plataformas de teleformación (LMS-acrónimo en inglés de Learning Management System) que aportan otras modalidades de enseñanza para utilizar en las aulas [35].

Como por ejemplo:

- **B-learning:** como apoyo a la enseñanza presencial, combina la enseñanza presencial con la tecnología no presencial [56].
- **E-learning:** es la formación totalmente a distancia, entendida como "el uso de tecnologías basadas en Internet para proporcionar un amplio abanico de soluciones que aúnan adquisición de conocimiento y habilidades o capacidades" [39].
- **M-learning:** o aprendizaje móvil, para la formación a distancia con el uso de tecnologías móviles como smartphone, tablets, lectores de MP3, ipad, etc.
- **T-learning:** es un sistema de aprendizaje transformativo, en el cual el uso de las nuevas tecnologías es parte del proceso, del desarrollo de contenidos y actividades que pueden darse de forma presencial o virtual. Se focaliza en el desarrollo de las habilidades en el "hacer" del estudiante.
- **W-learning:** La formación a distancia cooperativa, en base a herramientas colaborativas de la web 2.0.

El crecimiento de la tecnología y el uso masivo de los dispositivos móviles

en la ultima década, ha sido tan grande, que ha provocado el incremento de los estudiantes y docentes que adaptan a sus metodologías de estudio la enseñanza a distancia. Utilizando como eje las plataformas de e-learning.

3.1.1. Beneficios LMS

De entre sus múltiples ventajas se destacan los siguientes beneficios:

- Permiten estudiar en cualquier momento y lugar, anulando el problema de las distancias geográficas o temporales y ofreciendo una gran libertad en cuanto a tiempo y ritmo de aprendizaje.
- Posibilitan la capacitación de las personas con máxima flexibilidad y costos reducidos.
- Para su uso no se precisan grandes conocimientos (únicamente un nivel básico del funcionamiento de Internet y de las herramientas informáticas).
- Posibilita un aprendizaje constante y actualizado a través de la interacción entre tutores y alumnos.

3.1.2. Características de las plataformas LMS

Los componentes o características básicas de todo entorno virtual de aprendizaje, que además deben estar fuertemente ligados e interconectados, de forma que se influyan mutuamente y se retroalimenten pueden sintetizarse en los siguientes:

- Centralización y automatización: de la gestión del aprendizaje.
- Flexibilidad: La plataforma puede ser adaptada tanto a los planes de estudio de la institución, como a los contenidos y estilo pedagógico de la organización. También permite organizar cursos con gran facilidad y rapidez.
- Interactividad: La persona se convierte en el protagonista de su propio aprendizaje a través del autoservicio y los servicios autoguiados.
- Estandarización: Esta característica permite utilizar cursos realizados por terceros, personalizando el contenido y reutilizando el conocimiento.
- Escalabilidad: Estos recursos pueden funcionar con una cantidad variable de usuarios según las necesidades de la organización.

- Funcionalidad: Prestaciones y características que hacen que cada plataforma sea adecuada (funcional) según los requerimientos y necesidades de los usuarios.
- Usabilidad: Facilidad con que las personas pueden utilizar la plataforma con el fin de alcanzar un objetivo concreto.
- Ubicuidad: Capacidad de una plataforma para generar tranquilidad al usuario y provocarle la certeza de que todo lo que necesita lo va a encontrar en dicho entorno virtual.
- Integración: Las plataformas LMS deben poder integrarse con otras aplicaciones empresariales utilizadas por recursos humanos y contabilidad, lo que permite medir el impacto, eficacia, y sobre todo, el coste de las actividades de formación.

Además de las características generales, hay que tener presente que la intención con la que han sido diseñadas las plataformas contribuye activamente en su caracterización, en cuestiones como: las bases pedagógicas, los modelos de negocio, los modelos de gestión, las posibilidades tecnológicas de las propuestas o los perfiles de los usuarios finales [65].

3.1.3. Revisión de plataformas

Para abordar el presente trabajo, se realiza un análisis en las plataformas educativas existentes, en particular se expone en algunos puntos Blackboard y Moodle LMS, fue elegido Moodle ya que es utilizada actualmente en la Universidad de la República y Blackboard hasta el 2005 desarrolló y licenció aplicaciones de programas empresariales y servicios relacionados a más de 2200 instituciones educativas en más de 60 países [33], además de que fue un requerimiento inicial del proyecto de grado. El objetivo de este primer análisis es entender cómo trabajan, qué aporte le dan a la educación actual, las herramientas principales que emplean para facilitar el proceso educativo y por último, tener estas plataformas como referencias para el avance en la búsqueda de una solución que aporte valor en el estudio de cómo mejorar las herramientas actuales.

Blackboard LMS es un sistema de gestión de aprendizaje en línea, un ecosistema donde hay interacción de conocimiento entre tutores/estudiantes. Esta es una plataforma que tiene módulos de contenidos y herramientas de, comunicación interna, evaluación, seguimiento y gestión de aprendizaje.

En la actualidad los estudiantes se enfrentan a numerosos obstáculos complejos a lo largo de su trayectoria educativa. Un LMS es un componente importante para ayudarles a obtener el conocimiento buscado, pero este entorno de aprendizaje por sí solo únicamente es un aspecto de la experiencia del estudiante. Blackboard brinda complementos que enriquecen la plataforma. Como por ejemplo:

- Enseñanza y aprendizaje: Con las soluciones digitales de aprendizaje de Blackboard, se puede impulsar la efectividad académica y la participación de los estudiantes, así como permitir una mejora continua a través de información en materia de educación.
- Participación comunitaria: Las soluciones brindadas ayudan a las instituciones a compartir información específica sobre los estudiantes, noticias relevantes y actualizaciones en tiempo real con los miembros de la comunidad estudiantil.

Algunas soluciones contenidas en Blackboard:

Blackboard Learn, brinda un acceso rápido a la información más reciente e importante de todos los cursos.

Blackboard Collaborate Agrega, un sin fin de herramientas enfocadas en la interacción entre estudiantes y docentes de la plataforma, lo cual conlleva a que los estudiantes se sientan como si estuvieran en la misma aula.

Blackboard Analytics for Learn, combina los datos tanto del entorno de aprendizaje virtual como del sistema de información sobre los estudiantes, ofrece a las instituciones un rico conjunto de herramientas para identificar a los estudiantes con dificultades, descubrir las barreras que impiden completar el curso correctamente y realizar un seguimiento de la repercusión de las prácticas educativas en diferentes conjuntos de estudiantes a lo largo del tiempo.

Blackboard Ally ayuda a las instituciones a crear un entorno de aprendizaje más inclusivo y a mejorar la experiencia del estudiante al ayudarlos a tomar un claro control del contenido del curso teniendo en cuenta la facilidad de uso, la accesibilidad y la calidad.

Blackboard Digital Teaching & Learning Series

Es el programa digital de enseñanza y aprendizaje de Blackboard, permite establecer un estándar sobre cómo ofrecer la enseñanza y el aprendizaje en línea. Proporciona al personal académico capacitaciones y certificaciones de una clase [8].

Moodle LMS permite crear un espacio de aprendizaje privado en línea con un gran número de actividades y materiales interesantes. Se tiene en todo momento el control total de todos los datos y la forma en que el personal, estudiantes y clientes se incorporan al sistema.

Moodle es una plataforma de fuente abierta. Se es dueño del sitio y de su contenido en términos propios. Es un LMS de código abierto con todas las funciones y garantiza que siempre se tenga el control total.

Integración completa Se conecta con plataformas y servicios como Google Apps, Microsoft Office 365, NextCloud entre otros.

Personalización completa Usa una interfaz moderna que permite personalizar no solo el aspecto de la plataforma, sino también muchas configuraciones de administración.

Miles de extensiones Es fácil agregar funcionalidades adicionales para estudiantes, maestros y administradores con los complementos gratuitos creados por la comunidad.

Cumplimiento de accesibilidad Esta herramientas integra verificación de accesibilidad de texto que ayuda a crear cursos y contenido con soporte completo para todos los estudiantes, siguiendo estándares de accesibilidad comunes.

Listo para dispositivos móviles La herramienta ofrece flexibilidad de aprender desde cualquier dispositivo. Personaliza y adecua la información para que este disponible en cada uno de los dispositivos.

Perspectivas de Learning Analytics Brinda un monitoreo de alumnos con un seguimiento de finalización mejorado y planes de aprendizaje personalizados, con contenido de información para predecir y apoyar a los estudiantes en riesgo de reprobar.

Aprendizaje social Enfoque en el aprendizaje colaborativo entre alumnos a través de actividades grupales, foros, chats y otras herramientas para compartir conocimientos [42].

3.2. Learning Analytics Y Social Learning

El proceso de recolección y procesamiento de los datos de los estudiantes conforma lo que se conoce como Analítica del Aprendizaje, o en inglés, Learning Analytics (LA). Según el informe de Horizon Report (2016) [36], la Analítica del Aprendizaje es una aplicación educativa de analítica web dirigida

a un perfil de alumnos, que implica un proceso de recopilación y análisis de datos sobre la interacción individual de los estudiantes con las actividades de aprendizaje online.

Aplicar conceptos de análisis de datos al mundo educativo, se conoce como analítica académica (capturar y reportar datos dirigidos a los administradores de la educación) y la analítica de la acción (definir mejoras a partir de la evaluación de la efectividad de las instituciones educativas). La analítica del aprendizaje pretende ir un paso más adelante, evolucionando desde la perspectiva del simple análisis de datos, para informar a los responsables del cumplimiento de los objetivos institucionales, hacia la propuesta de nuevas herramientas dirigidas al docente para evaluar el entendimiento y la optimización no solo del aprendizaje, sino también del ambiente en el que tiene lugar.

Así surge la Analítica del Aprendizaje Social, el cual busca evidenciar que las nuevas ideas y habilidades no se producen de manera individual, sino que se estimulan y desarrollan a través de la interacción y la colaboración entre los miembros de una comunidad [26].

La Analítica del Aprendizaje Social (Social Learning Analytics), se centra en la manera en la que los estudiantes construyen el conocimiento de manera conjunta, en su espacio de interacción social y cultural.

En el aprendizaje social en línea, se consideran los ambientes educacionales tanto formales como informales, incluyendo las redes sociales y las comunidades en línea creadas con propósito de aprendizaje.

Capítulo 4

Metodología

El objetivo de este capítulo es explicar y justificar el diseño metodológico elegido. Describiendo así, los pasos realizados a lo largo del proceso de trabajo.

El proyecto de grado propuesto es de tipo investigación aplicada, ya que elabora propuestas innovadoras que podrán aplicarse de manera directa a la atención de la necesidad identificada por la institución educativa EviMed [24], actualmente enfocándose en aportar información valiosa en contenido social requerida para la toma de decisiones que conduzca a la mejora del desempeño académico de los estudiantes.

La investigación parte del relevamiento sistemático de la bibliografía sobre las mejores prácticas, los conocimientos más innovadores y las tecnologías más avanzadas en Analítica del Aprendizaje Social, para efectuar estudios de correlación que vinculen los patrones de interacción detectados en las redes sociales institucionales e informales, con el desempeño académico de los estudiantes. Una vez abordada la investigación inicial se orientada al análisis e interpretación de los grafos generados a partir de una red social, la cual se obtuvo partiendo de los datos de la plataforma. Como conclusión del relevamiento se propone un conjunto de soluciones prácticas que ofrecerá a los docentes y a las autoridades educativas de EviMed, los servicios de análisis que permitan conocer el perfil del comportamiento social de sus estudiantes y detectar oportunamente los datos críticos que reflejen situaciones de riesgo que pueden impactar en su desempeño académico. Aportando análisis, prueba de conceptos y herramientas automatizadas que brindan desde la extracción de datos, procesamiento y normalización de los mismos, carga de datos, y hasta generación de resultados en diversas formas (grafos, métricas asociadas al grafo,

visuales y conclusiones). Para lograrlo, se siguió la siguiente metodología:

- Análisis de otras plataformas de LMS.
- Revisión bibliográfica profunda sobre los temas clave involucrados en el proyecto, tales como: Analítica del Aprendizaje, Redes Sociales, Perfilado del Usuario, Teoría de grafos, Base de datos orientadas a grafos.
- Reuniones con autoridades de EviMed para recabar requerimientos y validarlos.
- Posteriormente se prosigue con la siguiente metodología:
    - Se parte de los supuestos de que los usuarios son docentes o autoridades de EviMed de la plataforma de estudio redEMC [25] que necesitan soluciones prácticas para sacar conclusiones acerca del comportamiento de los usuarios dentro de un determinado contexto, como por ejemplo la interacción de los usuarios dentro de un curso dado, obteniendo como resultado información útil a la hora del análisis orientado a la mejora del proceso de aprendizaje de los alumnos.
    - Como segundo paso se plantea detectar el modelo de datos existentes en la plataforma y cuáles son los componentes necesarios para llevar a cabo las transformaciones, carga, presentación de datos, y también las tecnologías y herramientas utilizadas en el análisis y obtención de métricas sobre las estructuras de grafos. Para la realización de este trabajo se procede explotando las siguientes etapas:
        - Investigación y evaluación de las herramientas y/o tecnologías a utilizar.
        - Validación mediante pruebas de concepto dado el objetivo/herramienta.
        - Análisis del resultado post pruebas.
    - Para la tarea de desarrollo del ambiente de software que ofrecen las soluciones integradas de Analítica del Aprendizaje Social las cuales permiten la construcción de un análisis enriquecedor del estudiante y las interacciones del mismo en un contexto dado, por ejemplo detección de las situaciones de riesgo que podrían impactar en el aprendizaje del mismo, a partir del análisis semántico de redes sociales, se aplicaron las metodologías de SCRUM [60] para desarrollo de software en grupos.

En la figura 4.1 se encuentra un diagrama de Gantt donde se observan las etapas y tiempos estimados al inicio del proyecto. A medida que se avanzó, resultó difícil definir el proyecto, por lo que la primeras etapas del mismo se extendieron más de lo previsto. Esto se ve reflejado en la figura 4.2 siendo una simplificación de las tareas realizadas para cumplir con el trabajo y el tiempo que requirió cada etapa.

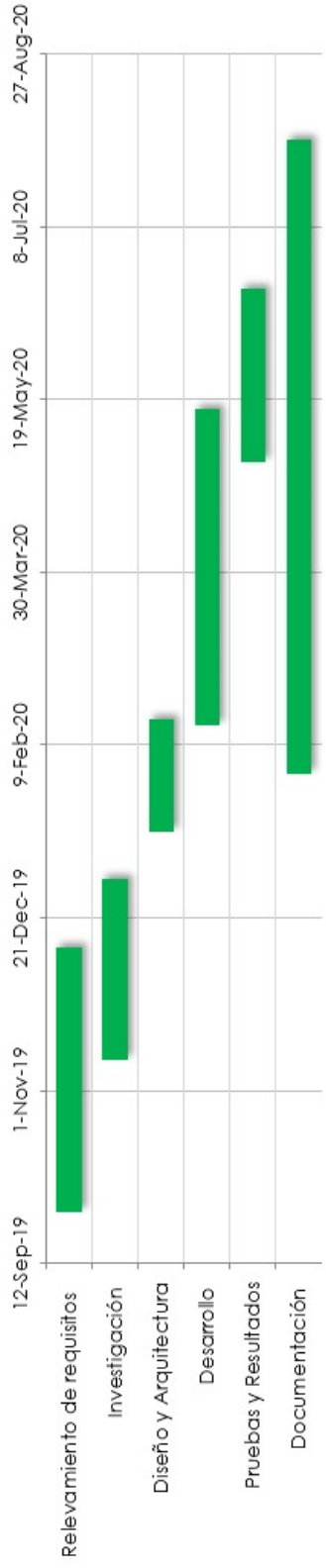


Figura 4.1: Diagrama de Gantt de tiempos estimados al inicio del proyecto

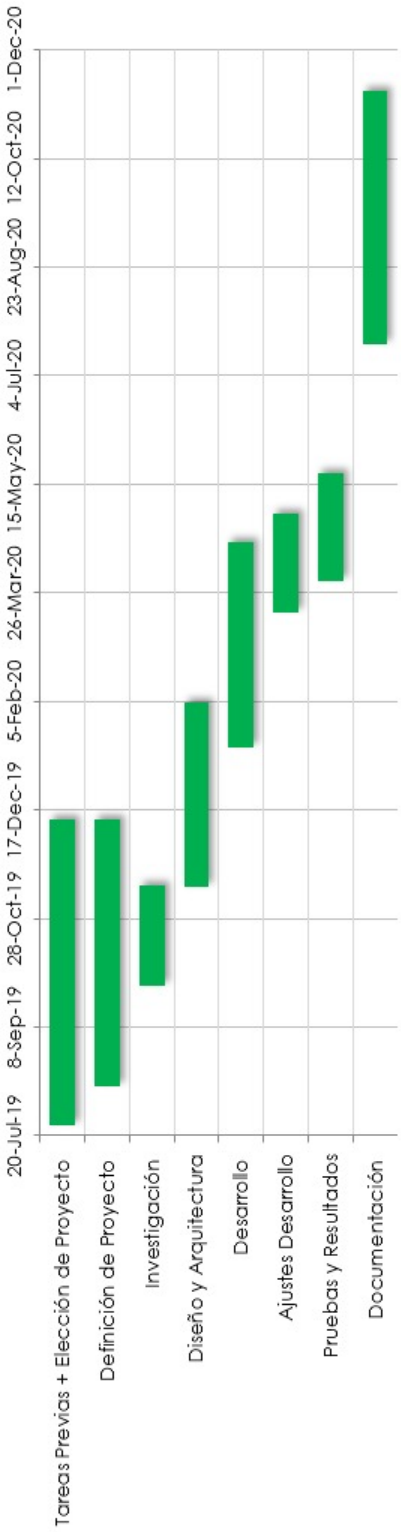


Figura 4.2: Diagrama de Gantt real del proyecto

Capítulo 5

Contexto de EviMed, necesidades y requisitos

5.1. Contexto

En esta sección se da a conocer la institución educativa EviMed [24] y la plataforma de aprendizaje redEMC [25], detallando sus características, y haciendo mención a su foco dentro del área del aprendizaje social. Explicando y dando una introducción de cómo aporta en el campo la plataforma, cómo está compuesta y cómo genera sus funcionalidades. Cabe destacar que en el resto del documento podrá aparecer mencionado query o consulta indistintamente para referirse a consultas SQL o Cypher.

5.1.1. La institución educativa EviMed

EviMed es una institución educativa radicada en Uruguay que provee una propuesta innovadora potenciada por las tecnologías de la información y las comunicaciones (TICs), en función de la educación continua para profesionales de la salud. Vale destacar que esta contribuye a la mejora de la atención médica a través de la educación del profesional en actividad. Está concebida como una institución socialmente responsable ya que su trabajo busca contribuir a mejorar la salud de la población, realizando proyectos en conjunto con organizaciones a través de las cuales sus productos y servicios llegan a la comunidad.

Es líder en la provisión de servicios y proyectos de educación médica continua (EMC) para América Latina.

Además cuenta con equipos multidisciplinarios, liderados por coordinadores académicos referentes en sus especialidades, con amplia experiencia y fuertes vínculos universitarios e intersociedades. Cada proyecto pedagógico es articulado por cuerpos docentes de expertos que imparten las videoconferencias y los conversatorios en línea, con la mentoría logística e informática de EviMed.

Por otro lado, redEMC es una plataforma virtual de aprendizaje a distancia, desarrollada por EviMed, la cual brinda programas permanentes de educación en línea, para profesionales y equipo de salud de toda América Latina. La misma cuenta con 35814 estudiantes, 350 docentes, 130 cursos, 63 instituciones y 63 foros.

La plataforma busca que el aprendizaje sea un proceso de participación social, a través de herramientas como los foros generales, que son abiertos a la creación de discusiones sobre diversos temas por parte de los participantes, y en el espacio para comentarios, que existe en el campus virtual debajo de cada elemento y material del curso.

También cuenta con una herramienta colaborativa de formato Wiki para la creación de informes, modelos, esquemas o proyectos [25]. Estos lazos son muy importantes y se fortalecen mediante actividades como el planteamiento de experiencias clínicas reales por parte de los participantes, con la posibilidad de que sus colegas retroalimenten su planteo por medio de sugerencias de intervención, lecturas de la realidad o modelos explicativos que enriquezcan a todo el grupo. Se persigue la inherencia de los participantes en su propio aprendizaje, esto se logra a través de instancias como el foro de discusión que se ofrece en cada módulo de aprendizaje y ruta de aprendizaje, herramienta de consulta individual durante todo el curso. La intención es crear las condiciones para que el conocimiento tácito de los profesionales de la salud, construido en su cotidiano vivir y trabajar, se integre a un conocimiento más explícito y pueda codificarse dentro de un marco conceptual definido. En particular, son los foros de discusión el espacio donde se produce el proceso de conversión social de los conocimientos, en el que éstos se expanden y crecen.

Se cuenta con una herramienta de simulaciones clínicas interactivas, en la que se presenta un caso clínico con múltiples opciones sucesivas de interpretación e intervención, de tal forma que el participante va realizando su propio proceso de descubrimiento al hacer elecciones. También se pueden establecer actividades de autoevaluación, donde los participantes descubren su propio avance y ganancia de conocimiento, así como las áreas del saber que aún re-

quieren de su atención para lograr los objetivos del curso.

Los cursos se dictan a través de la plataforma, los cuales tienen dos meses de duración y flexibilidad asincrónica. El progreso se da mediante lecciones temáticas que abren semanalmente, además de los módulos académicos, que incluyen una semana introductoria para facilitar la familiarización con la plataforma y el acceso al evento inaugural, así como una semana de cierre para actividades de consolidación.

5.1.2. Introducción al modelado y representación de los datos

La plataforma redEMC contiene una red social la cual permite a los usuarios conectarse entre sí, cualquier usuario puede conectarse con otro a partir de una interacción específica dentro de la plataforma.

La misma cuenta entre otros números, con más de 35814 estudiantes, 350 docentes, 130 cursos, 63 instituciones y 63 foros, esto genera innumerables interacciones. Estas se dan en distintos niveles y contextos. Por ejemplo, a nivel de un curso específico, los estudiantes intercambian mensajes en un foro para un determinado tema dentro de ese curso. Esto da a conocer datos que son de sumo interés para el aprendizaje como participación social.

Dado el mensaje de un estudiante, los demás evalúan el contenido con posibilidad de responder, reaccionar dando me gusta o no al mensaje y ver el perfil del estudiante y/o docente que hizo parte de la interacción. Como también se puede realizar una búsqueda puntual de un usuario dentro de un curso y/o red de usuarios, además cuenta con la posibilidad de responder mensajes a otro usuario. En particular, con estas interacciones se generan relaciones de sumo interés como la de “me gusta”, “sigue a”, “respuesta” y “búsqueda”. Estas interacciones son algunas de las que se llegan a dar en la plataforma, pero es bueno destacar que se pueden llegar a presentar más. Con estas últimas, es posible ver la riqueza de los datos contenidos en la plataforma en base a interacciones proporcionadas por el usuario.

Dentro del perfil de cada usuario, se puede recomendar y agregar a una red profesional a ese usuario. Por otro lado, en la plataforma, se pueden buscar usuarios de un determinado curso para agregar a la red de amistad y de esta manera ver en forma más enfocada datos e información de los mismos.

Las posibilidades de interacción social que provee redEMC, son sumamente

valiosas en el análisis y búsqueda de soluciones para las mejoras en relaciones interpersonales e interactividad entre miembros de una red de profesionales que pueden mejorar el proceso de aprendizaje en la educación médica continua.

Por otro lado, las interacciones que se registran en la plataforma, pueden evidenciar entre otras situaciones, subgrupos de estudiantes dentro de un curso o estudiantes aislados, estudiantes que son importantes a la hora de distribuir información, otros que responden de forma positiva o negativa a un mensaje dentro de un foro, actividad de un estudiante que puede derivar en el abandono/pérdida de un curso, o caso contrario, favorezcan el aprendizaje de otros estudiantes, etc.

Las posibilidades de interacción son diversas y cualquiera de ellas, si se analiza adecuadamente aporta desde su punto de vista información que es de sumo interés en el área de Social Learning.

La plataforma incorpora un visual que permite ver una representación gráfica de las amistades del usuario que está activo. Es un gráfico atractivo pero que carece de información analítica, esta información es sumamente importante para los docentes y administradores de la misma.

5.2. Necesidades

Dada la importancia de lograr estudiar y obtener datos inherentes a cómo se relacionan y conectan los usuarios dentro de la red de enseñanza que brinda redEMC.

Surge una primera problemática, dado el modelo de datos existente, ¿cómo obtener una nueva representación orientada a grafos?. Como se mencionó, la cantidad de datos relacionados a la interacción de un estudiante en la plataforma es enorme, pero la ausencia de soluciones que permitan la extracción y visualización de los mismos, de forma que, se puedan estudiar y entender de manera más intuitiva, gráfica y analítica es una desventaja. Es sumamente importante poder contar con soluciones para el estudio del comportamiento del estudiante y sobretodo que sea fácil la extracción, procesamiento y análisis de esta información. Para esto último, es bueno contar con métricas que puedan explicar los datos visualizados, esto implica no solo soluciones que permitan la visualización de los datos, sino que también algoritmos que devuelvan resultados precisos de un determinado valor de esa representación y modelo de datos.

La plataforma redEMC utiliza un modelo de datos definido por WordPress, el cual utiliza tablas [68] de datos globales al sistema donde almacena la información y las relaciones entre sí. Son un número de tablas fijas en el cual se puede personalizar su contenido y relacionar la información del usuario con otras entidades dentro de la plataforma. Esta base se implementa sobre un modelo relacional, particularmente es necesario la instalación de un manejador de base de datos MySQL [54] para la utilización de WordPress. Los datos a analizar vienen dados por las interacciones de los estudiantes con los docentes, cursos, foros, mensajes, etc.. por lo cual, para su representación es importante poder expresar los datos de una forma gráfica. Que permita entender, y analizar de forma eficaz y eficiente los datos.

Dada la cantidad de usuarios y las posibles relaciones en la plataforma, el camino para poder cumplir con lo anterior es considerar una Base de Datos no Relacional [66] más precisamente una base de datos orientada a grafos [66]. Esta elección se debe a que las base de datos relacionales [1] tienden a ser menos eficientes a la hora de generar datos que puedan representarse como una red. Y es más notoria aún la falta de capacidad de estas bases de datos sabiendo que en promedio se superan los 2 mil estudiantes activos por curso, esto genera cerca de 400 mil relaciones. Por último, y no menos importante, el cálculo de los algoritmos para encontrar métricas de utilidad dentro de una red es notoriamente más fácil realizarlo sobre una base de datos de grafos. Algunos algoritmos son casi inviables calcularlos en una base de datos que no esté orientada a grafos.

Los primeros grandes objetivos, por lo mencionado anteriormente son, elegir la base de datos no relacional (orientada a grafos) más adecuada, construir los artefactos necesarios para crear una base de datos orientada a grafos a partir del modelo actual de datos, esto trae implícito la resolución de problemas tales como, utilizar los datos actuales e identificar cómo llevarlos a un esquema que pueda ser aceptado como entrada para la creación de una base de datos de grafos.

También definir cómo a partir de tablas relacionales obtener nodos, aristas y las relaciones entre nodos. Esto, implica una transformación y posterior normalización de los datos. Es importante tener en cuenta todos estos puntos para poder avanzar con la solución deseada.

Con una base de datos como esta y aportándole información gráfica de analítica, tareas como, analizar la importancia de las relaciones en un conjunto

grande de estudiantes se tornan sumamente fáciles.

Por otra parte, es computacionalmente factible y performante, aplicar primitivas que arrojen resultados visuales y cuantitativos sobre las métricas [2.2] más importantes en una representación de grafos. Que es, en definitiva, la información que busca agregar la plataforma. En particular, las bases de datos de grafos tienen algoritmos que optimizan los cálculos de estas métricas y brindan herramientas de consulta para trabajar con los datos.

Esto genera otros dos nuevos desafíos. El primero es poder definir consultas que ayuden a sacarle mayor provecho al nuevo modelo de datos orientado a grafos y a su vez, aplicar algoritmos que permitan obtener las métricas de Aprendizaje Social. El segundo desafío es contextualizar los datos obtenidos y encontrar la mejor forma de presentarlos para su visualización y análisis.

5.3. Requerimientos

Mediante las reuniones mantenidas a lo largo del proyecto con los integrantes de EviMed y el estudio de la plataforma educativa, se pudieron recabar distintos requerimientos, entre los cuales se encuentran los siguientes:

- **Migrar a una base de datos de grafos:** El principal requerimiento identificado es el de migrar los datos que posee EviMed almacenados en una base de datos relacional a una base de datos orientada a grafos, con el objetivo de poder realizar un análisis profundo de las interacciones de los usuarios de la plataforma. Se busca que la base de datos sea capaz de soportar un gran volumen de datos sin inconvenientes y brinde primitivas que aporten valor y permitan realizar análisis para mejorar la plataforma educativa.
- **Alojar la base de datos en Amazon Web Services:** Con el objetivo de poder brindar una solución que se adapte de la mejor manera a los componentes que actualmente conforman la plataforma redEMC. Una característica deseada es que, los artefactos que forman parte de la solución, estén alojados en la nube. En particular en Amazon Web Services [5], esto es por que los artefactos principales que componen la plataforma de EviMed, es decir, la instancia de base de datos y el despliegue del sitio, cuentan con su ambiente de producción ejecutándose en AWS.
  Poder contar con la arquitectura en Amazon Web Services, simplifica

las configuraciones y permite una mayor escalabilidad. Esto lleva a que, la instalación de los componentes a desarrollar sea un requerimiento (no funcional) tan importante como cualquier otro. Se identifican los siguientes puntos a considerar:

  - Elegir y generar una instancia, de una máquina virtual en un servicio EC2 de Amazaon Web Services [3].
  - Poder acceder a través de internet a la instancia de EC2, para poder instalar y configurar la base de datos orientada a grafos y las herramientas necesarias para complementar la extracción, procesamiento y análisis de datos.
  - Exponer cada parte de la solución a través de internet.
  - Automatizar estas tareas.

- **Carga de datos a la base:** Se deben migrar los datos desde la base de datos relacional a la nueva base de datos orientada a grafos. Es necesario determinar un procedimiento óptimo el cual no interfiera con la performance de la base de datos original y sea fácil de ejecutar, indicando periodicidad del mismo y el horario en el cual se debe ejecutar. Así como también, la creación de un modelo de datos que pueda ser procesado por la base de grafos con información alojada en la base de datos de EviMed, para esto se debe identificar la información clave y necesaria, normalizarla e importarla en la base de datos de grafos para que luego permita realizar un análisis y obtener conclusiones de utilidad.
- **Poder ejecutar primitivas y algoritmos a demanda y obtener los resultados:** Para interactuar con la base de datos, se deben brindar mecanismos que permitan desde un sistema externo realizar consultas a la base, poder ejecutar los algoritmos a demanda y obtener los resultados de forma rápida y eficiente. Es necesario establecer un formato de respuesta de las consultas, estas pueden ser un valor numérico o se pueden obtener nodos y relaciones formando un subgrafo.
- **Brindar un análisis de los resultados obtenidos:** Agregarle valor a los resultados obtenidos, mediante la creación de consultas con un análisis, significado y objetivo claros, así como también identificar primitivas y algoritmos disponibles de la base de datos a elegir que permitan realizar razonamientos y lleven a conclusiones que ayuden a tomar pasos para mejorar el aprendizaje social.

Capítulo 6

Desarrollo e implementación de la solución

En este capítulo se pretende describir la elección de una base de datos de grafos para utilizar en el desarrollo del prototipo, así como también aspectos de la infraestructura utilizada, importación de la información en la nueva base de datos y se concluye con una descripción del prototipo desarrollado.

6.1. Elección Base de datos orientada a grafos

Antes de exponer las etapas de desarrollo necesarias para brindar una solución a cada uno de los requerimientos identificados y propuestos por EviMed, es necesario detenerse y entrar más en detalle en cuál es la base de datos orientada a grafos que se usará de aquí en más y el porqué elegimos dicha base de datos.

En esta sección se detalla el análisis realizado para la elección de la base de datos. Es necesario al elegir un software para servir necesidades, ver qué características ofrece, y las bases de datos de grafos no son una excepción.

6.1.1. Bases de datos analizadas

- **Neo4j**: Neo4j es una plataforma de base de datos de grafos nativa que está diseñada para almacenar, consultar, analizar y administrar datos altamente conectados de manera más eficiente que otras bases de datos. La base de datos de Neo4j es escalable tanto vertical como horizontalmente, sin introducir problemas de coherencia o integridad de los datos

mediante su arquitectura de agrupación causal, que ahora admite agrupaciones múltiples. Las relaciones de datos son entidades de primera clase y se pueden recorrer en tiempo constante sin búsquedas de índices (se pueden establecer índices nativos de alto rendimiento). Eso permite que incluso consultas complejas entreguen resultados en milisegundos en lugar de minutos. Diseñado desde cero hacia arriba para el modelo de grafo de propiedades desde el punto en que se concibe, dibuja o visualiza el gráfico, cómo se codifica en la sintaxis específica del grafo, a cómo calcula y ejecuta consultas hasta cómo persiste las conexiones en almacenamiento. Neo4j está trazando el camino hacia el paradigma de grafos al enfocarse en la conexión de datos, en lugar de cómo las bases de datos relacionales y NoSQL se enfocan en cómo se recopilan los datos.
Actualmente, la mayoría de las bases de datos se ejecutan a través de un servidor, al que se puede acceder a través de una biblioteca cliente. Neo4j puede ejecutarse en modo integrado, así como en modo servidor. Debe entenderse el modo integrado como una instancia que almacena toda la información en el disco, y no como una base de datos en memoria, que funciona exclusivamente con la memoria interna de la máquina. Neo4j embebido es ideal para dispositivos, aplicaciones de escritorio y para aplicaciones en servidores. Ejecutar Neo4j en el servidor es la forma más común de implementar una base de datos y es lo que más se usa en la actualidad [19] [50].
Ventajas:

- Schema flexible.
- HTTP API para administrar la base de datos.
- Soporte de drivers como Java, Spring, Scala, JavaScript.
- Backups online.
- Exportación de datos de consulta a formato JSON y XLS
- Cuenta con la comunidad activa de grafos más grande en el mundo
- Fácil de aprender, usar y de cargar la información en la base de datos.
- Permite distinguir entre usuarios, roles y permisos. Autenticación conectable con estándares compatibles (LDAP, Active Directory, Kerberos).
- Permite concurrencia.

- **AllegroGraph**: AllegroGraph es una base de datos orientada a grafos moderna, consistente y persistente con alto rendimiento, de código abierto y es utilizada por el Departamento de Defensa de EE. UU. Se caracteriza por el uso eficiente de la memoria, lo que permite escalar hasta mil millones de nodos, manteniendo siempre el máximo rendimiento. Básicamente, proporciona servicios que incluyen visión, construcción, creación rápida de prototipos y prueba de concepto, desarrollo y mejores prácticas para maximizar el valor de tecnologías semánticas. AllegroGraph proporciona una arquitectura a través del protocolo REST, que es un estilo arquitectónico que consta de un conjunto coordinado de restricciones aplicado a componentes, conectores y elementos de datos dentro de un sistema distribuido[28] [16].
  Ventajas de Allegro Graph
  - Los datos y metadatos se pueden administrar usando interfaces Java, Python, Lisp y HTTP, y se pueden consultar usando SPARQL y Prolog. AllegroGraph viene con análisis de redes sociales, capacidades geoespaciales, temporales y de razonamiento todos en una misma consulta.
  - AllegroGraph usa índices quíntuples ordenados que indexarán todos los campos primarios y no primarios. Por lo tanto, los usuarios nunca tienen que preocuparse por si un determinado campo está indexado o no.
  - Seguridad de tres niveles con filtros de seguridad.
  - Integración con SOLR y MongoDB.
  - Recuperación completa y rápida.
  - Se puede alojar en la nube.
- **ArangoDB**: ArangoDB es un sistema de base de datos multimodelo desarrollado por triAGENS GmbH. Los datos pueden ser almacenados como pares de claves o valores, documentos o grafos y todo esto se puede acceder con un único lenguaje de consulta (AQL - ArangoDB Query Language). ArangoDB usa el mismo núcleo y el mismo idioma de consultas para todos los modelos de datos. Si un nuevo producto es desarrollado, por lo general el modelo requiere cambios necesarios, por lo que ArangoDB brinda la oportunidad de mantener el Producto actualizado. Combinando diferentes modelos de datos en una consulta hace que estos

sean menos complicados [62] [17].
Ventajas y Características:

- Gestión de múltiples modelos de datos con un lenguaje de consulta y núcleo único (AQL).
- API HTTP para administrar la base de datos.
- Integración con JavaScript Foxx Framework
- Consolidación: Como base de datos nativa de múltiples modelos, se puede utilizar como un almacén de documentos completo, una base de datos de grafos, un motor de búsqueda o cualquier combinación de estas tecnologías.
- Fuerte consistencia de datos: cuando se utilizan varias bases de datos de un solo modelo, la consistencia de los datos se convierte en un problema. Estas bases de datos no están diseñadas para comunicarse entre sí, lo que significa que se debe implementar algún tipo de funcionalidad de transacción para mantener los datos consistentes entre diferentes modelos. Con ArangoDB, un solo back-end administra los diferentes modelos de datos con soporte para transacciones ACID.
- Escalado de rendimiento simplificado: las aplicaciones crecen y maduran con el tiempo. Con ArangoDB, es posible hacer frente a las crecientes necesidades de almacenamiento y rendimiento escalando de forma independiente con diferentes modelos de datos. ArangoDB escala tanto vertical como horizontalmente, y si las necesidades de rendimiento disminuyen, puede reducir fácilmente el sistema backend para ahorrar en hardware y requisitos operativos.

- **OrientDB**: OrientDB es un sistema multimodelo de gestión de bases de datos NoSQL de código abierto que admite datos en documentos, grafos, clave / valor y objetos. Fue lanzado comercialmente en 2011 por OrienTechnologies e implementado en Java. Es transaccional y admite arquitectura distribuida con replicación. La manipulación de la base de datos se puede hacer en Java, SQL o con Gremlin. El almacenamiento físico de datos se puede realizar en memoria y en disco. OrientDB proporciona seguridad de datos confidenciales, mediante el uso de autenticación, contraseña y los datos en reposo cifrados. Tiene una versión Community Edition que es gratuita (Licencia Apache 2) pero no admite

funciones como escalabilidad horizontal, tolerancia a fallos, agrupación, fragmentación y replicación. La Enterprise Edition es una extensión y admite funciones como perfiles de consulta, configuración de clústeres distribuidos, registro de métricas, monitoreo en vivo, una herramienta de migración y configuración de alerta. La arquitectura de OrientDB se puede distribuir en diferentes servidores y se utilizan de diversas formas para lograr máximo rendimiento, escalabilidad y robustez.
Ventajas y Características:

- Soporta lenguaje SQL.
- Instalación fácil y rápida.
- Soporte de tecnologías web: HTTP, RESTful protocol y librerías JSON.
- Versión gratuita con Apache License 2.
- Manipulación de bases de datos usando Java.
- Puede embeber documentos como cualquier otra base de datos de documentos, pero también admite relaciones.

OrientDB tiene algunas limitaciones como no tener una herramienta de importación, a veces la documentación está desactualizada y algunos usuarios han experimentado algunos errores con el editor de grafos [20].

6.1.2. Comparaciones entre las principales características de las bases de datos

Luego de hacer el análisis individual de las principales bases de datos se procedió a realizar una comparación de las características de las mismas, estudiar bibliografía y trabajos que hayan realizado este análisis. Para la comparación de las bases de datos se tomaron en cuenta los siguientes aspectos:

- Conocimientos previos de los integrantes del grupo.
- Requisito excluyente de la institución EviMed que la nueva base de datos se pueda hostear en Amazon Web Services.
- Facilidad de aprendizaje tanto del lenguaje propio de la base de datos, si tiene, como de su instalación y configuración.
- Primitivas y algoritmos de grafos que permitan obtener métricas de aprendizaje social.

- Documentación, tanto de la configuración de la base de datos como de las primitivas y algoritmos brindados.
- Esquema flexible, para poder agregar nuevos tipos de datos, lo que le permite a la organización agregar información a la base de datos y extender el análisis realizado en este estudio.
- Bakcups: si la base de datos proporciona funciones de planificación, ejecución y restauración de copias de seguridad. Una copia de seguridad completa contiene todos los archivos de datos y la información requerida para restaurar un repositorio al estado en el que estaba en el momento de la copia de seguridad.
- Escalabilidad: debido a que el hardware sigue avanzando a un ritmo rápido, es necesario un aumento de capacidad para las bases de datos, permitiendo el crecimiento en el número de transacciones por segundo. Hay dos enfoques para escalar una base de datos: Vertical y Horizontal. Escalar verticalmente implica agregar más recursos físicos o virtuales al servidor que aloja la base de datos: más CPU, más memoria o más almacenamiento. La escala horizontal implica agregar más instancias / nodos de la base de datos para tratar con la mayor carga de trabajo.

6.1.3. Conclusión y elección de Neo4j

Aunque todas las bases de datos analizadas tienen el mismo propósito, almacenar grandes volúmenes de datos con muchas relaciones entre ellos, son muy diferentes en las funcionalidades que ofrecen, presentando ventajas y debilidades.

Concluimos que Neo4j y ArangoDB ofrecen las mejores funcionalidades para implementar una base de datos orientada a grafos, ambas cuentan con un lenguaje de consultas propio y un esquema flexible. Y dentro de estas dos bases de datos el equipo optó por Neo4j. La decisión se tomó debido a que se contaba experiencias previas en la utilización de esta tecnología, cuenta con el lenguaje Cypher para realizar consultas que es fácil de aprender y existe buena documentación del mismo, en cuanto a la instalación y la configuración de Neo4j en Amazon Web Services es muy intuitiva y fácil de realizar, y funciona sobre IOS, Windows y Linux. Cuenta con un esquema flexible por lo que ingresar nuevos tipos de datos, nodos y relaciones no es un problema. Además, dado que es una base de datos de grafos de modelos múltiples, proporciona mucha

flexibilidad y admite fragmentación. Existen librerías de código abierto que se integran fácilmente con Neo4j y permiten realizar consultas y embeber el grafo en una aplicación web.

Otro punto de gran importancia para la elección de Neo4j es que la plataforma cuenta con un plugin de fácil instalación llamado Graph Algorithms que se utiliza en conjunto con GraphApp, herramienta que permite la ejecución y configuración previa de los algoritmos. Dicho plugin permite ejecutar los algoritmos mencionados anteriormente y muchos más, los cuales ayudan a obtener las métricas sobre los grafos, estas son particularmente útiles para el Aprendizaje Social y a partir de las cuales se puede realizar un análisis e interpretar los resultados de la ejecución de los distintos tipos de algoritmos para llegar a conclusiones muy interesantes y particularmente útiles a la hora de transmitir conocimiento. Por ejemplo, a través de cómo se relacionan las personas dentro de un curso y basándose en el cálculo de métricas de aprendizaje social, es posible identificar personas particularmente útiles para que el conocimiento fluya a través de los integrantes del curso de la mejor manera posible, también permite identificar estudiantes que tengan problemas para seguir un curso o adquirir el conocimiento que se esperaría, pudiendo atacar estas falencias en etapas prematuras, obteniendo así una mejor transferencia de conocimientos entre todos los integrantes del curso.

Otro ejemplo de la utilidad de estas métricas es la posibilidad de mejorar el curso año a año, si se estudia el comportamiento y la formación de distintas comunidades dentro del curso, se podrá implementar actividades específicas como una forma de fomentar y homogeneizar las comunidades de estudiantes. Es por esto que es fundamental, para el trabajo que se llevará a cabo, la posibilidad de ejecución de algoritmos que permitan calcular las métricas de aprendizaje social de un modo óptimo dado el volumen de datos que se maneja. Neo4j ayuda a realizar diversos tipos de análisis ya que permite ejecutar sus algoritmos sobre subgrafos e incluso precargar grafos para lograr un mejor rendimiento y posibilitar la obtención de las métricas, a continuación se listan los algoritmos que permite ejecutar Neo4j en conjunto con el plugin descrito anteriormente:

- Centralidad
  - Page Rank
  - Betweenness Centrality

    - ArticleRank
    - Closeness Centrality
    - Harmonic Centrality
    - Degree
    - Eigenvector
- Detección de comunidades
    - Louvain
    - Label Propagation
    - Weakly Connected Components
    - Triangle Count
    - Local Clustering Coefficient
    - K-1 Coloring
    - Modularity Optimization
    - Strongly Connected Components
- Similaridad
    - Node Similarity
    - Approximate Nearest Neighbors
    - Cosine Similarity
    - Euclidean Similarity
    - Jaccard Similarity
    - Overlap Similarity
    - Pearson Similarity
- Path Finding
    - Minimum Weight Spanning Tree
    - Shortest Path
    - Single Source Shortest Path
    - All Pairs Shortest Path
    - A*
    - Yen's K-shortest paths
    - Random Walk
    - Breadth First Search
    - Depth First Search

Cabe destacar que todos los algoritmos provistos cuentan con una excelente

documentación que facilita el entendimiento y la utilización de los mismos para luego mapear estos a las métricas deseadas [44].

También sitios web como db-engines [18], que estudian el ranking de bases de datos de grafos según su popularidad tienen en su análisis a Neo4j en la parte superior de sus listas como la primera opción.

Por lo tanto, debido a que Neo4j cumple con todos los requisitos como se justifico anteriormente, se tomó la decisión de utilizar Neo4j como base de datos orientada a grafos, lo cual será un pilar fundamental a lo largo de todo el trabajo.

6.2. Alojar Neo4j en la nube de Amazon.

Para cumplir con el requerimiento de acceder a la base de datos Neo4j a través de una instancia remota, se dan a conocer algunos de los pasos fundamentales para que esta tarea sea posible. Para empezar se debe crear una instancia EC2 en Amazon Web Services con sus correspondientes configuraciones, para permitir posteriormente, la correcta instalación de Neo4j. Por otro lado, la instalación de la base de datos de grafos Neo4j más sus respectivas configuraciones. Para el primer paso es necesario contar con una cuenta en Amazon Web Services.

El siguiente paso es dirigirse al recurso o servicio Amazon Elastic Compute Cloud (Amazon EC2). Allí, accediendo en la consola que disponibiliza AWS EC2, se elige, la Amazon Machine Image (AMI) [4] que incluye sistema operativo y softwares que se instalarán en la instancia a crear. Se prosigue con el resto de las configuraciones de la instancia, como por ejemplo las habilitaciones de Firewall, habilitaciones de puertos, especificaciones técnicas (Hardware), etc. Se pueden ver más detalles de estas configuraciones en el anexo de "Guía de instalación de Neo4j en AWS" 1.

Un punto a destacar, es que antes de poder ejecutar la instancia creada, Amazon Web Services obliga a definir una key pair, que más adelante se utilizará para autorizar al usuario que accede mediante conexión ssh a la instancia EC2. Una vez creada la key pair, se puede descargar y conservar de manera local.

6.2.1. Acceder por ssh a la instancia creada de EC2

Para empezar, se observan los datos necesarios para acceder via ssh a la instancia. La instancia tiene entre otros datos el dominio que se ha asignado (DNS) y la IP pública (IPv4).

Con estos datos, se puede referenciar la máquina local del usuario, y acceder a través de ssh. Esto es necesario para avanzar con la instalación de la base de datos.

6.2.2. Instalación Neo4j

Una vez dentro de la instancia, se procede a instalar Neo4j. Dependiendo del sistema operativo (Linux, Windows o Mac), se siguen los pasos de instalación adecuados a ese sistema. Entre ellos se incluyen instalaciones de paquetes Java Runtime Environment y el propio software de Neo4j, ver anexo 1.

Para poder acceder desde cualquier IP al browser de Neo4j, hace falta editar el archivo config de de Neo4j "neo4j.config", aquí, se habilitan las configuraciones necesarias para que la instancia de Neo4j quede publicada en la IP y puerto que se le indica.

Volviendo a la instancia de EC2, en el apartado de Configure Security Group (Configuración de grupo de seguridad) se debe habilitar los puertos definidos en el archivo "neo4j.config"para acceder al Browser de Neo4j.

Una vez terminada la instalación y configuración, se accede al servidor de la base de datos a través de **http://localhost:7474/browser/**.

A partir de este momento, se puede tanto desde el Browser como de la API de Neo4j realizar operaciones CRUD. En particular, desde el Browser se puede realizar tareas como visualizar los grafos generados e interactuar con Plugins (herramientas) de Neo4j útiles a la hora de trabajar con la base de datos.

Para ver con más detalles, el trabajo realizado en esta etapa, como también ver paso a paso cada parte de la instalación y configuración. Consultar el Anexo 1.

6.3. Modelo de grafo y carga de datos en Neo4j

Como primer paso para realizar la carga de datos es necesario definir qué información es relevante para el tipo de estudio a realizar, es decir identificar los datos a migrar desde la base de datos de EviMed a la nueva base de datos orientada a grafos. Luego de un análisis de la plataforma se llegó a la conclusión de que es fundamental identificar a los usuarios de la misma, y las relaciones que generan estos entre sí. Para esto se considera a cada usuario de la plataforma como un nodo del grafo a crear y a las interacciones entre sí como las aristas de este. Como cada usuario en EviMed cuenta con un identificador único, el mismo se mantendrá en Neo4j, por lo que cada nodo del grafo con la etiqueta "Usuario" tendrá un atributo clave del nodo que lo identifique llamado "idUsuario".

La idea es crear un grafo que represente a la red social, por lo que la primer relación entre usuarios a tener en cuenta es la de 'SIGUE_A', en EviMed se representa esta relación cuando un usuario de la plataforma agrega a su red de amistad a otro. Entonces dado un nodo Usuario A, tendrá una arista saliente con destino un nodo Usuario B, cuando en EviMed el nodo B se encuentre en la red de amistad del nodo A. De esta manera se representaría la red social en la base de datos Neo4j.

Otra relación entre usuarios presente en EviMed e interesante a representar en la nueva base de datos, es la de 'ME_GUSTA', esta se da cuando un usuario realiza un comentario en un foro y otro usuario marca que le gusta el mismo. Para esto se tendrá una arista saliente del nodo Usuario que marca "ME_GUSTA" a un comentario con destino al autor del mismo.

Debido a la cantidad de respuestas que tiene la plataforma se optó por crear la relación 'ME_GUSTA_CANT', en donde se genera solamente una arista con un atributo "Cantidad" correspondiente a la cantidad de "me gusta" que realiza un Usuario a los comentarios de otro. El objetivo de representar esta relación se debe a poder identificar aquellos usuarios que realizan comentarios valiosos para la comunidad ya sea dentro de un curso en específico o en varios, con el fin de poder identificar usuarios que guíen el aprendizaje dentro de los foros.

Para poder realizar el análisis por cursos se agregan al grafo nodos con etiquetas "Curso" y su correspondiente primary key representada en la plata-

forma de EviMed. La idea es poder identificar a los usuarios matriculados en determinado Curso identificado por el atributo “idCurso”, para esto se agrega la relación ’USUARIO_CURSO’ que tiene como origen un nodo Usuario, matriculado en el curso y como destino el propio Curso.

Por último se toma en cuenta la relación ’RESPUESTA’, la cual representa una respuesta a un comentario en la plataforma de EviMed. La forma de representar esta relación es mediante una arista saliente del nodo que responde a un comentario, y el destino de la arista es el nodo que realizó el comentario inicial. Dado la cantidad de respuestas que tiene la plataforma se optó por crear la relación ’RESPUESTA_CANT’, en donde se genera solamente una arista con un atributo “Cantidad” correspondiente a la cantidad de respuestas que realiza un Usuario a otro.

Se considera una relación importante ya que con la misma es posible detectar usuarios que interactúan asiduamente en la plataforma o en determinados cursos particulares, observar con qué otros usuarios interactúan, si estos suelen hacerlo en cursos determinados o son comunes sus interacciones y si estos pertenecen a su red de amistad o no. Además, es posible identificar de forma temprana nodos que no son activos en los foros ya que no realizan comentarios o respuestas en estos.

6.3.1. Métodos de Importación de Datos en Neo4j

En esta sección se estudian los diferentes métodos para importar datos en Neo4j, identificando el más óptimo para el caso de estudio. Neo4j cuenta con distintos métodos para importar información [45].

- **Importar Archivos CSV:** Uno de los formatos de datos más comunes es el de las hojas de cálculo el cual se utiliza para una variedad de importaciones y exportaciones de bases de datos relacionales, por lo que es fácil recuperar datos existentes de esta manera. El comando “LOAD CSV” en Cypher permite especificar la ruta del archivo a procesar y marcar diferentes delimitadores de valor y las declaraciones de Cypher sobre cómo se quiere modelar estos datos en un grafo.

- **Importar datos mediante una API:** En la actualidad, hay muchas fuentes de datos que utilizan una API para exponer datos a través de una URL, muchas de ellas en formato JSON. También se puede importar

este tipo de datos a Neo4j utilizando la biblioteca de extensión estándar APOC y ejecutando los comandos en la línea de comandos del navegador Neo4j o en un script. El comando "apoc.load.json" permite especificar una ruta URL y cualquier parámetro necesario, seguido de declaraciones Cypher para modelar esos datos en forma de árbol en un grafo.

- **Importar desde una Base de Datos Relacional:** Para obtener los datos de una base relacional existente se debe realizar un proceso de extracción, transformación y carga (ETL). Es decir, exportar datos de los sistemas existentes, manejar cualquier manipulación necesaria en los datos para la nueva estructura y luego importar los datos transformados a la nueva base de datos. Dependiendo del entorno particular en el que se esté trabajando, las diferentes herramientas para importar desde una base de datos relacional a una de grafos pueden proporcionar soluciones mejores o más rápidas que otras. La analizada es una herramienta llamada ETL Tool. La herramienta ETL de Neo4j proporciona una GUI simple que permite cargar datos desde casi cualquier tipo de base de datos relacional a una instancia de Neo4j. El proceso pide que se configure una conexión JDBC a casi cualquier tipo de base de datos relacional, luego realiza un mapeo automático a un modelo de datos de grafos. Esta herramienta proporciona un proceso simple y directo para una importación inicial desde una base de datos relacional a Neo4j de manera rápida y eficiente. Es una herramienta impulsada por la comunidad, por lo que las actualizaciones se realizan según sea necesario y no en un cronograma programado.

El método elegido para importar datos en el proyecto es mediante archivos CSV, para esto EviMed debe crear los mismos siguiendo el formato descrito en el anexo 2 y almacenarlos en un directorio identificado, dicho directorio puede pertenecer al mismo servidor en el que se encuentre alojado Neo4j o en otro el cual pueda ser accesible. Se debe crear un archivo para cada tipo de nodo que se quiera ingresar en el grafo y un archivo para cada relación indicando el nodo saliente y el nodo entrante. Mediante consultas Cypher ejecutadas a demanda en la base de Neo4j, se leen los archivos, se procesan y se inserta la información en el grafo [47].

Para ver con más detalles, el trabajo realizado en esta etapa consultar el

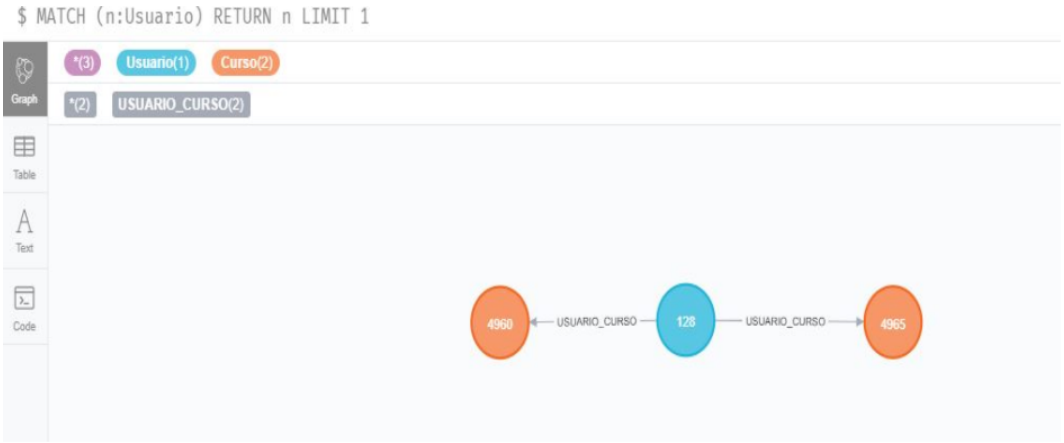


Figura 6.1: Relación entre nodos Usuario y nodo Curso

Anexo 2.

Luego de procesar los archivos y realizar la importación, se pueden apreciar ejemplos de los nodos y las relaciones creadas en la interfaz gráfica de Neo4j.

En la imagen se pueden observar las relaciones 'SIGUE_A', 'RESPUESTA_CANT' y 'ME_GUSTA_CANT' entre nodos con etiqueta 'Usuario'.

6.4. Visualización

En este punto, se hará enfoque a dos formas de cómo representar visualmente los resultados obtenidos de las consultas sobre los grafos. La idea es presentar formas de interactuar de manera fácil, intuitiva y visual con los datos, con el fin de poder deducir posibles interacciones o situaciones dentro de la red y que con una base de datos relacional no sería posible.

Una de las ventajas de trabajar con Neo4j es que la herramienta posee un Navegador, el cual es de fácil acceso para el usuario final, permitiendo entre otras cosas realizar las consultas sobre el grafo de forma sencilla, con una interfaz amigable en donde se ingresa la consulta en lenguaje Cypher y el resultado es un grafo con sus relaciones y propiedades.

Es una solución rápida que ofrece resultados visuales, cabe destacar que también existe la posibilidad de obtener los resultados en forma de tabla o archivo exportable, como lo puede ser un JSON.

Si bien es de gran utilidad la visualización que ofrece Neo4j, aparecen inconvenientes apuntando a otorgar el mayor valor a los colaboradores de EviMed.

El primer inconveniente detectado es que para poder obtener información

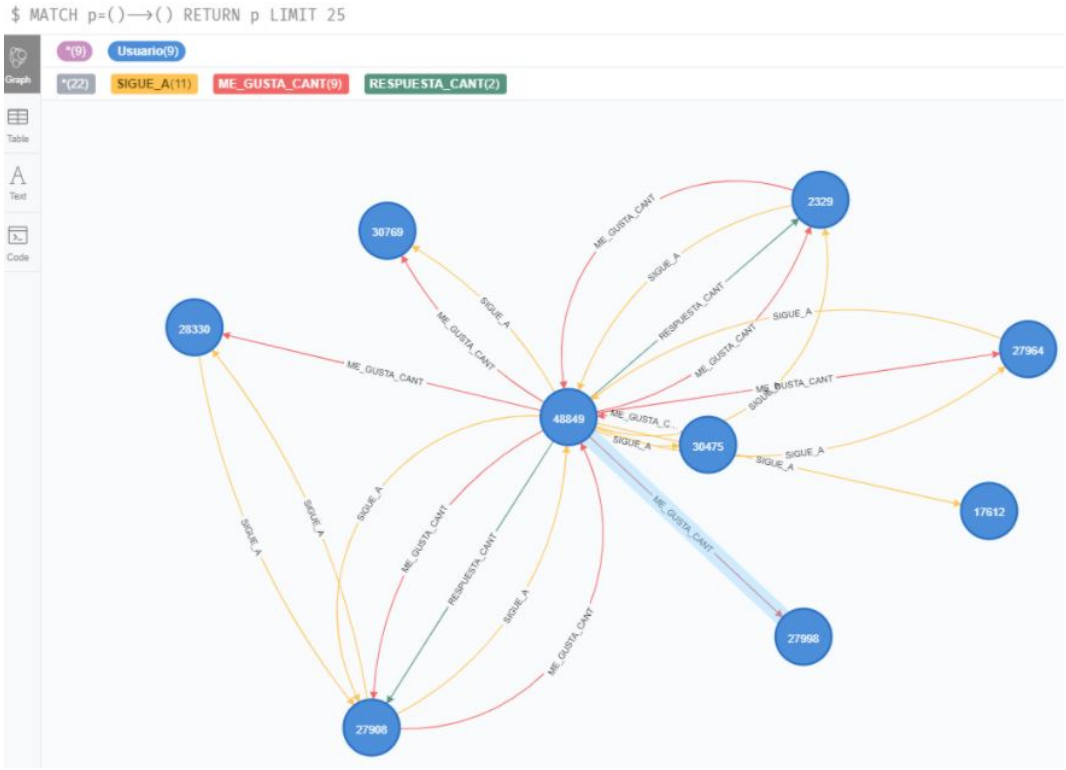


Figura 6.2: Relaciones agregadas a la base de datos

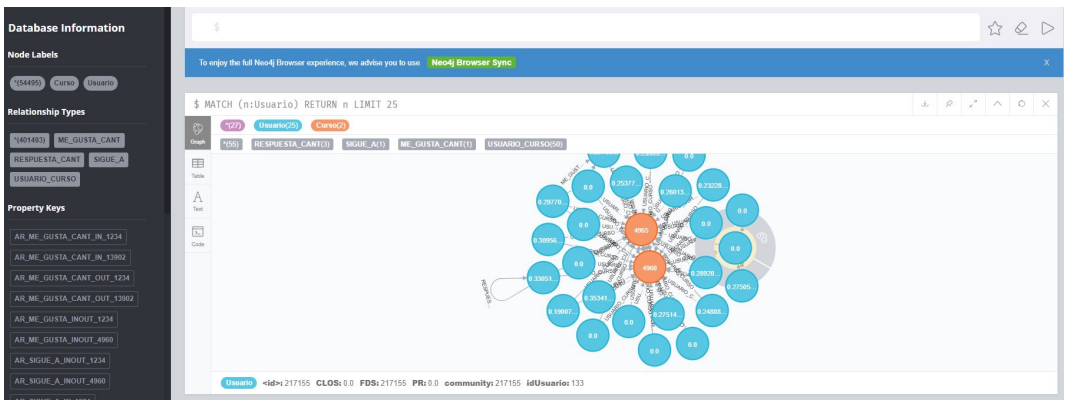


Figura 6.3: Interfaz gráfica de Neo4j

del grafo es necesario tener conocimientos en el lenguaje de consultas Cypher.

Otro inconveniente, es que al momento de generar consultas que incluyen una gran cantidad de nodos y aristas, el tiempo necesario para generar la visualización del resultado de estas consultas comienza a crecer y la manipulación del grafo una vez que se obtiene no es óptima.

En particular, en los requerimientos se menciona que los resultados se tienen que poder desplegar en tiempos de ejecución de forma tal que no perjudique la experiencia de navegación de una futura incorporación de esta solución en la plataforma.

Además, si bien no es un requerimiento en sí, no es eficiente que para poder incorporar esta solución, se acople la plataforma a la herramienta visual de Neo4j. Con todo lo que esto implicaría, por ejemplo la sincronización de visualizar un grafo, estaría por fuera de las pantallas de la plataforma ya que por el momento no se cuenta con una forma de embeber la herramienta visual de Neo4j en otros sitios web.

Para avanzar en una solución óptima, se decidió, buscar posibles alternativas a la interfaz brindada por Neo4j para generar una animación visual del grafo, la cual brinde una buena manipulación de una red con gran cantidad de nodos y aristas. A su vez, esta alternativa tiene que poder ser integrada de forma fácil y configurable. Esto último no es menor, ya que es un muy buen primer inicio pensando en integraciones con plataformas y sitios que quieran mostrar estos grafos.

6.4.1. Estudio de librerías

Para buscar una librería que permita dibujar un grafo en los distintos tipos de sitios, se orienta la búsqueda preferentemente por 3 características:

- Fácil configuración e integración con Neo4j y el sitio a desarrollar.
- Posibilidades de tener control y poder representar características más específicas de un grafo, por ejemplo: ver comunidades dentro del grafo representadas por un color.
- Que la visualización del grafo sea intuitiva y los tiempos de ejecución sean acordes a los buscados.

Analizando, se llegó a que la tecnología que mejor se adapta para integrar una visualización es JavaScript. Esto se debe a que JavaScript es el lenguaje

estándar de la web, de hecho es el lenguaje de facto. Integrable para cada aplicación, cualquier dispositivo o plataforma. Por otro lado, al ser código que ejecuta del lado del navegador, tiende a tener una mayor performance a la hora de dar respuesta en pantalla.

Siguiendo con el análisis, se puede detectar un punto de inflexión, una de las librerías que mejor se adaptan al renderizado de grafos es Vis.js [12], es una biblioteca de visualización dinámica basada en navegador. La biblioteca está diseñada para ser fácil de usar, manejar grandes cantidades de datos dinámicos y permitir la manipulación e interacción con los datos, la misma consta de los componentes DataSet, Timeline, Network, Graph2d y Graph3d [12]. Un detalle no menor de esta librería es que 18 organizaciones están apoyando a Vis.js, y además cuenta con una documentación muy buena.

Entre los componentes que incluye la librería, se encuentra Network, este es el encargado de proporcionar la visualización en pantalla.

La visualización es fácil de usar y admite formas, estilos, colores, tamaños, imágenes y más personalizados. Funciona sin problemas en cualquier navegador moderno para unos pocos miles de nodos y aristas. Para manejar una mayor cantidad de nodos, Network tiene soporte para clústeres. La red utiliza un lienzo HTML para renderizar [12].

En un principio, esta librería cumple con la mayoría de los requisitos. Pero, siguiendo con el análisis, una librería que cumple con todo esto y más lo que le agrega un plus a los requerimientos deseados es Neovis.js.

Neovis.js [12] trabaja basada en Vis.js pero hace énfasis en brindar una fácil integración y configuración con la base de datos Neo4j. Es decir, utiliza las cualidades de Vis.js para dibujar y representar grafos en pantalla, y además provee una capa de configuración a nivel programático, que es fácil de implementar y permite interpretar los resultados de las consultas Neo4j de forma sencilla, agregar información gráfica relacionada a la información almacenada en los nodos y aristas de Neo4j, por lo tanto esto permite a partir de la ejecución de los algoritmos provistos por Neo4j guardar en dichas entidades el resultado del algoritmo para ese nodo o esa arista y así posteriormente aprovechar esta información a la hora de visualizar el grafo logrando de este modo resaltar información a simple vista que de otra forma sería poco práctico y muy difícil de interpretar.

A modo de clarificar estas ideas se explicará con un ejemplo, un ciclo de vida podría ser ejecutar un algoritmo en Neo4j que identifique a qué comunidad

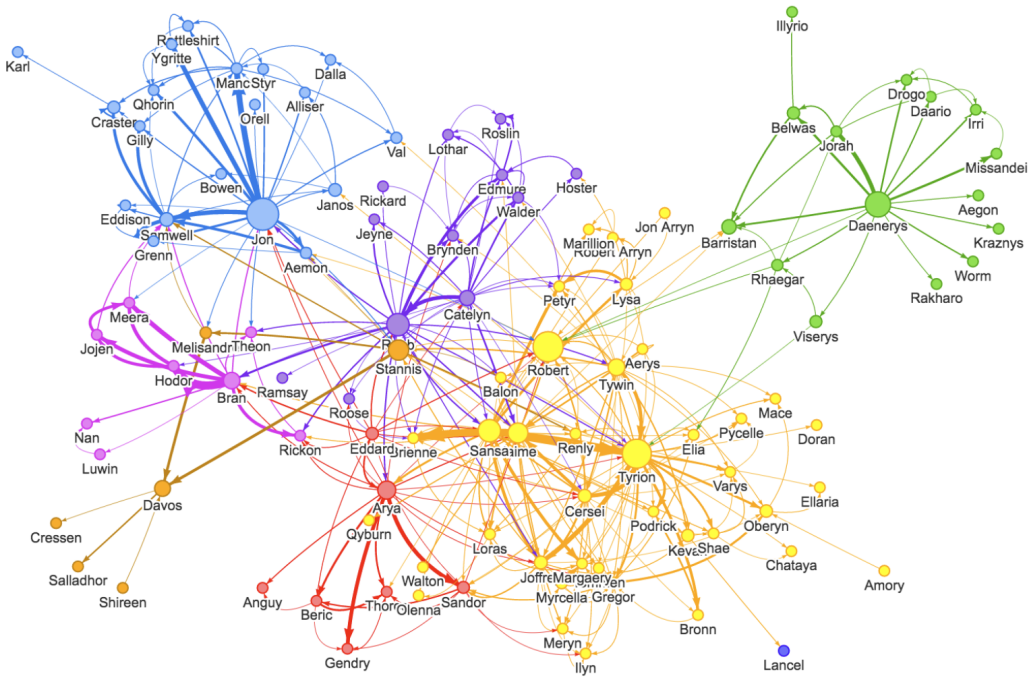


Figura 6.4: Ejemplo visualización Neovis.js

pertenece un nodo y se guarda este número identificador de comunidad en el propio nodo, luego es posible indicar a Neovis.js que utilice esta propiedad de nodo como un color, de esta forma es muy intuitivo visualizar las comunidades en el grafo, pero también si se quisiera utilizar otros algoritmos como por ejemplo el de centralidad y asignar el resultado al tamaño del nodo, obteniendo así un grafo con nodos de mayor importancia, más grandes, y pintados del color de su comunidad, además de esto gracias a Neovis.js y de la misma manera que se explica en el ejemplo anterior también es posible establecer el grosor de la arista.

Gracias a este ejemplo es claro ver cómo combinado la información recolectada por los algoritmos con las propiedades gráficas de un grafo es una excelente forma de visualizar gran cantidad de información, dando la posibilidad a realizar un análisis de la información sumamente valioso y sin demasiado esfuerzo.

Para poder evaluar el análisis y corroborar que la librería cumple con los requisitos que se plantearon, y además, explorar el sin fin de posibilidades con los que se cuenta teniendo una base de datos de grafos con las interacciones de los usuarios de la plataforma de EviMed. Se implementó una prueba de concepto. La misma tiene los siguientes objetivos:

- Explotar y explorar consultas sobre el grafo y sobre los algoritmos aplicados en el grafo
- Mostrar el desarrollo e integración entre un sitio web y la base de datos, consumiendo un API de Neo4j para crear consultas y aplicar algoritmos.
- Visualizar los resultados, poder analizar y obtener conclusiones de forma gráfica y en tiempo real.
- Exponer requisitos de EviMed de forma compacta en una solución entregable.

6.5. Desarrollo del Prototipo y funcionalidades básicas

A modo de completar los objetivos del presente trabajo y sintetizar un arduo trabajo de investigación, se decidió construir un prototipo web en el cual se permita realizar un análisis primario de la información provista por EviMed.

Con este prototipo se pretende brindarle a EviMed por un lado un código fuente que integra un método de comunicación con la base de datos Neo4j totalmente funcional, por otro, la posibilidad de mostrar la información que responde Neo4j de forma práctica para el desarrollador y amigable para el usuario final, posibilitando una personalización bastante completa del grafo resultado, lo cual facilita así el análisis de la información a simple vista.

Para concretar estos dos puntos fue necesario pensar en una lógica para las etiquetas que guardarán la información calculada por los algoritmos, que se pretende describir en las siguiente secciones.

También fue necesario crear una estructura en memoria que permita de forma relativamente sencilla agregar nuevas consultas a las ya preestablecidas sin necesidad de agregar lógica a la aplicación. Dividiremos esta sección en cuatro grandes funcionalidades que se describen a continuación.

6.5.1. Ejecución de los algoritmos por categoría

En esta etapa se detallan los pasos a seguir para ejecutar un algoritmo sobre el grafo. Para avanzar, se toma como ejemplo la ejecución del algoritmo Degree.

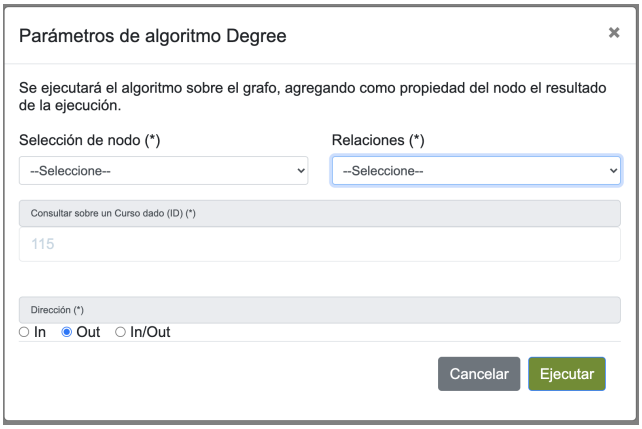


Figura 6.5: Modal donde se ejecuta Degree

Como se aprecia en la imagen anterior 6.5, la ejecución se aplica sobre el grafo, agregando a cada nodo el resultado del mismo. Es decir se agrega como propiedad del nodo el grado de centralidad con respecto a los parámetros seleccionados, para posteriormente quedar disponible en futuras consultas. Para lograr esto, en la pantalla se selecciona a qué conjunto de nodos se le aplicará el algoritmo, en este caso a los Usuarios. Luego se selecciona la relación que vincula a los nodos seleccionados, por ejemplo 'SIGUE_A'. Para terminar se selecciona el sentido de la arista, entrada (In), salida (Out) o ambas. Internamente, se genera y ejecuta una consulta en lenguaje Cypher sobre la API de Neo4j como la siguiente 6.1:

```
1 CALL algo.degree('MATCH p=((Curso {idCurso:123})--(u:Usuario))
2 RETURN id(u) as id', 'MATCH (u1:Usuario)<-[:SIGUE_A]-(u2:Usuari
    )
3 RETURN id(u1) as source, id(u2) as target',
4 {graph:'cypher', write: true,
5 writeProperty: 'DEGREE_SIGUE_A_IN_123'})
```

Listing 6.1: Codigo Cypher Algoritmo Degree

Que no es otra cosa que la conjunción de los parámetros proporcionados en la interfaz desarrollada para la ejecución de este algoritmo respecto a un subgrafo de la red completa, ya que es de interés estudiar por ejemplo la importancia de una persona en un curso y no a nivel de toda la red.

Cabe destacar el agregado de la propiedad writeProperty: 'DEGREE_SIGUE_A_IN_123' en la consulta que ejecuta el algoritmo. Esta propiedad es la que el algoritmo agrega sobre cada nodo con el resultado de la ejecución para ese nodo. La forma de identificar esa propiedad (Algoritmo + Relación + Sen-

tido + IdCurso) es importante a la hora de generar las consultas dinámicas predefinidas, las cuales muestran de forma visual, entre otras cosas el grado de centralidad de un nodo en base al tamaño del mismo. Esto es posible ya que se identifica la información de centralidad de un nodo mediante la propiedad mencionada anteriormente. Esta información es necesaria proporcionarla para configurar Neovis y este intérprete cómo dibujar los grafos mediante la información disponible en los nodos. Para ver en más detalle relacionado a este último tema ver sección de consultas dinámicas y predefinidas 6.5.3.

De forma similar al algoritmo anterior, se brinda también, una interfaz para ejecutar los siguientes algoritmos:

- Degree
- Article Rank
- Page Rank
- Closeness
- Detención de comunidad
- Label propagation
- Shortest Path
- Single Source Shortest Path

Todos siguen los mismos lineamientos para su ejecución, generar una consulta Cypher parametrizada que será impactada por medio de la API de Neo4j.

En particular, para la ejecución de los algoritmos "Single Source Shortest Path" y "Shortest Path" que brindan información acerca de caminos existentes entre un nodo origen y el resto del grafo, y desde un nodo origen y un nodo destino respectivamente, se incluyen los siguientes parámetros:
Nodo inicial/origen (nodo star) y nodo final/destino (nodo end)
generando la siguiente consulta:

```
1 MATCH (start:Usuario{idUsuario:333}), (end:Usuario{idUsuario
    :444})
2 CALL algo.shortestPath(start, end, 'cost',
3 { write:true,writeProperty:'sssp', direction:'OUTGOING' })
4 YIELD writeMillis, loadMillis, nodeCount, totalCost
5 RETURN writeMillis, loadMillis, nodeCount, totalCost
```

Listing 6.2: Codigo Cypher Algoritmo ShortetstPath

Esta ejecución agrega en los nodos involucrados en el camino (nodo inicio-nodo fin) una propiedad que identifica que es parte del camino consultado. Este, posteriormente puede ser obtenido mediante una consulta Cypher retornando la visualización del subgrafo resultante.

6.5.2. Query Cypher sobre Neo4j mediante API

Esta funcionalidad refiere a la ejecución de consultas que devuelven un grafo sin personalizar su visualización 6.6.

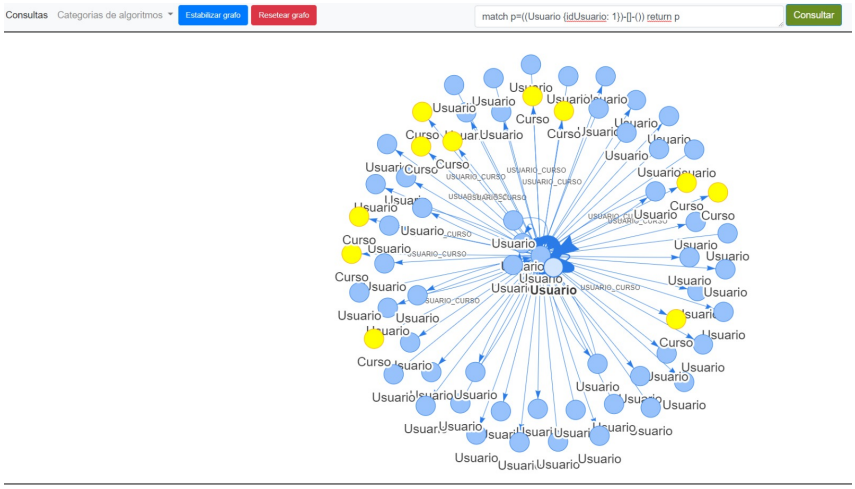


Figura 6.6: Ejemplo de query en lenguaje Cypher desde la interfaz

Brindar esta posibilidad es importante para la experiencia del usuario final al momento de realizar diversos análisis de uno o más grafos. Contar con la posibilidad de ingresar una consulta libre, habilita a que el usuario con conocimientos básicos en el lenguaje Cypher pueda realizar consultas específicas en cualquier momento para sacarle aún más jugo al análisis del grafo.

Por ejemplo, ejecutar un algoritmo que devuelva comunidades y dentro de ese subgrafo resultante (el de una comunidad específica) consultar por su identificador determinados nodos que son de interés. Esto es un ejemplo sencillo, de mucho valor al usuario final y es uno de un sin fin de análisis sobre un grafo que se pueden realizar.

6.5.3. Query dinámica sobre Neo4j mediante API

Esta funcionalidad es una de las más importantes del prototipo, donde se permite que un usuario no técnico y sin conocimientos del lenguaje Cypher

pueda seleccionar desde un combo consultas sobre el grafo en cuestión, luego de esto se despliega un conjunto de campos donde deberá ingresar ciertos datos para pre filtrar la información, aplicando la consulta a un subgrafo del toda la red, el hecho de que las consultas están predefinidas da la posibilidad de explotar al máximo la ejecución de los algoritmos y siendo estas ejecuciones precondición para ciertas consultas, esto hace posible como ya se ha mencionado con anterioridad modificar propiedad visuales del grafo, potenciando así una mejor interpretación de la información, además se implementó una estructura de datos donde se permite definir consultas sin necesidad de añadir código al prototipo, para lograr que estas consultas modifiquen las propiedades gráficas del grafo fue necesario apoyarse en una cierta nomenclatura donde quedará guardada la información calculada por los algoritmos como ya se mencionó en la funcionalidad de ejecución de los algoritmos, estas nomenclaturas se reemplazaran en el cuerpo de la consultas y en la configuración de Neovis.js al momento de ejecutar las mismas.

Se explicará a continuación en base a un ejemplo, la estructura de datos utilizada y que función cumple cada campo de la misma.

```
{
    id: 0,
    nombre: "Consulta degree por curso relacion",

    descripcion: "Esta consulta devuelve un grafo que
    muestra todos los nodos asociados al curso...",

    query: "MATCH ((Curso
    {idCurso:IDCURSO })--(u:Usuario)) MATCH
    ((Curso {idCurso:IDCURSO})--(u2:Usuario)) MATCH
    p=((u)-[r:IDRELACION]-(u2)) return u,u2, r",
    camposObligatorios: ["IDCURSO", "IDRELACION", "SENTIDO"],
    camposObligatoriosTooltip: ["Curso que se va a consultar
    (IDCURSO)", "relacion (SIGUE_A,RESPUESTA_CANT,
    ME_GUSTA_CANT)", "sentido
    de la reclacion (IN, OUT, INOUT)", "Algoritmos que se usan
    para el tama\~{n}o del nodo (DEGREE, PAGERANK..)",
    "algoritmo que se usa para la comunidad(LP..)"],

    camposOpcionales: ["LIMITE"],

    camposOpcionalesTooltip: ["Limite total de nodos y
```

```
    relaciones que se meustran en el grafo"],

    configLabels: {
      Usuario: {
        caption: "idUsuario",
        size: "DEGREE_IDRELACION_SENTIDO_IDCURSO",
        community: "LP_IDRELACION_SENTIDO_IDCURSO",
        title_properties: ["idUsuario", "
           DEGREE_IDRELACION_SENTIDO_IDCURSO", "
           LP_IDRELACION_SENTIDO_IDCURSO"]
      },
      Curso: {
        caption:"",
        size: "",
        community: ""
      }
    },
    configRelationships: {
      SIGUE_A: {
        caption: false,
        thickness: "nada"
      },
      RESPUESTA_CANT: {
        caption: false,
        thickness: "cantRespuestas"
      },
      ME_GUSTA_CANT: {
        caption: false,
        thickness: "cantMeGusta"
      }
    }
  }
```

Listing 6.3: Codigo generador de consulta dinámica

- id - Permite definir un identificador de la consulta
- nombre - Nombre que se muestra en el combo
- descripción - Descripción visual para el usuario de lo que se muestra en la consulta y sus precondiciones
- query - Consulta propiamente dicha, con etiquetas de referencia 6.4.

```
1 MATCH ((Curso {idCurso:IDCURSO })--(u:Usuario)) MATCH ((
    Curso {idCurso:IDCURSO})--(u2:Usuario)) MATCH p=((u)-[r
```

```
:IDRELACION]-(u2)) return u,u2, r
```

Listing 6.4: Consulta Cypher con etiquetas

Como se puede ver en el ejemplo IDCURSO posteriormente será reemplazado por el curso que ingrese el usuario, del mismo modo IDRELACION.

- camposObligatorios - Aquí se ingresan los campos que se le requieren al usuario para luego ser reemplazados, el usuario no podrá continuar la ejecución si no ingresa un valor en todos estos campos.
- camposObligatoriosTooltip - Pequeña explicación de cada campo obligatorio respetando el orden.
- camposOpcionales - Idem a los anteriores con la salvedad de que no son obligatorios para continuar con la ejecución
- camposOpcionalesTooltip - Pequeña explicación de cada campo opcional respetando el orden.
- configLabels - Configuración por tipo de nodo
  - Usuario - Configuración para nodos de tipo usuario
    - caption - Debe rellenarse con la propiedad que vendrá en el nodo con el valor que se quiere mostrar como etiqueta del nodo.
    - size - Campo que permite establecer la propiedad que vendrá en el nodo con el valor que le dará el tamaño al nodo
    - community - Campo que permite establecer la propiedad que vendrá en el nodo con el valor que le dará el color al nodo
  - Curso - Configuración para nodos de tipo Curso, la cual tiene la misma estructura que el tipo anterior (Usuario)
- configRelationships - Configuración por tipo de artista.
  - SIGUE_A Configuración establecida para aristas
    - caption - Propiedad que estará en la arista que se quiere mostrar en el grafo como label de la misma.
    - thickness - Este campo permite configurar qué propiedad de la arista dará el valor de grosor en ella.
  - RESPUESTA_CANT - Idem a SIGE_A pero para relaciones de tipo 'RESPUESTA_CANT'.
  - ME_GUSTA_CANT - Idem a SIGE_A pero para relaciones de tipo 'ME_GUSTA_CANT'.

Figura 6.7: Ejemplo consulta precargada

Las propiedades nueve y diez son propiedades de configuración de Neovis.js, la cual se cargará a la hora de mostrar el grafo resultado. Para más detalles de esta configuración puede consultar [13].

Cabe mencionar que en este trabajo de investigación y prototipado, se establecieron cinco consultas predefinidas, las cuales aplican a ciertas métricas de analítica social que se explicaran más adelante en la sección de resultados.

6.5.4. Resetear grafo

Dada la gran cantidad de información que se acumula en los nodos y aristas luego de la ejecución de los algoritmos, se permite borrar esas propiedades para abrir paso a nuevas ejecuciones y análisis sobre el grafo.

Capítulo 7

Evaluación y casos de ejemplos reales

En esta sección se pretende dar a conocer dos ejemplos reales de situaciones que se dan en una plataforma de aprendizaje, en particular en un curso dado con sus respectivos estudiantes analizando la centralidad de un curso. Posteriormente se evalúan resultados orientados a los tiempos de respuesta, tiempo de carga, test de carga y otros resultados correspondientes a la performance de la solución propuesta.

7.1. Aplicando los algoritmos Degree y Label Propagation

Para comenzar el análisis, se asume que se tiene en una instancia de Neo4j ejecutando con los datos correspondientes a los usuarios de un curso y una relación que los haga interactuar entre ellos. En este caso se filtra de entre 120 cursos uno solo para poder enfocar el análisis. Se utiliza el curso con identificador '1234'. Para obtener los usuarios a considerar en esta prueba, se realiza una consulta sobre el grafo, precisamente sobre el curso '1234' el cual contiene '1098' usuarios matriculados. Como relación a considerar se utiliza 'SIGUE_A'.

El primer paso a realizar es la ejecución del algoritmo. Este lo primero que realiza es una consulta que devuelve un subgrafo (el cual toma el algoritmo Degree como entrada), este subgrafo no es otra cosa que el filtro aplicado para obtener los usuarios relacionados con el curso '1234'.

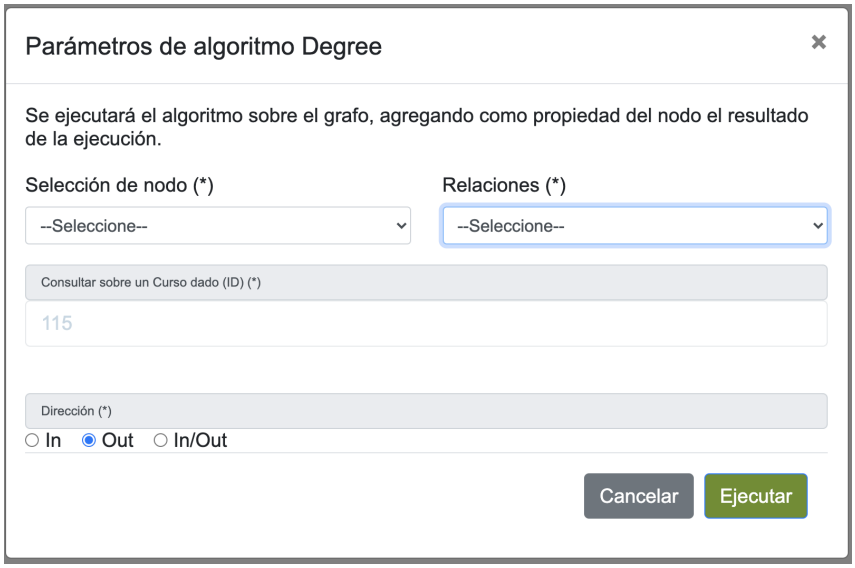


Figura 7.1: Parámetros del algoritmo Degree

Ahora sí, se aplica Degree sobre el subgrafo. Si la ejecución es exitosa en cada nodo se agrega el resultado de la ejecución, es decir el valor numérico de centralidad para cada nodo perteneciente al subgrafo.

Para ver el resultado obtenido, se ejecuta la query "Degree por curso relación"

Figura 7.2: Query predeterminada

y esto hace que se visualice el grafo en pantalla:

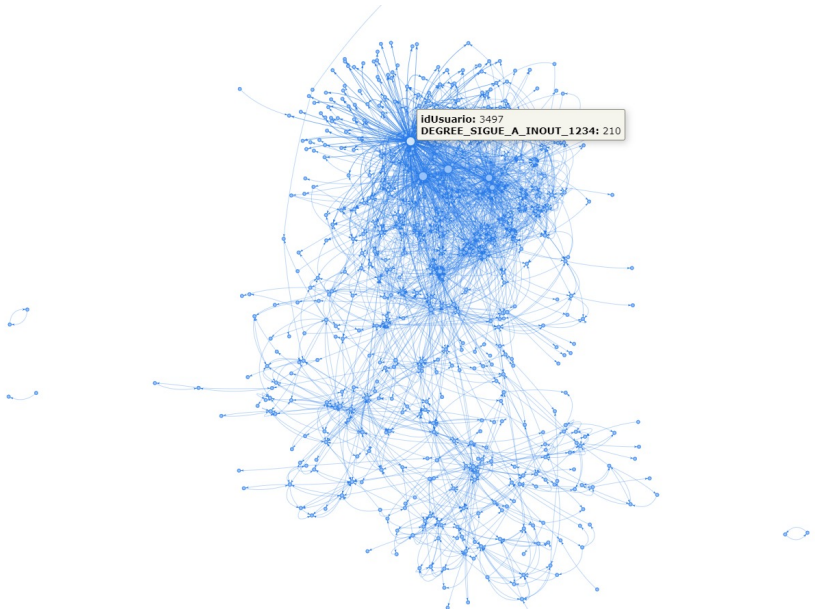


Figura 7.3: Grafo resultado de la query

Como se puede apreciar, en una primera observación del grafo ya se ven patrones interesantes. Para empezar, al ubicarse sobre un nodo se ve la información identificatoria del nodo Usuario y el resultado de la ejecución del algoritmo. En el caso anterior se realizó la ejecución considerando las aristas de entrada y de salida de cada uno de los nodos.

Siguiendo con la observación, vemos que sobresale el tamaño de algunos nodos, esto es por que la consulta visualiza los nodos con más centralidad dibujándolos más grande para brindar más información a simple vista. Entonces, esto nos muestra usuarios que dentro del contexto del curso '1234' son importante tenerlos en cuenta, ya que son usuarios que por la centralidad en la relación 'SIGUE_A' pueden llegar a influir en los otros ya sea en forma negativa o positiva, o simplemente son nodos que destacan a la hora de poder distribuir información.

Por otra parte, viendo el grafo resultado hay nodos que se ven aislados del resto que tienen muy pocas relaciones y una centralidad baja, también nodos que forman un ciclo están relacionados únicamente entre ellos, lo que puede llegar a dar a entender que forman grupos interactuando entre sí y no con el resto.

Siguiendo con el análisis, aplicando sobre el subgrafo el algoritmo de Label

Propagation (Propagación de etiquetas), que realiza el cálculo para decidir en qué comunidad se encuentra cada nodo y asignar el identificador de comunidad a la información del nodo. Brinda la posibilidad de sumar al análisis que se tiene de la centralidad de los nodos pero esta vez visualizando la información de las comunidades. Se ejecuta el algoritmo.

Figura 7.4: Parámetros algoritmo Label Propagation

Ahora se ejecuta nuevamente la siguiente consulta "Degree por curso relación" y esta vez el grafo resultante es el siguiente:

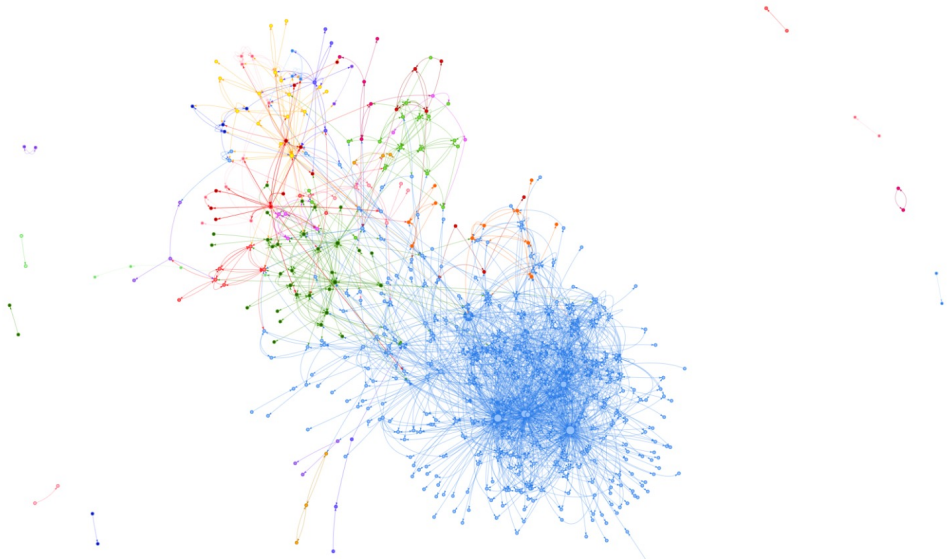


Figura 7.5: Grafo resultante separado por comunidades

Ahora que se tiene una vista de las comunidades dentro del curso, se ve claramente que el mayor porcentaje de usuarios se establece en la comunidad

de color azul, y además es la comunidad donde se encuentran los usuarios con mayor grado de centralidad.

Esto, conociendo un poco más la información de los usuarios, es decir sumándole datos por ejemplo patronímicos, académicos o realizando este análisis para otro tipo de relaciones se puede evaluar el por qué, por ejemplo un usuario tiene un mal o buen desempeño en el curso, o también ver quienes son y por qué los usuarios menos involucrados con el resto lo cual puede afectar de distintas formas.

Cabe destacar que la visualización tiene un dinamismo que permite arrastrar y agrandar el grafo, separar los nodos para ver más clara la información.

Continuando con el ejemplo, se realiza la ejecución de los algoritmos anteriores pero esta vez considerando solo las aristas entrantes (IN)

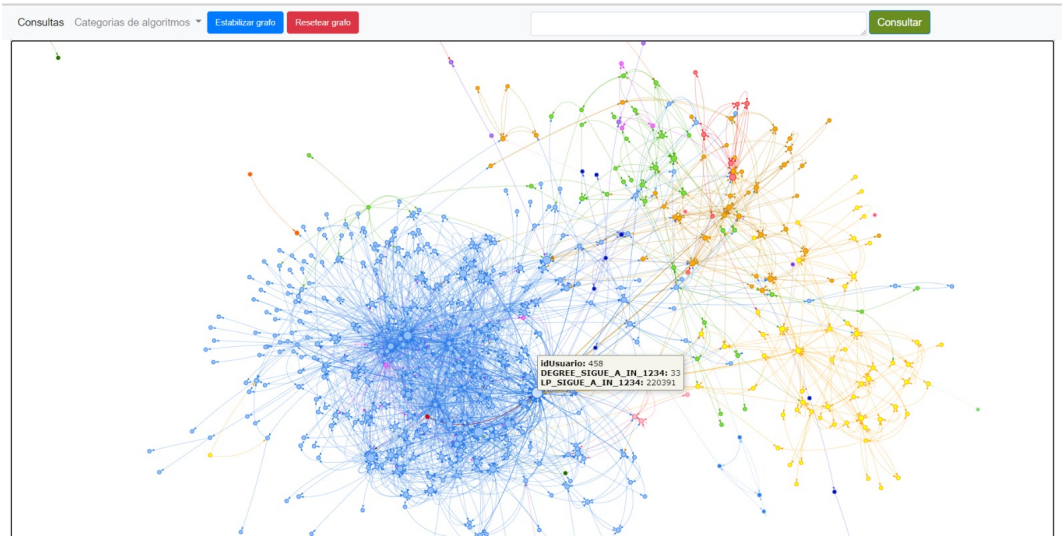


Figura 7.6: Grafo separado por comunidades curso '1234'

De esta forma, se puede enfocar el análisis por ejemplo, en cómo los usuarios de las comunidades o de una comunidad específica se relaciona con un usuario en particular y en contraposición, si se utilizan las aristas salientes (OUT) ver cómo un usuario se relaciona con el restos de los usuarios dispersos en las comunidades.

E sto es simplemente un análisis para demostrar el potencial de ejecutar dos algoritmos sobre un subgrafo. Es realmente grande la utilización que puede llegar a tener en un caso real donde se cuenta con más información acerca de las entidades que están representadas en el grafo.

Por último, sumado a lo anterior, si utilizamos consultas que no están

predefinidas en la interfaz desarrollada. Se puede por ejemplo ver con más claridad los usuarios relacionados con el usuario de mayor centralidad, ya sea tomando las aristas de entradas como las de salidas.

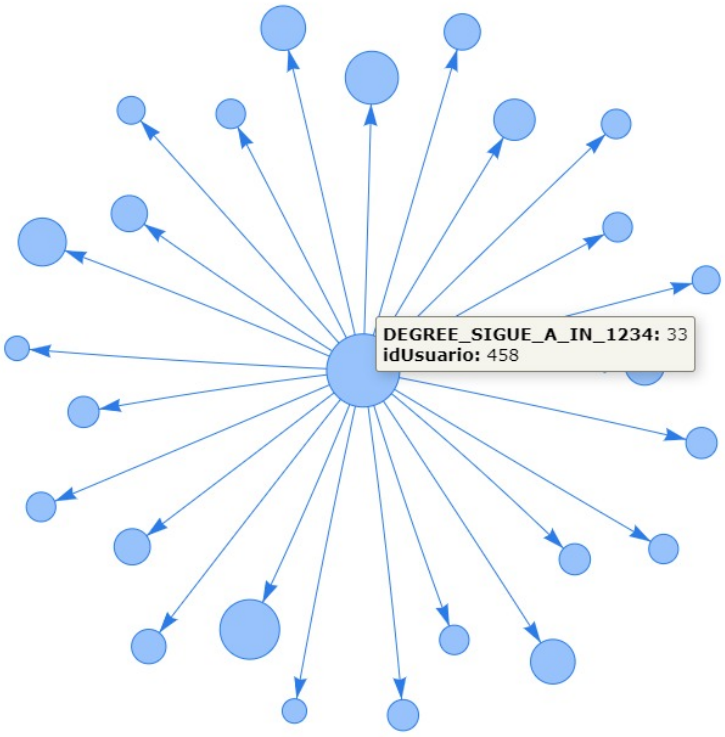


Figura 7.7: Nodo con mayor grado de centralidad y sus vecinos

7.2. Análisis Centralidad de un curso

En esta sección el estudio se centra en el análisis del curso ('13902') para poder estudiar e indagar sobre la centralidad del subgrafo formado por usuarios matriculados en el curso dado, obteniendo conclusiones con los datos brindados por EviMed.

Para realizar la evaluación se ejecutan los algoritmos sobre la relación 'ME_GUSTA_CANT' presente en el grafo, se debe recordar que esta relación se genera cuando el usuario de nodo origen marca con me gusta un comentario del nodo destino.

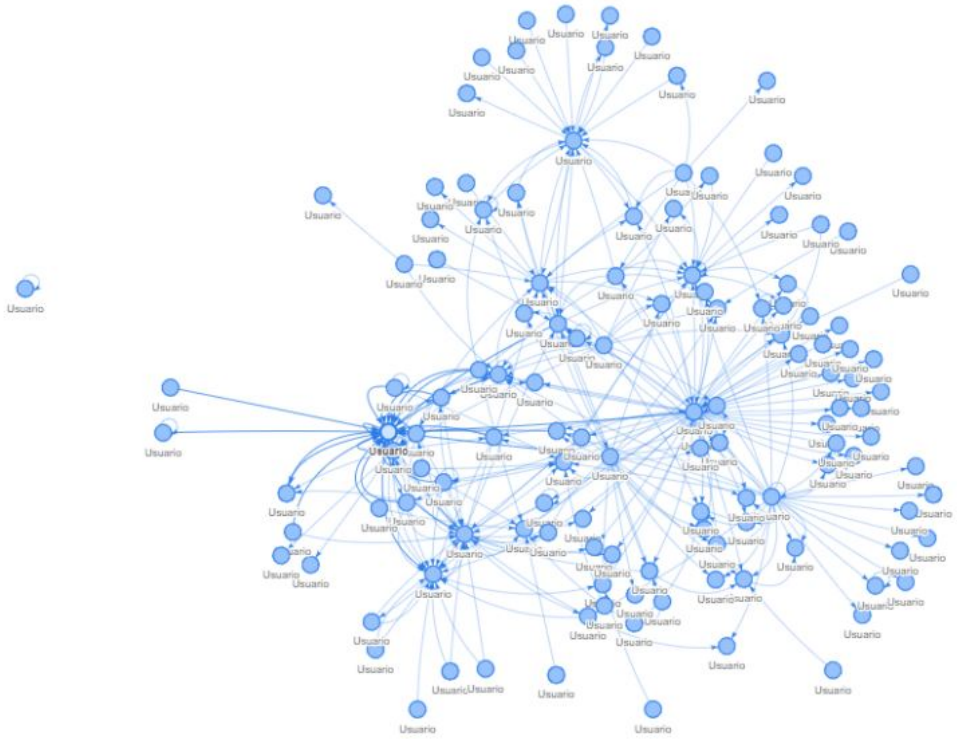


Figura 7.8: Grafo generado por la relación ME_GUSTA_CANT del curso 13902

Se ejecutan los algoritmos Degree, PageRank y Clossenes sobre el subgrafo de usuarios matriculados sobre el curso '13902' y sobre la relación 'ME_GUSTA_CANT'. Luego de la ejecución de estos se guarda en los nodos el valor de centralidad que arroja cada algoritmo con una etiqueta específica. Con la siguiente consulta

```
1 MATCH (u:Usuario) where exists(u.DEGREE_ME_GUSTA_CANT_IN_13902)
2 WITH  u
3 ORDER BY u.DEGREE\_ME\_GUSTA\_CAN\_IN\_13902 DESC LIMIT 1
4 RETURN u
```

Listing 7.1: Ejecución del algoritmo Degree

se obtiene el nodo que tiene la métrica de valor más alta para cada algoritmo y se procede con el análisis.

El nodo con mayor valor de centralidad luego de ejecutar el algoritmo degree es el '1198' y como se tiene en cuenta el sentido de la relación entrante (IN) se puede decir que el nodo '1198' es el usuario que recibió mayor cantidad de "me gusta" en sus mensajes. El mismo se puede ver coloreado con rojo en el grafo en la siguiente imagen.

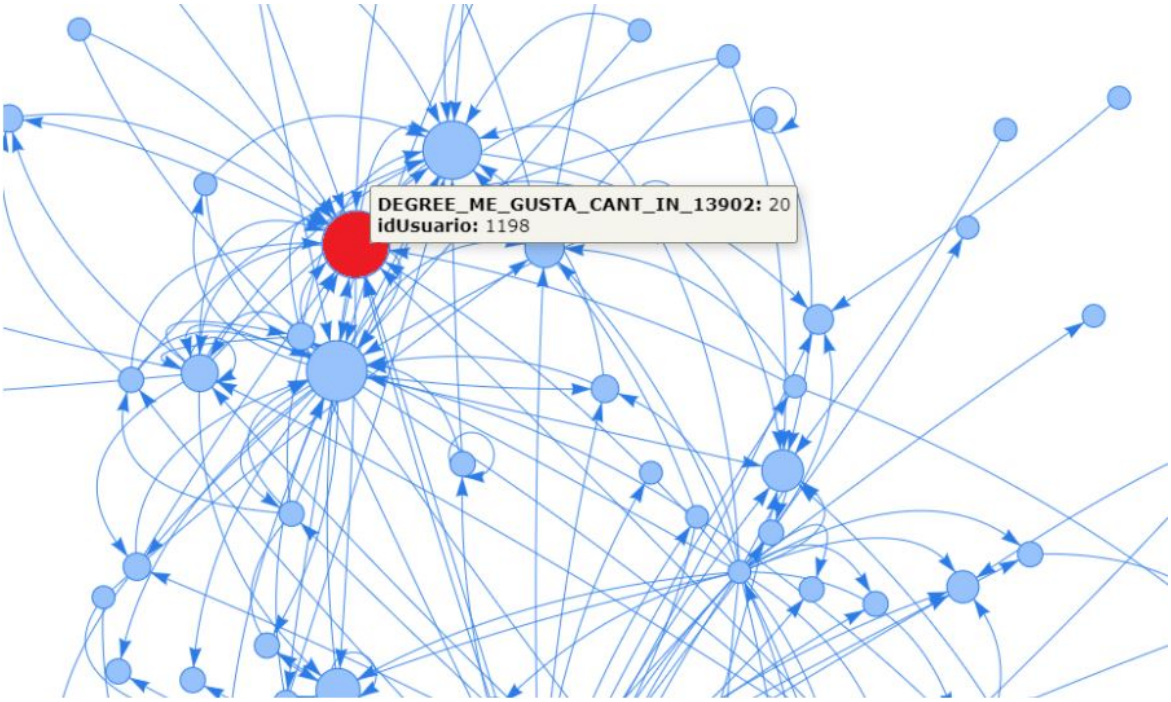


Figura 7.9: Nodo con mayor grado de Centralidad según algoritmo Degree

Por lo tanto se identificó a un nodo el cual interactúa con la red y esta reconoce que sus aportes son valiosos para toda la comunidad. Es de vital importancia identificar estos nodos ya que pueden ser los que guíen el aprendizaje dentro de la plataforma, y esto es valioso debido a la mecánica de aprendizaje que se lleva a cabo en la misma.

Otro de los algoritmos ejecutados fue el ArticleRank, el cual a diferencia del Degree mide la influencia transitiva de los nodos y la conectividad, es decir no tiene solamente en cuenta la cantidad de relaciones como lo hace el Degree sino también la calidad de esas relaciones y determina así la importancia de un nodo. ¿A qué se le dice calidad de la relación? llevándolo al caso de estudio, según la cantidad de relaciones 'ME_GUSTA_CANT' el nodo con mayor valor de centralidad es el "1198", pero si este nodo tiene una relación saliente a otro nodo con un número menor de aristas entrantes, esa relación tiene peso en el algoritmo ArticleRank, ya que un nodo con una gran cantidad de "me gusta" o aprobaciones por la comunidad, está "aprobando" el comentario de otro usuario por lo que el contenido de este tiene relevancia para la red.

De esta forma, el nodo con mayor valor de centralidad devuelta por el ArticleRank es el '6883', distinto al nodo '1198' arrojado por el algoritmo Degree. De esta forma se puede profundizar el análisis y tener en cuenta otras

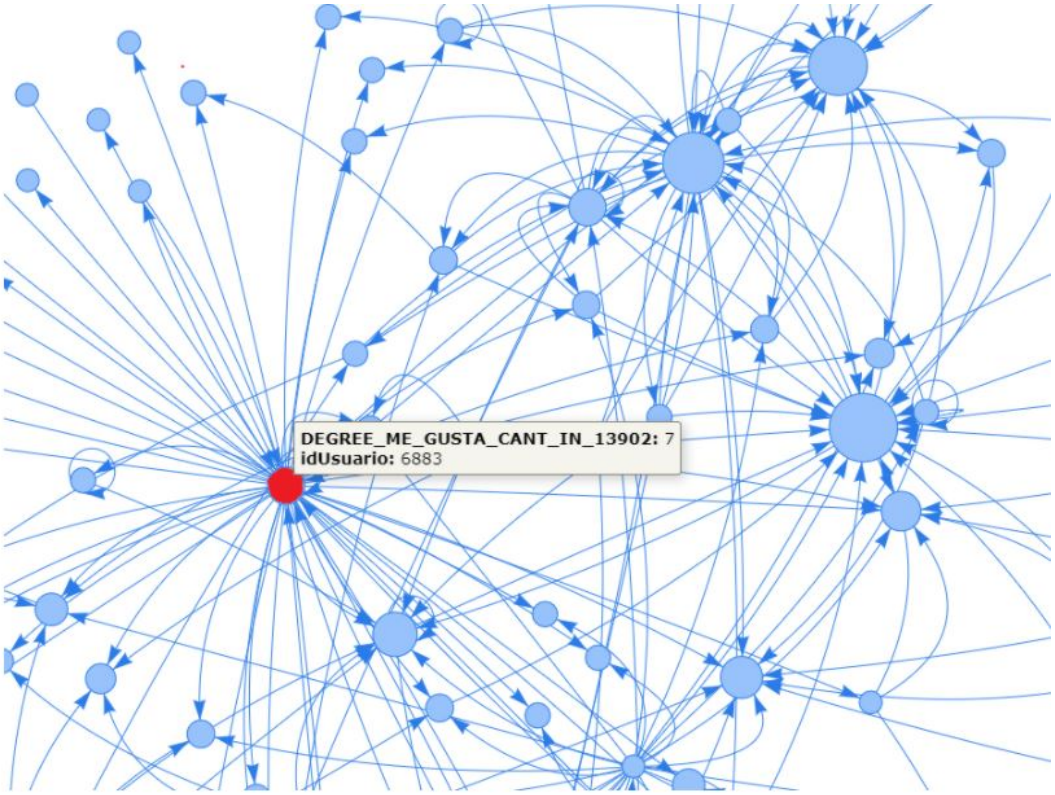


Figura 7.10: Nodo con mayor grado de centralidad según ArticleRank

variables.

Por último el algoritmo a analizar es el Closeness, el cual es un algoritmo capaz de detectar nodos que son capaces de esparcir información a través de toda la red de forma eficiente. La centralidad de proximidad de un nodo mide su distancia media (distancia inversa) a todos los demás nodos, es decir aquellos nodos con una puntuación de cercanía alta tienen distancias más cortas a todos los demás nodos. Esto lo hace calculando para cada nodo la suma de sus distancias a todos los demás nodos, basándose en el cálculo de las rutas más cortas entre todos los pares de nodos. Luego, la suma resultante se invierte para determinar la puntuación de centralidad de cercanía para ese nodo. Luego de la ejecución se observa que arroja otro nodo distinto a los dos algoritmos anteriores como el nodo con mayor centralidad de cercanía, el nodo es el '144'. De esta forma con los distintos algoritmos se tienen distintas herramientas para identificar nodos los cuales son buenos para transmitir información, nodos que son influyentes en la red y nodos los cuales no lo son, o nodos que no interactúan con el resto de la red. Además es posible realizar el análisis a toda la red o a cursos particulares ya que los algoritmos permiten realizar su ejecución sobre

subgrafos.

7.3. Análisis de tiempo de ejecución de los algoritmos

En esta sección se realiza el estudio de los tiempos de ejecución de los algoritmos para poder constatar que los mismos pueden ser ejecutados a demanda sin una gran demora y procesando una gran cantidad de nodos y aristas.

Para llevar a cabo el análisis se ejecutan una vez los algoritmos Degree, PageRank, ArticleRank, Closeness y Label Propagation sobre toda la red para que participen todos los nodos de etiqueta "Usuario" y sobre la relación 'SIGUE_A'. La pruebas se realizaron sobre un servidor que cuenta con 16 GB de memoria RAM, y dos CPU:

- -cpu:0
  - description: CPU
  - product: Intel(R) Xeon(R) CPU E5-2686 v4 @ 2.30GHz
  - vendor: Intel Corp.
  - physical id: 401
  - bus info: cpu@0
  - slot: CPU 1
  - size: 2700MHz
  - capacity: 3GHz
  - width: 64 bits
- -cpu:1
  - description: CPU
  - vendor: Intel
  - physical id: 402
  - bus info: cpu@1
  - slot: CPU 2
  - size: 2700MHz
  - capacity: 3GHz
  - capabilities: cpufreq

Neo4j brinda información al momento de ejecutar los algoritmos permitiendo desglosar el tiempo de ejecución de estos de la siguiente manera:

- Cantidad de nodos considerados.
- Milisegundos que tarda en cargar el grafo.
- Milisegundos que tarda en ejecutar el algoritmo.
- Milisegundos que tarda en escribir el resultado en cada nodo.

Los tiempos para cada algoritmo son los siguientes:

Tarea	Degree	ArticleRank	PageRank	Closeness	Label Propagation
Cantidad de Nodos	54375	54375	54375	54375	54375
Milisegundos que tarda en cargar el grafo	95	88	167	241	69
Milisegundos que tarda en ejecutar	19	17285	72	49218	670
Milisegundos que tarda en escribir	849	207	376	654	24
Total	963	17580	615	50113	763

Tabla 7.1: Tiempos en milisegundos por tarea de cada algoritmo

Como se puede ver en la fila Total de la tabla de tiempos anterior los algoritmos procesan los 54375 nodos de etiqueta "Usuario" y sus aristas de forma rápida y eficaz sin ningún tipo de demora, por lo que perfectamente los algoritmos pueden ser ejecutados a demanda, brindando así una herramienta muy útil a la plataforma de EviMed.

7.4. Algoritmos de predicción mediante redes neuronales sobre grafos

Un último punto a destacar del presente proyecto, es la implicancia directa en el área de Aprendizaje Automático [32], en particular, en la utilización de redes neuronales para la creación de algoritmos capaces de predecir eventos

dentro de las redes de aprendizaje. La solución generada en este proyecto de grado se está utilizado en trabajo de Doctorado en esta temática.

La idea inicial es, a partir de la solución desarrollada para generar grafos en base a los datos de la plataforma de EviMed, utilizarlos en algoritmos de predicción con base en Análisis de datos en redes complejas. Es claro el potencial que engloba el desarrollo de este proyecto, dando posibilidades de enriquecer la información analítica contenida en los grafos con componentes que se encarga de predecir eventos reales dentro de la plataforma. [57].

Esto es un ejemplo más de la importancia y el valor que aporta tener una representación en forma de red o grafo de las interacciones dentro de una plataforma de aprendizaje.

Capítulo 8

Conclusiones y trabajo futuro

En esta sección se presentan las conclusiones extraídas del trabajo realizado. En la sección 8.1 se mencionan los objetivos cumplidos y los desafíos encontrados, en la sección 8.2 se presentan líneas de trabajo a futuro.

8.1. Conclusiones

Durante el transcurso de este trabajo se estudió la forma de utilizar los datos contenidos en la plataforma de EviMed para agregar valor a la misma. Para esto se hizo primero una revisión de plataformas y bibliografía enfocada en Analítica de Aprendizaje Social, luego un estudio pormenorizado de las posibilidades que ofrece la plataforma elegida Neo4j, y finalmente la extracción y carga de los datos del modelo relacional original al de la base de datos orientada a grafos y su instalación y configuración en Amazon Web Services.

Se buscó brindar una solución automática en la generación de representaciones gráficas de las interacciones existentes en el proceso de aprendizaje. Esta solución toma como entrada datos relacionales que representan una red de aprendizaje muy completa y genera grafos que facilitan el estudio de las relaciones entre los diferentes actores como usuarios, docentes y cursos que participan en el proceso.

Por lo que el trabajo fue guiado por tres grandes objetivos, que cada uno engloba desafíos que se resolvieron de manera eficaz. Primero, el poder entender qué base de datos era la más adecuada, teniendo en cuenta que soluciones como estas suelen crecer de manera veloz, un requerimiento implícito y que de alguna forma facilita la integración con otros sistemas es la instalación y despliegue

de la base de datos en Amazon Web Services.

Por otro lado la elección de Neo4j ofrece las mejores funcionalidades para implementar una base de datos orientada a grafos, cuenta con un lenguaje de consultas propio fácil de aprender y un esquema flexible. Su instalación y configuración en Amazon Web Services resultó muy intuitiva y fácil de realizar. Cuenta con un esquema flexible por lo que ingresar nuevos tipos de datos, nodos y relaciones no es un problema. Además, dado que es una base de datos de grafos de modelos múltiples, proporciona flexibilidad y admite fragmentación. Existen librerías de código abierto que se integran fácilmente con Neo4j y permiten realizar querys y embeber el grafo en una aplicación web. Otra ventaja fundamental de Neo4j es la posibilidad de ejecutar los algoritmos mencionados permitiendo obtener las métricas sobre los grafos, estas son fundamentales para el trabajo que se llevó a cabo.

Segundo, y otro punto interesante, que llenó nuestras expectativas es el ver los resultados reales de analizar un grafo de 54375 nodos, contextualizando la información visual en interacciones de usuarios dentro de un curso. Poder explicar y entender cómo se comportan los usuarios, quiénes son relevantes en una red, cómo se dividen en comunidades, y saber como se separan los usuarios entre sí, entre otras cosas, que son realmente valioso. Más aún, partiendo de datos que no eran interpretables de esa forma. Es esto lo que nos convence de que utilizar esta solución en conjunto con las plataformas, en este caso EviMed, es sumamente valioso y aporta en gran medida en el área de Aprendizaje Social.

Como último objetivo, el proceso completo de extracción de datos de un modelo relacional, la generación de un esquema que si se aplica de forma adecuada contempla la gran mayoría de los grafos que son de interés estudiar. Además la generación de un grafo en una base de datos. Este primer paso por la automatización, es un gran aporte a los desarrolladores que quieran seguir el procedimiento y resolver otros problemas similares.

Basta con observar la contemporaneidad de la mayoría de las referencias, además del lanzamiento continuo de versiones de las herramientas utilizadas para notar que esta área en particular es relativamente nueva, e innovadora [31] [64]. Las herramientas utilizadas cambian constantemente logrando cada vez mejores resultados en todos los aspectos [49]. Por lo que creemos que con el paso del tiempo serán herramientas muy completas que de forma práctica permitan mejorar aún más este tipo de trabajos y análisis.

Como parte de esta conclusión nos gustaría mencionar que quedamos muy

satisfechos con el resultado obtenido y creemos que puede ser aprovechado en gran medida por EviMed, que conociendo más a fondo la lógica institucional, se podría explotar en varias áreas, como puede ser Aprendizaje Social tanto para mejorar los cursos o para seguimiento de estudiantes, así como también a nivel de venta y promoción de cursos.

8.2. Trabajo futuro

El proyecto realizado genera varias líneas de investigación que pueden ser abordadas en futuros trabajos.

En primera instancia y como se mencionó en la sección anterior, una de las principales contribuciones del presente trabajo es el poder generar a partir de los datos de la plataforma, un grafo con la información de las interacciones de los usuarios dentro de una red de usuarios en Evimed.

Siguiendo esa base, generar un grafo que contemple más situaciones de interacción incluyendo otras entidades adicionales a las ya contempladas, como por ejemplo se podría pensar incluir nodos de tipo recursos, documentos o actividades y relacionarlos, sería de gran importancia ya que sumaría información valiosa desde el punto de vista analítico y enriquecería la información que se brinda en esta solución.

También desarrollar grafos que permitan explorar otros niveles de la plataforma de aprendizaje.

Por ejemplo, generar grafos que abarquen no solamente interacciones de usuarios dentro de un curso, sino que también se puedan estudiar y visualizar qué sucede en los niveles donde se engloban varios cursos en áreas temáticas, como por ejemplo área de especialización, rama de medicina, etc.

Del punto anterior se desprende directamente la necesidad de trabajar en implementar nuevas querys que permitan reflejar métricas de aprendizaje social, o información que se pueda interpretar como un análisis que según la organización entienda pertinente para su negocio.

Por otra parte, es interesante sumar más algoritmos a los estudiados en esta solución. A la hora de sumergirse en el análisis de los grafos para poder extraer información valiosa relacionada al área del Social Learning, entran en juegos muchos más algoritmos que los tenidos en cuenta en el presente trabajo, que sin duda, también otorgan una gama muy variada e interesante de datos. Ejemplos de estos algoritmos son aquellos que permiten obtener información

acerca de la similaridad de dos nodos, es decir comparar a dos nodos estudiando si estos comparten determinados vecinos en el grafo utilizando la métrica de Jaccard.

Otro punto interesante es poder generar las herramientas necesarias para que la información analítica contenida en el estudio del grafo se pueda enriquecer con información "estática" de los usuarios. Al decir estática se hace referencia a los datos que no son considerados para formar el grafo, información de diferente índole (académica, patronímica, etc..). La idea es poder utilizarla para que ayude a entender lo que se visualiza en las métricas otorgadas en el estudio de cada grafo. Y también, que pueda ser utilizada en el aprendizaje de cómo actuar frente a las situaciones detectadas. Por ejemplo, agregar región o país al cual pertenece un nodo "Usuario" y poder realizar análisis con estos datos, observando si usuarios del mismo país interactúan más entre ellos generando comunidades o si se logra evitar esta situación en la plataforma logrando que los usuarios interactúen entre sí independientemente de su país o región.

Es importante destacar que si bien estas tareas son fundamentales para un análisis más profundo y de utilidad para EviMed, estos serían imposibles de no existir una versión simplificada de la realidad como es el caso del modelo que se emplea en este trabajo, y que de esta forma aporta información de gran valor para un primer análisis visual de los datos.

Referencias bibliográficas

[1] Alcolea Velázquez, J., F. P. Álvarez Arrieta and L. Moreno Iglesias, "Generador del modelo relacional y esquemas de bases de datos a partir del modelo entidad/relación," Universidad Complutense de Madrid Facultad de Informática, 2007.

[2] Álvarez Chappore, R., S. Weikert Perdomo, A. I. Rodríguez Techera, R. Balaguer Prestes, V. Zorrilla de San Martín Gründel, R. Martínez Barcellos and N. Geymonat Vignolo, *Educación y tecnología en el uruguay* (2015).

[3] AmazonWS, *Amazon elastic compute cloud (amazon ec2.*, https://aws.amazon.com/es/ec2/?nc2=h_ql_prod_fs_ec2&ec2-whats-new.sort-by=item.additionalFields.postDateTime&ec2-whats-new.sort-order=desc, [Online] Accedido; 15 de noviembre de 2020.

[4] AmazonWS, *Amazon machine images.*, https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/AMIs.html, [Online] Accedido; 15 de noviembre de 2020.

[5] AmazonWS, *Informática en la nube con aws.*, https://aws.amazon.com/es/what-is-aws/?nc1=f_cc, [Online] Accedido; 15 de noviembre de 2020.

[6] Ancízar Palacio N., J. O., "Detección de comunidades en redes: Algoritmos y aplicaciones," Universidad de Granada E.T.S. de Ingeniería Informática y de Telecomunicaciones Departamento de Ciencias de la Computación e Inteligencia Artificial Granada, España, 2013.
URL https://pdfs.semanticscholar.org/0b0c/b35ce617d48e1bb251afab2f541b1305399e.pdf

[7] Angles, R. and C. Gutierrez, "An introduction to graph data management," Springer, 2018.

[8] Blackboard., *North america — blackboard.com.*, https://www.blackboard.com/, [Online] Accedido; 15 de noviembre de 2020.

[9] Borgelt, C., M. Steinbrecher and R. Kruse, "Graphical Models: Representations for Learning, Reasoning and Data Mining." Cornwall, U.K.: Wiley, 2009.

[10] Bouadjenek, M. R., H. Hacid and M. Bouzeghoub, "Social networks and information retrieval, how are they converging? A survey, a taxonomy and an analysis of social information retrieval approaches and platforms," Elsevier, 2016.

[11] Brin, S. and L. Page, *The anatomy of a large-scale hypertextual web search engine*, in: *Proceedings of the Seventh International Conference on World Wide Web 7*, WWW7 (1998), p. 107–117.

[12] community edition., *A dynamic, browser based visualization library.*, https://visjs.org/, [Online] Accedido; 15 de noviembre de 2020.

[13] Contrib., N., *neovis.js. graph visualizations in the browser with data from neo4j.*, https://github.com/neo4j-contrib/neovis.js/, [Online] Accedido; 15 de noviembre de 2020.

[14] Costa, L. d. F., F. A. Rodrigues, G. Travieso and P. R. Villas Boas, "Characterization of complex networks: A survey of measurements," Taylor & Francis, 2007.

[15] Costa, L. d. F., F. A. Rodrigues, G. Travieso and P. R. Villas Boas, "Characterization of complex networks: A survey of measurements," Taylor & Francis, 2007.

[16] db engines, *Allegrograph system properties*, https://db-engines.com/en/system/AllegroGraph, [Online] Accedido; 15 de noviembre de 2020.

[17] db engines, *Arango system properties*, https://db-engines.com/en/system/ArangoDB, [Online] Accedido; 15 de noviembre de 2020.

[18] db engines, *Db-engines ranking - popularity ranking of graph dbms*, https://db-engines.com/en/ranking/graph+dbms, [Online] Accedido; 15 de noviembre de 2020.

[19] db engines, *Neo4j system properties*, https://db-engines.com/en/system/Neo4j, [Online] Accedido; 15 de noviembre de 2020.

[20] db engines, *Orientdb system properties*, https://db-engines.com/en/system/OrientDB, [Online] Accedido; 15 de noviembre de 2020.

[21] Documentation, N., *Introduction to graph databases.*, https://neo4j.com/graphacademy/online-training/introduction-to-neo4j-40/, [Online] Accedido; 15 de noviembre de 2020.

[22] Easley, D., J. Kleinberg et al., "Networks, Crowds, and Markets: Reasoning About a Highly Connected World." Cambridge, U.K.: Cambridge University Press., 2010.

[23] Elias, P., A. Feinstein and C. Shannon, "A note on the maximum flow through a network." IEEE, 1956.

[24] EviMed, *Evime sobre redemc*, https://redemc.net/evimed-sobre/, [Online] Accedido; 15 de noviembre de 2020.

[25] EviMed, *Redemc*, https://redemc.net/, [Online] Accedido; 15 de noviembre de 2020.

[26] Ferguson, R. and S. Buckingham Shum, "Social Learning Analytics: Five Aproaches. Proc. 2nd International Conference on Learning Analytics y Knowledge, (29 Apr-2 May, Vancouver, BC)," ACM Press: New York., 2012.

[27] Fortunato, S., "Community detection in graphs," Elsevier, 2010.

[28] Franz, I., *Introduction — allegrograph 7.0.3*, https://franz.com/agraph/support/documentation/current/agraph-introduction.html, [Online] Accedido; 15 de noviembre de 2020.

[29] Freeman, L. C., "Centrality in social networks conceptual clarification," North-Holland, 1978.

[30] Freeman, L. C., "Centrality in social networks conceptual clarification," North-Holland, 1978.

[31] Gabit., *Evolución del e-learning (infografía) — gabit.*, http://www.gabit.org/gabit/?q=es/evolucion-elearning-infografia/, [Online] Accedido; 20 de noviembre de 2020.

[32] Garat, D. and G. Moncecchi, *Curso: Aprendizaje automático.*, https://eva.fing.edu.uy/course/view.php?id=43/, [Online] Accedido; 15 de noviembre de 2020.

[33] González-Morales, L., *Metodología para el diseño instruccional en la modalidad b-learning desde la comunicación educativa*, Razón y Palabra (2017), p. 72.

[34] Hosseinmardi, H., A. Ghasemianlangroodi, R. Han, Q. Lv and S. Mishra, "Towards understanding cyberbullying behavior in a semi-anonymous social network," 2014.

[35] Ingrassia, C. and A. Giménez, "Aulas extendidas o ampliadas: ¿cómo y para qué usarlas?" 2016.

[36] Johnson, L., S. A. Becker, M. Cummins, V. Estrada, A. Freeman and C. Hall, "NMC Horizon Report: 2016 Higher Education Edition," Austin, Texas: The New Media Consortium., 2016.

[37] Luxburg., U. V., "A tutorial on spectral clustering." Statistics and Computing, vol. 17, 2007.

[38] Marsden, P. V., "Measures of Network Centrality." International Encyclopedia of the Social y Behavioral Sciences, 2015.

[39] Martínez, D. A., *Blended learning: modelo virtual-presencial de aprendizaje y su aplicación en entornos educativos*, Alicante: Departamento de Comunicación y Psicología Social, Universidad de Alicante (2007).
URL https://cutt.ly/Wg3mRmI

[40] Mihalcea, R. and D. Radev, "Graph-based natural language processing and information retrieval," Cambridge university press, 2011.

[41] Montes, A. and A. Espino Maldonado, "Bases de datos." Cobol en español, [Web; accedido el 02-11-2020].
URL http://www.escobol.com/modules.php?name=Sections&op=viewarticle&artid=104

[42] Moodle.TM., "La plataforma de aprendizaje de código abierto del mundo," 2020.
URL https://moodle.com/es/lms/

[43] Needham, M. and A. Hodler, "Graph Algorithms: Practical Examples in Apache Spark and Neo4j," O'Reilly Media, 2019.
URL https://books.google.com.uy/books?id=UwIevgEACAAJ

[44] Neo4j, *Chapter 1. introduction - the neo4j graph data science library manual v1.3*, https://neo4j.com/docs/graph-data-science/current/introduction/, [Online] Accedido; 15 de noviembre de 2020.

[45] Neo4j, *Data import - neo4j graph database platform*, https://neo4j.com/developer/data-import/, [Online] Accedido; 15 de noviembre de 2020.

[46] Neo4j, *Debian - operations manual*, https://neo4j.com/docs/operations-manual/current/installation/linux/debian/, [Online] Accedido; 15 de noviembre de 2020.

[47] Neo4j, *Importing csv data into neo4j - neo4j graph database platform*, https://neo4j.com/developer/guide-import-csv/, [Online] Accedido; 15 de noviembre de 2020.

[48] Neo4j, *Neo4j graph platform – the leader in graph databases*, https://neo4j.com/, [Online] Accedido; 15 de noviembre de 2020.

[49] Neo4j, *Neo4j supported versions- neo4j graph database platform*, https://neo4j.com/developer/kb/neo4j-supported-versions/, [Online] Accedido; 20 de noviembre de 2020.

[50] Neo4j, *Neo4j top ten reasons*, https://neo4j.com/top-ten-reasons/, [Online] Accedido; 15 de noviembre de 2020.

[51] Newman, M., "Networks: An Introduction." New York, NY, USA: Oxford University Press, Inc, 2010.

[52] Newman, M. and M. Girvan, "Finding and evaluating community structure in networks." Physical Review E - Statistical, Nonlinear, and Soft Matter Physics, vol. 69, no. 2 2., 2004.

[53] Nooy, W. d., A. Mrvar and V. Batagelj, "Exploratory Social Networks Analisis with Pajek." New York, United States of America: Cambridge University Press, 2005.

[54] Oracle, C. and affiliates., *Mysql documentation*, https://dev.mysql.com/doc/, [Online] Accedido; 15 de noviembre de 2020.

[55] Pallisé, J. S., "Campusvirtual UB: un nuevo entorno de enseñanzaaprendizaje," 2008.
URL http://www.untumbes.edu.pe/vcs/biblioteca/document/varioslibros/0317.%20Campus%20virtual%20UB%20%20Un%20nuevo%20entorno%20de%20ense%C3%B1anza-aprendizaje.pdf

[56] Pina, A. B., *Blended learning. conceptos básicos*, Pixel-Bit. Revista de medios y educación (2004).

[57] Rodriguez Bocca, P., *Curso: Análisis de datos en redes complejas.*, https://eva.fing.edu.uy/enrol/index.php?id=945/, [Online] Accedido; 15 de noviembre de 2020.

[58] Rosen., K. H., "Discrete Mathematics and Its Applications, 7 ed." New York, NY, U.S.A.: McGraw-Hill Companies, Inc., 2012.

[59] Sasaki, B., J. Chao and R. Howard, "Graph Databases for Beginners." 2018.

[60] Schwaber, K., "Agile project management with Scrum," Microsoft press, 2004.

[61] Symeon, P., K. Yiannis, V. Athena and S. Ploutarchos, "Community detection in social media, performance and application considerations," 2012.

[62] the native multi-model NoSQL database, A., *Introduction to arangodb documentation — arangodb documentation*, https://www.arangodb.com/docs/stable/, [Online] Accedido; 15 de noviembre de 2020.

[63] Tsvetovat, M. and A. Kouznetsov, "Social Network Analysis for Startups," .º'Reilly Media, Inc.", 2011.

[64] UCUR, *Programa de entornos virtuales de aprendizaje – portal del programa de entornos virtuales de aprendizaje (proeva).*, https://proeva.udelar.edu.uy/, [Online] Accedido; 20 de noviembre de 2020.

[65] Universidadviu, *Características, tipos y plataformas más utilizadas para estudiar a distancia*, https://www.universidadviu.com/caracteristicas-tipos-y-plataformas-mas-utilizadas-para-estudiar-a-distancia/, [Online] Accedido; 15 de noviembre de 2020.

[66] Vera, C. and S. Migani, "Introducción a las bases de datos de grafos: experiencias en Neo4j," 2019.

[67] Wenger, E., B. Trayner and M. De Laat, "Promoting and assessing value creation in communities and networks: A conceptual framework," The Netherlands: Ruud de Moor Centrum., 2011.

[68] WordPress.org., *Wordpress support*, https://wordpress.org/support/, [Online] Accedido; 15 de noviembre de 2020.

ANEXOS

Anexo 1

Guía para instalar Neo4j en una instancia EC2 de AWS

1.1. Introducción

Esta sección tiene dos objetivos. Por un lado mostrar el paso a paso para la creación de una instancia EC2 [5] con sus correspondientes configuraciones, para permitir posteriormente la correcta instalación de Neo4j. Y por otro lado, la instalación de la base de datos de grafos Neo4j más sus respectivas configuraciones.

1.2. Creación instancia EC2 en AWS

En primer lugar, se inicia sesión en Amazon Web Services y se accede a la consola de EC2 para crear una instancia. A continuación se selecciona la opción Launch instance

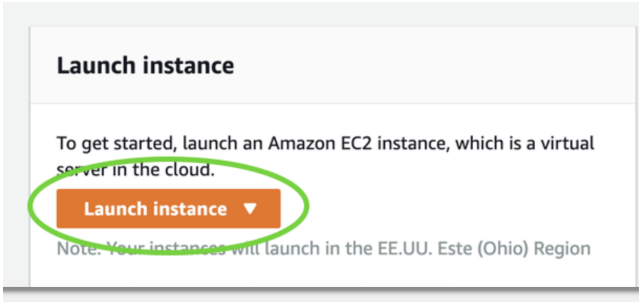


Figura 1.1: Iniciar instancia EC2

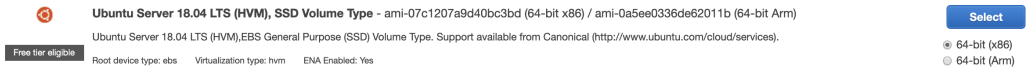


Figura 1.2: Ubuntu Server 18.04 LTS

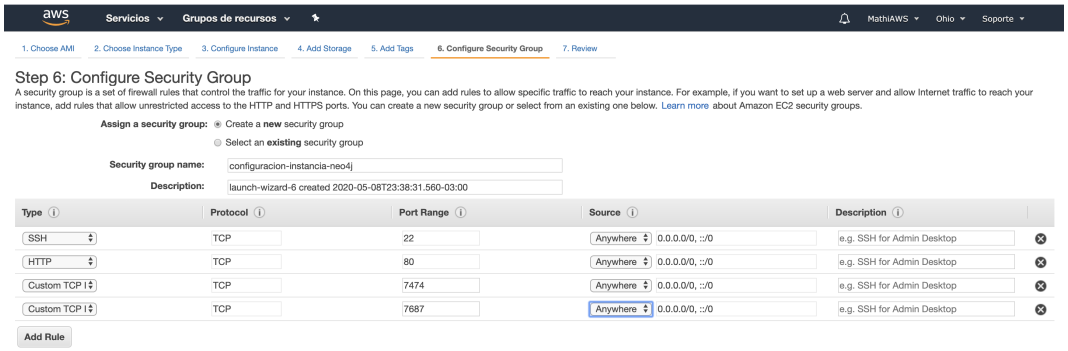


Figura 1.3: Reglas de acceso de la instancia

Ahora se tiene que elegir la imagen de la instancia que se va a levantar. Por ejemplo una imagen de Ubuntu Server 18.04 LTS

Luego, se deberán configurar las siguientes propiedades de la instancia a ejecutar:

- Choose AMI
- Choose Instance Type
- Configure Instance
- Add Storage
- Add Tags
- Configure Security Group

Para todos los pasos, a excepción de la sección de Configure Security Group se utilizan las propiedades que AWS otorga para una cuenta gratuita, es decir almacenamiento, memoria ram, etc.. básicos.

1.2.1. Configure Security Group

Este punto de la configuración de la instancia es importante, ya que es la que permite habilitar las IPs y puertos que son necesarios para acceder al Browser de Neo4j que posteriormente se instalará.

Aquí se definen las reglas de acceso a la instancia de la siguiente forma:

Como se puede ver en la imagen anterior, el puerto 22 es necesario que esté habilitado para poder acceder a la instancia de EC2 a través de la consola y

realizar todas las operaciones necesarias para la instalación de Neo4j

El puerto 80, es necesario para acceder desde el navegador a la interfaz de Neo4j

Por último, se habilitan los puertos 7474 y 7687. El primero es el puerto donde se levantará el Browser de Neo4j, y el segundo es necesario ya que Neo4j lo utiliza para que los SDK y lenguajes de programación puedan conectarse a la base de datos y poder ejecutar consulta sobre ella.

Nota: Para todos los casos, se permite el acceso de todas las IPs solamente por simplicidad y porque no había una restricción para este punto.

Una vez configurada esta sección, y terminado todos los pasos de configuración de la instancia. Se hace click en Launch y nos saldrá el siguiente Pop up

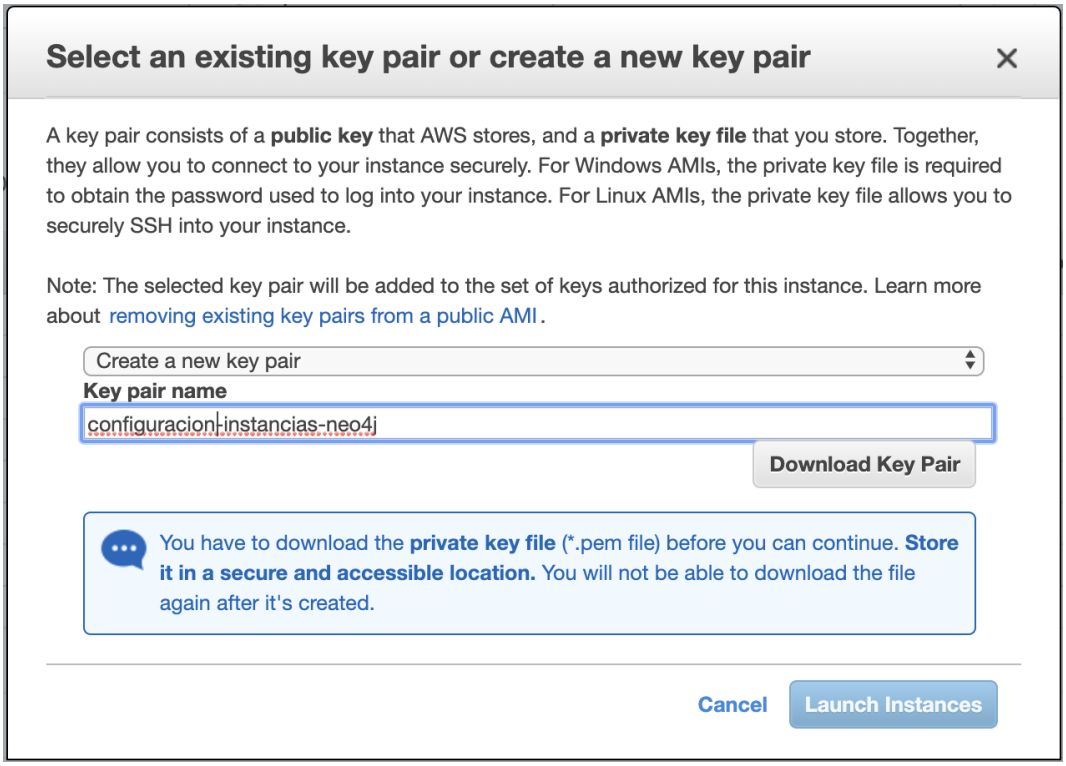


Figura 1.4: Creación de key pair

Aquí, es importante definir una key pair necesaria para autentificación para acceder mediante ssh a la instancia creada. Una vez que se crea, es posible descargarla haciendo click en Download Key Pair y conservarla en una pc. Se hace click en Launch Instance para terminar de levantar la instancia [3].

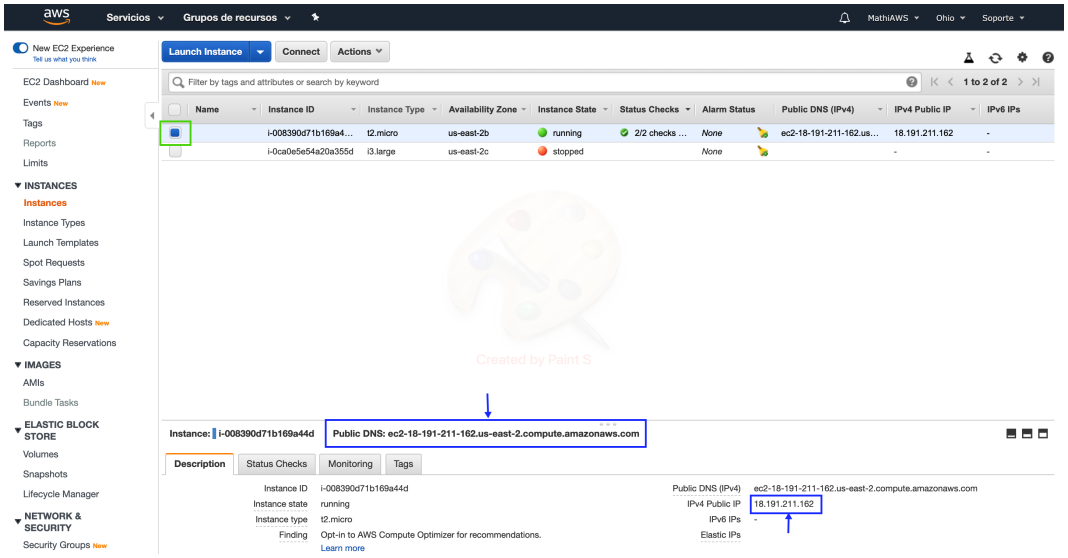


Figura 1.5: Información de la instancia

1.3. Acceder por conexión ssh a la instancia creada de EC2

Para empezar, se muestran los datos son necesarios para acceder vía conexión ssh a la instancia.

Cuando se observa en la instancia que se acaba de crear, se puede ver la descripción de la misma. Entre otros datos los que interesan por el momento son el dominio que se ha creado para la instancia (DNS) y la IP pública (IPv4).

Con estos datos se accede a la instancia por conexión ssh por ejemplo desde la consola del ordenador o desde alguna herramienta como Putty para Windows o similar. A continuación un ejemplo con la herramienta Termius de Mac. Primeros se crea la conexión con la instancia

En la imagen anterior se ve que se agrega la Key Pair generada anteriormente

Se termina de agregar la información para la conexión y se inicia la misma.

Nota: El nombre de usuario cuando se conecta varía según la instancia que se crea, en este ejemplo el usuario es “ubuntu”.

1.4. Instalación de Neo4j

- Neo4j se implementa en Java, por lo que deberá tener instalado Java Runtime Environment (JRE). Sino se tiene intentar lo siguiente:

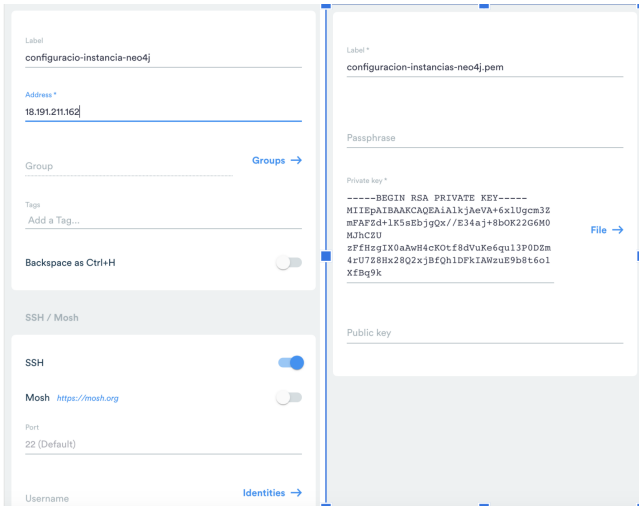


Figura 1.6: Configuración ssh con key pair

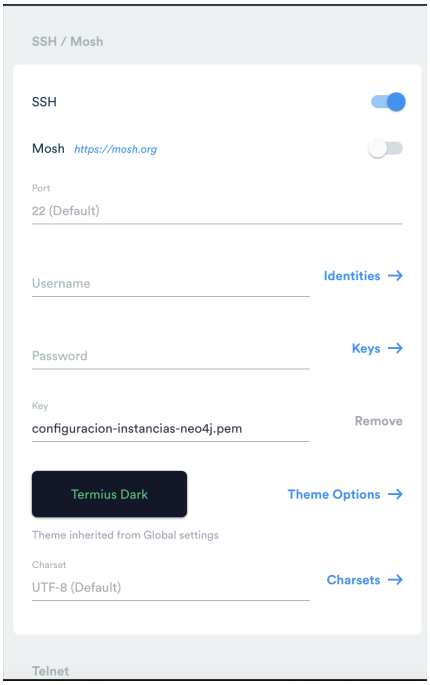


Figura 1.7: Configuración ssh

Figura 1.8: Inicio de sesión al servidor Ubuntu

```
ubuntu@ip-172-31-18-174:~$ sudo su
root@ip-172-31-18-174:/home/ubuntu# sudo apt install default-jre default-jre-headless
```

Figura 1.9: Instalación Java

Figura 1.10: Instalación Neo4j

Figura 1.11: Se agrega el repositorio a la lista de apt fuentes

Tal vez si su instancia es de Linux, antes deberá ejecutar apt-get update para actualizar los packages.

- Ahora se procede con la instalación de de Neo4j. Primero se agrega la clave del repositorio a nuestro llavero.
- Luego se agrega el repositorio a la lista de apt fuentes.
- Finalmente se actualiza la información del repositorio e instale Neo4j.
  sudo apt update
  sudo apt install neo4j
  El servidor debería haberse iniciado automáticamente y también debería reiniciarse en el arranque. Si es necesario, el servidor se puede detener con
  sudo service neo4j stop
  y reiniciar con
  sudo service neo4j start

- Accediendo a Neo4j Ahora se debería poder acceder al servidor de la base de datos a través de http://localhost:7474/browser/ Si se encuentran problemas para iniciar sesión con el nombre de usuario y contraseña predeterminados (neo4j y neo4j), se puede resolver fácilmente eliminando el archivo **/var/lib/neo4j/data/dbms/authy** reiniciando el servidor [46].

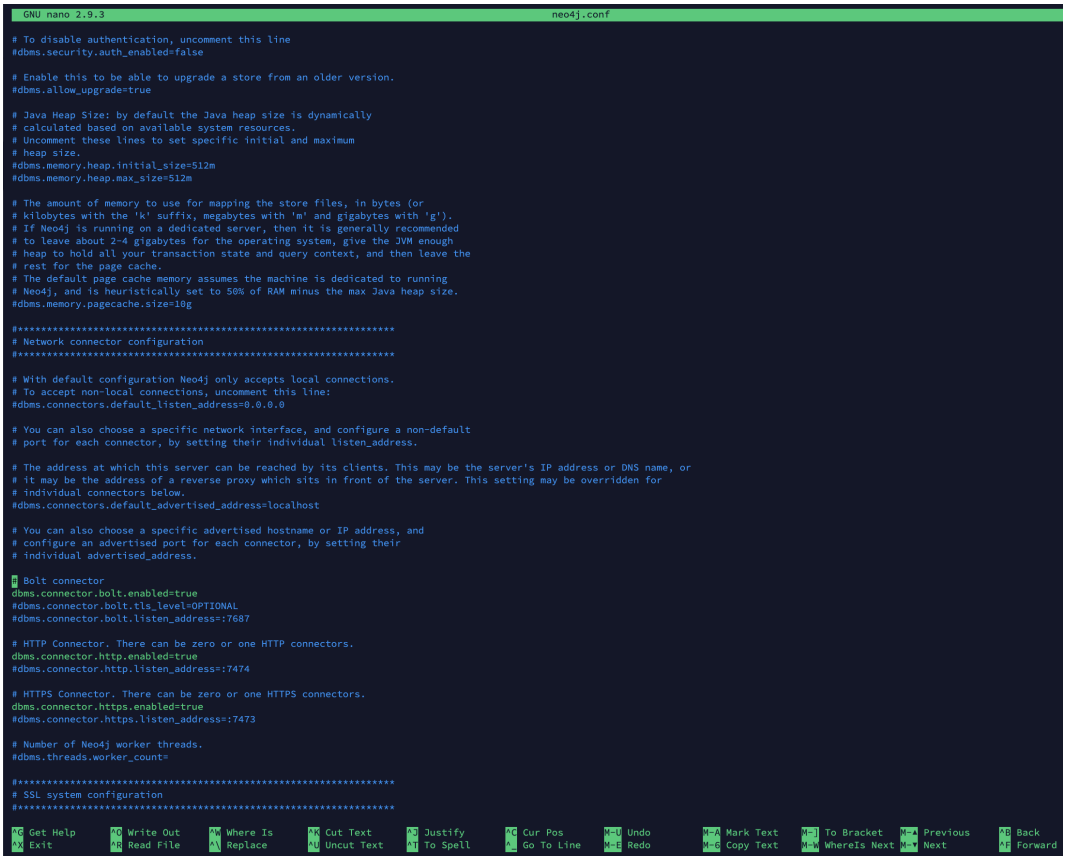


Figura 1.12: Archivo de configuración de Neo4j

1.5. Configuraciones de red necesarias para Neo4j

Para poder acceder desde cualquier IP al Browser de Neo4j, hace falta editar el archivo config de de Neo4j.

```
ubuntu@ip-172-31-18-174:/etc/neo4j$ ls
neo4j.conf
ubuntu@ip-172-31-18-174:/etc/neo4j$
```

Directorio de configuración de Neo4j

Se puede editar el archivo con el editor nano o vi de Linux. Una vez que esté abierto se observan configuraciones tales como SSL system configuration, Logging configuration y **Network connector configuration**. Este último es el que interesa por el momento. Se abre el archivo con el editor y se observa:

Aquí, se debe modificar las propiedades de:

- Default configuration Neo4j

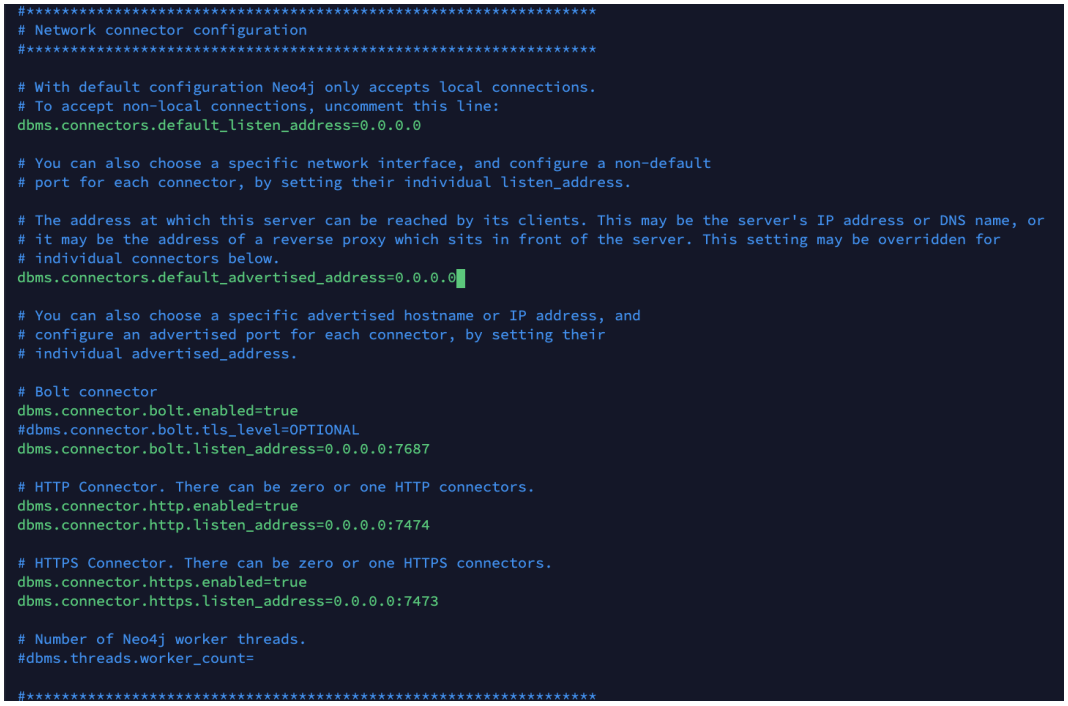


Figura 1.13: Archivo de configuración de Neo4j final

- dbms.connectors.default_listen_address
- dbms.connectors.default_advertised_address

- Bolt connector
  - dbms.connector.bolt.enabled
  - dbms.connector.bolt.tls_level
  - dbms.connector.bolt.listen_address

- HTTP Connector. There can be zero or one HTTP connectors.
  - dbms.connector.http.enabled
  - dbms.connector.http.listen_address

- HTTPS Connector. There can be zero or one HTTPS connectors.
  - dbms.connector.https.enabled
  - dbms.connector.https.listen_address

A continuación un ejemplo de como quedaria el archivo **neo4j.config**

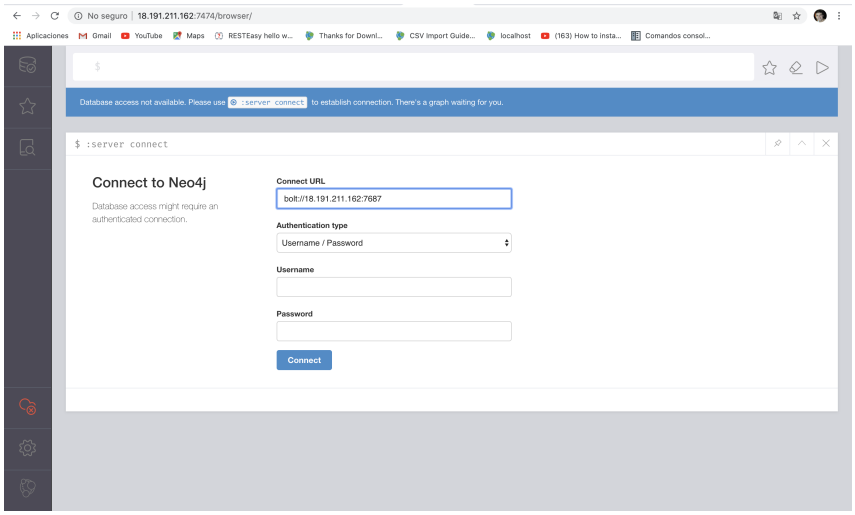


Figura 1.14: Inicio de sesión en la interfaz web de Neo4j

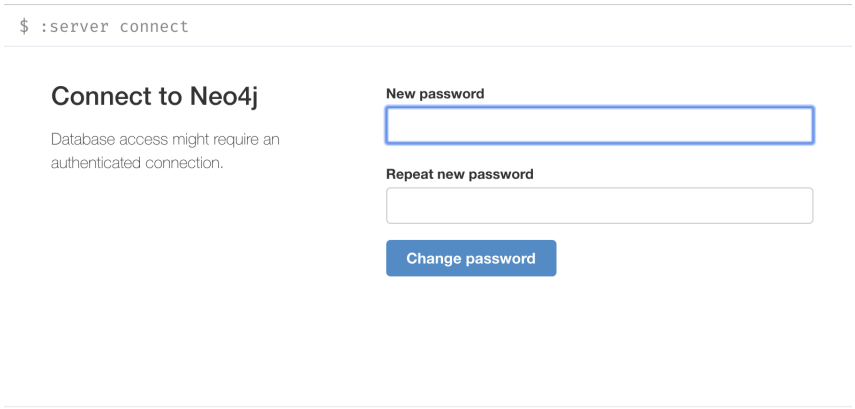


Figura 1.15: Cambio de claves

1.6. Iniciar Neo4j con usuario nuevo

Si la configuración de Neo4j ha sido exitosa se debería ser capaz de ver lo siguiente al tratar de acceder a http://localhost:7474/browser/

Aquí se utiliza **usr:neo4j y psw:neo4j** y luego se observa la oportunidad de cambiar las credenciales.

Una vez ingresadas las nuevas credenciales ya se puede empezar a usar la base de datos Neo4j [48] .

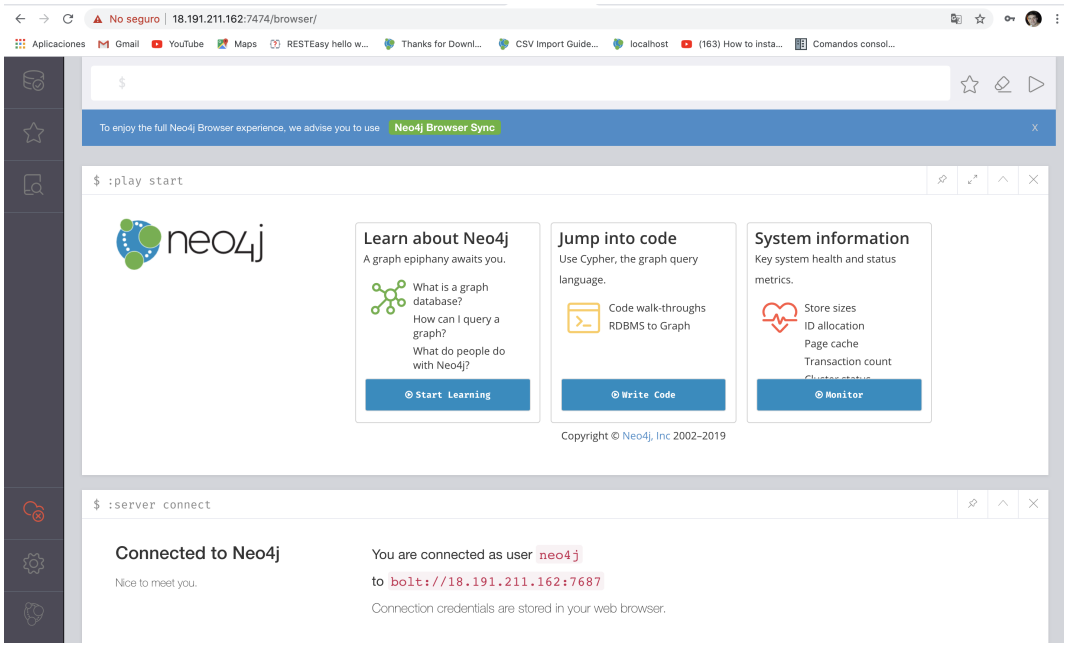


Figura 1.16: Acceso a la interfaz web luego de iniciar sesión en Neo4j

Anexo 2

Guía para carga datos en Neo4j

2.1. Introducción

En la siguiente sección se explica detalladamente el mecanismo propuesto por el equipo de proyecto de grado para importar datos en la base de datos Neo4j. Es necesaria la instalación previa de Neo4j en el servidor y que dicha instancia se esté ejecutando.

2.2. Metodología

La carga de la base de datos se realiza mediante archivos CSV los cuales contienen los datos, una vez que se disponga de dichos archivos se utiliza el lenguaje Cypher propuesto por Neo4j, el cual mediante querys ejecutadas en el motor de base de datos realizan la lectura de los archivos y la posterior carga. Por lo tanto se proponen los siguientes pasos:

- Creación de archivos CSV y ubicación dentro del servidor.
- Ejecución de querys sobre los archivos para la carga.

2.3. Creación de archivos CSV

Características generales del archivo CSV para la carga [47]:

- La codificación de caracteres es UTF-8
- La terminación de la línea final depende del sistema, por ejemplo, es \n en Unix o \r \n en Windows

Figura 2.1: Archivo con cabezales

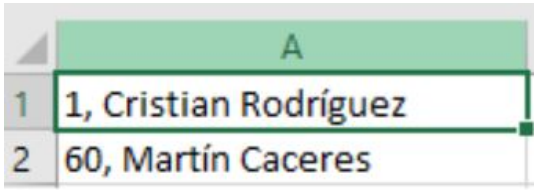


Figura 2.2: Archivo sin cabezales

- El archivo csv debe estar limitado por comas ","
- El comando de lectura del archivo lee todos los valores como strings, independientemente del valor en sí, por lo que se cuentan con las siguientes funciones en caso de que haya que convertir algún valor:
  - toInteger()
  - toFloat()
  - datetime()

Existen dos tipos de archivos, los que representan nodos de un grafo y los que representan aristas. Primero se deben procesar los archivos que representan nodos ya que sería un error intentar crear una arista sobre nodos que no existen aún, en la base de datos. A continuación se explica detalladamente cómo crear los distintos tipos de archivos.

Archivos Nodos:

- El nombre del archivo es indistinto, se aconseja crearlo con el nombre del tipo de nodo que se quiere ingresar, por ejemplo "Usuario.csv"
- Existen dos formas de crear el archivo:
  - Identificando las columnas con un nombre en la primer fila. Es importante que no exista espacio entre los nombres de las columnas separadas por ','.
  - Sin identificación de la columna en la primer fila.

- Se recomienda utilizar la primer forma y que dichas columnas correspondan con el nombre de los atributos del nodo que se pretenden ingresar.
- Tener en cuenta que el modo de creación utilizado afecta posteriormente a las querys que se utilizan para leer los archivos, por lo que se recomienda adoptar una sola forma de las antes mencionadas y mantenerla.

Archivos Aristas:

- El nombre del archivo es indistinto, se recomienda utilizar un nombre representativo de la relación entre los nodos.
- El archivo puede tener cabezales o no, se recomienda que tenga cabezal.
- Como lo que se busca representar es una relación entre nodos, el archivo debe tener mínimo dos columnas indicando cada una a un nodo. Luego si la relación lo amerita se pueden agregar columnas para guardar información en la arista.

2.4. Ejecución de querys para importar archivos

A continuación se procede a enumerar las querys necesarias para la lectura de los archivos CSV y la carga de los datos: Como se dijo anteriormente la query a ejecutar depende del archivo creado (con nombre en las columnas o sin ellas) y del tipo de archivo (nodos o aristas).

2.4.1. Querys para importar archivos de tipo Nodos

Si se va a importar un tipo de nodo nuevo a la base de datos, se deben crear primero las restricciones de unicidad del nodo y luego los índices del nodo para luego procesar el archivo:

Restricciones de unicidad:

```
1 CREATE CONSTRAINT ON (n:<LabelName>) ASSERT n.<propertyKey> IS
    UNIQUE
```

Listing 2.1: Restricción de unicidad

Donde se debe sustituir <LabelName> con el label del nodo y <propertyKey> con el nombre de la propiedad del nodo.

Creación de índice:

```
1 CREATE INDEX ON :<LabelName>(<propertyKey>)
```

Listing 2.2: Creación de índice

Donde se debe sustituir <LabelName> con el label del nodo y <propertyKey> con el nombre de la propiedad del nodo.

Una vez creadas las restricciones y los índices se procede a la lectura del archivo y la importación a la base de datos.

Lectura e importación del archivo con cabezales:
Para la lectura del archivo se utiliza el comando 'LOAD CSV' provisto por Cypher, el cual puede manejar archivos locales y remotos.

Los archivos locales se referencian con un prefijo 'file: ///', la seguridad de Neo4j tiene una configuración predeterminada la cual hace que los archivos locales solo se pueden leer desde el directorio de importación de Neo4j, el cual varía según el sistema operativo.

- Mac OS or Linux – <neo4j-home>/import
  - /Users/¡userid¿/Library/Application Support/Neo4j
  - Desktop/Application/neo4jDatabases/database-<id#>/installation-<neo4jVersion>/import
- Windows Desktop – < neo4j-home>/import

Si se requiere una ubicación de archivo diferente a la predeterminada, se puede actualizar la configuración en el archivo 'neo4j.conf' ubicando la siguiente propiedad en dicho archivo.

"This setting constrains all 'LOAD CSV' import files to be under the 'import' directory. Remove or comment it out to allow files to be loaded from anywhere in the filesystem; this introduces possible security problems. See the 'LOAD CSV' section of the manual for details. dbms.directories.import=import"

Se recomienda colocar los archivos en el directorio de importación de Neo4j, ya que mantiene el entorno seguro. Se puede hacer referencia a los archivos alojados en la web directamente con su URL, como 'https: //host/path/data.csv'. Sin embargo, se deben establecer permisos para que una fuente externa pueda leer el archivo.

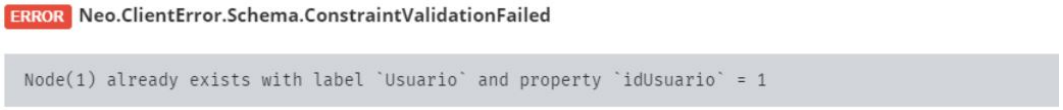


Figura 2.3: Error restricción de unicidad violada

A modo de ejemplo se importará un archivo CSV de nombre 'USUARIO.csv' para generar en el sistema los nodos de label Usuario, el archivo cuenta solamente con una columna denominada 'idUsuario' que corresponde con el nombre de la propiedad del nodo.

```
1 LOAD CSV WITH HEADERS FROM 'file:///USUARIO.csv' AS row
2 CREATE (p:Usuario {idUsuario: toInteger(row.idUsuario)})
3 RETURN count(*);
```

Listing 2.3: Procesamiento del archivo

Una vez ejecutada esta query se habrán ingresado todos los nodos que contenía el archivo.

Si el archivo contiene un nodo ya ingresado en el sistema y el tipo de nodo a ingresar cuenta con alguna restricción de unicidad que no cumple el archivo se obtendrá un mensaje de error como el siguiente.

Es por esto que es necesario ejecutar las querys que crean las restricciones para un tipo de nodo previamente a procesar los archivos.

Lectura de Archivo con gran cantidad de datos:

Para archivos que tienen gran cantidad de datos, es útil usar la etiqueta 'USING PERIODIC COMMIT' antes del comando 'LOAD CSV'. Esta etiqueta indica a Neo4j que la consulta puede generar cantidades excesivas de datos, por lo que debe confirmarse periódicamente.

```
1     USING PERIODIC COMMIT 500
2     LOAD CSV WITH HEADERS FROM 'file:///USUARIO.csv' AS row
3     CREATE (p:Usuario {idUsuario: toInteger(row.idUsuario)})
4     RETURN count(*);
```

Listing 2.4: Procesamiento de archivo con commit periódico

	A
1	idUsuario_A, idUsuario_B
2	900,37408
3	27386,16939
4	21955,18441
5	18922,18924

Figura 2.4: Archivo de tipo Arista

2.4.2. Querys para importar archivos de tipo Aristas

Para procesar un archivo que corresponde a aristas entre nodos, es necesario tener dos columnas que identifican de forma única a los tipos de nodos a los cuales se quiere relacionar y si es necesario demás columnas que agreguen información a la relación.

Se procede a explicar la metodología con tres ejemplos

1. Relación 'SIGUE_A' entre dos nodos del mismo tipo 'Usuario' sin información extra en la relación

```
1  USING PERIODIC COMMIT 2000
2  LOAD CSV WITH HEADERS FROM 'file:///
       SIGUE_A_SIN_REPETIDOS.csv' AS row
3  MATCH (u:Usuario { idUsuario: toInteger(row.
       idUsuario_A)}),(c:Usuario { idUsuario: toInteger(
       row.idUsuario_B)})
4  CREATE (u)-[:SIGUE_A]->(c)
5  RETURN count(*);
```

Listing 2.5: Ejemplo de procesamiento de archivo de relación

Analizando la consulta se puede ver que en la primer línea se indica que haga commit cada 2000 registros. En la segunda se lee el archivo con cabezales y se carga la fila como 'row'. En la tercer línea se realiza un 'MATCH' de la primer y segunda columna del archivo con los nodos de label 'Usuario' pre ingresados en la base de datos, mediante la propiedad del nodo 'idUsuario'. Es importante destacar que para una mayor eficiencia de la consulta se deben hacer los 'MATCH' sobre propiedades del nodo que tengan un índice previamente creado. En la cuarta línea de

la consulta se crea la relación de etiqueta 'SIGUE_A' entre el nodo u y el nodo c obtenidos en la línea anterior, cabe destacar que se le está dando dirección saliente a la relación desde el nodo u al nodo c.
Otro punto a tener en cuenta es que si en el archivo se tiene un dato que es utilizado para identificar a un nodo y este no existe en la base de datos, el registro no se procesa, pero sí se procesa el resto del archivo.

2. Relación 'RESPUESTA_CANT' entre dos nodos del mismo tipo 'Usuario' con información extra en la relación.

```
1    USING PERIODIC COMMIT 2000
2    LOAD CSV WITH HEADERS FROM 'file:///RESPUESTA_CANT.csv
        ' AS row
3    MATCH (u:Usuario { idUsuario: toInteger(row.
        idUsuarioResponde)}),(c:Usuario { idUsuario:
        toInteger(row.idUsuarioCreador)})
4    CREATE (u)-[:RESPUESTA_CANT{id: toInteger(row.id),
        cantidad: toInteger(row.cantidad)}]->(c)
5    RETURN count(*);
```

Listing 2.6: Ejemplo 2 con propiedad en la relación

La única diferencia con la query anterior se presenta en la línea 4 al momento de crear la relación en donde se puede ver que se le agregan dos propiedades a la misma.

3. Relación 'USUARIO_CURSO' entre un nodo de tipo 'Usuario' y un nodo de tipo 'Curso'

```
1        USING PERIODIC COMMIT 2000
2        LOAD CSV WITH HEADERS FROM 'file:///
            usuarios_matriculados.csv' AS row
3        MATCH (u:Usuario { idUsuario: toInteger(row.
            idUsuario)}),(c:Curso { idCurso: toInteger(row.
            idCurso)})
4        CREATE (u)-[:USUARIO_CURSO]->(c)
5        RETURN count(*);
```

Listing 2.7: Ejemplo 3 con dos propiedades en la relación