Instituto Tecnológico
y de Estudios Superiores de Occidente

Reconocimiento de validez oficial de estudios de nivel superior según acuerdo secretarial 15018, publicado en el Diario Oficial de la Federación del 29 de noviembre de 1976.

Departamento de Electrónica, Sistemas e Informática

Maestría en Sistemas Computacionales

Sistema de recomendación hibrido basado en grafos con enfoque a artículos científicos.

TRABAJO RECEPCIONAL que para obtener el **GRADO** de
MAESTRO EN SISTEMAS COMPUTACIONALES

Presenta: **ING. JORGE MANUEL ESCAMILLA ORNELAS**

Asesor **MTRO. VÍCTOR HUGO ORTEGA GUZMÁN**

Tlaquepaque, Jalisco. Junio de 2021.

AGRADECIMIENTOS

Primeramente, quiero agradecer al Consejo Nacional de Ciencia y Tecnología (CONACYT) por brindarme el apoyo económico necesario para la finalización de los estudios de mi posgrado a través del número de beca 642515.

El presente trabajo debe ser reconocido como una labor conjunta realizada con mi asesor de tesis el doctor Víctor Hugo Ortega Guzmán a quien le debo gran parte de mi aprendizaje durante estos dos años y mi gusto a la investigación en el área de bases de datos avanzadas y sistemas de recomendación.

Así mismo, quiero dar el más grande agradecimiento a mi gerente Luis Rocholl dentro de la empresa donde actualmente laboro (ORACLE) por confiar en mi desde el día que me entrevisto, así como apoyarme en la finalización de mis estudios y mi desarrollo profesional. Admiro mucho su manera de ser y dirigirse hacia las personas, es una persona con mucha templanza, conocimiento y vocación para ser líder.

Agradezco al doctor Luis Fernando Gutiérrez Preciado por haberme brindado la oportunidad de recurrir a su capacidad y conocimiento científico en el área de análisis de algoritmos y desarrollo de sistemas web, además por ser tan paciente y apoyarme incluso en horarios no habituales de estudio para atender las dudas que surgían en la elaboración del proyecto, muy pocas personas tienen la vocación de ser profesor y Luis Fernando es uno de ellos.

Al doctor José Francisco Cervantes Álvarez por enseñarme todo lo relacionado al área de *machine learning*, sin duda todos sus consejos y conocimientos para complementar el presente proyecto me inspiraron a seguirme desarrollando en este campo, y tener la capacidad de implementar la aplicación de modelos de aprendizaje automático en mi trabajo.

Agradezco a la Universidad Jesuita de Guadalajara (Instituto Tecnológico y de Estudios Superiores de Occidente) ITESO por darme la oportunidad de estudiar mi posgrado con los mejores profesores de la industria en el tema académico y por el apoyo económico ofrecido a través de la beca por convenio empresarial con mi actual compañía, ORACLE.

A todas las personas que directa o indirectamente hicieron posible el presente trabajo.

DEDICATORIA

A Dios, por permitirme llegar a este momento tan especial en mi vida, por los triunfos y adversidades que me han tocado vivir para llegar a esta etapa, que me han ayudado a formar mi persona y valorar lo que realmente importa en esta vida, la familia.

A mi madre Patricia Ornelas y a mi padre Víctor Escamilla por demostrarme siempre su amor y apoyo incondicional, todos mis logros se los debo a ustedes incluyendo la finalización de mi posgrado, me han dado todo lo que soy como persona, mis valores y mis principios.

A mi hermana Claudia Escamilla por enseñarme constantemente a sonreírle a la vida y disfrutarla, cuidándome siempre como mi hermana mayor y transmitiéndome toda la bondad de su corazón.

RESUMEN

El presente trabajo da evidencia del desarrollo de un sistema de recomendación de artículos científicos, y sus entidades relacionadas como autores y journals a través de una interfaz web utilizando como fuente de información una base de datos de grafos (Neo4j).

Hasta hace no mucho tiempo internet era solamente una repositorio donde solo las personas con capacidad de entender y desarrollar código de programación eran las encargadas de generar contenido; ahora todo el mundo con acceso a la red, tiene la posibilidad de producir información directa o indirectamente.

Desde la década de 1980, el internet ha crecido su demanda hasta el punto de incluir el potencial informativo de universidades y centros de estudio que ahora tienen la capacidad de publicar sus artículos de investigación en los repositorios de la web.

Debido a este aumento de demanda y generación de información, los sistemas de búsqueda se han vuelto ineficientes al momento de satisfacer las necesidades de un usuario que desea encontrar artículos científicos que los ayuden a desarrollarse profesionalmente.

Es así como surge la motivación de desarrollo de la presente solución, un sistema de recomendación. Las principales características de nuestra propuesta incluyen la utilización de una base de datos no relacional como fuente de datos, así como un sistema web con interfaz gráfica amigable que le permite al usuario final hacer una búsqueda de palabras clave y obtener como resultado un conjunto de artículos científicos acorde a sus necesidades.

En la revisión de la literatura se mencionan diversos tipos de sistemas de recomendación que son analizados en el presente documento, nuestro sistema de recomendación es considerado hibrido debido a que analiza el texto interno de cada uno de los artículos científicos y las participaciones colaborativas que tuvieron los autores para la creación de las entidades en la base de datos de grafos.

Existe una amplia variedad de algoritmos enfocados a este tipo de topología, como el caso de métodos de búsqueda, evaluación de importancia de nodos dentro de la estructura y generadores de comunidades que fueron utilizados para el desarrollo del algoritmo interno de recomendación como es el caso de *Louvain* y *PageRank*.

En los siguientes capítulos se presenta la descripción de funcionamiento y utilización del sistema creado, así como un caso de prueba particular donde el usuario final escribe su búsqueda con palabras clave y al finalizar obtuvo sugerencias directas e indirectas de manera interactiva en el sistema web.

Se concluye que el uso de un sistema hibrido de recomendaciones basada en grafos cumple eficazmente con las sugerencias resultantes y nos provee de cualidades como el análisis en tiempo real, persistencia de información y la capacidad de generalizar el tópico esencial de recomendación independientemente de la estructura del grafo.

SUMMARY

The present work provides evidence of developing a recommendation system for scientific articles and their related entities such as authors and journals through a web interface using a graph database (Neo4j) as a source of information.

Some years ago, the internet was only a repository where only people with the ability to understand and develop programming code were in charge of generating content; now, everyone with access to the network has the possibility of producing information directly or indirectly.

Since the 1980s, the internet has grown in demand to include the information potential of universities that now can publish their research articles in web repositories.

Due to this increase in demand and generation of information, search systems have become inefficient trying to comply with a user who wants to find scientific articles that help them develop professionally.

According to that, this is how the motivation for the development of this solution arises, a recommendation system. The main characteristics of our proposal include the use of a non-relational database as a data source and a web system with a friendly graphical interface that allows the end-user to search for keywords and obtain a set of scientific articles as a result, according to their needs.

In the literature review, various types of recommendation systems are mentioned that are analyzed in this document, and our recommender system is considered hybrid because it analyzes the internal text of each of the scientific articles and the collaborative participations that the authors have in the creation of the entities in the graph database.

There is a wide variety of algorithms focused on this type of topologies, such as the case of search methods, evaluation of the importance of nodes within the structure, and community generators that were used for the development of the internal recommendation algorithm, as is the case of Louvain and PageRank.

The following chapters show the description of the operation and use of the system created and a particular test case where the end-user writes his search with keywords and obtained direct and indirect suggestions interactively on the web system.

It is concluded that the use of a hybrid system of recommendations based on graphs effectively complies with the resulting suggestions and provides us with real-time analysis, the persistence of information, and the ability to generalize the essential topic of recommendation regardless of the structure of the graph.

TABLA DE CONTENIDO

LISTA DE FIGURAS

LISTA DE TABLAS

LISTA DE ACRÓNIMOS Y ABREVIATURAS

TFIDF	*Term Frequency – Inverse Document Frequency*
TOG	Trabajo de Obtención de Grado
SR	Sistema de recomendación.
NEO4J	*Native graph database*
ML	*Machine Learning*
WWW	*World Wide Web*
GUI	*Graphical User Interface*
CYPHER	*Graph query language*
LDA	*Latent Dirichlet Allocation*
SQL	*Structured Query Language*
NOSQL	*Not Only SQL*
ACID	*Atomicity, Consistency, Isolation, Durability*
API	*Application Programming Interface*
SCC	*Strongly Connected Components*
LPA	*Label Propagation*
CSV	*Comma Separated Values*
HTML	*Hypertext Markup Language*
CSS	*Cascading Style Sheets*
JSON	*Javascript Object Notation*
DOM	*Document Object Model*

1. INTRODUCCIÓN

1.1. Antecedentes

La aparición de los sistemas de recomendación está ligada al nacimiento de la Web 2.0 y al cambio de paradigma del *marketing* en Internet que ello supuso. En un inicio, el *marketing* en internet se producía de manera unidireccional. Las páginas webs de las empresas eran solamente catálogos donde se presentaban sus productos o servicios, pero los usuarios empezaron a interactuar con estas páginas, con las redes sociales, en páginas de música y películas.

Así de esta manera, comenzamos a generar información muy valiosa para las empresas, que empezaron a contar con retroalimentación directa de los usuarios sobre los contenidos de su interés. Así nacieron los sistemas de recomendación, estos sistemas recogen esas retroalimentaciones, esas interacciones de los usuarios y los utilizan para realizar recomendaciones de ítems que creen que van a ser de su agrado.

Existen diversos tipos de sistemas de recomendación que han surgido a lo largo de los años y que se han ido perfeccionando, los más relevantes son:

- Colaborativos. El sistema realiza las recomendaciones basándose en las valoraciones positivas de usuarios con un perfil de gustos similar al que se quiere recomendar, el usuario activo.
- Filtro basado en contenido. El sistema recomienda ítems al usuario basándose exclusivamente en su experiencia pasada.
- Filtro basado en conocimiento. En estos sistemas se utiliza información proporcionada por el usuario sobre sus preferencias de manera directa, así como el conocimiento que se pueden obtener sobre los ítems para realizar las recomendaciones.
- Hibrido. Comprende la combinación de más de un sistema de los mencionados previamente.

1.2. Justificación

Los motivos que impulsaron el objeto de esta investigación contemplan el hecho de evaluar un sistema hibrido (basado en citas y contenido) utilizando una base de datos no relacional de grafos y conocer el rendimiento que puede tener así como la calidad de las recomendaciones hechas enfocado a un tema de aplicación real para los investigadores, al tratar de buscar artículos científicos que cumplan con sus expectativas y necesidades.

1.3. Problema

Hoy en día, el internet contiene mucha información relacionada a diferentes temas que seguirá creciendo al paso de los años, lo que empieza a crear un conflicto con la búsqueda de la información de valor para el usuario final.

En el caso específico de los artículos de investigación, los sistemas de búsqueda que hay en el mercado no muestran una representación gráfica de las relaciones existentes entre los artículos, los autores, coautores y journals, de manera que las sugerencias para esta información altamente relacionada pueda visualizarse de mejor manera para que el usuario encuentre sus necesidades más rápido.

Si los actuales sistemas de recomendación de artículos científicos empiezan a ser ineficientes al momento de hacer búsquedas relevantes, podríamos padecer de falta de información, pérdida de tiempo e incluso dudar de la validez de las fuentes tecnológicas de la información.

Estos sistemas de recomendación de artículos científicos manejan millones de registros de información no estructurada de manera que se buscara optimizar la obtención de información utilizando una base de datos no relacional basada en grafos.

1.4. Objetivos

1.4.1. Objetivo General:

Desarrollar un sistema de recomendación de artículos científicos hibrido que considere las citas entre los autores, el contenido de los documentos y utilice una base de grafos.

Codificar el algoritmo interno utilizado para generar las recomendaciones con técnicas de agrupamiento de nodos y evaluación de relevancia dentro de la topología del grafo universal, de manera que pueda ser utilizado para otros casos de uso que no sean artículos científicos.

1.4.2. Objetivos Específicos:

Construir o adquirir los *datasets* necesarios para implementar un sistema de recomendación de artículos científicos.

Implementar los *datasets* en una base de datos basada en grafos (NEO4J).

Realizar las consultas necesarias para la generación de sugerencias.

Desarrollar la interfaz gráfica amigable con el usuario.

1.5. Novedad científica, tecnológica o aportación

El valor agregado del presente trabajo se centra en la utilización de un sistema hibrido basado en citas, contenido y grafos que a través de algoritmos de centralidad, generación de comunidades y relevancia de nodos, sea posible obtener recomendaciones de una fuente masiva de artículos científicos, pero que al mismo tiempo pueda aplicarse con otra estructura de base de datos sin la necesidad de realizar muchos cambios, así como en tener la capacidad de visualizar en la interfaz de usuario la sección (sub-grafo) de recomendaciones para un análisis profundo de los resultados entregados por el sistema final.

En la siguiente sección se realiza un análisis comparativo entre la solución propuesta en el presente documento y los diferentes tipos de sistemas de recomendación que han surgido a lo largo de los años, para poder un mejor entendimiento del valor agregado de nuestra solución.

2. ESTADO DEL ARTE O DE LA TÉCNICA

En un inicio los sistemas de recomendación eran conocidos tan solo como filtros colaborativos y los primeros trabajos datan de principios de los años 90. El término fue acuñado en 1992 para un sistema de filtrado de correo electrónico no automatizado donde se identificaron algunas cuestiones importantes para el desarrollo de los sistemas de recomendaciones como:

- Escalabilidad.
- Análisis de la información.
- Consistencia de las recomendaciones [1].

El punto más relevante dentro de los mencionados previamente es la escalabilidad y la cantidad inmensa de información que va surgiendo en las librerías digitales y que ha creado un inconveniente a los usuarios en la búsqueda de recursos en línea. Hoy en día a los usuarios con ciertos intereses se le puede proveer la misma información en respuesta a las consultas de búsqueda que ellos realizan pero las historias individuales del uso de librerías de información no son tomadas en cuenta para sugerir las mejores opciones a los usuarios. Para recuperar documentos relevantes con una búsqueda en específico, existen librerías digitales con los elementos necesarios para colectar información relevante.

Los sistemas de recomendación son herramientas que generan recomendaciones sobre un determinado objeto de estudio, a partir de las preferencias y opiniones dadas por los usuarios. El uso de estos sistemas se está poniendo cada vez más de moda en Internet debido a que son muy útiles para evaluar y filtrar la gran cantidad de información disponible en la Web con objeto de asistir a los usuarios en sus procesos de búsqueda y recuperación de información [2].

Existen diversos tipos de sistemas de recomendación que se adaptan a las diferentes necesidades que la plataforma y usuario final tengan. Desde la fuente de información que buscará almacenar todo el contenido para tener acceso desde los algoritmos como la base de datos; Dentro de esta categoría se encuentran las bases de datos no relacionales (Neo4j es una base de datos basada en grafos muy utilizada) y las relacionales. Así mismo existen tipos de sistemas de recomendación basado en sus tipos de motor de búsqueda, los más relevantes que han surgido a lo largo de los años son:

- Sistemas de popularidad.
- Basado en contenido.
- Colaborativos.
- Basado en grafos.
- Híbrido.

El funcionamiento de los sistemas de recomendación ha evolucionado gracias a las técnicas de *Machine Learning*. Anteriormente los motores de búsqueda, plataformas de contenido y ventas de producto funcionaban con *rankings* o listas de popularidad.

Estos sistemas eran funcionales hasta cierto punto, pero no podían personalizar la experiencia del usuario y mostraban elementos que no se correspondían a nuestros intereses [3].

A continuación, se listan algunos de los ejemplos de sistemas de recomendación que han sido creados a lo largo de los años.

2.1. Xerox PARC Tapestry (1992)

Information Tapestry fue un sistema experimental que se consideraba hibrido debido a que utilizaba un filtrado colaborativo y un filtrado basado en el contenido, así como la valoración y el resaltado automático, para adaptar la entrega y presentación de la información a los intereses personales los usuarios de NetNews [4].

Este sistema permitía almacenar los datos de los usuarios, en concreto, sobre los artículos o noticias que éstos habían leído y posteriormente esta información era utilizada por otros usuarios que aún no habían leído el artículo o noticia, para establecer si un documento era relevante o no. En sus inicios este tipo de sistemas fue adoptado con el nombre de filtro colaborativo dado que permite que los usuarios creen filtros a través de sus ítems de interés, en el caso de *Tapestry* artículos o noticias, y colaborativo pues los usuarios añadían las anotaciones con las opiniones sobre los documentos. Las opiniones añadidas eran utilizadas

para la búsqueda personalizada de otros usuarios. En la tabla 1, se pueden apreciar las principales características diferenciadoras del sistema Xerox contra la propuesta desarrollada en este documento.

2.2. Movie Lens (1996)

MovieLens es de los primeros sistemas de recomendación elaborados por el grupo Lens (GroupLens) y basa sus recomendaciones en la información proporcionada por los usuarios del sitio web, como la clasificación de películas. El sitio utiliza una variedad de algoritmos de recomendación, incluidos algoritmos de filtrado colaborativo [5].

En el año de 1996, GroupLens formó una empresa comercial llamada Net Perceptions, que atendía clientes que incluían las empresas EOnline y Amazon.com. EOnline utilizó los servicios de Net Perceptions para crear el sistema de recomendaciones para Moviefinder.com, mientras que Amazon.com utilizó la tecnología de la compañía para formar su motor de recomendación inicial para las compras de los consumidores. GroupLens utiliza también metodologías de análisis de información masiva dentro de un grupo de estudiantes científicos en Minnesota. En la tabla 1, se pueden apreciar las principales características diferenciadoras del sistema MovieLens contra la propuesta desarrollada en este documento.

2.3. Fab (1997)

FAB es un sistema de recomendación hibrido diseñado para ayudar a los usuarios a examinar la enorme cantidad de información disponible en la *World Wide Web* y creado en el año de 1997 por Balavanovic y Shoham.

Se considera como un sistema de recomendación hibrido debido a que es basado en contenido y co-ocurrencia. El sistema trabaja modelando el perfil del usuario basado en el contenido de los análisis cuando un usuario otorga calificación a una página web y compara estos perfiles para determinar similitudes entre usuarios para una recomendación colaborativa.

De esta manera el usuario recibirá páginas, tanto las que ha calificado relevantes (con respecto a su perfil) como las que han recibido calificaciones altas por usuarios con un perfil similar al suyo (vecinos cercanos).

Existen tres componentes principales dentro de la arquitectura del sistema de recomendación FAB: agentes de colección, que seleccionan páginas de un tema específico; agentes de selección, los cuales encuentran páginas para un usuario específico, y un conector central que enlaza ambos agentes para realizar la recomendación. Cada agente mantiene un perfil del usuario basado en las palabras que contienen las páginas de Web que el usuario ha calificado [6]. En la tabla 1, se pueden apreciar las principales características diferenciadoras del sistema FAB contra la propuesta desarrollada en este documento.

2.4. Scienstein (2015)

Es un sistema de recomendación hibrido que utiliza un filtrado basado en contenido y la técnica colaborativa, que se quedó en fase de desarrollo, y con el enfoque de ser una aplicación de escritorio con la capacidad de recomendar artículos universitarios.

La principal característica de este sistema de recomendación es que no trabaja en base a palabras separadas de búsqueda ingresadas por el usuario final, si no que se hace la carga de un archivo universitario de investigación así como las referencias bibliográficas del mismo y el algoritmo interno se encarga de analizar en contenido, las citas y las calificaciones obtenidas por otros usuarios para hacer una recomendación final; Algunos procesos internos como el de citación se hacían manualmente, lo que hacía tedioso el proceso del sistema [6].

Existen también varios sistemas de recomendaciones que usan correlación usuario a usuario, conocido como aproximación de filtros colaborativos. Estos sistemas dan recomendaciones basadas en correlaciones entre usuarios buscando comportamientos en el sistema. Los enfoques basados en contenido y filtrado colaborativo no son mutuamente excluyentes uno del otro y existen esfuerzos para integrarlos para obtener una recomendación más precisa [7].

Una manera simple de combinar ambos es ejecutar recomendaciones usando dos esfuerzos por separado y combinar los resultados. Algunos sistemas son híbridos, los cuales combinan dos esfuerzos por un nivel más bajo generados por artículos o usuarios. . Un esfuerzo puro de filtros colaborativos usualmente confía en la información de transacciones, mientras un contenido basado en el enfoque usualmente utiliza datos factuales del artículo.

Otros sistemas incorporan información transaccional en la representación de usuarios, así como artículos. En el sistema reportado por Basu, Hirsh y Cohen, los usuarios están representados por un conjunto de elementos y elementos están representados por un conjunto de usuarios, El método de aprendizaje inductivo se usa para predecir la película del usuario preferencias basadas en el par usuario – película.

Estos sistemas híbridos informaron diferentes grados de ganancia de precisión de predicción al utilizar información de múltiples fuentes, que van desde modestos beneficios a importantes mejora. Aunque es necesario agregar que el exceso de información no siempre conduce a mejorar resultados [8]. En la tabla 1, se pueden apreciar las principales características diferenciadoras del sistema *Scienstein* contra la propuesta desarrollada en este documento.

2.5. *The Movie critic* (1998)

The Movie Critic es un sistema de recomendación de películas, de diversos géneros, basado en filtrado colaborativo. El sistema tiene un proceso de evaluación, el usuario califica películas de acuerdo a un grado

de aceptación (de varios niveles). De este proceso el sistema crea las relaciones entre personas para realizar las recomendaciones. La evaluación puede ser modificada en cualquier momento.

El sistema provee varios tipos de recomendaciones, cuáles son las películas que más le gustarán, las que no le gustarán y películas para dos personas. El usuario puede consultar, por género, cuáles son sus recomendaciones. Las recomendaciones muestran también cuál fue la evaluación del grupo de "vecinos cercanos" y el posible grado de aceptación de usuario. El grado de aceptación del usuario puede variar dependiendo si la película tuvo o no controversias en el ambiente cinematográfico [9]. En la tabla 1, se pueden apreciar las principales características diferenciadoras del sistema *Movie Critic* contra la propuesta desarrollada en este documento.

2.6. *PHOAKS* (1997)

People Helping One Another Know Stuff es un sistema de recomendación que reconoce y redistribuye recomendaciones de recursos de Web buscando en mensajes electrónicos. Este sistema está basado en filtrado colaborativo, lo que hace posible que un grupo de personas hagan y reciban recomendaciones entre sí.

Se distingue de otros sistemas por dos características principales: el rol de especialización y reutilización. *PHOAKS* recomienda páginas de Web, busca en los mensajes las opiniones que los participantes dejen acerca de estas páginas, y las selecciona si pasan ciertos requerimientos.

La arquitectura de *PHOAKS* consiste en tres procesos principales: buscar mensajes con un patrón específico, clasificación de las instancias de los patrones y disposición de la información encontrada [10].

En la tabla 1, se pueden apreciar las principales características diferenciadoras del sistema *Phoaks* contra la propuesta desarrollada en este documento.

2.7. *Referral* Web (1997)

Es un sistema interactivo para la construcción, visualización y búsqueda de redes sociales en el *World Wide Web*. Este sistema modela una red social mediante un grafo en cuyos nodos son representados los individuos y las conexiones entre nodos indican una relación directa descubierta entre ellos. Para identificar la relación directa entre individuos, se utilizan métodos tales como referencias encontradas en las páginas personales, referencias de co-autores en publicaciones técnicas, citas en las publicaciones y organigramas.

Referral Web no intenta crear nuevas comunidades sino más bien ayudar a los usuarios a hacer un uso más eficiente de sus redes existentes de colegas profesionales. Perteneciendo a una comunidad, el usuario puede descubrir contactos a gente o a información que de otra manera le estaría oculta [11]. En la tabla 1,

se pueden apreciar las principales características diferenciadoras del sistema *Referral web* contra la propuesta desarrollada en este documento.

2.8. *Citeseer (1997)*

Es un sistema que utiliza los registros de páginas favoritas (*bookmarks*) de un usuario y la organización de éstos registros para la recomendación de páginas de Web relevantes [Rucker y Polanco 1997]., puesto que los registros representan interés en el contenido y su organización indica relevancia entre los elementos. C*iteseer* utiliza en método de filtrado colaborativo y recomienda al usuario las páginas de electrónicas de sus "vecinos cercanos" [12]. En la tabla 1, se pueden apreciar las principales características diferenciadoras del sistema *Citeseer* contra la propuesta desarrollada en este documento.

2.9. *Footprints (1998)*

En este trabajo, *Wexelblat* [1998]. presenta un marco de investigación y algunas herramientas para mostrar cómo se puede utilizar la historia de la interacción como parte de la interfaz del usuario para la navegación social. Utilizando las pistas que han dejado usuarios anteriores, se les ayuda a los usuarios actuales a encontrar y a entender información que requieren. *Footprints* pretende que la historia de la interacción en el mundo digital sea tan fácil de seguir como en el mundo físico, mediante mapas, caminos y notas.

La historia de la interacción implica la presencia de un usuario y de un objeto, y se define como el historial acumulado de las acciones (enfatizando la secuencia), la relación que el usuario ha detectado entre los elementos y la organización resultante. Las modificaciones a estos objetos afectan nuestra percepción sobre los mismos. El que el usuario pueda saber, qué se ha hecho con la información, quién ha interactuado con ella, por qué lo ha hecho y que ha sido revisada, le ayuda a identificar, autentificar y entender la información.

Footprints ayuda al usuario en su navegación por el *World Wide Web* sin mostrar al usuario tanta historia de la interacción que pueda distraerlo de su tarea principal [13]. En la tabla 1, se pueden apreciar las principales características diferenciadoras del sistema *Footprints* contra la propuesta desarrollada en este documento.

El análisis de todos los sistemas de recomendación que han surgido a lo largo de los años, sirven como punto de comparación con la solución propuesta por la presente propuesta para hacer una recomendación de artículos científicos. En la tabla 1, se pueden apreciar las principales características diferenciadoras del sistema propuesto contra otras nueve soluciones.

El análisis de todos los sistemas de recomendación que han surgido a lo largo de los años, sirven como punto de comparación con la solución propuesta por la presente propuesta para hacer una recomendación de artículos científicos. En la tabla 10, se pueden apreciar las principales características diferenciadoras del sistema propuesto contra otras nueve soluciones, así mismo se puede concluir que no se encontró un

sistema de recomendación de artículos científicos que cumpla con las mismas características de nuestra solución propuesta, es decir, utilizando una base de datos de grafos con algoritmos de agrupamiento y evaluación de relevancia dentro de la topología del grafo. De los sistemas analizados en la tabla 1.

Table 1 Resumen de comparación con otros sistemas de recomendación

Sistema de recomendación	Tipo	Características
Xerox	Colaborativo y contenido	Recomendador de noticias. Importancia a las opiniones. Sistema experimental.
Movie Lens	Colaborativo	Recomendador de películas. Importancia a las búsquedas de los usuarios en sitios web. Utilizado ampliamente en prácticas de ciencia de datos.
FAB	Popularidad, Contenido y co-ocurrencia	Recomendador de páginas web. Importancia en rankings otorgados por los mismos usuarios Compara la relación entre los perfiles de usuario.
Scienstein	Contenido y colaborativo	Recomendador de artículos universitarios. La entrada del usuario es un documento completo. Se quedó en fase de desarrollo. Aplicación de escritorio.
Movie critic	Colaborativo	Recomendador de películas. Relaciones basadas en rankings de los usuarios. Evaluación con capacidad a modificarse.
PHOAKS	Colaborativo	Recomendador de recursos web basado en el análisis de mensajes electrónicos, y opiniones de las páginas web que hacen los usuarios.
Referral web	Grafos, Co-ocurrencia	Recomendador de usuarios para redes sociales. Analizador de citas en las publicaciones de las redes.
Citeseer	Colaborativo	Recomendador de *bookmarks.* Analizador de usuarios similares para la recomendación.
Footprints	Contenido	Recomendador de términos de búsqueda en la web. Analiza el tipo de información buscada por los usuarios previos.

Sistema de recomendación hibrido de artículos científicos	Contenido, Co-ocurrencia, Grafos	Recomendador de artículos científicos, journals y autores (búsqueda del usuario). Uso de técnicas de minería de texto. Análisis de palabras clave Representación gráfica del sub-grafo resultante. Interfaz GUI web.

En el siguiente capítulo analizaremos el significado de cada uno de los términos necesarios para entender el problema que resuelve la solución presentada en el presente documento, un sistema de recomendación de artículos científicos basado en co-ocurrencias, contenido y grafos.

3. MARCO TEÓRICO/CONCEPTUAL

Dado que Sarahí Partida Ochoa y Jorge Escamilla Ornelas trabajamos juntos en el desarrollo general de la solución, pero cada uno desarrolló un escenario especial de implementación, compartimos la misma información de esta sección en nuestros documentos de obtención de grado.

El internet es la fuente más completa que existe y ha crecido aceleradamente en los últimos años, debido a esto existe una gran cantidad de información para cualquier tipo de tópico existente y la búsqueda de diferentes temas que hacemos como usuarios finales, nos devuelve más cada vez más artículos que pueden ser relevantes y muchos no relevantes, de ahí la importancia comercial de implementar sistemas de recomendación eficientes.

El proyecto consiste en la elaboración de un sistema de recomendación basado en grafos el cual contenga la información necesaria para predecir y ofrecer recomendaciones eficientes a los usuarios finales, logrando recomendaciones para la lectura de artículos científicos, orientados a un tema en particular.

Los sistemas de recomendación tienen aplicación para diversas áreas como economía, educación, arte, ocio, artículos científicos y métodos de aprendizaje adaptativos, donde se estudiará más a fondo los últimos dos temas para el desarrollo de este documento.

Un sistema de recomendación debe ser parte de la colaboración entre investigadores enriqueciendo la información obtenida. La investigación académica trata de la relevancia entre artículos y búsqueda de consultas. Ambas son similares, pero son utilizadas para la recomendación de sistemas. Su arquitectura ayuda a entender cómo está construida la recomendación y el conjunto de datos para tener una muestra de cómo crear la recomendación [9].

3.1. Sistema de recomendación

"Los sistemas de recomendación son herramientas de software o técnicas que ayudan a proveer sugerencias de diferentes ítems a diferentes usuarios" [14].. Estos sistemas de recomendación han sido bien aceptados por la sociedad, debido a que desde antes de sus primeras apariciones como *software*, se ha utilizado a manera de consejos entre amigos, personas relacionadas a cada uno y esto se ve reflejado en los productos y tecnologías que consumimos, las cuales se han convertido parte de nuestra vida cotidiana.

Existen diversos algoritmos y métodos de recomendación que toman en cuenta diversas características, al momento de buscar artículos relevantes para los usuarios, por ejemplo unos consideran más importante la relación que pueda existir entre usuarios categorizados como semejantes y hay sistemas de recomendación que te sugieren artículos dependiendo del contenido relacionado a las preferencias de cada usuario. En la figura1 se puede observar los diferentes tipos de métodos o algoritmos de recomendación, que pueden ser aplicados a artículos científicos u otros tópicos.

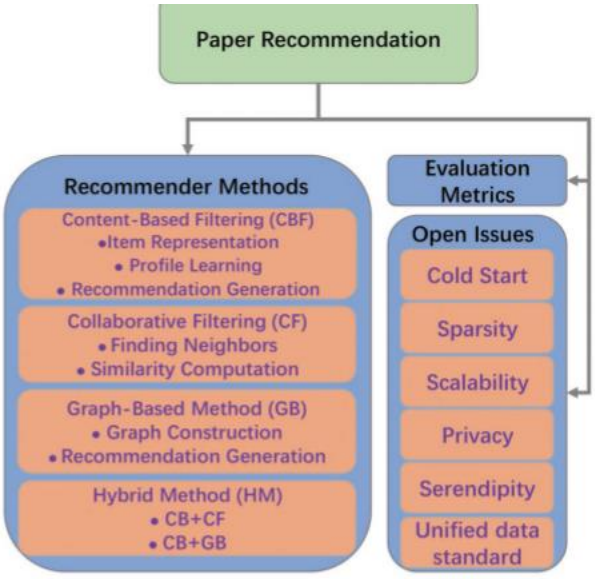


Figura 1 Tipos de sistemas de recomendación [3].

3.2. Métodos de recomendación

Para generar una recomendación, el sistema tiene que aprender las preferencias del usuario o analizar las características de cada ítem relacionado al tópico [15]., dependiendo de la acción que se haga, se pueden dividir estas técnicas o métodos, los más utilizados pueden categorizarse en cuatro tipos, basado en contenido, filtro colaborativo, hibrido y basado en grafos.

3.2.1. Filtro basado en contenido

Esta técnica se basa en el ítem o producto para hacer la predicción y posteriormente la recomendación en lugar de utilizar al usuario, así mismo, considera el historial de preferencias del mismo, construyendo un perfil y buscando similitudes con las características de diferentes ítems [16].

Entre las ventajas de esta técnica, a comparación con la búsqueda basada en palabras clave, se consideran los intereses de cada individuo, y no de los demás. Así mismo, si sus intereses llegan a cambiar en un futuro, también lo harán sus recomendaciones, donde se ofrece recomendaciones fortuitas para que el usuario pueda obtener información que no estén actualmente en sus intereses, pero puedan estarlo.

Su proceso se divide principalmente en 3 etapas:

- *Item representation* Tiene el objetivo de estructurar la información no especifica, representar las características de los ítems como vectores para posteriormente computar la similitud con el *profile* del usuario, se llegan a utilizar métodos de representación como el TF-IDF en el caso de documentos.
- *Profile learning* El objetivo de este paso es construir el perfil del usuario utilizando algoritmos de extracción de tópicos como el LDA o Latent *Dirichlet Allocation*
- *Recommendation generation* Obtiene la lista de posibles recomendaciones relevantes para el usuario, evaluando la similitud de los vectores resultantes de los procesos previos [16]. como se puede observar en la figura 2.

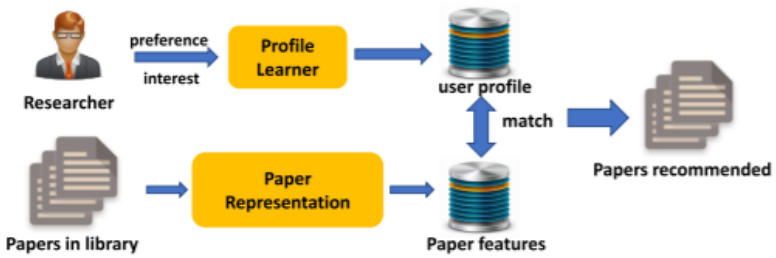


Figura 2 Proceso de un sistema de recomendación basado en contenido [2].

3.2.1.1. Método de representación TF-IDF

El termino TF-IDF (*Term Frequency – Inverse document frequency*) es una medida estadística utilizada para evaluar la importancia de una palabra en un documento y en una colección de documentos. TF-IDF es el producto de dos factores, la frecuencia de término y la frecuencia inversa de documento.

La frecuencia de término es el número de veces que un término t ocurre en un documento D, y la frecuencia inversa de documento es una medida que expresa si el término es común o no en la colección de documentos [17].

3.2.1.2. Algoritmo LDA

Este algoritmo es un ejemplo de modelado de tópicos o temas, y es utilizado para clasificar texto de un documento en un tópico particular. Este construye un tópico por cada modelado de cada documento y las palabras generadas en cada uno de estos son modeladas como distribuciones de *Dirichlet*. Básicamente se utiliza este algoritmo para la separación e interpretación de temas de un documento [18].

3.2.2. Filtro colaborativo

Este tipo de técnica se centra en las acciones, *ratings* o calificaciones hechas en los ítems de parte de otros usuarios que tengan un perfil similar (usuarios vecinos). Al igual que la técnica basada en contenido, este método también necesita conocer los intereses del usuario, en otras palabras, El sistema de recomendación de filtro colaborativo es el proceso de recomendar ítems usando la opinión de otros usuarios vecinos [19].

Se puede dividir en dos categorías al momento de realizar sus predicciones: basado en usuarios y basado en ítems.

3.2.2.1. Basado en usuario

Los sistemas de recomendación utilizan el perfil de otros usuarios similares para hacer la predicción. En este, los usuarios se dividen en categorías o grupos, los usuarios en el mismo grupo comparten intereses similares.

3.2.2.2. Basado en ítems

Principalmente se enfoca en la relación que existe entre los ítems, en lugar de entre los usuarios. Si los usuarios otorgan calificaciones positivas sobre algunos ítems, el sistema podrá

recolectar los ítems candidatos a ser elegidos basándose en el *raiting* histórico de calificaciones del usuario.

Entre las desventajas de esta metodología se encuentran el arranque en frio, es decir, para los ítems nuevos que no tienen una calificación asignada, no pueden ser recomendados.

En general, como se muestra en la figura 3 [16]. la técnica de recomendación de filtro colaborativo se basa en la construcción de una matriz de *raitings*, donde se debe considerar que no se tienen matrices densas, es decir, no todos los usuarios dan su calificación a todos los ítems, por lo tanto es más óptimo el basado en ítems al manejar matrices más pequeñas.

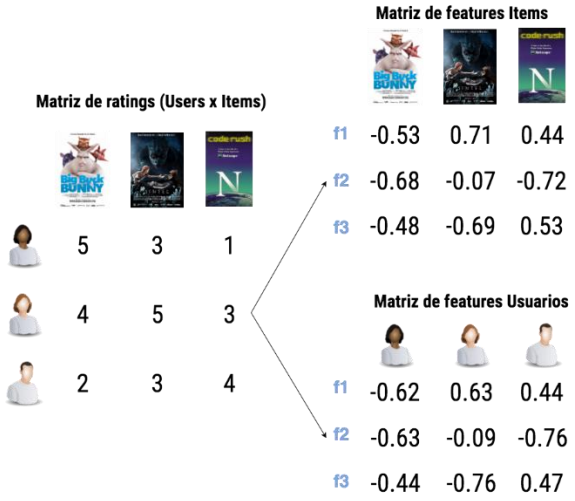


Figura 3 Matrices de índices para un sistema de recomendación de filtro colaborativo [9].

3.2.3. Basado en grafos

Como lo dice su nombre, esta técnica de recomendación se centra en la construcción de un grafo para hacer las predicciones y recomendaciones donde cada ítem / usuario puede verse como entidad o nodo y las posibles relaciones que existen entre ellos se pueden representar como aristas [20]. La figura 4 representa a un grafo utilizado como fuente de información para el sistema de recomendación de artículos científicos de nuestra presente propuesta, como se puede observar los nodos en color azul son los autores, los nodos en color verde son los artículos y el nodo rosa es el journal donde se han publicado algunos de los artículos. Existe una alta relación entre las entidades lo que hace más factible el uso de una base de datos no relacional como lo es Neo4j.

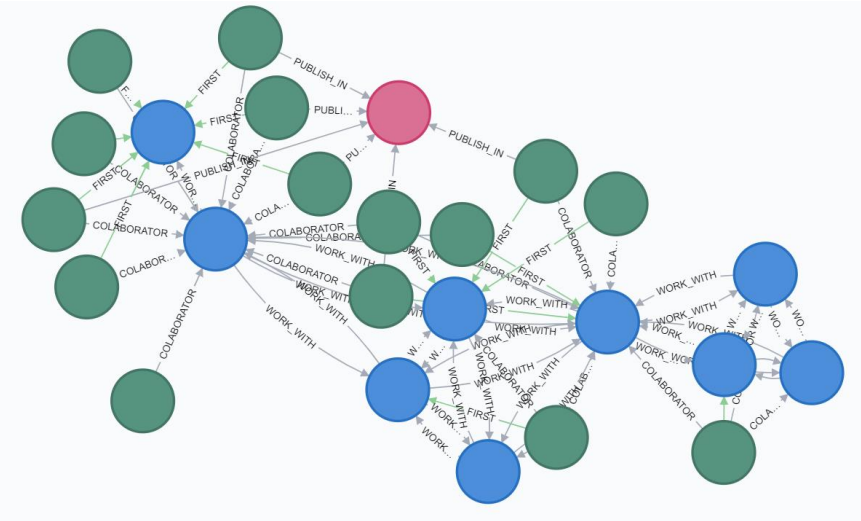


Figura 4 Grafo usado para un sistema de recomendación básico.

Para generar la recomendación se suele utilizar algoritmos de recorrido / búsqueda como el de caminata aleatoria con reinicio, el cual evalúa la probabilidad de estando en un vértice inicial i, llegar a un vértice final j pasando por nodos adyacentes y se debe considerar el caso de que se pueda llegar al nodo inicial i, por eso el nombre de reinicio [21].

3.2.4. Hibrido

Los diferentes métodos antes vistos tienen ventajas y desventajas; con el método hibrido se busca fusionar el comportamiento de 2 o más métodos para complementar sus debilidades, como las mostradas en la figura 5. La principal ventaja del método hibrido es que se puede hacer la combinación de diferentes técnicas de recomendación y obtener información de diferentes fuentes [16].

Como se puede observar en la imagen 5, un tipo de sistema hibrido utilizado es el paralelo de los sistemas basado en contenido y basado en filtro colaborativo, de esta manera se toma en cuenta los intereses del usuario a través de un perfil creado y el rating hecho por el usuario a diferentes ítems. Al final, se terminan recomendando la coincidencia de las 2 listas diferentes de ítems posibles relevantes para el usuario [22].

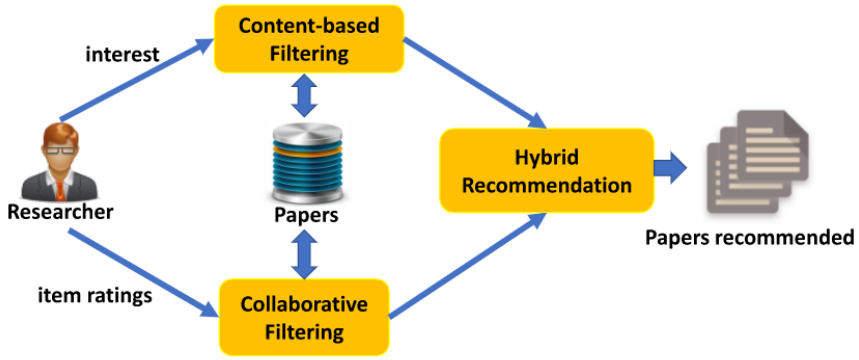


Figura 5 Método hibrido (CF + CFB) paralelo [15].

Existen diversas técnicas de recomendación, las más utilizadas con la basada en contenido y la de filtro colaborativo, pero En la tabla 2 [16] se pueden observar las principales ventajas y desventajas de la utilización de los mismos.

Tabla 2 Ventajas y desventajas de las técnicas de recomendación

Técnica	Ventaja	Desventaja
Filtro basado en contenido	Cada ítem puede ser descubierto para calcular la similitud. Resultados relacionados a preferencias del usuario	Solo considera la frecuencia de un ítem, no la calidad. Problema con nuevos usuarios, no hay que recomendar.
Filtro colaborativo	Las recomendaciones pueden ser fortuitas. Calidad de resultados garantizada	Problema de arranque frio para nuevos ítems o nuevos usuarios. Problema de escasez
Basado en grafos	Considera diversas fuentes para hacer la recomendación.	No considera los intereses del usuario.

3.3. Métodos de evaluación

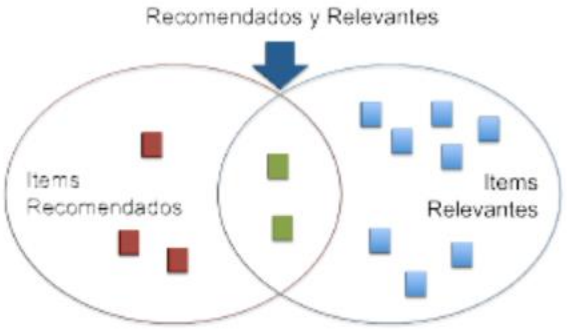
Los sistemas de recomendación al final de cuentas son modelos de predicción, que tienen algoritmos que buscan minimizar el error de dicha función. Por lo tanto es importante medir y determinar si la técnica de recomendación aplicada para dicho sistema de recomendación es efectiva. Los métodos de evaluación más utilizados son Precisión, Recuperación y Medida F (F1) [16]., [23].

3.3.1. Precisión

Es usado para medir la exactitud de recomendar ítems relevantes para los usuarios en un sistema de recomendación, es decir, dentro de los ítems recomendados al usuario, cuales fueron relevantes para él. Su fórmula viene representada en la figura 6 [16].

3.3.2. Recuperación

Cuantifica la fracción de ítems relevantes dentro de un conjunto de ítems recomendados al usuario final. Es decir, de los ítems que resultaron relevantes para el usuario, cuantos fueron seleccionados. Su fórmula viene representada en la figura 6 [16].



$$Precision = \frac{|Recomendados \cap Relevantes|}{|Recomendados|}, y$$

$$Recall = \frac{|Recomendados \cap Relevantes|}{|Relevantes|}$$

Figura 6 Precisión vs Recuperación [8].

3.3.3. Medida F

Representa el promedio armónico entre Precisión y recuperación. Ya que se puede dar el caso en el que se pueden contradecir los resultados de estos dos factores, caso dado cuando el número de ítems recomendados se hace mayor, entonces la recuperación tiende a crecer y la precisión tiende a disminuir.

3.4. Medidores de eficiencia

Todos los sistemas de recomendación tienen problemas que se han tratado de combatir a lo largo del tiempo y como han ido evolucionando. Claro ejemplo del modelo hibrido que busca combinar 2 o más técnicas de recomendación para anular los siguientes problemas muy comunes [22].

3.4.1. Arranque en frío

Se refiere al estado inicial de las entidades de un sistema de recomendación, es decir cuando un usuario o ítem son recién creados, es complicado modelar un perfil (método basado en contenido) o relacionarlo con usuarios supuestamente similares (método de filtro colaborativo), ya que no se cuenta con información necesaria.

3.4.2. Escasez

En la mayoría de los sistemas de recomendación, el número de usuarios es mayor al número de ítems, de esta manera se hacen predicciones efectivas, pero puede darse el caso en el que el número de usuarios es menor al número de ítems o incluso que el número de ítems sea muy poco calificado por los usuarios, es decir, que no tengan muchas relaciones. Este problema provoca que se tenga escasez de información para poder realizar una recomendación de calidad, en especial para la técnica de filtro colaborativo que utiliza matrices de calificación de elementos [16].

3.4.3. Escalabilidad

Esta definición se refiere a la capacidad de un sistema de recomendación de trabajar efectivamente en diferentes ambientes donde existe una gran cantidad de usuarios y de ítems.

3.4.4. Hallazgo fortuito

El comportamiento común de un sistema de recomendación, es predecir ítems candidatos a los intereses de los usuarios, pero de vez en cuando es útil tener recomendaciones fortuitas para incrementar el área de conocimiento. El número de estas recomendaciones fortuitas no puede ser elevado, ya que el sistema de recomendación pierde credibilidad.

3.5. Bases de datos NOSQL

La llegada de las bases de datos NOSQL (*NOT Only SQL*) aparecieron a partir de la llegada de la WEB2.0 cuando empresas como Facebook, Twitter, y Youtube tenían que soportar las visitas de millones de usuarios y dar respuestas a millones de consultas, provocando que las bases de datos relacionales empezaran a fallar [24].

Las bases de datos NoSQL utilizan una variedad de modelos para acceder y administrar grandes cantidades de información [25]. Su principal objetivo es la alta escalabilidad de los sistemas, no llegan a suplir una base de datos relacional pero si mejoran su comportamiento cuando se maneja información excesiva. Existen diversos tipos como las basadas en documentos, llave - valor y columnas, sin embargo, para el desarrollo de este proyecto, nos enfocaremos en las basadas en grafos.

Grafos: el propósito de una base de datos de grafos es facilitar la creación y la ejecución de aplicaciones que funcionan con conjuntos de datos altamente conectados. Los casos de uso típicos para una base de datos de gráficos incluyen redes sociales, sistemas de recomendaciones, detección de fraude y gráficos de conocimiento [25].

3.5.1. NEO4J

En una base de datos basada en grafos. La información se representa como nodos de un grafo, y sus relaciones con las aristas del mismo modelo. Se suelen utilizar grafos multicapa o grafos heterogéneos donde cada capa contiene un tipo de entidad y existen relaciones entre ellas [20].

Neo4j es un *software open source* de base de datos orientado a grafos implementado en Java y desarrollado por Neo *Technology*. Entre las principales características de esta base de datos se encuentran:

- Intuitivo. Usa un modelo de datos grafico para la representación del modelo.
- Confiable. Soporta transacciones ACID. Buscando en todo momento la atomicidad, consistencia y persistencia del modelo.
- Altamente escalable. Soporte para miles de nodos, relaciones y propiedades.
- Expresivo. Potente lenguaje de consulta (CYPHER).
- Simple. Accesible a través de una API orientada a objetos (JAVA) [26].

3.5.1.1. CYPHER

Es un lenguaje declarativo basado en el lenguaje de bases de datos relacionales SQL, que permite manipular la información en NEO4J. Cypher es el lenguaje propio de Neo4j, su sintaxis utiliza un estilo ascii – art, lo que lo hace muy intuitivo. Los nodos se representan con círculos y las relaciones con flechas, por ejemplo, su representación consiste en poner nodos entre paréntesis y relaciones como flechas encerradas entre corchetes. Las propiedades de las entidades se indican con una estructura similar a los diccionarios.

```
(nodo) – [:RELACIÓN]. -> (nodo)
(nodo {nombre:'Oscar', apellido:'García'})
```

3.5.1.2 NEOVYS

Neovys es una librería de javascript para la visualización de nodos en una aplicación de UI. Existen diferentes motivaciones y herramientas para crear visualizaciones de gráficos. Esto incluye herramientas para explorar el gráfico, el tipo de visualizaciones interactivas que puede ver en el navegador Neo4j. o visualizaciones para mostrar los resultados de algún análisis. Estos pueden ser interactivos (algo que se incrustará en una aplicación web o incluso en una aplicación independiente) o estáticos, destinados a transmitir un significado específico que podría usarse en forma impresa o en una publicación de blog [27].

En la Figura 7 podemos ver un ejemplo de subgrafo perteneciente al grafo principal que hace de fuente de información de nuestro sistema de recomendación. Se puede observar los diferentes tipos de relación existentes entre las entidades de artículos científicos, autores y journals. Todos los nodos son del mismo tamaño pero cada entidad y cada relación están coloreados de una única manera.

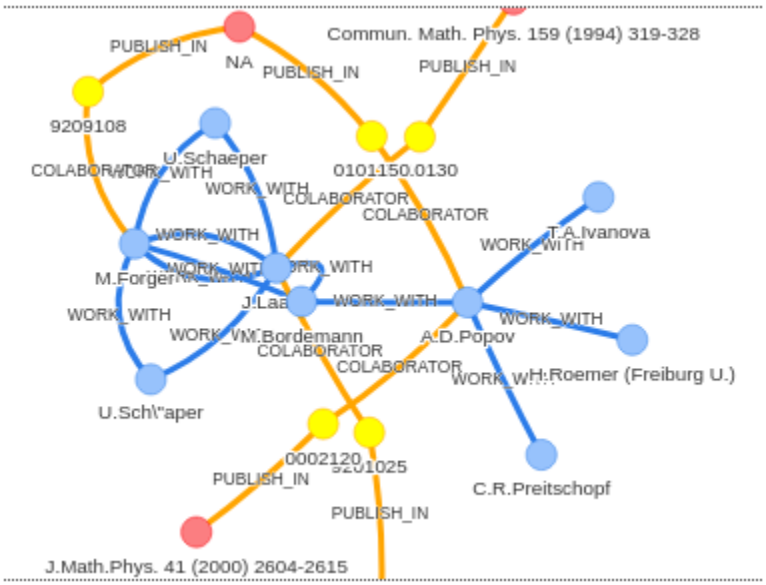


Figura 7 Visualización de subgrafo del dataset de artículos científicos usando Neovis.

Específicamente nos gustaría analizar con Neovis.js:

- El tamaño del nodo debe ser proporcional a la puntuación de *pagerank* del personaje. Esto nos permitirá identificar rápidamente nodos importantes en la red.
- El color del nodo lo determina la propiedad comunitaria. Esto nos permitirá visualizar clústeres.
- El grosor de la relación debe ser proporcional a la propiedad de peso en la relación INTERACTS.
- Neovis.js, al combinar el controlador JavaScript para Neo4j y la biblioteca de visualización vis.js, nos permitirá construir esta visualización.

3.6. Algoritmos de centralidad

Los algoritmos de centralidad se utilizan para comprender los roles de nodos particulares en un grafo y su impacto en cierta red. Son útiles porque identifican los nodos más importantes y nos ayudan a comprender la dinámica del grupo, como la credibilidad, la accesibilidad, la velocidad a la que se propagan las cosas y las relaciones entre los grupos. Aunque muchos de estos algoritmos se inventaron para el análisis de redes sociales, desde entonces han encontrado usos en una variedad de industrias y campos [28].

3.6.1.1. Page Rank

PageRank es el más conocido de los algoritmos de centralidad. Mide la influencia transitiva (o direccional) de los nodos. *PageRank* considera la influencia de los nodos vecinos de un nodo. Por ejemplo, tener algunos amigos muy poderosos puede hacerte más influyente que tener muchos amigos menos relevantes. *PageRank* se calcula distribuyendo iterativamente el rango de un nodo entre sus vecinos o atravesando aleatoriamente el grafo y contando la frecuencia con la que se golpea cada nodo durante estos recorridos [28].

PageRank lleva el nombre del cofundador de Google, Larry Page, quien lo creó para clasificar los sitios web en los resultados de búsqueda de Google. La suposición básica es que una página con más enlaces entrantes y más influyentes es más probable una fuente creíble. *PageRank* mide el número y la calidad de las relaciones entrantes a un nodo para determinar una estimación de la importancia de ese nodo. Se presume que los nodos con más influencia sobre una red tienen más relaciones entrantes de otros nodos influyentes [28].

El objetivo de este algoritmo es el de asignar un puntaje de importancia a cada uno de los nodos del grafo dependiendo de las conexiones entrantes y salientes en su ubicación dentro de la topologia del grafo resultante. Para el caso especifico de la solucion propuesta, se puede observar que con 30 iteraciones y utilizando solamente la relacion "*colaborator*" entre los articulos cientificos, los primeros 5 nodos mejor evaluados serian los mostrados en la figura 8.

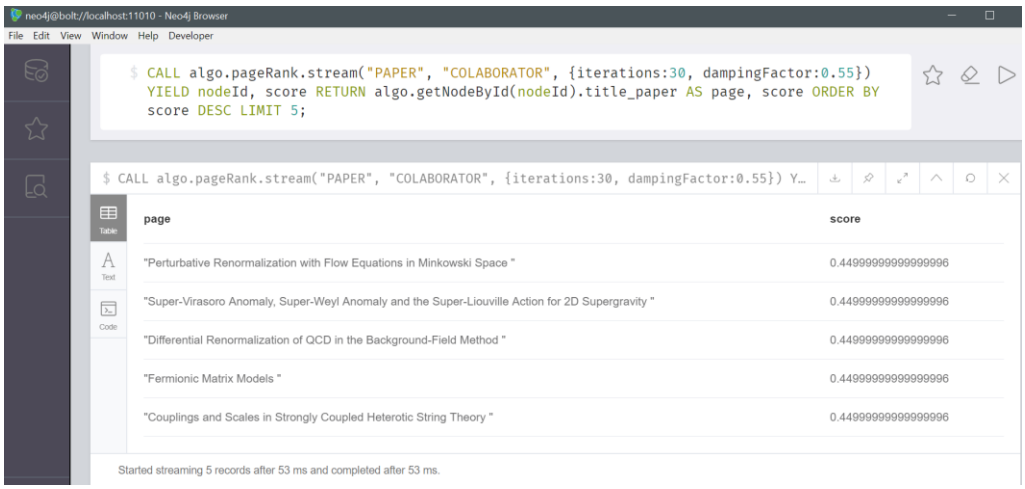


Figura 8 *PageRank* para obtener los artículos científicos con mejor puntaje

3.6.1.1.1. Influencia

La intuición detrás de la influencia es que las relaciones con los nodos más importantes contribuyen más a la influencia del nodo en cuestión que las conexiones equivalentes a los nodos menos importantes. La medición de la influencia generalmente implica puntuar nodos, a menudo con relaciones ponderadas, y luego actualizar las puntuaciones en muchas iteraciones. A veces, se puntúan todos los nodos y en otras se utiliza una selección aleatoria como distribución representativa.

3.6.1.1.2 Fórmula de PageRank

PageRank se define en el documento original de Google de la siguiente manera:

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

Donde:

- Asumimos que una página u tiene citas de las páginas T1 a Tn.
- d es un factor de amortiguación que se establece entre 0 y 1. Por lo general, se establece en 0,85. Puede pensar en esto como la probabilidad de que un usuario continúe haciendo clic. Esto ayuda a minimizar el descenso de rango, que se explica en la siguiente sección.
- 1-d es la probabilidad de que se llegue a un nodo directamente sin seguir ninguna relación.
- C (Tn) se define como el grado de salida de un nodo T [29].

En la Figura 9 muestra un pequeño ejemplo de cómo *PageRank* continuará actualizando el rango de un nodo hasta que converja o cumpla con el número establecido de iteraciones.

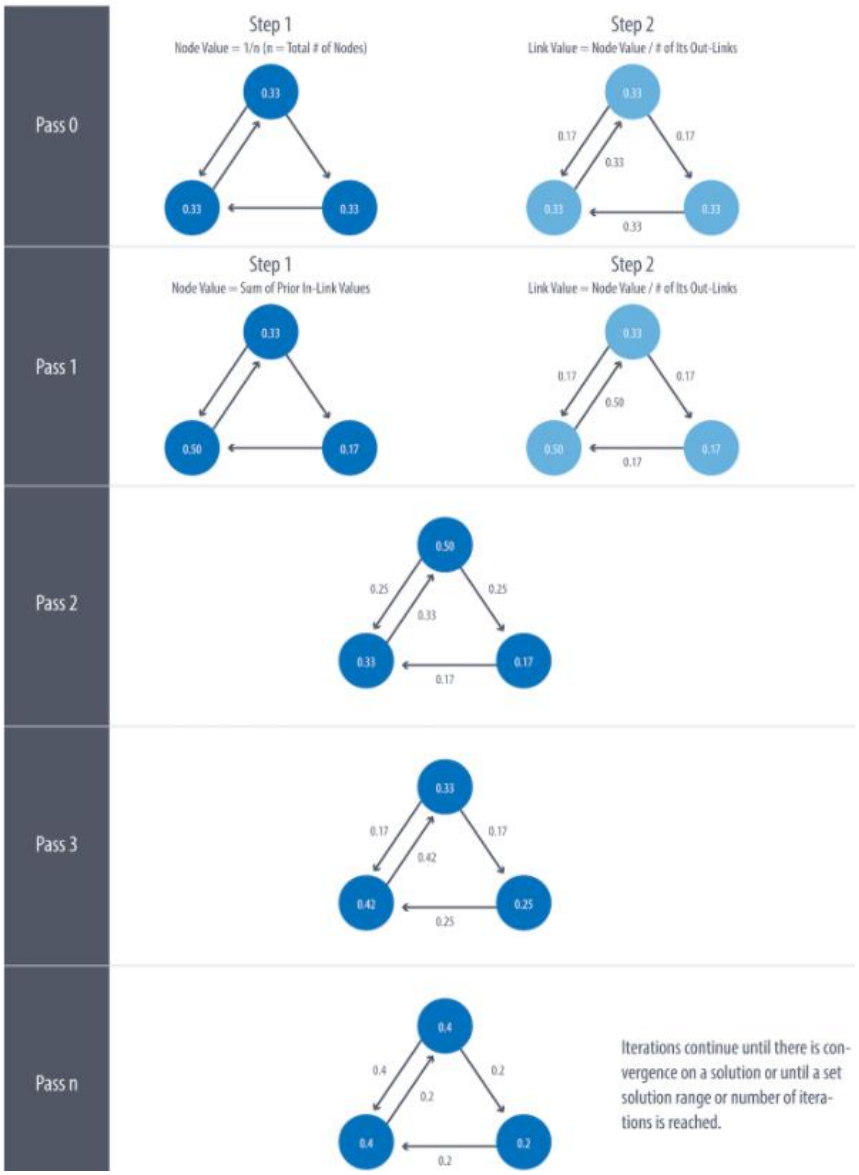


Figura 9 Iteraciones de PageRank [28].

3.6.1.2 Centralidad de intermediación

Centralidad de intermediación o mejor conocido en inglés como "*Betweenness Centrality*". El algoritmo *Betweenness Centrality* primero calcula la ruta más corta (ponderada) entre cada par de nodos en un gráfico conectado. Cada nodo recibe una puntuación, basada en el número de estos caminos más cortos que pasan por el nodo. Cuantos más cortos sean los caminos en los que se encuentra un nodo, mayor será su puntuación [21].

Intermediación La centralidad se consideró una de las "tres concepciones intuitivas distintas de centralidad" cuando fue presentada por *Linton C. Freeman* en su artículo de 1971, "Un conjunto de medidas de centralidad basadas en la intermediación" [28].

Un puente en una red puede ser un nodo o una relación. En un gráfico muy simple, puede encontrarlos buscando el nodo o la relación que, si se elimina, haría que una sección del gráfico se desconectara. Sin embargo, como eso no es práctico en un gráfico típico, usamos un algoritmo de centralidad de intermediación. También podemos medir la intermediación de un clúster tratando al grupo como un nodo.

Un nodo se considera fundamental para otros dos nodos si se encuentra en cada ruta más corta entre esos nodos, como se muestra en la Figura 10.

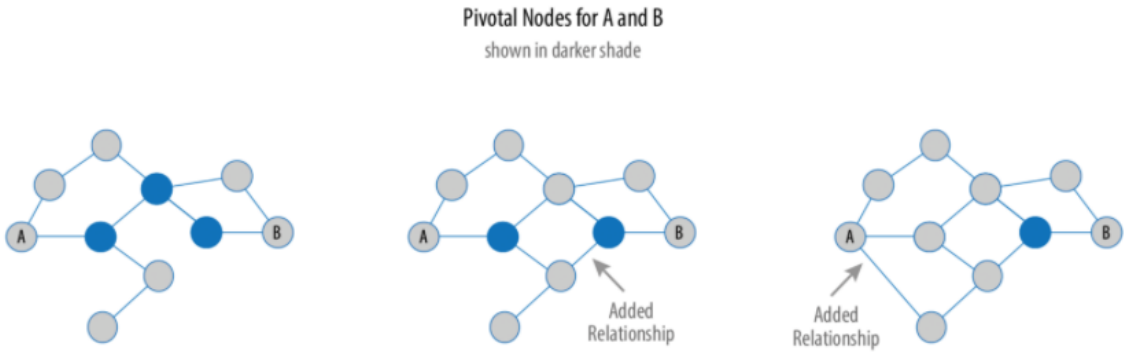


Figura 10 Los nodos fundamentales se encuentran en cada camino más corto entre dos nodos. La creación de rutas más cortas puede reducir la cantidad de nodos fundamentales para usos como la mitigación de riesgos [28].

3.6.1.3 Calculando centralidad de intermediación

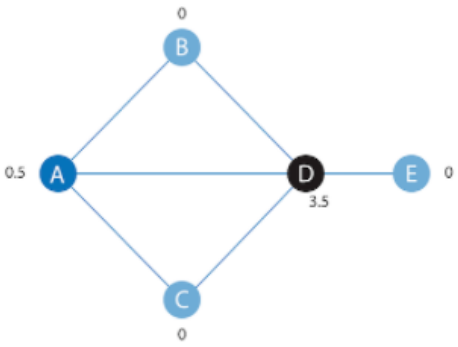
La centralidad de intermediación de un nodo se calcula sumando los resultados de la siguiente fórmula para todas las rutas más cortas:

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

Donde:

- u es un nodo.
- p s el número total de caminos más cortos entre los nodos s y t.
- p (u) es el número de caminos más cortos entre los nodos syt que pasan por el nodo u [28].

La Figura 11 ilustra los pasos para resolver la centralidad de intermediación.



Node D Calculation

Pairs with Shortest Paths Through D	Total Possible Shortest Paths for That Pair	% of Total Through D (1/Total)
A, E	1	1
B, E	1	1
C, E	1	1
B, C	2 (through D & A)	0.5
Betweenness Score		3.5

Figura 11 Conceptos básicos para calcular la centralidad de intermediación [28].

Este es el procedimiento:

- Para cada nodo, busque los caminos más cortos que lo atraviesen.
- B, C, E no tienen rutas más cortas y se les asigna un valor de 0.
- Para cada ruta más corta en el paso 1, calcule su porcentaje del total de rutas más cortas posibles para ese par.
- Sume todos los valores del paso 2 para encontrar la puntuación de centralidad de intermediación de un nodo. La tabla de la Figura 11 ilustra los pasos 2 y 3 para el nodo D.
- Repita el proceso para cada nodo [28].

En la Figura 12, se puede observar a los autores mejor valuados con el algoritmo de centralidad de intermediación utilizando solamente las entidades de tipo autor y las relaciones de colaboración entre los mismos.

Figura 12 Visualización de la centralidad de intermediación aplicada para obtener los autores mejor valuados de nuestro dataset [28].

3.7. Algoritmos detección de comunidades

La formación de comunidades es común en todo tipo de redes, e identificarlas es fundamental para evaluar el comportamiento grupal y los fenómenos emergentes. El principio general para encontrar comunidades es que sus miembros tendrán más relaciones dentro del grupo que con nodos fuera de su grupo. La identificación de estos conjuntos relacionados revela agrupaciones de nodos, grupos aislados y estructura de red. Esta información ayuda a inferir comportamientos o preferencias similares de grupos de pares, estimar la resiliencia, encontrar relaciones anidadas y preparar datos para otros análisis. Los algoritmos de detección de comunidades también se utilizan comúnmente para producir visualización de redes para inspección general [28].

3.7.1.1. Componentes Fuertemente Conectados

El algoritmo Componentes Fuertemente Conectados o mejor conocido en inglés "*Strongly Connected Components*" (SCC) es uno de los algoritmos más nuevos de grafos. SCC encuentra conjuntos de nodos conectados en un grafo dirigido donde cada nodo es accesible en ambas direcciones desde cualquier otro nodo en el mismo conjunto. Sus operaciones en tiempo de ejecución escalan bien, proporcionalmente al número de nodos. En la Figura 13, puede ver que los nodos de un grupo SCC no necesitan ser vecinos inmediatos, pero debe haber rutas direccionales entre todos los nodos del conjunto [28].

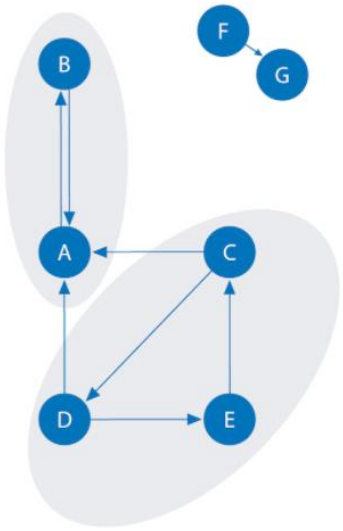


Figura 13 Componentes Fuertemente Conectados [28].

Utilizar Componentes Fuertemente Conectados como un paso inicial en el análisis de grafos para ver cómo está estructurado o para identificar grupos estrechos que pueden justificar una investigación independiente. Un componente que está fuertemente conectado se puede utilizar para perfilar comportamientos o inclinaciones similares en un grupo para aplicaciones como motores de recomendación [28].

Muchos algoritmos de detección de comunidades, como SCC, se utilizan para encontrar y colapsar clústeres en nodos individuales para un mayor análisis entre clústeres. También puede usar SCC para visualizar ciclos para análisis, como encontrar procesos que pueden bloquearse porque cada subproceso está esperando que otro miembro tome acción [28].

En la figura 14 se puede observar algunas de las comunidades formadas con el algoritmo SCC dentro de nuestro subconjunto de autores interconectados por las citas bibliográficas que los relacionan. Así mismo se cumple con la regla de accesibilidad entre los miembros de cada comunidad.

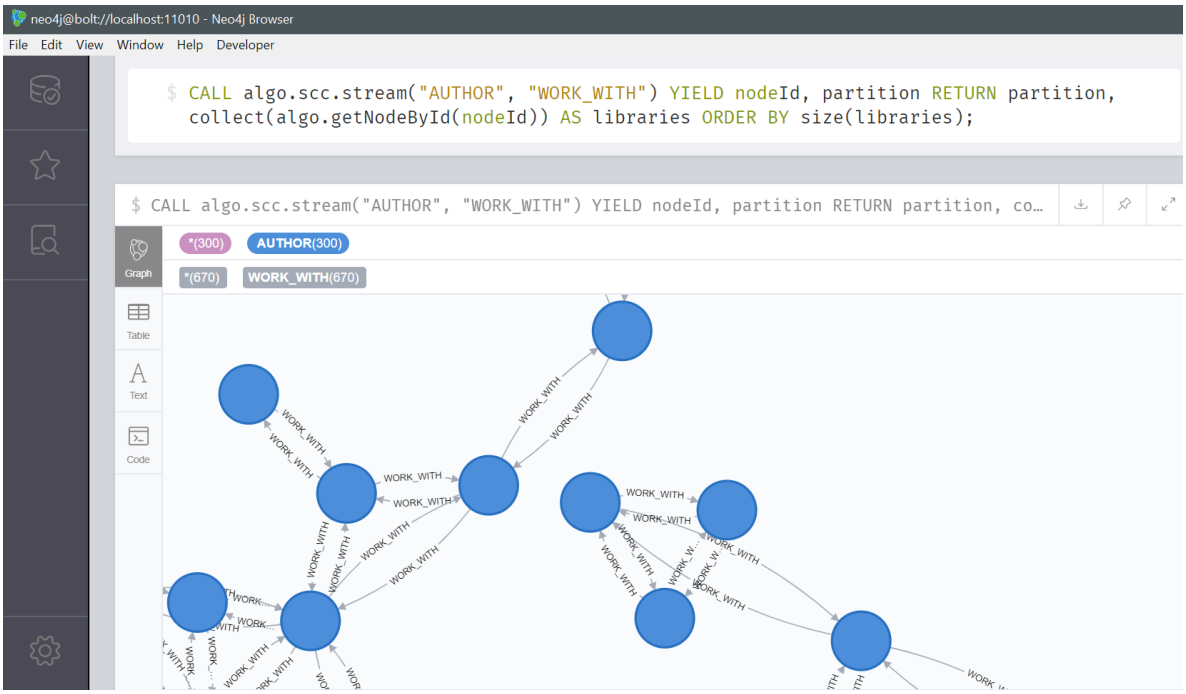


Figura 14 SCC aplicado al subgrafo de autores de la presente solución.

3.7.1.2. Label propagation

El algoritmo de Propagación de Etiquetas o mejor conocido en inglés como "*Label Propagation*" (LPA) es un algoritmo rápido para encontrar comunidades en un grafo. En LPA, los nodos seleccionan su grupo en función de sus vecinos directos. Este proceso se adapta bien a las redes donde las agrupaciones son menos claras y las ponderaciones se pueden usar para ayudar a un nodo a determinar en qué comunidad ubicarse. También se presta bien al aprendizaje semi supervisado porque puede sembrar el proceso con etiquetas de nodo indicativas asignadas previamente [28].

La intuición detrás de este 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 quedan atrapadas dentro de un grupo de nodos densamente conectados y los nodos que terminan con la misma etiqueta cuando finaliza el algoritmo se consideran parte de la misma comunidad. El algoritmo resuelve superposiciones, donde los nodos son potencialmente

parte de múltiples clústeres, asignando membresía al vecindario de etiquetas con la relación combinada y el peso de nodo más altos. LPA es un algoritmo relativamente nuevo propuesto en 2007 por U. N. *Raghavan, R. Albert y S. Kumara*, en un artículo titulado "Algoritmo de tiempo casi lineal para detectar estructuras comunitarias en redes a gran escala" [28].

La Figura 15 y la Figura 16 muestran dos variaciones de la Propagación de Etiquetas, un método de empuje simple y el método de tracción más típico que se basa en pesos de relación. El método de extracción se presta bien a la paralización [28].

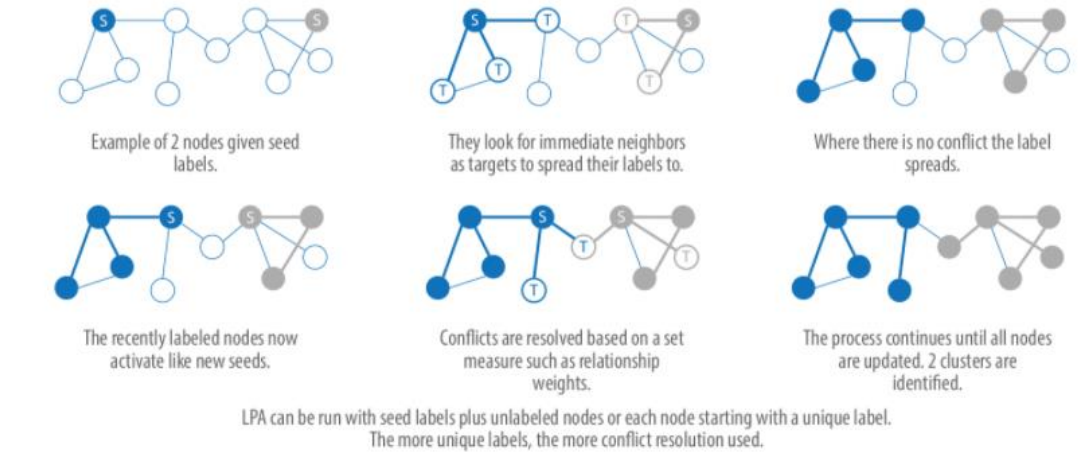


Figura 15 Propagación de Etiquetas método de empuje simple [28].

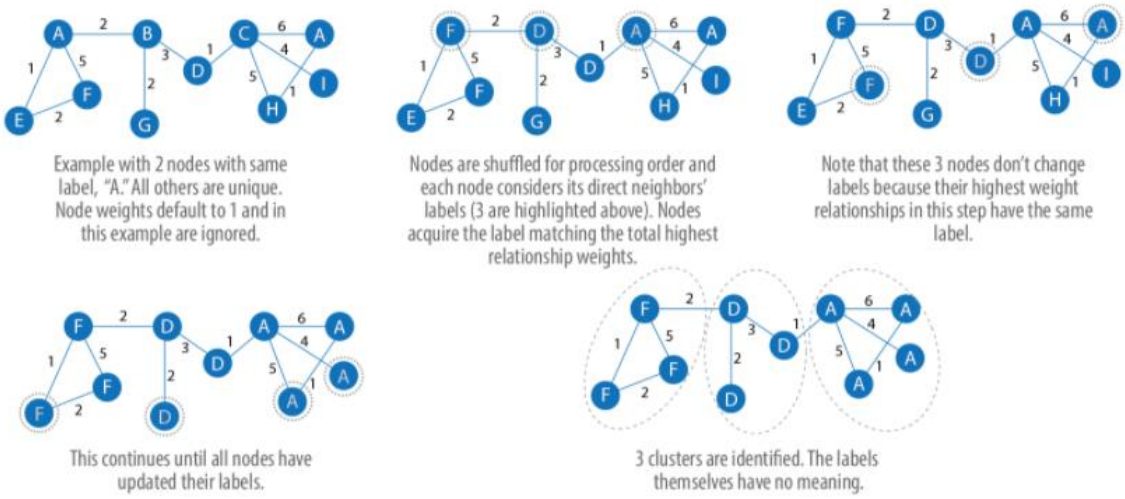


Figura 16 Propagación de Etiquetas método de tracción [28].

Los pasos que se utilizan a menudo para el método de extracción de propagación de etiquetas son:

- Cada nodo se inicializa con una etiqueta única (un identificador) y, opcionalmente, se pueden utilizar etiquetas de "semilla" preliminares.

- Las etiquetas se propagan a través de la red.
- 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.
- LPA alcanza la convergencia cuando cada nodo tiene la etiqueta mayoritaria de sus vecinos [28].

A medida que las etiquetas se propagan, los grupos de nodos densamente conectados llegan rápidamente a un consenso sobre una etiqueta única. Al final de la propagación, solo quedarán unas pocas etiquetas, y los nodos que tienen la misma etiqueta pertenecen a la misma comunidad [28].

En la figura 17 se muestra un ejemplo de una comunidad generada con el algoritmo de Louvain utilizando nuestro grafo de artículos científicos, tomando como base las entidades de tipo autor y las relaciones de colaboración *work_with* que hay entre ellos.

Figure 17 Comunidad generada con el algoritmo de Louvain

3.7.1.2.1 Aprendizaje semi-supervisado y etiquetas de semillas

A diferencia de otros algoritmos, Propagación de Etiquetes puede devolver diferentes estructuras de comunidad cuando se ejecuta varias veces en el mismo gráfico. El orden en el que LPA evalúa los nodos puede influir en las comunidades finales que devuelve [28].

El rango de soluciones se reduce cuando algunos nodos reciben etiquetas preliminares (es decir, etiquetas de semillas), mientras que otros no están etiquetados. Es más probable que los nodos sin etiqueta adopten las etiquetas preliminares [28].

Este uso de la Propagación de Etiquetas puede considerarse 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 evolucionan [28].

Finalmente, LPA a veces no converge en una única solución. En esta situación, los resultados de nuestra comunidad cambiarán continuamente entre algunas comunidades notablemente similares y el algoritmo nunca se completará. Las etiquetas de semillas ayudan a guiarlo hacia una solución. *Spark* y Neo4j usan un número máximo establecido de iteraciones para evitar una ejecución interminable [28].

3.7.1.3. Modularidad de Lovian

El algoritmo de Modularidad de *Louvain* encuentra clústeres comparando la densidad de la comunidad a medida que asigna nodos a diferentes grupos. Se puede pensar en esto como un análisis de "qué pasaría si" para probar varias agrupaciones con el objetivo de alcanzar un óptimo global [28].

Propuesto en 2008, el algoritmo *Louvain* es uno de los algoritmos basados en modularidad más rápidos. Además de detectar comunidades, también revela una jerarquía de comunidades a diferentes escalas. Esto es útil para comprender la estructura de una red en diferentes niveles de granularidad [28].

Louvain cuantifica qué tan bien se asigna un nodo a un grupo al observar la densidad de conexiones dentro de un grupo en comparación con una muestra promedio o aleatoria. Esta medida de asignación comunitaria se llama modularidad. Es el más conocido de los algoritmos de centralidad. Mide la influencia transitiva (o direccional) de los nodos. *PageRank* considera la influencia de los nodos vecinos de un nodo. Por ejemplo, tener algunos amigos muy poderosos puede hacerte más influyente que tener muchos amigos menos poderosos. *PageRank* se calcula distribuyendo iterativamente el rango de un nodo entre sus vecinos o atravesando aleatoriamente el grafo y contando la frecuencia con la que se golpea cada nodo durante estos recorridos [28].

3.1.7.3.2 Agrupación basada en la calidad mediante modularidad

La modularidad es una técnica para descubrir comunidades al dividir un gráfico en módulos (o grupos) más generales y luego medir la fuerza de las agrupaciones. A diferencia de simplemente observar la concentración de conexiones dentro de un grupo, este método compara las densidades de relación en grupos determinados con las densidades entre grupos. La medida de la calidad de esas agrupaciones se llama modularidad [28].

Los algoritmos de modularidad optimizan las comunidades a nivel local y luego a nivel mundial, utilizando múltiples iteraciones para probar diferentes agrupaciones y aumentar la aspereza. Esta estrategia identifica las jerarquías de la comunidad y proporciona una comprensión amplia de la estructura general. Sin embargo, todos los algoritmos de modularidad adolecen de dos inconvenientes:[28].

- Fusionan comunidades más pequeñas en comunidades más grandes.

- Puede ocurrir una meseta donde varias opciones de partición están presentes con modularidad similar, formando máximos locales e impidiendo el progreso.

Un cálculo simple de modularidad se basa en la fracción de las relaciones dentro de los grupos dados menos la fracción esperada si las relaciones se distribuyeran al azar entre todos los nodos. El valor está siempre entre 1 y –1, los valores positivos indican más densidad de relación de la que cabría esperar por azar y los valores negativos indican menos densidad. La Figura 18 ilustra varias puntuaciones de modularidad diferentes basadas en agrupaciones de nodos.

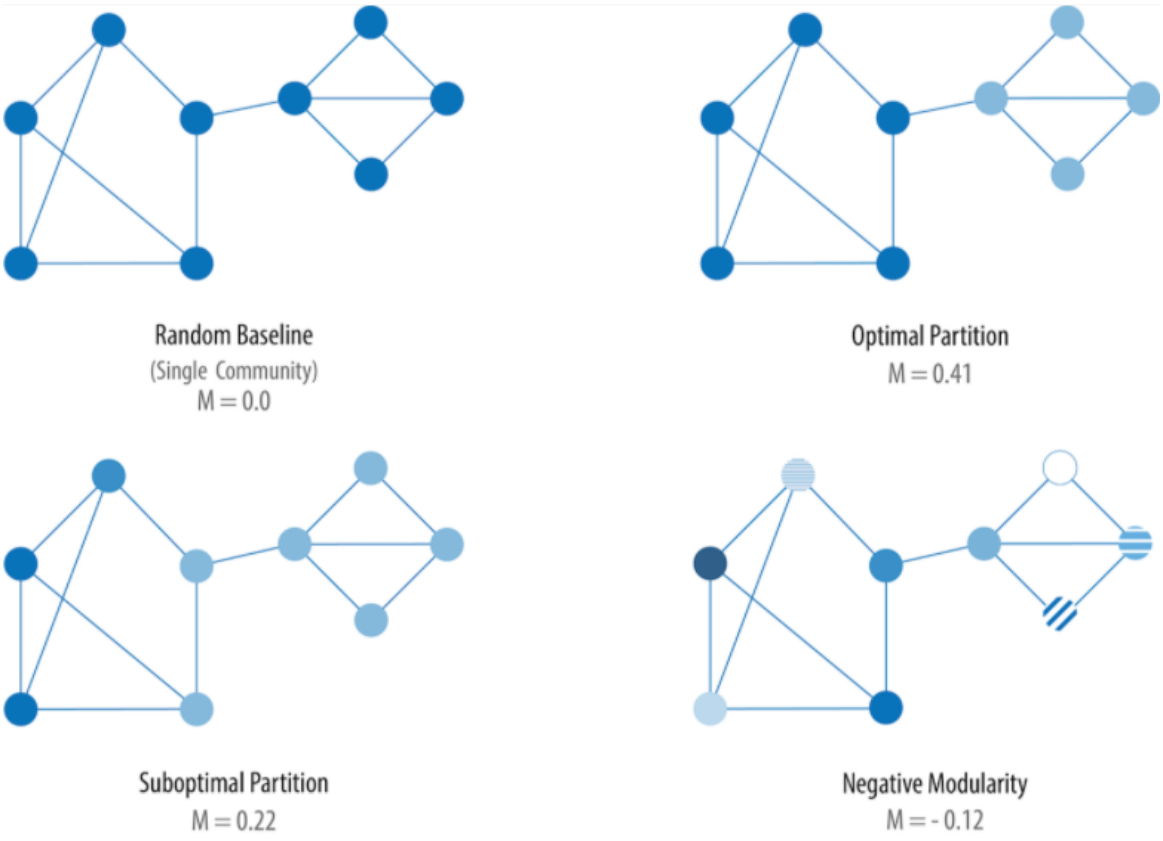


Figura 18 Cuatro puntuaciones de modularidad basadas en diferentes opciones de partición [28].

La fórmula para la modularidad de un grupo es:

$$M = \sum_{c=1}^{n_c} [\frac{Lc}{L} - \left(\frac{k_c}{2L}\right)^2]. [28].$$

Dónde:

- L es el número de relaciones en todo el grupo.

- Lc es el número de relaciones en una partición.
- kc es el grado total de nodos en una partición.

El cálculo de la partición óptima en la parte superior de la Figura 18 es el siguiente:

- La partición oscura es $(\frac{7}{13} - \left(\frac{15}{2(13)}\right)^2) = 0.205$
- La partición ligera es $(\frac{5}{13} - \left(\frac{11}{2(13)}\right)^2) = 0{,}206$
- Estos se suman para M = 0,205 + 0,206 = 0,41

El algoritmo consiste en la aplicación repetida de dos pasos, como se ilustra en la Figura 19.

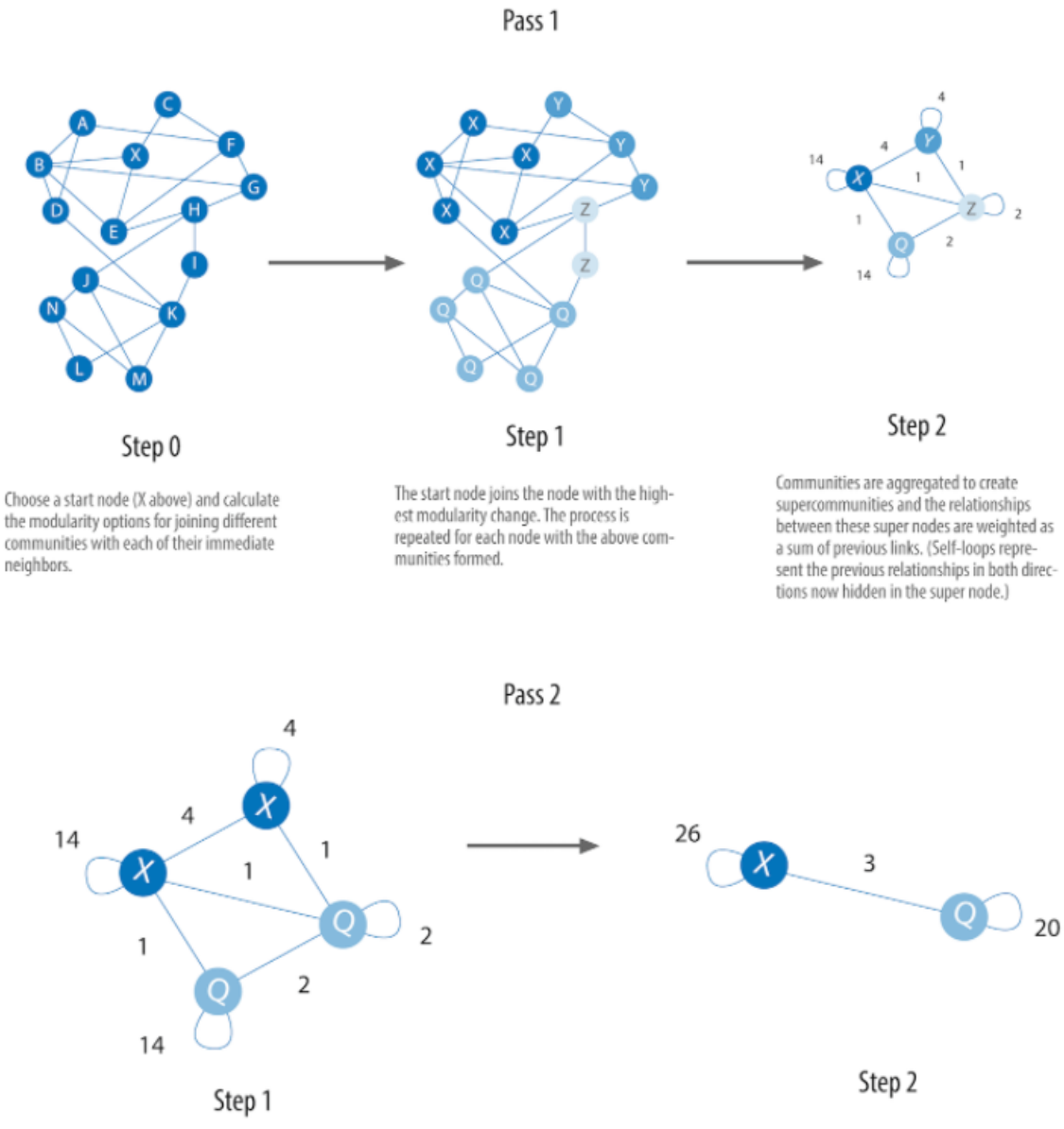


Figura 19 Proceso de algoritmo de Louvian [28].

3.8. Minería de texto.

Para el análisis, obtención e interpretación de la información del proyecto, es necesario aplicar ciertas técnicas de minería de texto que nos permiten recuperar solo la información necesaria para el sistema de recomendación desde las fuentes crudas de texto y así guardar la información dentro de nuestra base de datos basada en grafos (Neo4j).

La minería de textos es el conjunto de procesos necesarios para transformar documentos o recursos de texto desestructurados en información relevante y estructurada. La información estructurada puede utilizarse para descubrir automáticamente patrones ocultos y predecir resultados futuros utilizando una combinación de técnicas estadísticas, lingüísticas y de reconocimiento de patrones. Un proceso típico de minería de textos consta de los siguientes pasos:

- Identificar el texto y someterlo a pretratamiento: limpieza del texto para eliminar información innecesaria, división en "*tokens*" individuales (componentes más pequeños) e identificación de las categorías gramaticales según la gramática del idioma utilizado.
- Extraer información pertinente y transformarla en datos estructurados. La información se recupera buscando en todo el texto convertido en "*tokens* y almacenando los resultados de una manera más estructurada y organizada que permita el análisis del mismo.
- Seleccionar características importantes para crear modelos y categorías conceptuales. El número de conceptos presentes en los datos no estructurados suele ser muy grande. La clave es identificar las características más relevantes y utilizarlas para construir modelos significativos basados en categorías y relaciones de datos [30].

3.8.1 Tokenización.

La tokenización es una parte interesante del análisis de texto. Un "*token*" en términos de lenguaje natural es "una instancia de una secuencia de caracteres en un documento en particular que se agrupan como una unidad semántica útil para su procesamiento". Como las raíces y las ramas de un árbol, todo el lenguaje humano es un lío de excrecencias naturales: divididas, en descomposición, vibrantes y florecientes. La tokenización es parte de la metodología que usamos cuando enseñamos a las máquinas sobre palabras, la parte fundamental de nuestro invento más importante [31].

En el caso de nuestro proyecto, es necesario este procedimiento para detectar la información relevante de nuestra fuente principal de recursos, y de esta manera utilizando expresiones regulares, recopilar fragmentos de la información que será guardada en la base de datos no relacional.

3.8.2 Normalización.

La normalización es un proceso que convierte una lista de palabras en una secuencia más uniforme. Esto es útil para preparar texto para su posterior procesamiento. Al transformar las palabras a un formato estándar, otras operaciones pueden trabajar con los datos y no tendrán que lidiar con problemas que puedan comprometer el proceso. Por ejemplo, convertir todas las palabras a minúsculas simplificará el proceso de búsqueda.

Este proceso está relacionado con nuestro proyecto porque es necesaria la eliminación de caracteres especiales, y palabras sin contexto para reducir así la dimensionalidad del texto crudo inicial y optimizar el rendimiento temporal de nuestros algoritmos de búsqueda y análisis [32].

3.9. Desarrollo WEB.

El desarrollo web es un término que define la creación de sitios web para Internet o una intranet. Para lograr esto, la tecnología de software se usa en el lado del servidor y del lado del cliente, lo que implica una combinación de procesos de base de datos utilizando un navegador para realizar ciertas tareas o mostrar información. Este tipo de desarrollo implica la creación de un sitio web, por lo general, se refiere al lado de codificación y programación de la producción del sitio web en lugar del lado del diseño web.

Abarca todo, desde una simple página de texto HTML hasta aplicaciones complejas y ricas en características diseñadas para acceder desde varios dispositivos conectados a Internet. Ejemplos de desarrollo web rico en funciones incluyen sitios web de comercio electrónico, sistemas de gestión de contenido (CMS) y redes sociales. Los lenguajes y software de programación de desarrollo web más comunes incluyen lenguaje de marcado de hipertexto (HTML), hojas de estilo en cascada (CSS), JavaScript, PHP, Drupal y MySQL [33].

Para el desarrollo de nuestro sistema de recomendación se hizo un sistema web local con la capacidad de atender las peticiones de los usuarios (*Backend*) y devolver en formato JSON la información resultante de las consultas que se realizan.

3.9.1 Servidor Frontend

El frontend es el cliente y es el que se encarga de toda la lógica de este cuando desea realizar alguna petición, podemos decir que este concepto surge en el 2008 debido a las herramientas que surgieron a partir de ese momento: HTML5, CSS3 (2008), JSON (2013 - 2015), *AngularJS* (2010), *Ember*, *Backbone*, *Rest* (2000) y *Nodejs*, estas tecnologías hicieron posible el *Frontend*.

El *Frontend* ha ido evolucionando con nuevas tecnológicas como lo son: ES6 (2015), *React* (2013), *Vue* (2014), *Angular* (2016), *GraphQL* (2015; en el panorama actual entra en el juego *WebAssembly* que es un nuevo tipo de código que se ejecuta en los navegadores web modernos y proporciona nuevas funciones y grandes ganancias en el rendimiento.

3.9.2 Servidor Backend

Backend es la capa de acceso a datos de un *software* o cualquier dispositivo, que no es directamente accesible por los usuarios, además contiene la lógica de la aplicación que maneja dichos datos. El *Backend* también accede al servidor, que es una aplicación especializada que entiende la forma como el navegador solicita cosas.

Algunos de los lenguajes de programación para *Backend* son Python, Node.js, PHP, Go, Ruby y C#. Y así como en el *frontend*, todos estos lenguajes tienen diferentes *frameworks* que te permiten trabajar mejor según el proyecto que estás desarrollando, como *Django, Flask, Express.js, Laravel, Symphony Framework, Ruby on Rails* y *ASP.Net* [34].

Existen diversas formas de configuración de servidores para hacer la conexión entre el *frontend* y el *backend*, donde la mayoría de las veces se trabajan en servidores separados con diferentes capacidades de almacenamiento pero en el caso de este proyecto, se decidió utilizar un solo servidor local donde se encuentran *backend* y *frontend* pero trabajando de la misma manera en cuanto a envío y recepción de peticiones se refiere.

3.9.3 JSON

JSON, cuyo nombre corresponde a las siglas *JavaScript Object Notation* o Notación de Objetos de *JavaScript*, es un formato ligero de intercambio de datos, que resulta sencillo de leer y escribir para los programadores y simple de interpretar y generar para las máquinas.

Una de las características de JSON, al ser un formato que es independiente de cualquier lenguaje de programación, es que los servicios que comparten información por este método no necesitan hablar el mismo idioma, es decir, el emisor puede ser Java y el receptor Python, pues cada uno tiene su propia librería para codificar y decodificar cadenas en este formato [35].

Por estas razones decidimos utilizarlo como el medio de comunicación entre nuestro servidor *backend* y nuestro servidor *frontend* simulados, con la finalidad de enviar y recibir la información recolectada de la base de datos para la atención de las peticiones del usuario final.

4. DESARROLLO METODOLÓGICO

En este capítulo se presenta en detalle el desarrollo metodológico de la presente propuesta de solución, en la cual se utilizó una metodología de prototipo incremental con enfoque a la creación de software, donde su principal característica es el seguimiento de las etapas de manera secuencial donde se va iniciando una nueva fase del proyecto en cuanto se termina la previa. Las fases utilizadas para elaboración del presente proyecto son: planeación, diseño, implementación, verificación e instalación.

4.1 Planeación.

Dentro de esta etapa se tiene contemplado la investigación de los temas necesarios para el entendimiento del problema, así como el desarrollo e implementación de la propuesta, como es el caso de los algoritmos más utilizados por los sistemas de recomendación, los tipos de base de datos que aplican y características específicas de su metodología analizada para observar las áreas de oportunidad que se podrían implementar en nuestra solución.

Así mismo se definió los pasos de diseño necesarios para la implementación de nuestro sistema de recomendación específico en artículos científicos y las materias útiles ofrecidas dentro de la maestría en sistemas computacionales para complementarnos de los conocimientos necesarios para el desarrollo del proyecto.

En la figura 20 se puede observar el diagrama planeado para la presente propuesta, la cual consiste en la creación de una herramienta de recomendación de artículos científicos, autores y journals en base a las búsquedas realizadas por los usuarios finales.

Todo el proceso inicia con la extracción de información relevante de la fuente de información SNAP, utilizando técnicas de minería de texto como normalización y tokenización en base a expresiones regulares.

Posteriormente se toma esta información relevante y se vuelca a archivos de formato CSV que se usarán para la importación de información dentro de la base de datos de grafos Neo4j. Una vez teniendo el esquema cargado en Neo4j, se planifica la realización de pruebas con diferentes algoritmos de comunidad y relevancia mencionados en el capítulo 3, tomando como resultado final la decisión de implementar la combinación de Louvain y PageRank.

Como fase final se diseña la estructura del funcionamiento del sistema web entre los usuarios finales y nuestro algoritmo interno, donde los usuarios harán búsquedas por campos específicos para encontrar el artículo científico, autor o journal de su interés y la presente solución le proporcionará como resultado un subgrafo de sugerencias relacionadas a la búsqueda que realizo el usuario, de manera que no solamente se cumplirá con la búsqueda de sus necesidades, también se le mostrara sugerencias de las entidades relacionadas.

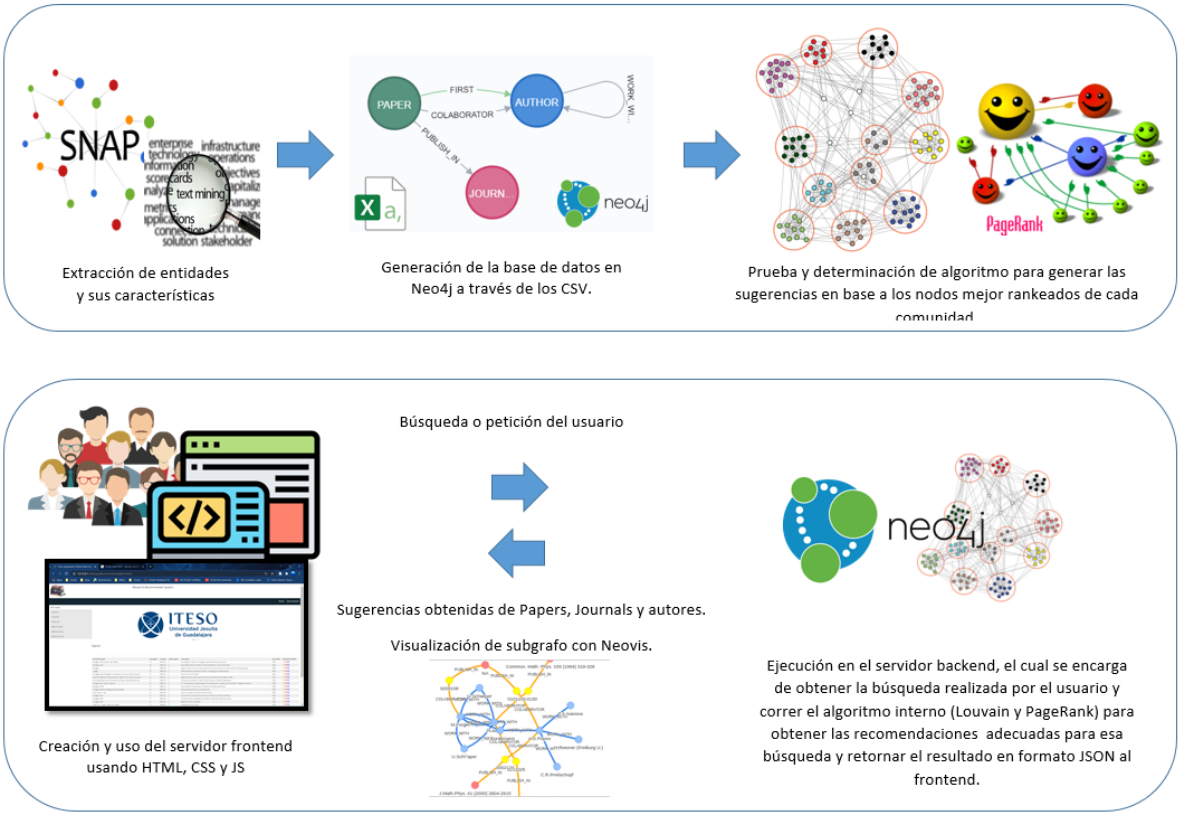


Figura 20 Diagrama de la solución propuesta

4.2 Diseño.

4.2.1 Obtención de la información cruda

Contempla el análisis de diferentes fuentes de información dentro de la web, para elegir la que tenga más consistencia en su información; Esto se refiere a que el *corpus* sea coherente, informativo y tenga las referencias necesarias entre sus artículos para hacer una representación similar en la base de datos basada en grafos.

En la figura 21 se puede observar la sección de la página de SNAP donde se obtuvo la información en crudo con los artículos científicos, sus relaciones, autores y journals.

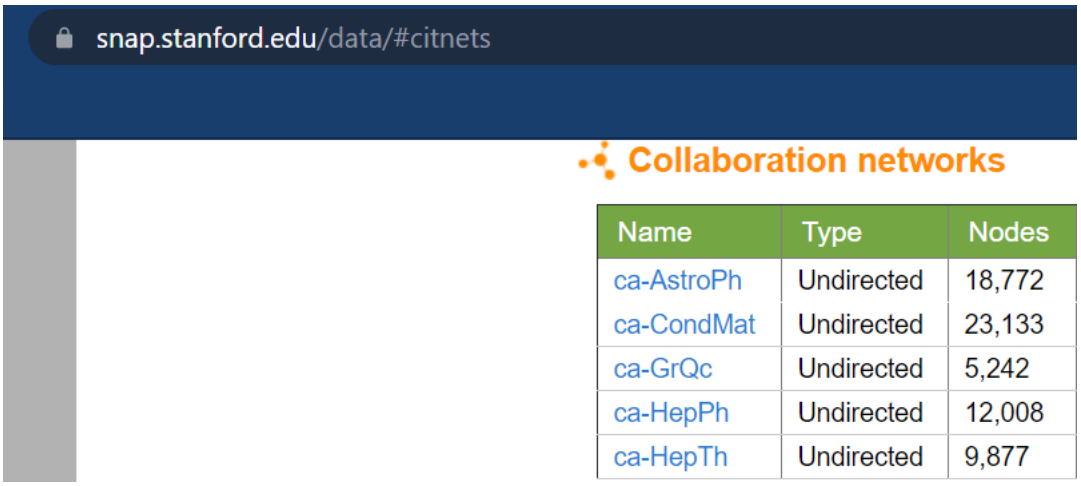


Figura 21 Página de SNAP con la información en crudo.

4.2.2 Minería de texto

Una vez que se ha elegido el *corpus* de documentos a utilizar, se eligieron los métodos de minería de texto más útiles para extraer solo la información con más contexto significativo para nuestro sistema de recomendación de artículos científicos, este paso implica un análisis exhaustivo de los documentos y sus propiedades, de tal manera que puedan ser transformadas a entidades en el contexto de una base de datos basada en grafos.

4.2.3 Normalización

Este paso consiste en la limpieza del *corpus*, es decir, eliminar todos los signos de puntuación, caracteres especiales, y palabras sin contexto que estén presente en nuestro *dataset* inicial y que no ayudan a realizar un análisis completo del problema que se busca resolver con nuestra propuesta de solución. Dentro de los pasos intermedios se puede contemplar una lista negra de palabras por ser ignoradas en nuestro algoritmo.

4.2.4 Tokenización

Posterior a conseguir la información más relevante del texto crudo previamente analizado y limpio para nuestro sistema de recomendación, el siguiente paso consiste en hacer una separación de *tokens* o palabras del texto, con la finalidad de hacer una fácil interpretación por parte de nuestro procesamiento de transformación y carga de información en la base de datos no relacional de Neo4j.

4.2.5 Creación formato csv

Una parte importante dentro del diseño de la presente propuesta es elegir el formato de archivo con el cual se hará la transformación de nuestro texto crudo a un grafo con entidades y aristas en Neo4j, y debido a la simpleza de la importación de la información, se decide utilizar el formato CSV para esta parte del proceso. Se creó un archivo para cada tipo de entidad así como para cada tipo de relación, de manera que en la figura 22 se observa el total de archivos generados para la creación de la base de datos.

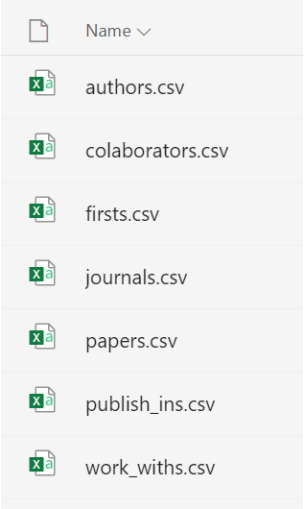


Figura 22 Archivos CSV generados con la información procesada.

Es necesario especificar para cada entidad y relación, sus respectivos atributos, en la imagen 23 se puede apreciar un fragmento del archivo CSV relacionado a los artículos científicos, donde además se describe en la última columna la etiqueta o nombre de la entidad a la que pertenece la información.

	A	B	C	D	E	F	G
1	id_paper:ID(PAPER)	title_paper	year_paper	month_pa	day_pape	comments	:LABEL
2	9201001	Combinat	1991	12	31	46 pages	PAPER
3	9201002	Inomogen	1992	1	2	5 pags. 0 f	PAPER
4	9201003	Intersectio	1992	1	2	73 pages,	PAPER

Figura 23 Archivo CSV con información de artículos científicos

4.2.6 Generación de la base de datos

El principal objetivo de este paso tiene que ver con el diseño del grafo a nivel entidad – relación, definiendo los atributos de cada tipo de entidad, así como los atributos de cada arista y los tipos de relaciones que existirán en nuestro grafo. Por último, buscar e implementar el comando de *CYPHER* que sea capaz de hacer esta transformación e importación a nuestra base de datos.

La imagen 24 describe la topología del grafo a utilizar, además se consideran los siguientes atributos para cada uno de los nodos y relaciones.

Papers. *Title, year, month, day, comments.*

Author. *Id, name.*

Journal. *Id, title.*

Colaborator. *Id_colaborator, weight_c.*

Work_with. *Id_work_with, weight_w.*

Publish_in. *id_publish_in, weight_p.*

First. *Id_first, weight_f.*

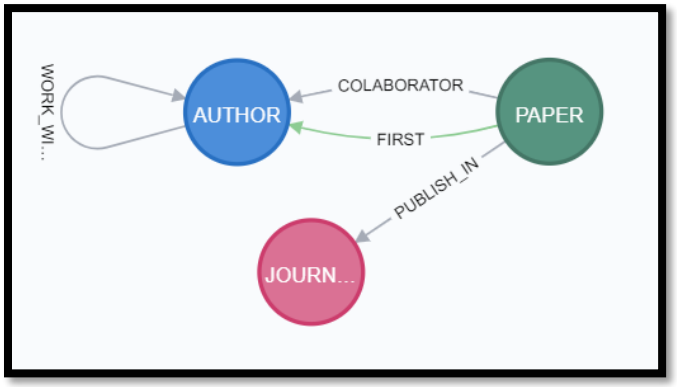


Figura 24 Topología del grafo creado.

4.2.7 Algoritmos

Es la parte de diseño más importante de la presente propuesta de solución, porque implica la colección, análisis y prueba de diferentes algoritmos basados en grafos que sean de utilidad para generar recomendaciones de calidad y con coherencia. Dentro de los tipos de algoritmos analizados se contemplan los de relevancia de entidades dentro de la topología del grafo, recorrido del grafo, modularidad, *clustering* y análisis de similitud. Una vez que se tienen seleccionados los algoritmos a utilizar, se busca la creación de un algoritmo hibrido entre los mismos de manera que se complementen sus cualidades.

En esta fase del programa, se corrieron funciones integradas en la librería apoc para observar el comportamiento de cada algoritmo mencionado en el capítulo 3. La conclusión fue implementar primero un algoritmo de generación de comunidades de nodos similares en base a su modularidad como lo es Louvain y ordenarlos por relevancia dentro de su topología, de manera que al buscar un nodo en específico, las recomendaciones sean los nodos que están dentro del mismo *cluster* y los nodos relacionados a estos mismos.

4.2.8 Creación del servidor

Se decide utilizar el servidor local de la computadora de desarrollo como un sistema de *backend* y *frontend* pero simulando el mismo proceso que se tendría que hacer en dos servidores independientes con solicitudes y respuestas vía protocolo http. Así mismo se eligen las tecnologías a ser utilizadas para la programación del comportamiento de los servidores, en este caso *javascript* y *nodejs*.

4.2.9 Interfaz gráfica

Contempla la creación de plantillas con diseños para ser utilizados dentro de nuestro sistema web, la ubicación de los componentes del *DOM*, la interacción con el usuario buscando en todo momento que sea intuitivo y fácil de utilizar. Así mismo se hace la recolección de imágenes y elección de librerías para su desarrollo.

En la figura 25 se puede observar la página principal del sistema de recomendación de artículos científicos montado en un servidor local y con la capacidad de encontrar *papers*, autores y *journals* con diferentes criterios de búsqueda, uno por cada atributo de los previamente mencionados en la sección 4.2.6.

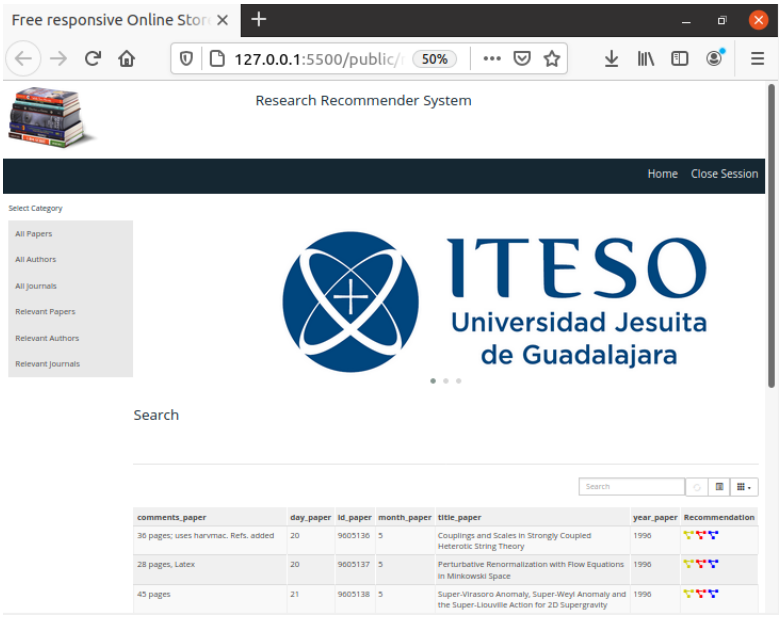


Figura 25 Página principal del sistema de recomendación.

4.2.10 Método de conexión entre Frontend y Backend

Es necesario la configuración de puertos para ser usado por cada uno de los servidores, así como los códigos de respuesta por ser enviados desde el servidor *backend* hasta el servidor *frontend* y la estructura de los objetos JSON que sirven como método de comunicación entre ambas partes.

4.2.11 Visualización de recomendaciones

Una vez que se tiene listo el diseño de la topología de nuestra base de datos, se hace un análisis de las diferentes formas de representar los sub-grafos de recomendación y las librerías que pueden ser de ayuda para su visualización, de tal manera que nos decidimos por un método de representación y los pasos que se deben seguir para su implementación dentro del sistema web de nuestra propuesta de solución.

La herramienta seleccionada fue Neovis debido a que su implementación con Neo4j es práctica, solamente especificando las propiedades de conexión con la base de datos y el *query* a representar en formato HTML dentro del árbol DOM. En la figura 26 se puede ver la página principal de la herramienta y un ejemplo grafico de su representación.

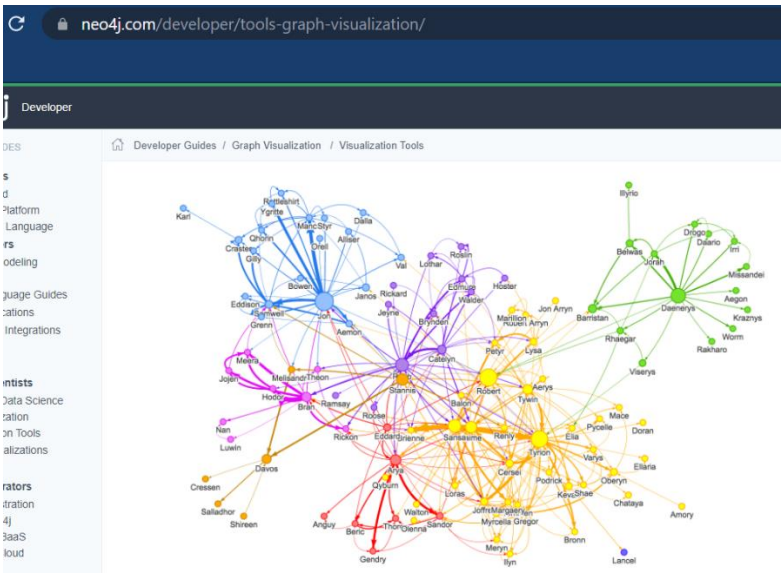


Figura 26 Página oficial de Neovis y ejemplo de uso.

4.3 Implementación.

Para la implementación de la solución utilizamos los pasos descritos en la etapa de desarrollo del proyecto, donde analizamos primeramente la información cruda relacionada a los artículos científicos, colectamos la información más relevante dentro de nuestro contexto con métodos de minería de texto para hacer la importación de las entidades en nuestra base de datos a través de un archivo separado por comas. Esta base de datos de grafos creada es la fuente de alimentación del servidor *backend* el cual se encarga de atender las peticiones que hace el servidor *frontend* relacionando dichas peticiones con la interacción del usuario final con el sistema web.

4.4 Verificación.

Dentro de esta fase, se hace la validación del funcionamiento completo del sistema, es decir, considerando los algoritmos que corren directamente en Neo4j, el servidor *frontend* encargado de interactuar con el cliente y el servidor *backend* de atender las peticiones del usuario para obtener así la información de la base de datos. Dado que la propuesta de solución es un sistema de recomendación, es necesario observar analíticamente las recomendaciones obtenidas por el sistema web y comparar contra la perspectiva que podría tener nuestro cliente final y de esta manera decidir si las recomendaciones son eficientes o el sistema completo necesita algún tipo de ajuste.

4.5 Instalación.

Una vez que el sistema ha sido ajustado correctamente y las recomendaciones son válidas dentro de nuestro contexto de artículos científicos, el siguiente paso es hacer el levantamiento de los servidores localmente para observar al sistema trabajando como si estuviera en el punto final de su ciclo de vida. La instalación y montaje de la aplicación se realizó en el sistema operativo Ubuntu v20, donde se instalaron los componentes necesarios para la ejecución del sistema web, como nodejs, Neovis, y algunos *frameworks* relacionados a *visual studio code*. En la figura 27 se puede observar el sistema instalado en una máquina virtual del sistema operativo Ubuntu.

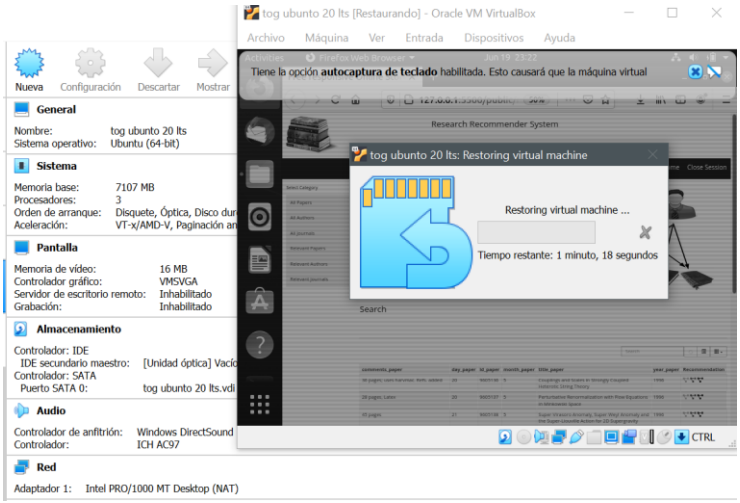


Figura 27 Instalación del sistema de recomendación en máquina virtual Ubuntu

El presente proyecto está constituido por la creación de un sistema de recomendación completo, desde la producción de la base de datos desde archivos de texto en crudo, la codificación de las REST APIS, la creación del backend, el frontend del sistema web y la instalación del software en una máquina virtual Ubuntu. La parte principal de la propuesta y el valor agregado consiste en la creación de un algoritmo genérico para obtener las recomendaciones, el cual es independiente a cualquier topología de grafo y utiliza la combinación de Louvain y PageRank.

Como trabajo faltante o a futuro se sugiere elaborar un sistema frontend independiente a la topología del grafo, es decir, que se adapte y obtenga en tiempo de ejecución los atributos de los nodos en algunas secciones donde actualmente está fijo a los atributos del grafo de artículos científicos utilizado.

En el siguiente capítulo se hablará de la implementación de los pasos previos de manera específica sobre nuestra fuente de información, los artículos científicos. Además se podrá observar el ejemplo de caso de prueba que fue analizado para evaluar nuestra solución.

5. RESULTADOS Y DISCUSIÓN

El problema a resolver por nuestra propuesta de solución consiste en la incapacidad de encontrar información relevante entre grandes cantidades de documentos para el caso específico de artículos científicos, y es que haciendo mención al estado del arte mostrado en el capítulo 2, solamente hay un sistema de recomendación de artículos pero son con enfoque universitario y es necesario introducir un archivo completo como criterio de búsqueda, por lo que al desarrollar un sistema de recomendación hibrido con la capacidad de mostrarle al usuario sugerencias relacionadas a las búsquedas de sus necesidades en formato tabular y en formato de grafo dinámico, el objetivo general de la solución ha sido cumplido.

En el presente capítulo se presentan los pasos de implementación mencionados en el capítulo anterior pero específicos para lograr la creación de nuestro sistema de recomendación utilizando una base de datos de grafos.

5.1. Resultados

El desarrollo de la propuesta de solución del presente documento se realizó en base al flujo de la figura 28, la cual consiste en la obtención de la información relevante relacionada a artículos científicos mediante técnicas de minería de texto, creando una serie de documentos con formato separado por coma para la importación de la información a la base de datos basada en grafos. Esta base de datos es la fuente de información para el servidor *backend* que se mantiene en constante comunicación con el servidor *frontend* para interactuar con el usuario y ofrecer una serie de recomendaciones que ayude a satisfacer su búsqueda o necesidades.

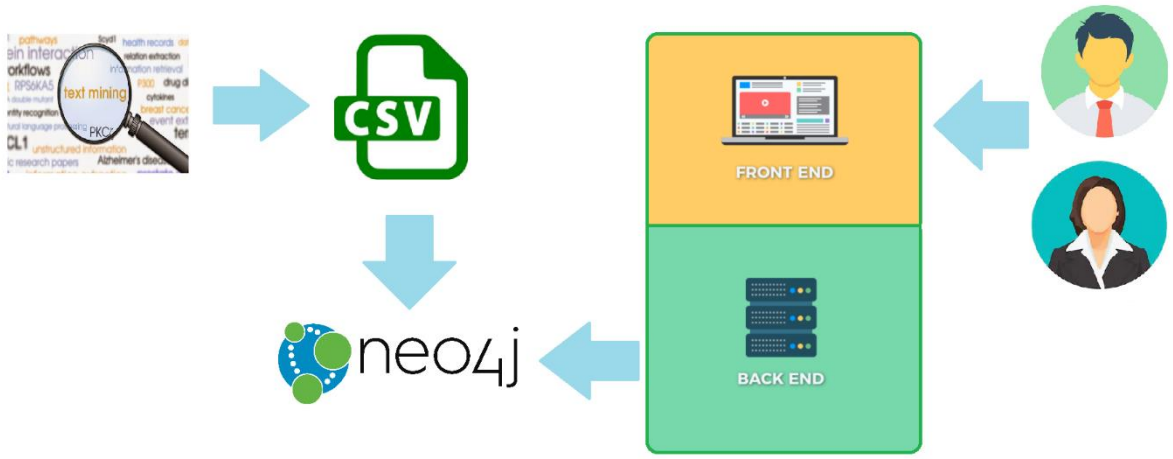


Figura 28 Funcionamiento del sistema de recomendación

5.1.1 Obtención de la información cruda.

Después de hacer un exhaustivo análisis de diferentes *datasets* en SNAP (apéndice A), el que cumple con la información necesaria para resolver nuestro problema de análisis es *"Collaboration network"* de la universidad de Stanford [36]. Este *dataset* consiste en una seria de documentos con extensión abs, donde cada uno de ellos es la representación o resumen de un artículo científico con número de identificación específico.

En la figura 29 se puede observar un ejemplo de estos archivos. Se cuenta con un rango de año de edición entre 1993 y 2003. De cada uno de estos archivos se tiene información relevante para la creación de nuestro grafo que se divide en 3 tipos de entidad: *Papers*, *Journals* y *Authors* con atributos relacionados como título, autores secundarios (coautor), comentarios del articulo y fecha de publicación que serán almacenados en nuestra base de datos de Neo4j.

```
------------------------------------------------------------------------------
\\
Paper: hep-th/9201001
From: zuber@poseidon.saclay.cea.fr (C. Itzykson)
Date: 31 Dec 91 23:54:17 MET 1991 +0100   (37kb)

Title: Combinatorics of the Modular Group II: the Kontsevich integrals
Authors: C. Itzykson and J.-B. Zuber
Comments: 46 pages
Subj-class: High Energy Physics - Theory; Quantum Algebra
Journal-ref: Int.J.Mod.Phys. A7 (1992) 5661-5705
\\
  We study algebraic aspects of Kontsevich integrals
as generating functions for intersection theory over moduli space
and review the derivation of Virasoro and KdV constraints.
 1. Intersection numbers
 2. The Kontsevich integral
  2.1. The main theorem
  2.2 Expansion of Z on characters and Schur functions
  2.3 Proof of the first part of the Theorem
 3. From Grassmannians to KdV
 4. Matrix Airy equation and Virasoro highest weight conditions
 5. Genus expansion
 6. Singular behaviour and Painlev'e equation.
 7. Generalization to higher degree potentials
\\

```

Figura 29 Ejemplo de archivo crudo de información relacionado a un artículo científico

5.1.2 Minería de texto.

De la figura 29 podemos analizar que el contenido de los archivos están perfectamente resumidos a nuestras necesidades, y fue necesaria la aplicación de métodos de minería de extracción de características para colectar específicamente la información necesaria para la creación de nuestro grafo. Hay ciertos encabezados que fueron de utilidad en la integración de expresiones regulares para el objetivo previamente mencionado.

5.1.3 Normalización

En la figura 30 mostramos la representación del documento previo relacionado a un artículo científico después del proceso de normalización, prácticamente eliminamos todas las palabras que no son de utilidad para la creación de nuestras entidades en Neo4j utilizando expresiones regulares también eliminamos algunos signos de puntuación que no son representativos.

```
Paper: hep-th/9201001
From: C. Itzykson
Date: 31 Dec 1991
Title: Combinatorics of the Modular Group II: the Kontsevich integrals
Authors: C. Itzykson and J.-B. Zuber
Comments: 46 pages
Journal-ref: Int.J.Mod.Phys. A7 (1992) 5661-5705
```

Figura 30 Texto después del proceso de normalización

5.1.4 Tokenización

Ahora que teníamos la información resumida, se utilizó la tokenizacion para separar el texto en palabras y así tener más facilidad de exportarlas a un archivo separado por comas que fue interpretado por Neo4j para la creación de la base de datos de grafos.

```
hep-th/9201001
C. Itzykson
31 Dec 1991
Combinatorics of the Modular Group II: the Kontsevich integrals
C. Itzykson and J.-B. Zuber
46 pages
Int.J.Mod.Phys. A7 (1992) 5661-5705
```

Figura 31 Texto después del proceso de tokenización

5.1.5 Creación formato CSV

Posteriormente se hizo la creación de diferentes archivos delimitados por coma de manera automatizada con un *script* de Python, con el objetivo de crearlos en una cantidad pequeña de tiempo cercana a los 60 minutos considerando la gran cantidad de información. Es importante resaltar que estos archivos CSV deben tener un encabezado específico para la correcta interpretación por Neo4j. En la figura 32 se puede observar un ejemplo de archivo CSV generado con el *script* automático.

```
id_paper:ID(PAPER),title_paper,year_paper,month_paper,day_paper,comments_paper,:LABEL
9201001,"Combinatorics of the Modular Group II: the Kontsevich integrals
",1991,12,31,46 pages,PAPER
9201002,"Inomogeneous Quantum Groups as Symmetries of Phonons
",1992,1,2,5 pags. 0 figs,PAPER
9201003,"Intersection Theory, Integrable Hierarchies and Topological Field Theory
",1992,1,2,"73 pages, most figures are not included. Lectures given at the",PAPER
9201004,"The Heterotic Green-Schwarz Superstring on an N=(2,0) Super-Worldsheet
",1992,1,2,33 pages,PAPER
9201005,"Ward Identities in Two-Dimensional String Theory
",1992,1,3,12 pages,PAPER
9201006,"On Symmetries of Some Massless 2D Field Theories
",1992,1,6,13 pages,PAPER
9201007,"Static Domain Walls in N=1 Supergravity
",1992,1,6,29 pages + 11 figures (not included),PAPER
9201008,"Coulomb Gas Representations and Screening Operators of the N=4
  Superconformal Algebras
",1992,1,7,16 pages,PAPER
```

Figura 32 Archivo CSV para la generación de la base de datos

5.1.6 Generación de la base de datos

Neo4j (Apéndice B) tiene un *framework* capaz de importar grandes cantidades de información de archivos separados por coma en una base de datos existente (En la figura 33 se puede observar el comando de CYPHER (Apéndice C) y los argumentos utilizados para este caso de solución especifico), por lo que se creó una base de datos limpia y se ejecutó el *framework neo4j-admin* para la importación de la información, lo que nos da como resultado el esquema que se puede observar en la figura 34 dentro del explorador de Neo4j.

```
bin\neo4j-admin.bat import --nodes import\authors.csv --nodes import\papers.csv --nodes import\journals.csv
--relationships:PUBLISH_INS=import\publish_ins.csv --relationships:WORKS_WITH=import\work_withs.csv
--relationships:FIRST=import\firsts.csv --relationships:COLABORATOR=import\colaborators.csv --ignore-missing-nodes=true
--multiline-fields=true
```

Figura 33 Comando para la importación del dataset

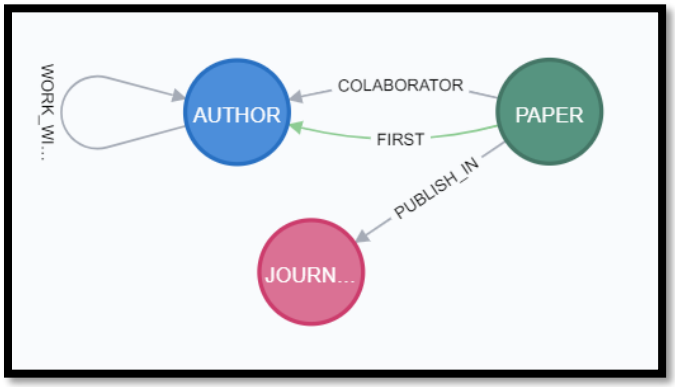


Figura 34 Esquema de la base de datos

5.1.7 Implementación de algoritmos de *clustering* (*Louvain*), relevancia (*PageRank*)

La parte más importante de la presente propuesta fue la elección de los algoritmos a utilizar para generar recomendaciones útiles para el usuario final dependiendo de sus búsquedas. Después de una exploración exhausta de los diferentes tipos de algoritmos enfocados a grafos como búsqueda, centralidad y detección de comunidades se decidió utilizar una metodología hibrida en la cual la búsqueda realizada por el usuario final sería de utilidad para generar comunidades con entidades similares al nodo resultante utilizando el algoritmo de *Louvain*, de esta manera se obtuvo nodos con las mismas características que posteriormente se relacionan con la entidad que el usuario desea buscar para ofrecerle una lista de sugerencias, donde aplicamos el algoritmo de centralidad *PageRank* para ordenar los nodos por jerarquía de importancia dentro de la topología y ese fue el resultado final de recomendación.

El algoritmo de *Louvain* está basado en la modularidad de los datos. Este realiza una evaluación del conjunto de datos y compara la densidad de aristas que están presentes dentro o fuera de la comunidad. Al optimizar este valor de iteración se obtiene un estimado de agrupación de los nodos que pertenecen a una red. En la figura 35 se puede observar un ejemplo breve del funcionamiento del algoritmo de *Louvain* [37].

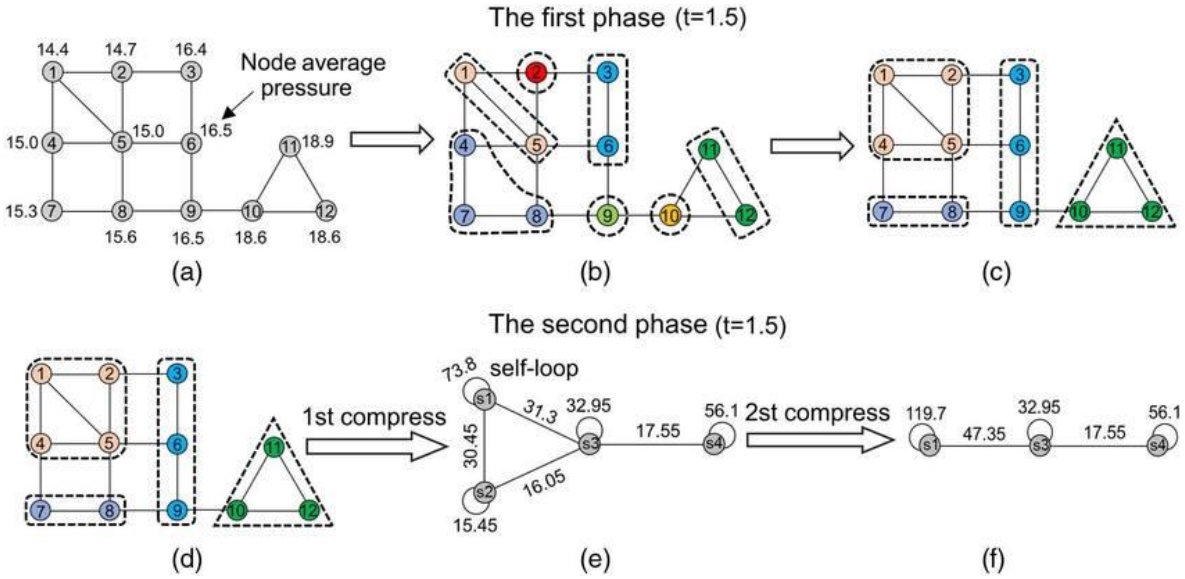


Figura 35 Proceso algoritmo de Louvain [37].

Ahora que se obtienen los nodos categorizados por comunidad, se obtuvieron todas las entidades similares a la búsqueda inicial del usuario final y se ordenaron con el algoritmo de *PageRank* para obtener los más importantes dentro de nuestra topología. En la figura 36 se puede observar su funcionamiento interno, que implica el analizar el número de aristas entrantes y salientes de cada uno de los nodos, asignándoles una ponderación hasta que converge el proceso, deja de haber cambios y obtienes una ponderación final con

los nodos más importantes de la red de tu comunidad. Para la aplicación de estos algoritmos se utilizaron las funciones de la librería APOC (Apéndice D).

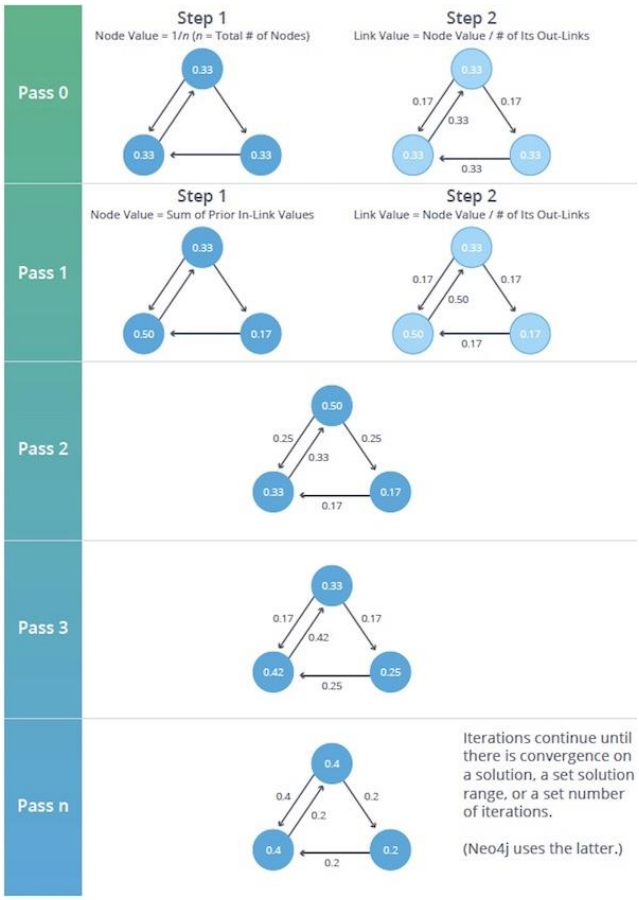


Figura 36 Pasos para implementar PageRank [21].

5.1.8 Creación de servidor *backend* para atender las peticiones de los *queries*.

El código que se ejecuta dentro del servidor es el encargado de atender las peticiones de los usuarios en el sistema web, de tal manera que obtiene acceso a los datos de Neo4j, aplica los algoritmos previamente mencionados y regresa una respuesta en formato json. Se utilizó el lenguaje *javascript* junto con el *framework* de *nodejs* (Apéndice E) para la creación del servidor *backend*, utilizando como base la programación orientada a objetos y el *driver* principal de neo4j para la creación de nuestra *rest api.*

5.1.9 Desarrollo de la interfaz gráfica UI

En todo momento se diseñó la interfaz gráfica para ser amigable con el usuario, con el objetivo principal de obtener las recomendaciones con pocos *clicks* y navegación simplificada. En la figura 37 se puede observar la página principal del sistema web, donde tenemos del lado izquierdo un menú lateral para seleccionar el tipo de entidad a encontrar, ya se artículos científicos, autores o journals. En la parte central se visualiza la tabla inicial de búsqueda donde se muestran todos los resultados de artículos científicos sin ningún tipo de filtro, para que el usuario pueda ver todas las opciones registradas del sistema. En la parte superior se encuentra un carrusel de imágenes que es representativo de la escuela ITESO y un menú de encabezado derecho para cerrar la aplicación.

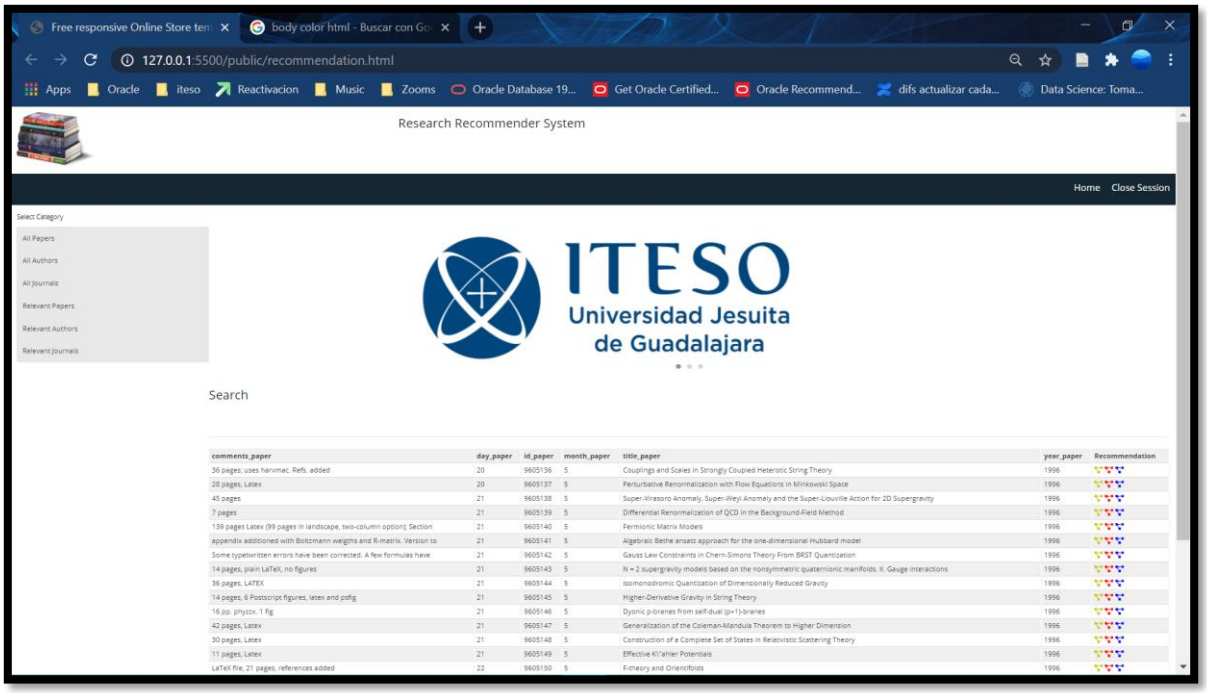
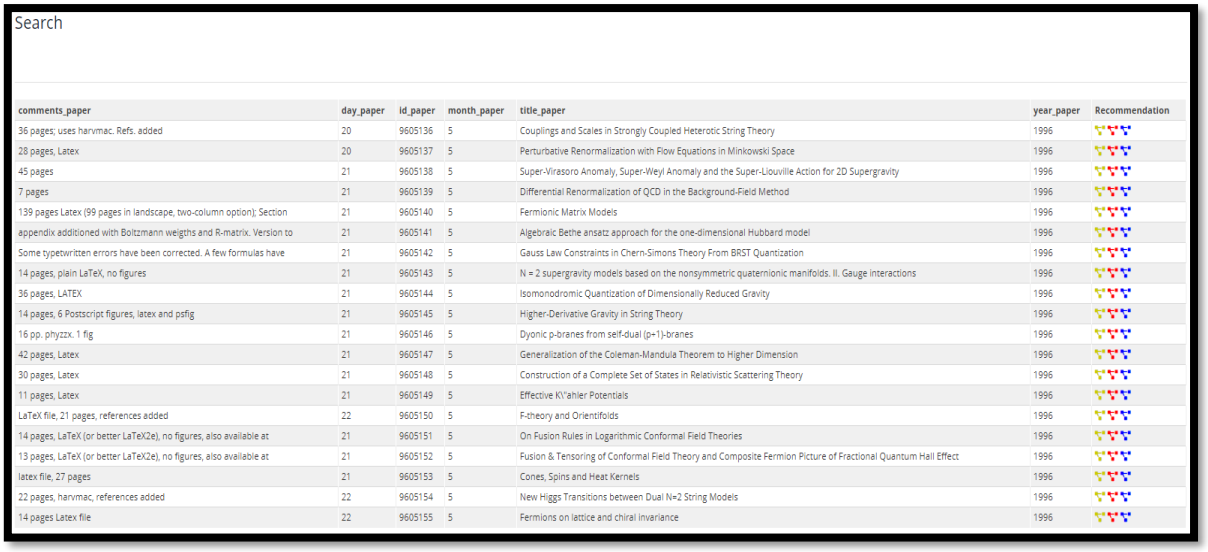


Figura 37 Página principal de la solución propuesta

En la figura 38 se puede observar los diferentes tipos de filtro que tenemos para la entidad de artículos científicos dentro de nuestro grafo, hicimos posible la creación de un filtro especializado en el cual se puede especificar el id del artículo, titulo, comentarios relacionados, un rango de fecha de edición o búsqueda por *keywords* para encontrar rápidamente aquellos artículos que esté buscando el usuario final.

Figura 38 Filtros avanzados de búsqueda de artículos científicos

Una vez que se hayan encontrado los artículos, autores o *journals* especificados por el usuario en su búsqueda, se llenara la tabla principal de la aplicación web con toda la información detallada de cada tipo de entidad, y en la última columna se encuentran 3 tipos diferentes de botón, los cuales deben ser seleccionados para obtener una serie de sugerencias realizadas con nuestro algoritmo interno.



comments_paper	day_paper	id_paper	month_paper	title_paper	year_paper	Recommendation
36 pages; uses harvmac. Refs. added	20	9605136	5	Couplings and Scales in Strongly Coupled Heterotic String Theory	1996	
28 pages, Latex	20	9605137	5	Perturbative Renormalization with Flow Equations in Minkowski Space	1996	
45 pages	21	9605138	5	Super-Virasoro Anomaly, Super-Weyl Anomaly and the Super-Liouville Action for 2D Supergravity	1996	
7 pages	21	9605139	5	Differential Renormalization of QCD in the Background-Field Method	1996	
139 pages Latex (99 pages in landscape, two-column option); Section	21	9605140	5	Fermionic Matrix Models	1996	
appendix additioned with Boltzmann weigths and R-matrix. Version to	21	9605141	5	Algebraic Bethe ansatz approach for the one-dimensional Hubbard model	1996	
Some typewritten errors have been corrected. A few formulas have	21	9605142	5	Gauss Law Constraints in Chern-Simons Theory From BRST Quantization	1996	
14 pages, plain LaTeX, no figures	21	9605143	5	N = 2 supergravity models based on the nonsymmetric quaternionic manifolds. II. Gauge interactions	1996	
36 pages, LATEX	21	9605144	5	Isomonodromic Quantization of Dimensionally Reduced Gravity	1996	
14 pages, 6 Postscript figures, latex and psfig	21	9605145	5	Higher-Derivative Gravity in String Theory	1996	
16 pp. phyzzx. 1 fig	21	9605146	5	Dyonic p-branes from self-dual (p+1)-branes	1996	
42 pages, Latex	21	9605147	5	Generalization of the Coleman-Mandula Theorem to Higher Dimension	1996	
30 pages, Latex	21	9605148	5	Construction of a Complete Set of States in Relativistic Scattering Theory	1996	
11 pages, Latex	21	9605149	5	Effective K\"ahler Potentials	1996	
LaTeX file, 21 pages, references added	22	9605150	5	F-theory and Orientifolds	1996	
14 pages, LaTeX (or better LaTeX2e), no figures, also available at	21	9605151	5	On Fusion Rules in Logarithmic Conformal Field Theories	1996	
13 pages, LaTeX (or better LaTeX2e), no figures, also available at	21	9605152	5	Fusion & Tensoring of Conformal Field Theory and Composite Fermion Picture of Fractional Quantum Hall Effect	1996	
latex file, 27 pages	21	9605153	5	Cones, Spins and Heat Kernels	1996	
22 pages, harvmac, references added	22	9605154	5	New Higgs Transitions between Dual N=2 String Models	1996	
14 pages Latex file	22	9605155	5	Fermions on lattice and chiral invariance	1996	

Figura 39 Tabla principal de resultados encontrados

En la figura 39 se visualiza los botones mencionados, donde cada color está representado por una entidad diferente del grafo (*Papers* – amarillo), (*Journals* - rojo) y (*Authors* - azul) y solamente se seleccionan con un *click.*

En el menú lateral de la figura 40, añadimos opciones específicas para que el usuario pueda ver en su tabla principal de resultados todos los nodos por cada tipo de entidad, que serían las 3 primeras opciones. Así mismo en las siguientes 3 opciones aplicamos el algoritmo de centralidad *PageRank* para hacer una ordenación previa de los resultados de búsqueda de la tabla principal de tal manera que obtenemos en orden descendente de importancia el detalle de cada nodo.

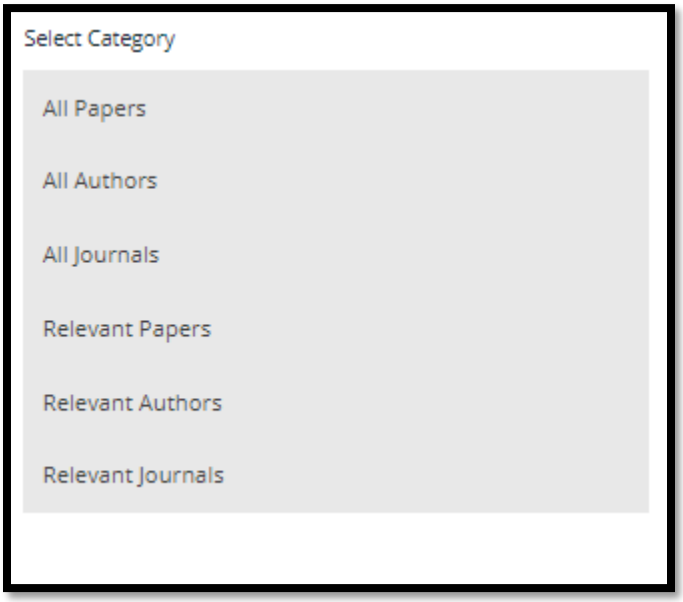


Figura 40 Menú lateral de filtros

En la figura 41 se visualiza un ejemplo de la tabla secundaria con recomendación de autores seleccionados por el usuario. Así mismo, una vez que se consiguen las sugerencias, existe la opción de representarlas mediante un sub-grafo de la base de datos a través de una nueva ventana emergida gracias a oprimir el botón rojo '*Show network recommendation graph*'.

Figura 41 Tabla secundaria con recomendaciones finales

Para finalizar esta sección, en la figura 42 está representada la página web completa, con los filtros de búsqueda, la tabla principal de resultados, la tabla secundaria de sugerencias y el menú lateral de opciones.

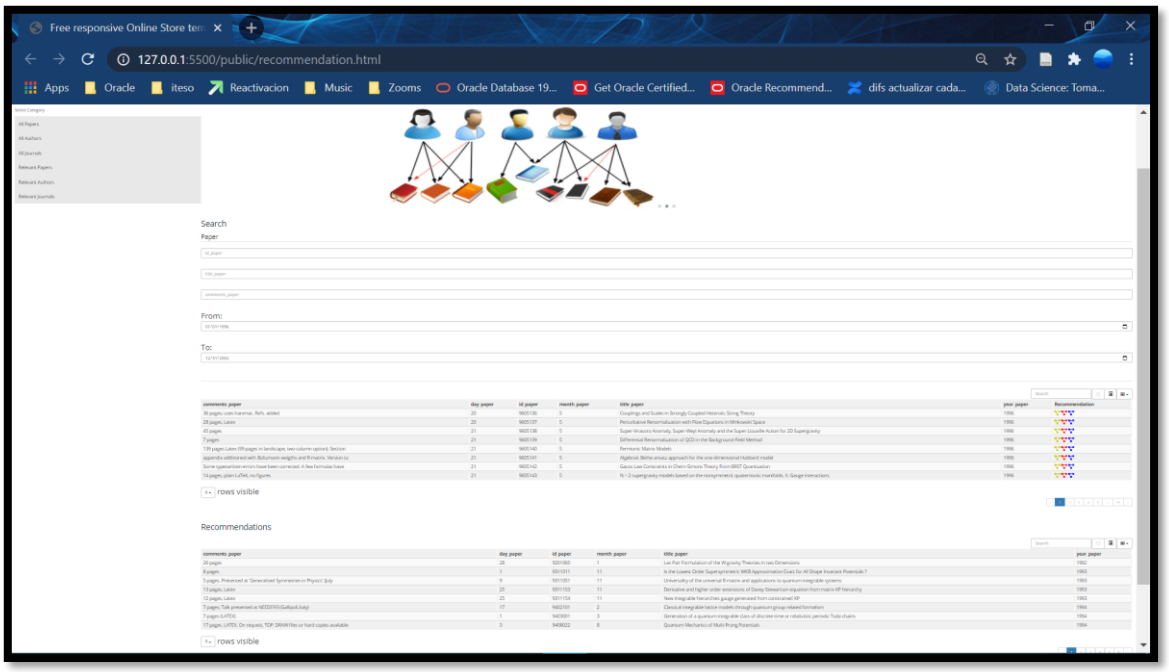


Figura 42 Visualización de la página web completa

5.1.10 Conexión entre *Frontend* y *Backend* (Interacción con el usuario final)

La manera en la que se comunican ambos servidores es a través de puertos específicos y datos en formato JSON con los resultados de las recomendaciones obtenidas desde la base de datos. Es importante mencionar que se utilizaron funciones asíncronas para la colección de la información que posteriormente fue representada en el sistema web.

5.1.11 Visualización de las recomendaciones (sub-grafo) usando Neovis.js

Para representar el resultado de las recomendaciones, se utilizó la librería Neovis.js (Apéndice F) la cual tiene como característica principal el conectarse directamente a la base de datos, solamente especificando el usuario y la contraseña de la misma para obtener acceso. Así mismo para la configuración de Neovis.js se creó un método el cual contiene el nombre de las entidades a mostrar (*Papers*, *Authors*, *Journals*) y el nombre de las relaciones que hay entre ellos. En la figura 43 se muestra un ejemplo de sub-grafo de recomendaciones.

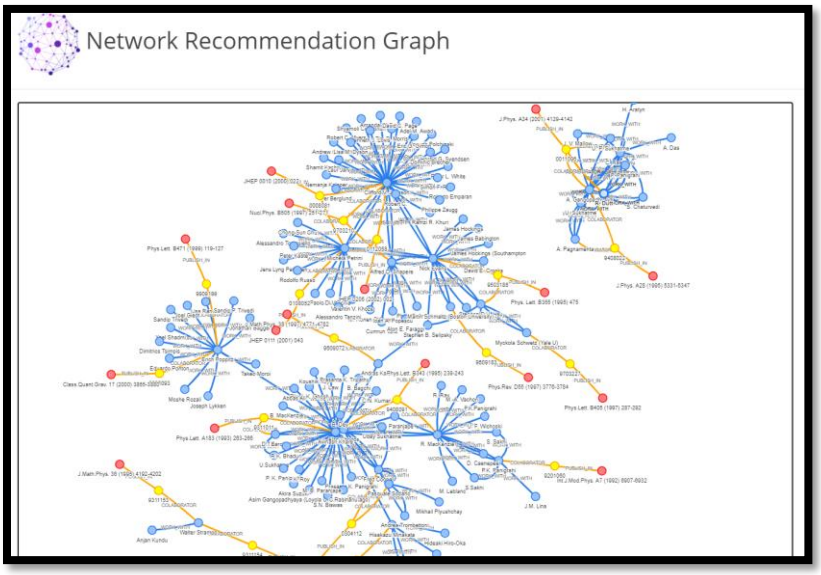


Figura 43 Sub-grafos de recomendación generados por el sistema web

5.1.12 Ejemplo de caso de uso.

Una vez que hemos comprendido las partes fundamentales que componen la presente solución, se muestra un ejemplo de uso específico de recomendación para un artículo científico. El primer paso se muestra en la figura 44, donde se realiza una búsqueda inicial de artículos que tengan la palabra clave *graph* dentro de su título y su fecha de publicación este entre 1997 y 1998, teniendo como resultado 2 artículos.

Figura 44 Búsqueda inicial de artículos comprendidos entre 1997 y 1998 que contengan la palabra clave *graph* en su titulo

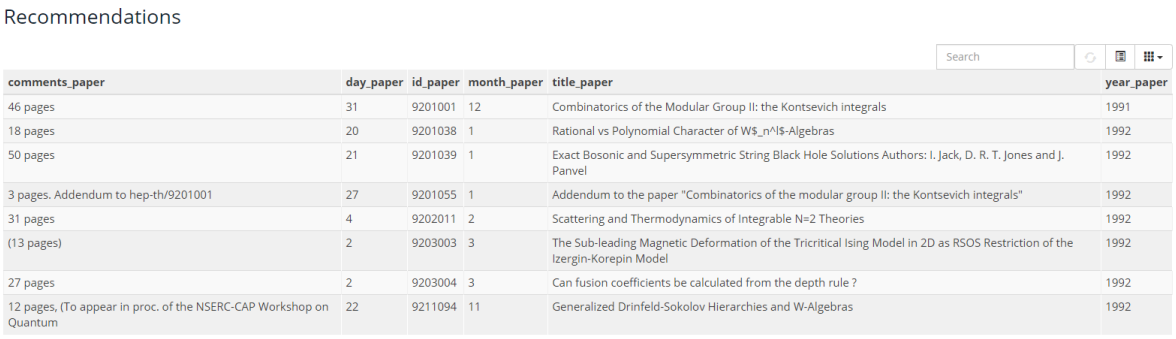
Ahora se puede observar en la figura 45 los resultados de la búsqueda previa, y en la última columna que contiene 3 diferentes tipos de botón, se puede elegir la entidad a ser recomendada por nuestros resultados de búsqueda. El botón amarillo es para obtener sugerencias de artículos relacionados al articulo seleccionado, el botón rojo para recomendación de journals y el azul para obtener sugerencias de autores.


Search

comments_paper	day_paper	id_paper	month_paper	title_paper	year_paper	Recommendation
Plain Tex, 7 pages; 2 figures. Talk given at the 21st International	21	9701103	1	Conformal field theory and graphs	1997	
21 pages, LaTeX2e, requires feynmf package to draw Feynman graphs	16	9805098	5	On Kreimer's Hopf algebra structure of Feynman graphs	1998	

Figura 45 Resultados del filtro de búsqueda

Una vez que se selecciona el *graph-button* de color amarillo representativo de los *papers*, aparecerá en la tabla secundaria de la parte inferior, todos los artículos científicos recomendados por nuestro algoritmo interno y que podrán ser visualizados en una nueva ventana. En la figura 46 se muestra el resultado de la sugerencia.

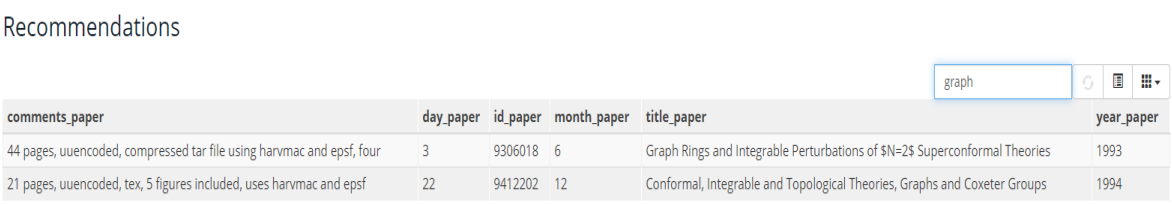

Recommendations

Search

comments_paper	day_paper	id_paper	month_paper	title_paper	year_paper
46 pages	31	9201001	12	Combinatorics of the Modular Group II: the Kontsevich integrals	1991
18 pages	20	9201038	1	Rational vs Polynomial Character of W$_n^l$-Algebras	1992
50 pages	21	9201039	1	Exact Bosonic and Supersymmetric String Black Hole Solutions Authors: I. Jack, D. R. T. Jones and J. Panvel	1992
3 pages. Addendum to hep-th/9201001	27	9201055	1	Addendum to the paper "Combinatorics of the modular group II: the Kontsevich integrals"	1992
31 pages	4	9202011	2	Scattering and Thermodynamics of Integrable N=2 Theories	1992
(13 pages)	2	9203003	3	The Sub-leading Magnetic Deformation of the Tricritical Ising Model in 2D as RSOS Restriction of the Izergin-Korepin Model	1992
27 pages	2	9203004	3	Can fusion coefficients be calculated from the depth rule ?	1992
12 pages, (To appear in proc. of the NSERC-CAP Workshop on Quantum	22	9211094	11	Generalized Drinfeld-Sokolov Hierarchies and W-Algebras	1992

Figura 46 Recomendaciones finales del artículo seleccionado

Ahora que se obtienen las recomendaciones de la tabla secundaria, es posible hacer búsquedas más específicas dentro de este sub-segmento en la parte superior derecha de la tabla, dicha búsqueda se hace por palabras clave como en el paso inicial sobre todo el segmento de información registrada en la base de datos. En la figura 47 se puede observar un ejemplo de aplicación.


Recommendations

graph

comments_paper	day_paper	id_paper	month_paper	title_paper	year_paper
44 pages, uuencoded, compressed tar file using harvmac and epsf, four	3	9306018	6	Graph Rings and Integrable Perturbations of $N=2$ Superconformal Theories	1993
21 pages, uuencoded, tex, 5 figures included, uses harvmac and epsf	22	9412202	12	Conformal, Integrable and Topological Theories, Graphs and Coxeter Groups	1994

Figura 47 Búsqueda por palabras clave en tabla secundaria de recomendaciones

Por último se visualiza en la figura 48 la ventana emergente que se puede lanzar conteniendo el sub-grafo resultante relacionado a la sugerencia final, donde cada tipo de entidad está representada por un color

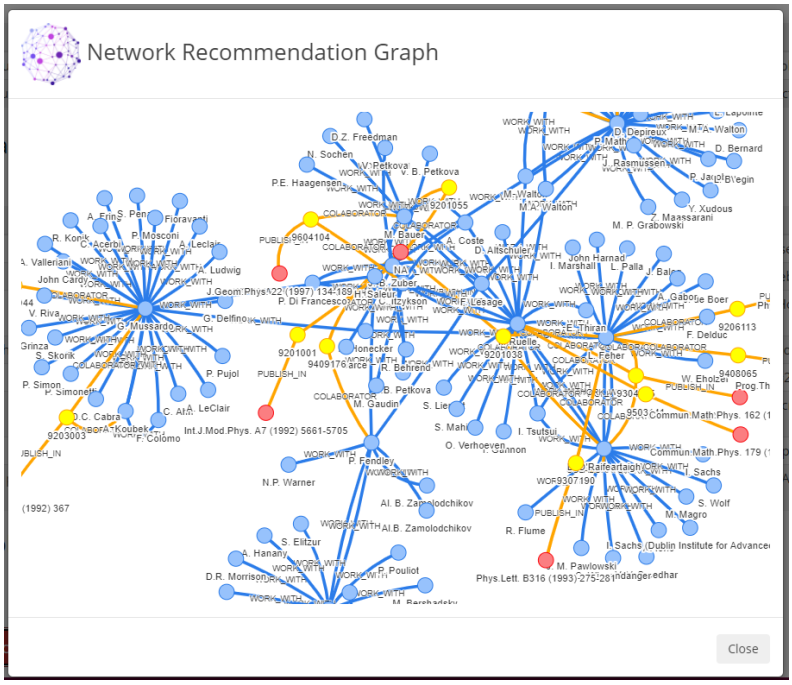


Figura 48 Sub-grafo resultante de recomendación del ejemplo de caso de uso

5.2. Discusión

Los resultados de la sección previa indican que la implementación de un sistema de recomendación basado en bases de datos de grafos, algoritmos de centralidad y detección de comunidades es factible para obtener sugerencias acordes a las necesidades de los usuarios finales, con la principal ventaja de visualizar una red de sugerencias colaborativas en la representación del sub-grafo resultante.

Con el modelo orientado a grafos se puede almacenar cualquier tipo de información de manera natural y en tiempo real debido a que la topología es muy similar a la del pensamiento humano. Una base de datos orientada a grafos tiene una presentación intuitiva y resumida de las relaciones con estructuras flexibles y agiles.

Como trabajo complementario al presente proyecto se sugiere la implementación de sistemas de recomendación genéricos basado en sumarizaciones lo cual puede facilitar aún más el análisis y permite realizar acciones de manera interactiva en super-nodos, analizando así patrones o tendencias en cantidades más grandes de información, hablando más en términos de escalabilidad de este tipo de sistemas de recomendación basados en grafos.

6. CONCLUSIONES

6.1. Conclusiones

Con la elaboración de este proyecto se demostró que la implementación de un sistema de recomendación hibrido utilizando una base de datos de grafos es eficiente independientemente del tema analizado, que en este caso fue orientado a artículos científicos y sus entidades relacionadas (Autores y Journals).

Los objetivos que se cumplieron con nuestra propuesta de solución son.

- **Desarrollar un sistema de recomendación basado en grafos para el escenario de artículos científicos.**

Fue el objetivo general del objeto de estudio. Se desarrolló un sistema web donde usuario final puede hacer búsquedas específicas por palabras clave de sus necesidades en 3 diferentes tipos de entidad *(Papers, Journals, Authors)* a través de las propiedades de cada uno de ellos como nombre, titulo, año de publicación, y comentarios. Internamente el sistema web interpretaba las búsquedas del usuario y hacia uso del algoritmo interno que se elaboró para la solución, donde se analiza el contenido de cada artículo científico, se agrupa por similaridad con otros que posteriormente son recomendados.

Con el modelo orientado a grafos que se utilizó en la presente solución, se puede almacenar diferentes tipos de información de forma natural, ya que la estructura utilizada es muy similar al pensamiento humano. Por lo mismo decidimos mostrar un sub-grafo interactivo con las sugerencias resultantes de tal manera que se obtuvieran más recomendaciones colaborativas indirectas a las recomendaciones principales.

- **Construir o adquirir los *datasets* necesarios para implementar un sistema de recomendación de artículos científicos.**

Para cumplir con este objetivo específico fue necesaria la implementación de técnicas de procesamiento de lenguaje natural para interpretar y clasificar la información relevante de los archivos crudos, la parte más difícil de la solución. Dentro de estas técnicas se encuentran la normalización y la tokenizacion; Procesos encargados de eliminar palabras sin contexto de nuestra información y separar por unidades más pequeñas los enunciados para una mejor interpretación y utilización al momento de importarlo a la base de datos de grafos.

- **Implementar los *datasets* en una base de datos basada en grafos (NEO4J).**

En esta fase se realizó un diseño previo de la topología resultante, y nos enfrentamos a algunos problemas al momento de hacer la importación a Neo4j por la gran cantidad de información que se estaba tratando.

Después de más investigación, se optó por hacer s*plit* de la información cada entidad (nodo) en dos diferentes CSV de manera que se especificaron los encabezados necesarios para la interpretación de las aristas del grafo y mediante la herramienta de línea de comandos de neo4j-admin fue exitosa la carga completa de información en cuestión de unos cuantos minutos.

- **Realizar las consultas necesarias para la generación de sugerencias.**

Dentro de nuestro análisis, se estudiaron diferentes tipos de algoritmos proporcionados en librerías de Neo4j como recorrido de grafos, generación de comunidades y evaluación de relevancia dentro de la topología del grafo. Fue la fase más exhaustiva de investigación de nuestro proyecto debido a que teníamos que encontrar la combinación de algoritmos que nos entregaran recomendaciones más acordes a las muestras de prueba utilizadas de artículos científicos. Terminamos optando por un algoritmo de clasificación no supervisada basada en el cálculo de la modularidad de diferentes sub conjuntos de grafos de nuestra búsqueda como es el caso de *Louvain* y una vez seleccionada la comunidad objetivo, recomendamos las entidades dentro del mismo, ordenadas jerárquicamente con la ponderación ofrecida por el algoritmo de *PageRank*. Pudimos observar variaciones en los sub-grafos de recomendación resultantes en cuanto a tamaño, cantidad de nodos y aristas pero en todas aparecieron nuestros resultados de búsqueda directos y sugerencias colaborativas indirectas, lo que nos demostraba una buena precisión en las recomendaciones.

- **Desarrollar la interfaz gráfica amigable con el usuario.**

Para el desarrollo y cumplimiento de este objetivo específico, se utilizaron las tecnologías html, css y javascript para ofrecerle al usuario final un sistema web dinámico con el que pudiera interactuar, así mismo se evaluaron diferentes estructuras de página antes de decidir el final de tal manera que con unos cuantos clicks, el usuario pueda tener sugerencias de la búsqueda de sus necesidades.

6.2. Trabajo Futuro

Se propone el análisis, desarrollo e implementación de sistemas de recomendación utilizando algoritmos de sumarizacion de grafos de tal manera que sea posible el incremento de información en tiempo real con ambientes de *big data*. Con el desarrollo de la presente solución se crearon *drivers* de utilidad para la implementación de recomendadores sin importar el tema de estudio, es decir, independiente a sus topologías.

Así mismo, se observa factible el estudio de otros tipos de recomendadores híbridos contemplando los tipos existentes (contenido, ítems, popularidad, colaborativo) con diferentes algoritmos de generación de comunidades o jerarquización de nodos incluso con otro gestor de base de datos diferente a Neo4j para hacer una comparación de rendimiento entre los mismos.

BIBLIOGRAFÍA

[1]. R. R. Carrillo, "*Revisión, selección e implementación de un algoritmo de recomendación de material bibliográfico utilizando tecnología j2ee*", B.S. Thesis, Universidad del BIO-BIO, Chile, 2008.

[2]. E.H. Viedma. 2004. "*Sistemas de recomendaciones: herramientas para el filtrado de información en Internet*". [Hipertext Online]. Available: http://www.upf.edu/hipertextnet/numero-2/recomendacion.html. [Accessed Jan. 16, 2021.]

[3]. 2019. "*Sistemas de recomendación | Qué son, tipos y ejemplos*," *GraphEverywhere*. [Online]. Available: https://www.grapheverywhere.com/sistemas-de-recomendacion-que-son-tipos-y-ejemplos/. [Accessed Jan. 16, 2021].

[4]. A. Grasso. 2004. "*Recommender system and method - Xerox Corporation*." [Online]. Available: https://www.freepatentsonline.com/y2004/0254911.html. [Accessed Jan. 16, 2021].

[5]. S. Chancellor. 2021. "*Recommendations with MovieLens.*" [Online]. Available: https://movielens.org/ [Accessed Jan. 16, 2021].

[6]. B. Gipp, J. Beel, and C. Hentschel, "*Scienstein: A Research Paper Recommender System*". Fraunhofer Institute for Telecommunications. Berlin Germany. 2009.

[7]. R. Burke, "*Hybrid Recommender Systems: Survey and Experiments*," *User Model. User-Adapt. Interact*, vol. 12, Nov. 2002,

[8]. R. Vences-Nava, V. H. Menéndez-Domínguez, and S. Medina-Peralta, "*Evaluating a hybrid recommender system of theses,*" *Ing. Investig. Tecnol.*, vol. 20, no. 3, Sep. 2019.

[9]. S. Galán, "*Filtrado Colaborativo y Sistemas de Recomendación*". B.S. Thesis. Universidad Calos III de Madrid. Madrid, España. 2007.

[10]. L. Terveen, W. Hill, B. Amento, D. McDonald, and J. Creter, "*PHOAKS: A system for sharing recommendations*," *Commun ACM*, vol. 40, pp. 59–62, Mar. 1997, doi: 10.1145/245108.245122.

[11]. "*Referral Web: combining social networks and collaborative filtering: Communications of the ACM: Vol 40, No 3.*" [Online]. Available: https://dl.acm.org/doi/abs/10.1145/245108.245123 [Accessed Jan. 24, 2021].

[12]. K. Labille, S. Gauch, and A. Smittu, "*Conceptual Impact-Based Recommender System for CiteSeer x,*" Sep. 2015, vol. 1448.

[13]. P. Larasatie and S. Setiowati, "*From Fingerprint to Footprint: Using Point of Interest (POI) Recommendation System in Marketing Applications,*" vol. 12, pp. 118–131, Aug. 2019, doi: 10.12695/ajtm.2019.12.2.4.

[14]. Ms.Uma.N, "*A technical review on recommendation systems in a framework of job recommender system using naive bayes algorithm,*" *Glob. J. Eng. Sci. Res.*, vol. 5, no. 6, pp. 253–257, Jun. 2018, doi: 10.5281/zenodo.1302263.

[15]. A. Pujahari and V. Padmanabhan, "*Group Recommender Systems: Combining User-User and Item-Item Collaborative Filtering Techniques,*" in *2015 International Conference on Information Technology (ICIT)*, Dec. 2015, pp. 148–152, doi: 10.1109/ICIT.2015.36.

[16]. X. Bai, M. Wang, I. Lee, Z. Yang, X. Kong, and F. Xia, "*Scientific Paper Recommendation: A Survey*," *IEEE Access*, vol. 7, pp. 9324–9339, 2019, doi: 10.1109/ACCESS.2018.2890388.

[17]. A. Mishra and S. Vishwakarma, "*Analysis of TF-IDF Model and its Variant for Document Retrieval*," in *2015 International Conference on Computational Intelligence and Communication Networks (CICN)*, Dec. 2015, pp. 772–776, doi: 10.1109/CICN.2015.157.

[18]. Y. Wang, Z. Hong, and M. Shi, "*Research on LDA Model Algorithm of News-oriented Web Crawler*," in *2018 IEEE/ACIS 17th International Conference on Computer and Information Science (ICIS)*, Jun. 2018, pp. 748–753, doi: 10.1109/ICIS.2018.8466502.

[19]. Y. Park, S. Park, S. Lee, and W. Jung, "*Fast Collaborative Filtering with a k-nearest neighbor graph*," in *2014 International Conference on Big Data and Smart Computing (BIGCOMP)*, Jan. 2014, pp. 92–95, doi: 10.1109/BIGCOMP.2014.6741414.

[20]. I. Popescu, K. Portelli, C. Anagnostopoulos, and N. Ntarmos, "*The case for graph-based recommendations*," in *2017 IEEE International Conference on Big Data (Big Data)*, Dec. 2017, pp. 4819–4821, doi: 10.1109/BigData.2017.8258553.

[21]. D. Le, "*Random walk with restart: A powerful network propagation algorithm in Bioinformatics field*," in *2017 4th NAFOSTED Conference on Information and Computer Science*, Nov. 2017, pp. 242–247, doi: 10.1109/NAFOSTED.2017.8108071.

[22]. A. Wijonarko, D. Nurjanah, and D. S. Kusumo, "*Hybrid recommender system using random walk with restart for social tagging system*," in *2017 International Conference on Data and Software Engineering (ICoDSE)*, Nov. 2017, pp. 1–6, doi: 10.1109/ICODSE.2017.8285875.

[23]. A. Tharwat, "*Classification assessment methods*," *Appl. Comput. Inform.* vol. 17 No. 1, pp. 168-192. Aug. 2018, doi: 10.1016/j.aci.2018.08.003.

[24]. W. K. Hauger and M. S. Olivier, "*NoSQL databases: forensic attribution implications*," *SAIEE Afr. Res. J.*, vol. 109, no. 2, pp. 119–132, Jun. 2018.

[25]. "*Bases de datos no relacionales | Bases de datos de gráficos | AWS*," *Amazon Web Services, Inc* [Online]. Available: https://aws.amazon.com/es/nosql/, [Accessed Oct. 26, 2019].

[26]. "*Neo4j Graph Platform – The Leader in Graph Databases.*" [Online]. Available: https://neo4j.com/. [Accessed Feb. 07, 2021].

[27]. W. Lyon. "*Graph Visualization With Neo4j Using Neovis.js*". [Online]. Available: *https://medium.com/neo4j/graph-visualization-with-neo4j-using-neovis-js-a2ecaaa7c379* Neo4j Contrib, 2020. [Accessed Jan. 2021].

[28]. M. Needham. O. Media, "*Graph Algorithms*." 1st ed. Gravenstein Highway North, Sebastopol: O'Reilly Media. 2019.

[29]. E. Roberts and K. Schroeder, "*The Google PageRank Algorithm,*" B.S. Thesis. Baruch, New York. p. 3. 2016.

[30]. E. Botta-Ferret and J. E. Cabrera-Gato, "*Minería de textos: una herramienta útil para mejorar la gestión del bibliotecario en el entorno digital,*" *ACIMED*, vol. 16, no. 4, pp. 1-12, Oct. 2007.

[31]. 2020. "*What is Tokenization | Tokenization In NLP*," *Analytics Vidhya* [Online]. Available: https://www.analyticsvidhya.com/blog/2020/05/what-is-tokenization-nlp. [Accessed Jan. 17, 2021].

[32]. "*Text Normalization. Why, what and how. | by Tiago Duque | Towards Data Science.*" [Online]. Available: https://towardsdatascience.com/text-normalization-7ecc8e084e31. [Accessed Jan. 17, 2021].

[33]. A. Javier, Sep. 2020. "*¿Que es la programación web?,*" *Open Analytics - SEO y Programación*, [Online]. Available: https://openanalytics.es/programacion-web. [Accessed Jan. 17, 2021].

[34]. H. M. Abdullah and A. M. Zeki, "*Frontend and Backend Web Technologies in Social Networking Sites: Facebook as an Example*," in *2014 3rd International Conference on Advanced Computer Science Applications and Technologies*, Dec. 2014, pp. 85–89, doi: 10.1109/ACSAT.2014.22.

[35]. Nov 10, 2019. "*JSON: ¿Qué es y para qué sirve?,*" *NextU LATAM*, [Online]. Available: https://www.nextu.com/blog/que-es-json. [Accessed Jan. 17, 2021].

[36]. "*SNAP: Network datasets: Astro Physics collaboration network*." [Online]. Available: https://snap.stanford.edu/data/ca-AstroPh.html (accessed Feb. 06, 2021).

[37]. "*Algoritmo de Louvain | Algoritmos de detección de comunidades*." [Online]. Available: https://www.grapheverywhere.com/algoritmo-de-louvain. [Accessed Feb. 07, 2021].

[38]. "*SNAP: Stanford Network Analysis Project.*" [Online]. Available: http://snap.stanford.edu/ [Accessed Feb. 20, 2021].

[39]. "*SNAP: Network datasets: Condense Matter collaboration network*." [Online]. Available: https://snap.stanford.edu/data/ca-CondMat.html. [Accessed Feb. 20, 2021].

[40]. "*Cypher Query Language - Developer Guides*," [Online]. Available: https://neo4j.com/developer/cypher. [Accessed Feb. 20, 2021].

[41]. "*Neo4j APOC Library - Developer Guides,*" [Online]. Available: https://neo4j.com/developer/neo4j-apoc. [Accessed Mar. 3, 2021].

[42]. Node.js, "Node.js," *Node.js*. [Online]. Available: https://nodejs.org/es. [Accessed Feb. 20, 2021].

APÉNDICE A. *Stanford Network Analysis Platform Dataset*

Stanford Network Analysis Platform (SNAP) es una biblioteca de análisis de redes y minería de gráficos de uso general. Está escrito en C ++ y se escala fácilmente a redes masivas con cientos de millones de nodos y miles de millones de bordes. Manipula de manera eficiente gráficos grandes, calcula propiedades estructurales, genera gráficos regulares y aleatorios y admite atributos en nodos y bordes. SNAP también está disponible a través de NodeXL, que es un *front-end* gráfico que integra el análisis de red en *Microsoft Office* y Excel [38].

Dentro de la elaboración del presente proyecto, se utilizó como fuente de información la red colaborativa de artículos científicos, que está compuesta por nodos y relaciones de diferentes tópicos, entre los cuales se encuentran:

- La red de colaboración Arxiv COND-MAT (*Condense Matter Physics*) es de e-print arXiv y cubre las colaboraciones científicas entre los artículos de los autores enviados a la categoría Condense Matter. Si un autor i es coautor de un artículo con el autor j, el gráfico contiene un borde no dirigido de i a j. Si el artículo es coautor de k autores, esto genera un (sub) grafo completamente conectado en k nodos. Los datos cubren artículos en el período de enero de 1993 a abril de 2003 (124 meses). Comienza a los pocos meses del inicio de arXiv y, por lo tanto, representa esencialmente la historia completa de su sección COND-MAT [39].
- La red de colaboración Arxiv GR-QC (Relatividad General y Cosmología Cuántica) es de e-print arXiv y cubre las colaboraciones científicas entre los artículos de los autores enviados a la categoría Relatividad General y Cosmología Cuántica.
- La red de colaboración Arxiv HEP-PH (Física de altas energías - Fenomenología) es de e-print arXiv y cubre las colaboraciones científicas entre los artículos de los autores presentados en la categoría Física de altas energías - Fenomenología.
- La red de colaboración Arxiv HEP-TH (High Energy Physics - Theory) es de e-print arXiv y cubre las colaboraciones científicas entre los artículos de los autores presentados en la categoría High Energy Physics - Theory.

En la figura 49 se muestran las estadísticas generales del grafo implementado con el dataset.

Dataset statistics	
Nodes	69678
Edges	146450
Nodes in largest WCC	17903 (0.954)
Edges in largest WCC	197031 (0.995)
Nodes in largest SCC	17903 (0.954)
Edges in largest SCC	197031 (0.995)
Average clustering coefficient	0.6306
Number of triangles	1351441
Fraction of closed triangles	0.1345
Diameter (longest shortest path)	14
90-percentile effective diameter	5

Figura 49 Estadísticas del grafo implementado [39].

APÉNDICE B. *Neo4j*

Neo4j es una base de datos de grafos nativa NoSQL de código abierto que proporciona un backend transaccional compatible con ACID para sus aplicaciones. El desarrollo inicial comenzó en 2003, pero ha estado disponible públicamente desde 2007. El código fuente, escrito en Java y Scala, está disponible de forma gratuita en GitHub o como descarga de una aplicación de escritorio fácil de usar. Neo4j tiene una edición comunitaria y una edición empresarial de la base de datos. *Enterprise Edition* incluye todo lo que *Community Edition* tiene para ofrecer, además de requisitos empresariales adicionales como copias de seguridad, agrupación en clústeres y capacidades de conmutación por error.

Neo4j se conoce como una base de datos de grafo nativa porque implementa de manera eficiente el modelo de gráficos de propiedades hasta el nivel de almacenamiento. Esto significa que los datos se almacenan exactamente como los escribe en la pizarra y la base de datos utiliza punteros para navegar y recorrer el gráfico. A diferencia del procesamiento de gráficos o las bibliotecas en memoria, Neo4j también proporciona características de base de datos completas, incluido el cumplimiento de transacciones ACID, el soporte de clústeres y la conmutación por error en tiempo de ejecución, lo que lo hace adecuado para usar gráficos para datos en escenarios de producción.

Algunas de las siguientes características particulares hacen que Neo4j sea muy popular entre desarrolladores, arquitectos y administradores de bases de datos:

- Cypher, un lenguaje de consulta declarativo similar a SQL, pero optimizado para gráficos. Ahora lo utilizan otras bases de datos como SAP HANA Graph y Redis Graph a través del proyecto openCypher.
- Recorridos de tiempo constantes en grandes grafos tanto en profundidad como en amplitud debido a la representación eficiente de nodos y relaciones. Permite la ampliación a miles de millones de nodos en hardware moderado.
- Esquema de grafo de propiedad flexible que se puede adaptar con el tiempo, lo que permite materializar y agregar nuevas relaciones más adelante para atajar y acelerar los datos de dominio cuando las necesidades comerciales cambien [27].

En la figura 50 se muestran las principales características de Neo4j.

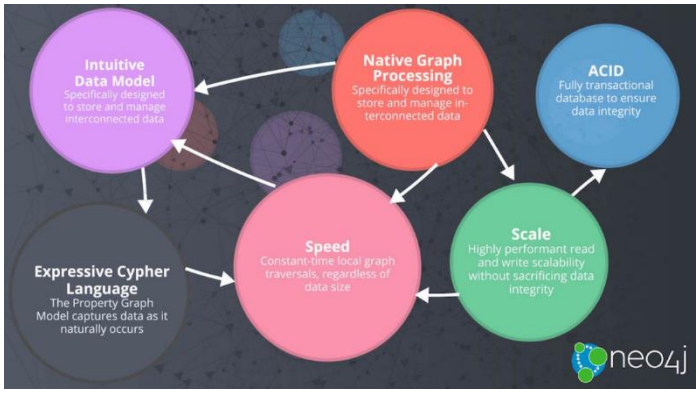


Figura 50 Características Neo4j [26].

APÉNDICE C. *CYPHER*

Cypher es el lenguaje de consulta de gráficos de Neo4j que permite a los usuarios almacenar y recuperar datos de la base de datos de grafos. Neo4j quería que la consulta de datos de grafos fuera fácil de aprender, comprender y usar para todos, pero también incorporar el poder y la funcionalidad de otros lenguajes estándar de acceso a datos. Esto es lo que Cypher pretende lograr.

La sintaxis de Cypher proporciona una forma visual y lógica de hacer coincidir patrones de nodos y relaciones en el gráfico. Es un lenguaje declarativo inspirado en SQL para describir patrones visuales en gráficos utilizando la sintaxis ASCII-Art. Nos permite indicar qué queremos seleccionar, insertar, actualizar o eliminar de los datos de nuestro gráfico sin una descripción de cómo hacerlo exactamente. A través de Cypher, los usuarios pueden construir consultas expresivas y eficientes para manejar la funcionalidad necesaria de creación, lectura, actualización y eliminación.

Cypher no solo es la mejor manera de interactuar con los datos y Neo4j, ¡también es de código abierto! El proyecto openCypher proporciona una especificación de lenguaje abierto, un kit de compatibilidad técnica e implementación de referencia del analizador, el planificador y el tiempo de ejecución de Cypher. Está respaldado por varias empresas de la industria de las bases de datos y permite a los implementadores de bases de datos y clientes beneficiarse, utilizar y contribuir libremente al desarrollo del lenguaje openCypher [40].

En la imagen 51 se puede observar un ejemplo de implementación de *sub-queries* con CYPHER, una cualidad del lenguaje que fue utilizada en la presente solución para elaborar nuestro algoritmo interno.

Cypher

```
MATCH (p:Person)-[r:IS_FRIENDS_WITH]->(friend:Person)
WHERE exists((p)-[:WORKS_FOR]->(:Company {name: 'Neo4j'}))
RETURN p, r, friend
```

Figura 51 *Sub-queries* con CYPHER [40].

APÉNDICE D. *APOC*

APOC son las siglas de *Awesome Procedures* en Cypher. Antes del lanzamiento de APOC, los desarrolladores necesitaban escribir sus propios procedimientos y funciones para una funcionalidad común que Cypher o la base de datos Neo4j aún no habían implementado como soporte. Cada desarrollador puede escribir su propia versión de estas funciones, lo que genera mucha duplicación.

Entonces, uno de nuestros desarrolladores de Neo4j creó la biblioteca APOC como una biblioteca de utilidades estándar para procedimientos y funciones comunes. Esto permitió a los desarrolladores de todas las plataformas e industrias usar una biblioteca estándar para procedimientos comunes y solo escribir su propia funcionalidad para la lógica empresarial y las necesidades específicas de casos de uso.

Se cree que la biblioteca APOC es la biblioteca de extensión más grande y más utilizada para Neo4j. Incluye más de 450 procedimientos estándar que brindan funcionalidad para utilidades, conversiones, actualizaciones de gráficos y más. Están bien soportados y son muy fáciles de ejecutar como funciones independientes o de incluir en consultas Cypher.

Existen dos maneras de instalar APOC dentro de Neo4j:

- A nivel proyecto, instalando la herramienta para todas las bases de datos del sistema, como se muestra en la figura 52.
- A nivel de la base de datos, seleccionando dentro de sus propiedades la librería a ser instalada [41].

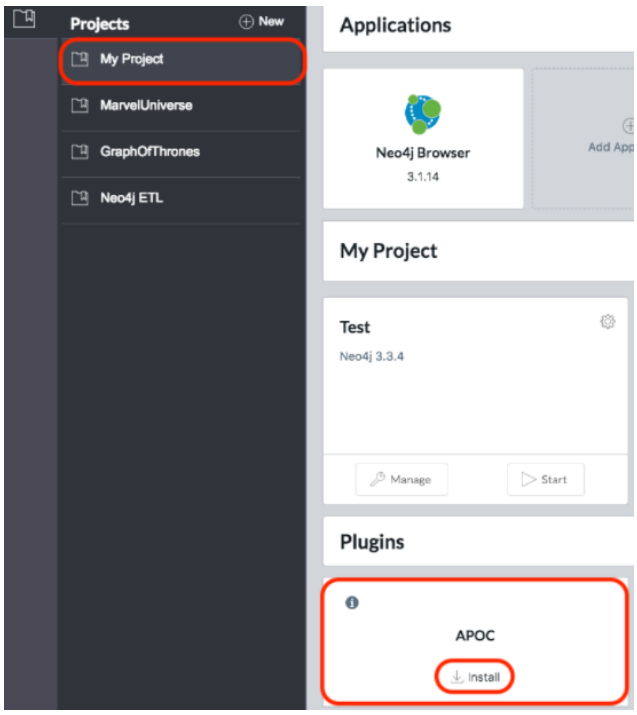


Figura 52 Instalación de APOC[41].

APÉNDICE E. *NODEJS*

Node.js es una plataforma basada en el tiempo de ejecución de JavaScript de Chrome para crear fácilmente aplicaciones de red rápida y escalable. Node.js utiliza un modelo de E / S sin bloqueo controlado por eventos que lo hace liviano y eficiente, perfecto para aplicaciones en tiempo real con uso intensivo de datos que se ejecutan en dispositivos distribuidos.

A continuación se muestran algunas de las características mostradas en la imagen 53 que hacen de Node.js la primera opción de los arquitectos de software.

- Asíncronas y controladas por eventos: todas las API de la biblioteca Node.js son asincrónicas, es decir, sin bloqueo. Básicamente, significa que un servidor basado en Node.js nunca espera a que una API devuelva datos. El servidor pasa a la siguiente API después de llamarlo y un mecanismo de notificación de Eventos de Node.js ayuda al servidor a obtener una respuesta de la llamada API anterior.
- Muy rápido: al estar construido sobre el motor JavaScript V8 de Google Chrome, la biblioteca Node.js es muy rápida en la ejecución de código.
- Un solo subproceso pero altamente escalable: Node.js usa un modelo de un solo subproceso con bucle de eventos. El mecanismo de eventos ayuda al servidor a responder de forma no bloqueante y hace que el servidor sea altamente escalable en comparación con los servidores tradicionales que crean subprocesos limitados para manejar solicitudes. Node.js utiliza un programa de un solo subproceso y el mismo programa puede proporcionar servicio a un número mucho mayor de solicitudes que los servidores tradicionales como el servidor HTTP Apache.
- Sin almacenamiento en búfer: las aplicaciones Node.js nunca almacenan datos en búfer. Estas aplicaciones simplemente generan los datos en fragmentos.
- Licencia: Node.js se publica bajo la licencia MIT [42].

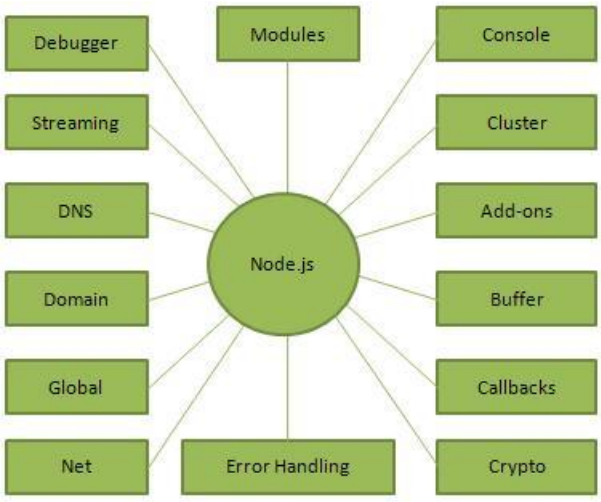


Figura 53 Características de Nodejs[42].

APÉNDICE F. *Neovis framework*

Esta librería fue diseñada para combinar la visualización de *JavaScript* y Neo4j en una integración perfecta. La conexión a Neo4j es simple y directa, y debido a que se construyó teniendo en cuenta el modelo de gráfico de propiedades de Neo4j, el formato de datos que Neovis espera se alinea con la base de datos. La personalización y coloración de estilos basados en etiquetas, propiedades, nodos y relaciones se define en un único objeto de configuración. Neovis.js se puede usar sin escribir *Cypher* y con un mínimo de *JavaScript* para integrarse en su proyecto.

Para maximizar la funcionalidad y las capacidades de análisis de datos a través de la visualización, también puede combinar esta librería con la biblioteca de algoritmos de grafos en Neo4j para diseñar la visualización y alinearla con los resultados de algoritmos tales como rango de página, centralidad, comunidades y más. A continuación, vemos una visualización gráfica de las interacciones de los personajes de *Game Of Thrones* representadas por neovis.js y mejoradas con los algoritmos de gráficos de Neo4j aplicando algoritmos de detección de *pagerank* y comunidad al estilo de la visualización en la imagen 54 [27].

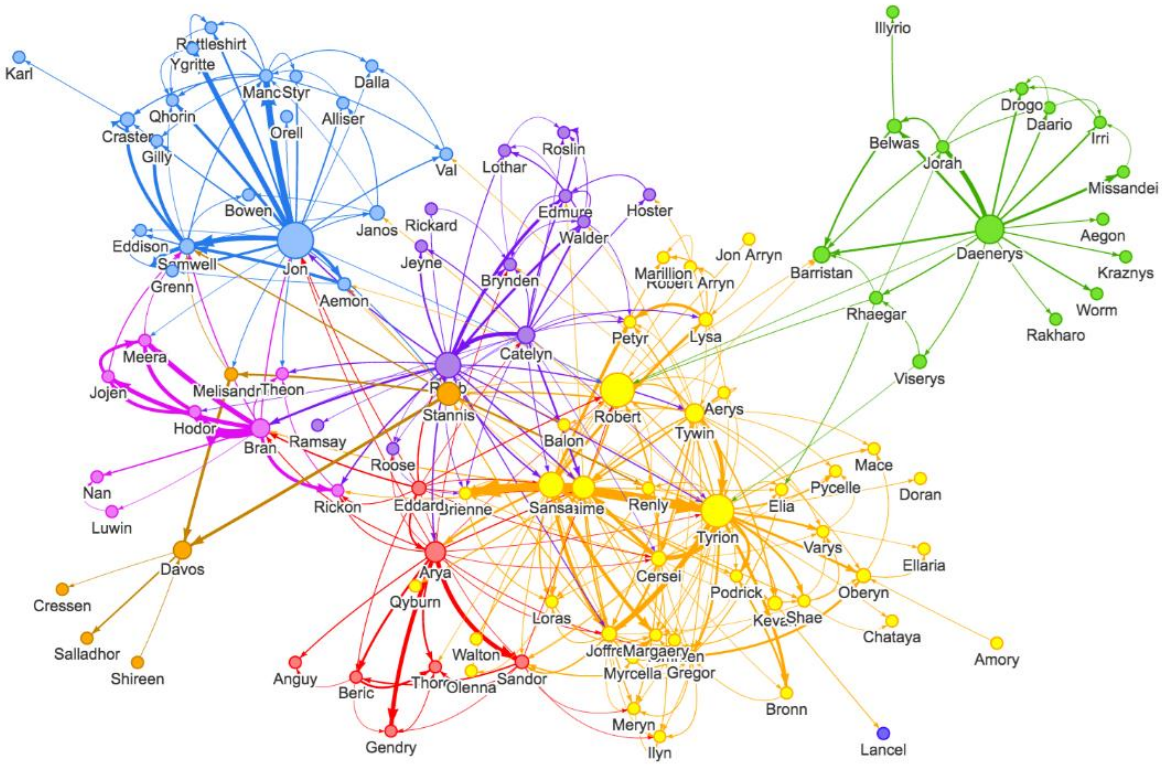


Figura 54 Grafo generado con Neovis [27].