UNIVERSIDAD NACIONAL DEL ALTIPLANO

FACULTAD DE INGENIERÍA MECÁNICA ELÉCTRICA, ELECTRÓNICA Y SISTEMAS

ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS

MODELO DE DATOS ORIENTADO A GRAFOS PARA EL PROCESAMIENTO DE RECOMENDACIONES DE LIBROS BASADOS EN EL ESTÁNDAR *MACHINE READABLE CATALOGING*

TESIS

PRESENTADA POR:

VICTOR JHAMPIER CAXI MAQUERA

PARA OPTAR EL TÍTULO PROFESIONAL DE:

INGENIERO DE SISTEMAS

PUNO – PERÚ

2023

Reporte de similitud

NOMBRE DEL TRABAJO

MODELO DE DATOS ORIENTADO A GRAFOS PARA EL PROCESAMIENTO DE RECOMENDACIONES DE LIBROS BASADOS EN EL ESTÁNDAR MACHINE READABLE CATALOGING

AUTOR

VICTOR JHAMPIER CAXI MAQUERA

RECUENTO DE PALABRAS

23941 Words

RECUENTO DE CARACTERES

135413 Characters

RECUENTO DE PÁGINAS

124 Pages

TAMAÑO DEL ARCHIVO

4.3MB

FECHA DE ENTREGA

Nov 29, 2023 7:32 AM GMT-5

FECHA DEL INFORME

Nov 29, 2023 7:34 AM GMT-5

11% de similitud general

El total combinado de todas las coincidencias, incluidas las fuentes superpuestas, para cada base

- 8% Base de datos de Internet
- 2% Base de datos de publicaciones
- Base de datos de Crossref
- Base de datos de contenido publicado de Cross
- 8% Base de datos de trabajos entregados

Excluir del Reporte de Similitud

- Material bibliográfico
- Material citado
- Material citado
- Coincidencia baja (menos de 8 palabras)

Universidad Nacional del Altiplano
E.P. DE INGENIERIA DE SISTEMAS
Dr. Elmer Coyla Idme
DIRECTOR DE ESCUELA

V°B°. Dra. Guina G. Sotomayor A.
Sub Directora Investigación EPIS

DEDICATORIA

A Dios, por guiarme a lo largo de esta travesía académica. A mis amados padres, Víctor y Elvira, cuyo amor y convicción en mí han sido fundamentales en cada paso que he dado. A mi pequeña Flavia, quien llenó mis noches de desvelo mientras dedicaba horas al estudio.

Victor Jhampier Caxi Maquera

AGRADECIMIENTO

Ante todo, expreso mi gratitud a Dios por iluminar mi camino a lo largo de este desafiante proceso académico.

A mis padres, quienes realizaron incontables esfuerzos para facilitarme oportunidades educativas.

A los profesores de la Escuela Profesional de Ingeniería de Sistemas, por proporcionarme una sólida y valiosa formación académica.

De manera muy especial, agradezco a mi asesor, D.Sc. Elmer Coyla Idme, por su experta orientación y paciencia durante mi desarrollo académico e investigación.

Deseo agradecer de manera especial al D.Sc. Oliver Vilca Huayta por su papel a lo largo de mi formación académica, desde ser mi docente preuniversitario hasta el actual presidente del jurado.

Finalmente, me gustaría expresar mi más sincera gratitud a dos personas excepcionales, Julia y Flavia, por su apoyo inquebrantable a lo largo de este viaje.

Victor Jhampier Caxi Maquera

ÍNDICE GENERAL

CAPÍTULO IV

RESULTADOS Y DISCUSIÓN

ÁREA: Ingeniería de software, bases de datos e inteligencia de negocios

TEMA: Bases de datos orientado a grafos

FECHA DE SUSTENTACIÓN: 06 de diciembre del 2023

ÍNDICE DE FIGURAS

Pág.

ÍNDICE DE TABLAS

Pág.

ÍNDICE DE ANEXOS

Pág.

ÍNDICE DE ACRÓNIMOS

ACID	Atomicity, Consistency, Isolation, Durability (Atomicidad, Consistencia, Aislamiento y Durabilidad).
BNE	Biblioteca Nacional de España.
CNRI	Corporación Nacional de Investigación de Internet.
CWI	Centrum Wiskunde & Informatica (Centro para la Investigación de Álgebra Computacional).
DBMS	Database Management System (Sistema de Gestión de Bases de Datos).
DDC	Dewey Decimal Classification (Clasificación Decimal Dewey).
GNN	Graph Neural Network (Red Neuronal de Grafos).
HTTP	Hypertext Transfer Protocol (Protocolo de Transferencia de Hipertexto).
IDE	Integrated Development Environment (Entorno de Desarrollo Integrado).
IFLA	International Federation of Library Associations and Institutions. (Federación Internacional de Asociaciones de Bibliotecarios y Bibliotecas)
ILD	Intra-List Diversity (diversidad intra-lista).
IoT	Internet of Things (Internet de las Cosas).
ISBN	International Standard Book Number (Número estándar internacional de libros)**.**
JSON	JavaScript Object Notation (Notación de objetos de JavaScript).
KNN	K-Nearest Neighbors (Vecinos más cercanos).
MARC	Machine-Readable Cataloging (Catálogo legible por máquina)**.**
NoSQL	Not Only SQL (No solo SQL).
OWL	Web Ontology Language (Lenguaje de ontologías de la Web).
RDA	Resource Description and Access (Descripción y acceso de recursos).
RDF	Resource Description Framework (Marco de descripción de recursos).

ROC	Receiver Operating Characteristic (Característica Operativa del Receptor).
SQL	Structured Query Language (Lenguaje de Consulta Estructurado).
TF-IDF	Term Frequency-Inverse Document Frequency (Frecuencia de Término-Frecuencia Inversa de Documento).
UML	Unified Modeling Language (Lenguaje de Modelado Unificado).
UNIMARK	Universal Machine Readable Cataloging (Catalogación Universal Legible por Máquina).

RESUMEN

El procesamiento de recomendaciones implica analizar grandes conjuntos de datos para identificar patrones y similitudes entre elementos y proporcionar sugerencias. El procesamiento de recomendaciones es ampliamente utilizado en diversos campos, como en el comercio electrónico, medicina, entretenimiento, etc. Sin embargo, en el contexto de las recomendaciones de libros basadas en el estándar MARC, su aplicación sigue siendo limitada. El objetivo principal de esta investigación fue la utilización de un modelo de base de datos basado en grafos para generar recomendaciones de libros catalogados mediante el estándar MARC, con un énfasis particular en lograr diversidad en los resultados a partir de características de similitud. Se implementaron tres métodos distintos (Índice de Jaccard, Autoría excluida y Categoría principal). Los datos utilizados en esta investigación consistieron en información bibliográfica procedente de dos bibliotecas públicas de la región, con un total de 2265 libros. Se empleó la métrica de la diversidad para evaluar la calidad de las recomendaciones. Los resultados obtenidos indicaron que el modelo demostró ser altamente eficiente. Los experimentos revelaron que la combinación de los tres métodos generó una notable diversidad en las recomendaciones, medida por la métrica de la diversidad, con valores que oscilaron entre 53% y 95%. Además, se observó un destacable desempeño en términos de tiempos de carga de datos y ejecución de consultas, con un máximo de 162 milisegundos, lo que evidencia la eficiencia en la generación de recomendaciones. Este estudio también logró un exitoso preprocesamiento de los datos, así como la representación efectiva de la información bibliográfica en el modelo de base de datos orientado a grafos.

Palabras clave: Procesamiento de recomendaciones, Modelo de base de datos orientadas a grafos, filtrado basado en similitud, estándar MARC.

ABSTRACT

The processing of recommendations involves analyzing large datasets to identify patterns and similarities among items and provide suggestions. Recommendation processing is widely used in various fields such as e-commerce, medicine, entertainment, etc. However, in the context of MARC-based book recommendations, its application remains limited. The main objective of this research was to utilize a graph database-oriented model to generate recommendations for cataloged books based on the MARC standard, with a particular emphasis on achieving diversity in the results based on similarity features. Three different methods were implemented (Jaccard Index, Excluded Authorship, and Main Category). The data used in this research consisted of bibliographic information from two public libraries in the region, totaling 2265 books. Diversity metrics were employed to evaluate the quality of recommendations. The results obtained indicated that the model proved to be highly efficient. The experiments revealed that the combination of the three methods generated significant diversity in recommendations, as measured by the diversity metric, with values ranging from 53% to 95%. Additionally, there was a notable performance in terms of data loading and query execution times, with a maximum of 162 milliseconds, demonstrating efficiency in recommendation generation. This study also successfully preprocessed the data and effectively represented bibliographic information in the graph database-oriented model.

Keywords: Recommendation processing, Graph-oriented database model, similarity-based filtering, MARC standard.

CAPÍTULO I

INTRODUCCIÓN

Los libros desempeñan un papel fundamental en la vida humana al ser una fuente inagotable de conocimiento que abarca una amplia variedad de temas, autores y propósitos. Una amplia mayoría de bibliotecas públicas de la región catalogan sus libros mediante el formato estandarizado conocido como Formato MARC. Los datos bibliográficos que este formato contiene atributos valiosos que pueden ser aprovechados con el fin de generar recomendaciones.

Acorde con los avances actuales en el ámbito computacional, han surgido modelos más eficientes para la generación de recomendaciones. Específicamente, los modelos de datos orientados a grafos han demostrado ser particularmente efectivos en la representación y el procesamiento de información altamente conectada, tal como la que se encuentra en los datos bibliográficos. Estos modelos permiten el almacenamiento y procesamiento eficiente de información al enfocarse en las relaciones entre las entidades, representándolas como nodos y relaciones de un grafo.

Esta investigación se enfoca en la aplicación de un modelo de base de datos orientado a grafos para la generación de recomendaciones de libros mediante técnicas de similitud. El propósito central radica en aprovechar las capacidades de este modelo de base de datos para representar de manera efectiva la información bibliográfica y emplearla como fundamento sólido para el procesamiento de recomendaciones. La premisa subyacente en este estudio es que, mediante el uso de este modelo de base de datos, se pueden obtener recomendaciones diversificadas.

La estructura de esta tesis consta de los siguientes capítulos: El primer capítulo se centra en los objetivos de la investigación. El segundo capítulo presenta la base teórica, revisa los antecedentes de la investigación y establece el marco conceptual. El tercer capítulo expone detalladamente los métodos y materiales utilizados en el estudio. El cuarto capítulo presenta los resultados de la investigación, los evalúa y concluye con una discusión, así como las conclusiones derivadas del estudio.

1.1. OBJETIVOS DE INVESTIGACIÓN

1.1.1. Objetivo General

Implementar un modelo de base de datos orientado a grafos para procesar recomendaciones de libros basados en el estándar MARC que obtenga diversidad en los resultados.

1.1.2. Objetivos Específicos

- Implementar un algoritmo de preprocesamiento para mejorar la calidad de los datos de entrada para el modelo de base datos.
- Representar información bibliográfica en forma de nodos y relaciones en el modelo de base de datos.
- Implementar métodos de recomendación basadas en similitud en el modelo de datos que cuya combinación generen diversidad en los resultados.
- Evaluar la diversidad de las recomendaciones procesadas por el modelo de base de datos.

CAPÍTULO II

REVISIÓN DE LITERATURA

En este capítulo se establecen las bases teóricas, revisa los antecedentes a nivel nacional e internacional y presenta el marco conceptual. Se exploran conceptos clave relacionados con modelos de recomendación, análisis de datos y estructuras de datos orientadas a grafos.

2.1. BASES TEÓRICAS

2.1.1. Procesamiento de recomendaciones

El procesamiento de recomendaciones implica el empleo de algoritmos y técnicas de análisis de datos para proporcionar recomendaciones personalizadas a los usuarios, teniendo en cuenta sus preferencias y comportamientos anteriores. Este enfoque se utiliza mucho en las aplicaciones de comercio electrónico, donde se sugieren productos basándose en las compras anteriores del usuario o en los productos que ha visto (GraphEverywhere, 2020).

En este contexto, el procesamiento de recomendaciones se refiere a las tareas necesarias para filtrar, priorizar y proporcionar información relevante y personalizada a los usuarios en un sistema de recomendaciones. Esto incluye recopilar y limpiar datos, aplicar algoritmos de filtrado, evaluar y mejorar continuamente el sistema para aumentar la precisión y eficacia de las recomendaciones. En otras palabras, el procesamiento de recomendaciones abarca los procesos que permiten a un sistema de recomendaciones ofrecer elementos personalizadas y relevantes a los usuarios (Kavin *et al*., 2021).

Es así que, existen sistemas de recomendación que emplean técnicas de minería de datos y aprendizaje automático para analizar la relación de datos de los usuarios y los elementos, y posteriormente generar recomendaciones personalizadas para cada usuario, como se menciona en el artículo de (Qassimi *et al.*, 2021). Así mismo, los sistemas de recomendación emplean algoritmos para analizar datos de artículos disponibles con el fin de sugerir artículos que puedan ser de interés o utilidad para el usuario objetivo. Estos artículos pueden ser explícitos o implícitos y también se utilizan para identificar los artículos que cumplen las normas establecidas en relación con un conjunto específico de requisitos del usuario (Chicaiza & Valdiviezo, 2021).

De esta manera, se entiende que el procesamiento de recomendaciones involucra la recopilación de datos, tales como preferencias, historial, calificaciones y comentarios respecto a un contenido, y la aplicación de algoritmos de procesamiento automático para analizar estos datos y generar recomendaciones.

2.1.1.1. Procesamiento de recomendaciones de libros

Los procesos de recomendación de libros involucran la evaluación de datos relacionados con los libros, así como las interacciones históricas con los usuarios, lo que posibilita la generación de recomendaciones. Este proceso implica la habilidad de anticipar las preferencias de los usuarios respecto a los libros, basándose en sus interacciones previas con la literatura (Zhiyuli *et al*., 2023).

Estos sistemas pueden emplear diversas técnicas, como el filtrado colaborativo y el filtrado basado en el contenido, para analizar los datos de

los usuarios y los libros, y ofrecer recomendaciones personalizadas. El objetivo principal es ayudar a los usuarios a descubrir nuevos libros que puedan interesarles y mejorar la utilización de los recursos de la biblioteca (Tian *et al*., 2019). Asimismo, los sistemas de recomendación de libros pueden estar basados en grafos de conocimiento, cuyo proceso de recomendación depende de datos externos y se fundamenta en la inferencia entre nodos para descubrir nuevas conexiones (Chicaiza & Valdiviezo, 2021).

Considerando a los autores previamente mencionados, se infiere que un proceso de recomendación de libros se fundamenta en la calificación de estos, haciendo uso de interacciones históricas para evaluar la calidad de un libro. Posteriormente, esta información se emplea para anticipar las preferencias de libros de los usuarios, culminando en la entrega de una recomendación apropiada para cada usuario.

2.1.1.1.1 Fases del proceso de recomendación

a) Recopilación de Datos

Se utilizan diferentes fuentes de información para recopilar datos, como interacciones históricas (como clics, navegación, préstamos, colecciones, comentarios, calificaciones, etc.), así como datos sobre los usuarios y los libros. Luego, se aplican diversos modelos de aprendizaje automático para generar recomendaciones (Zhiyuli *et al*., 2023).

La recopilación de datos se efectúa a través de diversas técnicas, como el etiquetado colaborativo y la minería de datos. La técnica de etiquetado colaborativo se emplea con el fin de recabar las opiniones de

los usuarios y así mejorar la calidad del sistema de recomendación. Paralelamente, se recurre a técnicas de minería de datos para analizar tanto los datos de los usuarios como los de los elementos, y seguidamente generar recomendaciones personalizadas para cada usuario (Qassimi *et al*., 2021).

Según lo señalado por los autores, la recopilación de datos el primer paso, esto incluyen información sobre usuarios, productos, interacciones pasadas, calificaciones, reseñas, preferencias, historiales. Estos datos se utilizan posteriormente para predecir las preferencias de los usuarios.

b) Preprocesamiento de Datos

El preprocesamiento de datos implica limpiar y organizar los datos sin procesar para que sean adecuados para la construcción y el entrenamiento de modelos, el preprocesamiento de datos es el primer y más importante paso en la construcción de un sistema de recomendación de libros (Kavin *et al*., 2021).

Es una etapa crítica en el proceso de aprendizaje automático, ya que los datos suelen ser ruidosos, incompletos o inconsistentes, y pueden contener información redundante o irrelevante que puede afectar negativamente el rendimiento del modelo (Liu *et al*., 2022).

De esta manera se entiende que el preprocesamiento de datos se refiere al conjunto de técnicas y operaciones que se aplican a los datos antes de ser utilizados para entrenar un modelo.

c) Almacenamiento de datos

Los sistemas de recomendación utilizan el almacenamiento de datos para guardar información sobre los usuarios, los elementos y las interacciones entre ellos, y cómo esta información se utiliza para generar recomendaciones *(Salau et al*., 2022).

Entre ellas se encuentran las bases de datos orientadas a grafos, que utilizan un sistema de almacenamiento de datos nativo, optimizado específicamente para almacenar y gestionar grafos. No todas las bases de datos orientadas a grafos emplean este tipo de almacenamiento nativo; sin embargo, algunas de ellas serializan los datos de los grafos en bases de datos relacionales, bases de datos orientadas a objetos o bases de datos de propósito general (Amézquita Llerena, 2020).

De los autores, los sistemas de recomendación aprovechan el almacenamiento de datos para registrar datos sobre usuarios, elementos y sus interacciones la cual se utiliza para crear recomendaciones personalizadas.

d) Diseño del Modelo

En el contexto de los sistemas de recomendación, el modelado se refiere al proceso de construir un modelo matemático o estadístico que pueda predecir las preferencias o intereses de un usuario en función de sus interacciones pasadas con los elementos (Salau *et al*., 2022).

En síntesis, durante esta fase se seleccionan las técnicas de recomendación que pueden sugerir elementos similares a los que el usuario

ha consumido anteriormente, basadas en el contenido o en el comportamiento de usuarios similares, entre otras técnicas.

e) Generación de Recomendaciones

La fase de generación de recomendación se refiere al proceso de sugerir elementos que puedan ser interesantes o útiles para un usuario objetivo. Esto se logra mediante el uso de técnicas de análisis de datos y algoritmos que analizan los datos de los usuarios y los elementos disponibles para recomendar elementos que puedan ser de interés para el usuario. Los sistemas de recomendación utilizan diferentes métodos de recolección de información para captar el interés de sus usuarios (Chicaiza & Valdiviezo, 2021).

Esta fase se centra en la selección de elementos que un usuario puede encontrar interesantes o relevantes, basándose en sus preferencias y comportamientos previos. Estas técnicas son herramientas útiles para una variedad de dominios, incluyendo el comercio electrónico, la música, las películas, los libros, entre otros (Silveira *et al*., 2019).

En definitiva, los sistemas de recomendación utilizan el modelo entrenado y los datos del usuario para crear recomendaciones personalizadas que pueden adoptar la forma de productos o contenidos que sean relevantes para cada usuario.

Figura 1

Fases del proceso de recomendación

Elaboración propia.

2.1.1.1.2 Técnicas de recomendación

a) Recomendación en base al nivel de personalización

- **No personalizados**

El enfoque no personalizado no utiliza información de usuarios específicos para hacer predicciones. En otras palabras, estos sistemas de recomendación no tienen en cuenta las preferencias o el historial de compras de un usuario en particular. En cambio, se basan en información general sobre los productos o servicios que se están recomendando, como su popularidad o su relevancia para una categoría determinada. Estos sistemas pueden ser útiles en situaciones en las que no se dispone de información sobre los usuarios o cuando se desea proporcionar recomendaciones más generales (Silveira *et al*., 2019).

Es decir, la recomendación de elementos que son iguales para todos los usuarios, no se adapta a las preferencias individuales de cada usuario. Estas recomendaciones se basan en la popularidad o relevancia general de los elementos recomendados y están destinadas a un grupo de usuarios en general (Chicaiza & Valdiviezo, 2021). En este enfoque, no se tiene en cuenta la información específica de cada usuario, como sus preferencias individuales o su historial de interacciones (Qassimi *et al*., 2021).

Esta técnica busca ofrecer recomendaciones que sean adecuadas o relevantes para un público más amplio en función de la popularidad o utilidad general de los elementos recomendados.

- **Personalizados**

La recomendación personalizada utiliza información sobre el comportamiento y las preferencias previas de un usuario para hacer recomendaciones más específicas. Por ejemplo, si un usuario ha comprado libros de ciencia ficción en el pasado, es probable que se le recomienden más libros de ciencia ficción en el futuro. Esta estrategia ha demostrado mejorar significativamente la calidad de las recomendaciones en comparación con los sistemas de recomendación no personalizados (Silveira *et al*., 2019).

Así como también, se basa en las preferencias, intereses y necesidades específicas de cada usuario, sugiriendo libros en función de sus interacciones históricas con los libros, así como de sus atributos básicos y los atributos de los libros. Para lograr esto, se utilizan varios modelos de aprendizaje automático que construyen recomendaciones precisas y personalizadas para cada usuario (Zhiyuli *et al*., 2023).

En resumen, la recomendación personalizada es el proceso de recomendar elementos que se adaptan a las preferencias y necesidades particulares de un usuario. Este proceso se realiza mediante técnicas de análisis de datos y algoritmos que analizan los datos de los usuarios y los elementos disponibles para sugerir elementos que pueden ser de interés para el usuario (Chicaiza & Valdiviezo, 2021).

Teniendo en cuenta a los autores, la recomendación personalizada se basa en el historial y las preferencias individuales de cada usuario, lo que la hace más efectiva y precisa.

- **Semi personalizados**

Las recomendaciones semi personalizadas representan una combinación de ambos enfoques de recomendación. Por ejemplo, consideremos un sistema de recomendación de películas que utiliza tanto el historial de visualización del usuario como las tendencias de búsqueda de los usuarios para generar recomendaciones (Li *et al*., 2023).

En lugar de depender exclusivamente de la información de calificación histórica, este modelo también incorpora información adicional, como los perfiles de usuario y los atributos de los elementos. Esta inclusión de datos adicionales mejora la calidad de las recomendaciones al permitir que el modelo tenga en cuenta tanto los intereses y preferencias individuales del usuario como las características específicas de los elementos recomendados (Zhang *et al*., 2020).

En conclusión, las recomendaciones semi personalizadas combinan datos personales y datos agregados para generar recomendaciones adaptadas a cada usuario. Este enfoque proporciona ventajas en términos de precisión, relevancia y escalabilidad (Li *et al*., 2023).

b) En base al patrón de consumo

- **Recomendación basada en métricas de similitud**

Índice de Jaccard

El coeficiente de Jaccard es uno de los métodos utilizados para calcular la similitud entre dos elementos el cálculo de este método se basa en la medida de similitud del espacio vectorial (Wahyuningsih, 2021).

Es una medida de similitud entre conjuntos que se utiliza en una variedad de campos, incluyendo la informática, la estadística, la biología y la geología, es una medida de similitud entre dos conjuntos (Shtovba & Petrychko, 2019).

$$Jaccard(A, B) = \frac{A \cap B}{A \cup B}$$

Similitud del coseno - Distancia del coseno

La similitud del coseno es un algoritmo para medir la similitud entre dos vectores. Se utiliza para clasificar y ordenar vectores en función del valor de similitud del coseno obtenido. El valor de similitud del coseno indica la similitud entre dos vectores, y el valor máximo posible es 1, y este se utiliza en la filtración basada en contenido para recomendar elementos similares a los que un usuario ha interactuado previamente (Sukestiyarno *et al*., 2023).

La similitud del coseno es un método de cálculo de similitud utilizado en análisis de datos y minería de datos. Es una medida de similitud que se calcula como el producto interno de dos vectores dividido por el producto de las magnitudes de los vectores (Wang *et al*., 2021).

$$Cos(A,B) = \frac{AB}{\|A\|\|B\|} \frac{\sum_{i=1}^{n} Ai\ Bj}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}}$$

Similitud de Pearson

El coeficiente de similitud de Pearson es un algoritmo utilizado en sistemas de recomendación para medir la similitud entre dos conjuntos de datos estos también se utilizan para clasificar y ordenar conjuntos de datos en función del valor de similitud de Pearson obtenido. El valor de similitud de Pearson indica la correlación entre dos conjuntos de datos, y el valor máximo posible es 1. Se utiliza para recomendar elementos a los usuarios en función de las calificaciones y preferencias de otros usuarios con gustos similares (Sukestiyarno *et al*., 2023).

$$R = \frac{\sum_{i=1}^{n}(xi - \bar{x})\ (yi - \bar{y})}{\sqrt{\sum_{i=1}^{n}(xi - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(yi - \bar{y})^2}}$$

KNN – K-Vecinos más cercanos

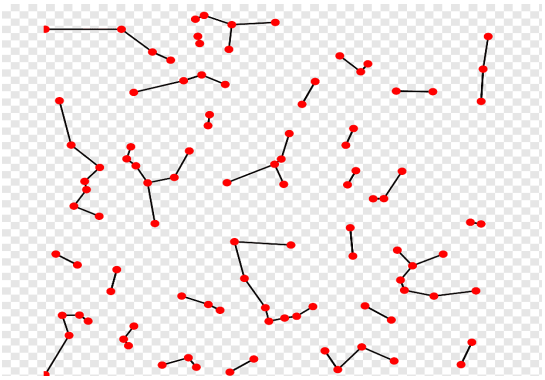
K-Vecinos más cercanos (K-NN) es un algoritmo utilizado en sistemas de recomendación para encontrar los elementos más similares a un elemento dado. Se utiliza en la filtración basada en contenido y colaborativa para recomendar elementos similares a los que un usuario ha interactuado previamente (Sukestiyarno *et al*., 2023).

En la clasificación K-NN, se utiliza un conjunto de datos de entrenamiento para clasificar nuevos puntos de datos en una o varias categorías, además se utiliza un conjunto de datos de entrenamiento para predecir el valor de una variable continua para nuevos puntos de datos

(Wang *et al*., 2021). El algoritmo funciona encontrando los K elementos más cercanos al elemento dado y luego recomendando elementos similares a esos K elementos. Los elementos más cercanos se determinan en función de las calificaciones y preferencias de otros usuarios con gustos similares (Sukestiyarno *et al*., 2023).

Figura 2

Clasificación por el vecino cercano



Nota: PngWing (2022).

- **Recomendación basada en contenido**

Es la recomendación de elementos similares a los que un usuario ha consumido anteriormente en términos de contenido. Esta técnica se basa en la suposición de que los usuarios tienen preferencias similares en términos de contenido y que los elementos que han consumido en el pasado son buenos indicadores de sus preferencias actuales (Qassimi *et al*., 2021).

Esta recomendación se basa en el contenido, por lo que aprende sobre el interés del usuario en función de la característica que cada usuario le interese, ya sea un libro leído o por un autor especifico, cuyo propósito es recomendar un libro de su interés en el futuro (Ramovecchi & García, 2021). Asimismo, utiliza interacciones históricas (clics, navegación,

préstamos, recopilación, comentarios, calificación, etc.) entre lectores y libros como fuentes de datos de características, las combina con atributos básicos de usuario y atributos de libro, y utiliza varios Modelos de aprendizaje automático para crear recomendaciones precisas (Zhiyuli *et al.*, 2023).

- **Recomendación basada en filtrado colaborativo**

Esta tiene su origen en un artículo publicado por Breese *et al*.,(1998) en Microsoft. Donde esta recomendación mayormente se da en la opinión de los demás usuarios es el que predice la utilidad de un ítem en base a la opinión de los demás usuarios, en el que se crea un conjunto de vecino cercanos, para ello la evaluación de los usuarios anteriores tiene una semejanza en el otro usuario, es por ello que el resultado predice una combinación de punto conocido como vecino cercano.

El filtrado colaborativo es uno de los principales algoritmos de filtrado comúnmente utilizados en los sistemas de recomendación. Este algoritmo se basa en la idea de que, si a un grupo de usuarios les gustan los mismos elementos, es probable que les gusten otros elementos similares. El filtrado colaborativo utiliza la información de las interacciones entre los usuarios y los elementos para generar recomendaciones personalizadas para cada usuario (Qassimi *et al*., 2021). Estos funcionan identificando usuarios que tienen intereses similares a los del usuario activo. Si los usuarios son similares, entonces es más probable que el usuario activo esté interesado en el contenido que el usuario similar ha calificado o revisado positivamente (Li *et al*., 2023).

- **Recomendación basada en conocimiento**

Una recomendación basada en conocimiento es un tipo de recomendación que se genera utilizando un modelo de conocimiento que representa las relaciones entre el contenido, los usuarios y otros factores relevantes. Este modelo de conocimiento se puede utilizar para identificar recomendaciones que son relevantes para el usuario y que satisfacen sus necesidades (Li *et al*., 2023).

Por otro lado, la recomendación basada en conocimiento se refiere a la recomendación de elementos que se ajustan a las necesidades y preferencias de un usuario en función de su perfil de conocimiento, para poder lograr esto se utiliza técnicas de minería de datos y análisis de contenido para comprender el perfil de conocimiento del usuario y recomendar elementos que se ajusten a ese perfil. Estos elementos pueden incluir recursos educativos, cursos en línea, libros, artículos, entre otros (Qassimi *et al*., 2021).

Un sistema de recomendación basado en el conocimiento podría recomendar libros basándose en su género, autor u otros atributos, en lugar de en el historial de lectura o las preferencias del usuario (Zhiyuli *et al*., 2023).

- **Recomendación basa en el contexto o conscientes del contexto**

Este tipo de recomendación considera el contexto en el que se encuentra el usuario, como su ubicación, la hora del día, o el evento en el que está participando. Por ejemplo, un sistema de recomendación basado en la situación podría recomendar un restaurante cercano a un usuario que

está caminando por la calle, o una película romántica a un usuario que está en una cita (Sattar & Bacciu, 2022).

En el contexto de la recomendación de libros, un sistema de recomendación basado en el contexto podría recomendar libros según la ubicación actual del usuario o la hora del día, como recomendar un libro sobre historia local cuando el usuario visita una nueva ciudad (Zhiyuli *et al*., 2023).

Para generar recomendaciones basadas en el contexto, los sistemas de recomendación pueden utilizar técnicas de minería de datos y análisis de contenido para comprender el contexto del usuario y recomendar elementos que se ajusten a ese contexto. Estos elementos pueden incluir recursos educativos, cursos en línea, libros, artículos, entre otros (Qassimi *et al*., 2021).

- **Recomendación hibrida**

La recomendación híbrida es un enfoque que combina diferentes técnicas de recomendación para mejorar la precisión y la calidad de las recomendaciones. En lugar de depender de un solo método de recomendación, la recomendación híbrida utiliza múltiples enfoques, como la recomendación basada en contenido, la recomendación colaborativa y la recomendación basada en conocimiento, para proporcionar recomendaciones más precisas y personalizadas (Gao *et al*., 2023).

c) En Base al tipo de retroalimentación

La retroalimentación, también conocida como feedback, se refiere a la información que se proporciona a una persona o sistema sobre su desempeño o comportamiento, la retroalimentación puede ser tanto positiva como negativa y puede ser proporcionada de manera explícita o implícita.

- **Retroalimentación implícita**

La retroalimentación implícita se refiere a la retroalimentación que el usuario no proporciona explícitamente, sino que se infiere de su comportamiento o acciones. la retroalimentación implícita puede incluir cosas como el historial de navegación del usuario, consultas de búsqueda o historial de compras. Este tipo de comentarios se puede utilizar para mejorar la precisión de las recomendaciones, incluso cuando no haya comentarios explícitos (como calificaciones o reseñas) disponibles. La retroalimentación implícita es importante en los sistemas de recomendación porque puede ayudar a mejorar la precisión de las recomendaciones al proporcionar información adicional sobre las preferencias y necesidades de los usuarios (Qassimi *et al*., 2021; Zhiyuli *et al*., 2023).

- **Retroalimentación explicita**

Es la información que se recopila directamente de los usuarios a través de encuestas, calificaciones o comentarios. Esta retroalimentación se utiliza para evaluar la calidad de las recomendaciones y mejorar el rendimiento del sistema. En general, la retroalimentación explícita es una forma importante de obtener información directa de los usuarios y mejorar la calidad de las recomendaciones (Silveira *et al*., 2019).

Esta retroalimentación puede ser utilizada para mejorar la precisión de las recomendaciones futuras y proporcionar una mejor experiencia de usuario. Por ejemplo, si un usuario califica positivamente un recurso educativo recomendado, el sistema de recomendación puede utilizar esta información para recomendar recursos similares en el futuro. Por otro lado, la retroalimentación explícita puede incluir cosas como la calificación de un libro por parte del usuario, su reseña de un libro o su comentario sobre un libro. Este tipo de comentarios se puede utilizar para mejorar la precisión de las recomendaciones indicando directamente las preferencias e intereses del usuario (Zhiyuli *et al*., 2023).

d) En base al tipo de técnica

- **Basados en memoria (Reglas)**

La recomendación basada en memoria es un tipo de método que puede ser de dos tipos: basado en usuarios y basado en elementos, y utiliza medidas de similitud para encontrar elementos o usuarios similares y recomendar elementos que hayan sido apreciados por estos usuarios o elementos similares (Chicaiza & Valdiviezo, 2021).

- **Basados en Modelos (Machine Learning)**

La recomendación basada en modelos es un enfoque de los sistemas de recomendación que utiliza técnicas de aprendizaje automático para construir un modelo de preferencias del usuario a partir de sus interacciones pasadas con el sistema. Este modelo se utiliza para hacer recomendaciones de elementos que el usuario podría estar interesado en función de sus preferencias previas como lo sugiere el artículo de (Jyothi, 2020). Estos modelos se emplean para hacer recomendaciones de elementos que el usuario podría estar interesado en función de sus preferencias previas, y utilizan el aprendizaje automático para analizar los datos de los usuarios y los elementos disponibles con el fin de recomendar elementos de interés para el usuario (Chicaiza & Valdiviezo, 2021).

2.1.1.2. Los problemas del procesamiento de recomendaciones

a) Escasez de Datos:

Los sistemas de recomendación requieren grandes cantidades de datos para entrenar modelos precisos. Sin embargo, en muchos casos, no hay suficientes datos disponibles para entrenar un modelo efectivo. Esto puede ser un problema particular para los sistemas de recomendación que se ocupan de nichos o áreas de interés poco comunes (Zhang et al., 2019).

Este problema afecta negativamente a la precisión de las predicciones en el proceso de recomendación. Surge en los sistemas de recomendación porque los usuarios suelen proporcionar sólo un número limitado de valoraciones de los artículos (Ahmadian *et al*., 2022).

b) Arranque en frio:

Es un desafío común en los sistemas de recomendación. Se refiere a la dificultad de generar recomendaciones adecuadas cuando no se dispone de suficiente información sobre usuarios y elementos. Este problema afecta principalmente a los sistemas de filtrado colaborativo, debido que requieren datos previos sobre las preferencias de los usuarios para funcionar correctamente. En situaciones en las que no se cuenta con un historial de actividad de usuarios, es complicado proporcionar recomendaciones de calidad (Kawai et al., 2022).

c) Dispersión:

La dispersión en el procesamiento de recomendaciones puede generar problemas como la escasez de datos, la dificultad para identificar patrones de comportamiento, el sesgo en las recomendaciones y la falta de diversidad en las mismas, los cuales pueden impactar negativamente la calidad de las recomendaciones y la experiencia del usuario (Joyanes Aguilar & Zahonero Martínez, 2007).

El problema ocurre especialmente en catálogos extensos con numerosos usuarios activos, la construcción de la matriz de usuarios por ítems en estos casos suele ser muy dispersa, con muchos valores faltantes, lo que dificulta encontrar grupos de usuarios que tengan calificaciones en común con el usuario objetivo (Rojas Rojas, 2019).

d) Comprensión del lenguaje natural:

En el ámbito de los sistemas de recomendación, surgen retos con el procesamiento del lenguaje natural. Estos retos se derivan de la necesidad de razonar eficazmente sobre diversos tipos de datos y modalidades para generar recomendaciones precisas y útiles para los usuarios. Estos obstáculos abarcan la necesidad de razonar sobre modalidades únicas o múltiples (como opiniones, texto, imágenes y metadatos) para producir recomendaciones impactantes (Zhang *et al*., 2019).

e) Sobre especialización:

El problema de la sobre especialización surge en los sistemas de recomendación cuando no se tiene en cuenta la variación potencial de los gustos de los usuarios. Esto sucede cuando los algoritmos de recomendación basados en contenido sugieren productos directamente relacionados con el perfil del cliente en lugar de ofrecer cosas nuevas (Stitini *et al., 2022)*.

En específico, la sobre especialización los sistemas de recomendación basados en contenido se refieren a la incapacidad inherente de estos sistemas para descubrir elementos inesperados. Estos sistemas tienden a sugerir elementos que tienen altas calificaciones en comparación con el perfil del usuario, lo que resulta en la recomendación de elementos similares a los que el usuario ya ha calificado (Castells *et al*., 2022).

En síntesis, la sobre especialización se refiere a la situación en la que un sistema de recomendación se vuelve demasiado enfocado en las

preferencias pasadas de un usuario y no ofrece variedad en las recomendaciones

f) Ataques al recomendador:

Un sistema de recomendación funciona como un negocio, lo que lleva a las personas a presentar opiniones sesgadas al ofrecer millones de reseñas positivas de sus propios productos, al tiempo que ofrecen reseñas negativas de los productos y artículos de la competencia conocido como Silling-attacks (Hossain *et al*., 2022).

Por otro lado, hay usuarios con preferencias y gustos únicos que son difíciles de encontrar o comparar con otros usuarios en un sistema de recomendación, a éstos se les conoce como Gray-sheep. Estos usuarios tienen gustos inusuales o exóticos, lo que dificulta la creación de perfiles precisos para ellos. Como resultado, el enfoque tradicional de buscar similitudes en el proceso de recomendación no funciona bien para estos usuarios (Alabdulrahman & Viktor, 2021).

Del mismo modo, hay usuarios Black-Sheep que muestran un comportamiento poco convencional, lo que hace que las recomendaciones sean casi imposibles. Esta situación se produce cuando los individuos son conscientes de que sus valoraciones pueden influir en el sistema de recomendaciones, lo que provoca la desconfianza de otros usuarios que perciben la manipulación de las recomendaciones (Hossain *et al*., 2022).

g) Ausencia de información vs compromiso con la privacidad:

Existe el derecho de las personas a mantener la autoridad sobre la información recopilada y su uso. Sin embargo, estos datos pueden ser utilizados indebidamente por terceros con fines no deseados (Müllner, 2023).

El conflicto entre privacidad y recopilación de información puede compararse con una balanza: cuando se inclina hacia la privacidad, salvaguarda la información personal del usuario pero limita la personalización de las recomendaciones; y cuando se inclina hacia la recopilación de información, las recomendaciones son más precisas pero a costa de la privacidad del usuario (Isufi *et al*., 2021).

La creación de sistemas de recomendación plantea el reto permanente de encontrar un equilibrio entre proporcionar recomendaciones precisas y personalizadas, lo que implica recopilar y analizar grandes cantidades de datos de los usuarios, y proteger la privacidad de los usuarios, para lo que es necesario limitar la recopilación y el uso de datos personales (Müllner, 2023).

2.1.1.3. Estándar MARC

Son las siglas de Machine Readable Cataloging (Catalogación legible por máquina), que indica que una máquina puede leer y comprender los datos contenidos en un registro de catalogación. Esta iniciativa comenzó con el "Proyecto Piloto MARC" en 1968 y constituye una importante herramienta destinada a lograr niveles óptimos de normalización, compatibilidad y transferencia de información

bibliográfica legible por ordenador. Permite la cooperación y el intercambio de recursos entre bibliotecas (BNE, 2021).

Este formato, hasta 1994, estaba compuesto por diversos formatos que se aplicaban a diferentes tipos de materiales, como publicaciones periódicas (1969), libros (1970), mapas (1970), películas (1971), manuscritos (1973) y música (1976). En otras palabras, se seguía la misma metodología para la catalogación de estos materiales. Posteriormente, surgieron propuestas para la integración de estos formatos, y finalmente, en 1994, se publicó una edición que incluía todos los cambios aprobados, garantizando así la unificación de los formatos. Esto significa que se estableció un único formato simple y claro para gestionar una variedad de recursos que una unidad de información pueda contener (Gavilán, 2008).

- **MARC-21**

El ultimo hito del avance del Formato es la armonización de los formatos MARC para dar paso al formato MARC-21. Esto significa que se unieron muchos formatos que se tenía como por ejemplo libros, publicaciones seriadas, para todas estas clases de recursos se unieron en uno solo y también se unieron formatos que se habían creado en otros países. Todos los países fueron armando sus propias normas para poder normalizarlas que permitieron el uso adecuado para conformar registro de excelente calidad (Gavilán, 2008).

MARC-21 ha avanzado de acuerdo con las necesidades de compatibilidad con la normativa RDA. Donde se incluyó nuevos campos

y subcampos para facilitar la aplicación de nuevos requerimientos como por ejemplo la descripción bibliográfica de recursos digitales.

- **UNIMARC**

La Federación Internacional de Asociaciones de Bibliotecarios y Bibliotecas (IFLA) creó este formato para facilitar el intercambio de datos bibliográficos a nivel internacional. Se basa en los principios de París, que establecen los estándares para la descripción bibliográfica, y se utiliza para describir una variedad de recursos, como libros, artículos, grabaciones sonoras, videos, mapas, etc. Un registro bibliográfico de cada recurso contiene información sobre el título, autor, fecha de publicación, tema, etc. (IFLA, 2001).

Se utiliza en bibliotecas de todo el mundo, y facilita el intercambio de información bibliográfica entre diferentes instituciones. Esto permite a los usuarios acceder a recursos de bibliotecas de todo el mundo, independientemente de su ubicación. Este formato se utiliza para describir personas, entidades corporativas y temas, facilitando la colaboración entre bibliotecas de todo el mundo, además es una herramienta crucial para mejorar el acceso a la información a nivel global en bibliotecas (IFLA, 2001).

Características

- Es un formato internacional facilita el intercambio de información bibliográfica entre bibliotecas.
- Es un formato flexible, que puede utilizarse para describir una amplia gama de recursos.

- Es un formato estandarizado, lo que implica la fácil comprensión de los registros bibliográficos.
- Es un formato actualizado, que se mantiene al día con los últimos avances en la catalogación bibliográfica.

- **Sistema decimal de clasificación decimal Dewey**

Es un sistema de clasificación bibliográfica creado por Melvil Dewey en 1876 que organiza los materiales de la biblioteca en categorías numéricas. Es uno de los sistemas de clasificación más utilizados en todo el mundo y se usa para organizar y clasificar libros y otros materiales en bibliotecas públicas, escolares y académicas (Patterson, 2020).

La clasificación decimal de Dewey (DDC) se basa en el sistema decimal, lo que significa que los números de clasificación se componen de una serie de dígitos, cada dígito representa un grado específico. Los números de clasificación del se utilizan para organizar los libros y otros materiales en las bibliotecas. La *tabla 1* muestra las principales categorías de este sistema decimal.

2.1.1.4. Métricas de evaluación

Son medidas utilizadas para evaluar la calidad de las recomendaciones proporcionadas por un sistema de recomendación, estas se utilizan para medir el rendimiento del sistema y determinar si las recomendaciones son útiles, relevantes y precisas para el usuario, también pueden incluir medidas de precisión, como la tasa de aciertos y la tasa de error, así como medidas de satisfacción del usuario, como la retroalimentación explícita e implícita (Silveira *et al*., 2019).

Tabla 1

Categorías principales del sistema de clasificación decimal Dewey

Num.	Clasificación	Temas
000	Obras Generales	Enciclopedia, periodismo, bibliotecas, museos.
100	Filosofía y Psicología	Sentimientos, emociones
200	Religión	Biblia, cristianismo y otras religiones
300	Ciencias Sociales	Economía, educación, adivinanzas, leyendas, días feriados.
400	Lenguaje	Diccionarios, lenguaje de señas
500	Ciencias Naturales	Matemática, plantas, animales, volcanes, huracanes, experimentos, electricidad.
600	Tecnología y ciencia de la salud	Cuerpo humano, medicina, enfermería, inventos, agricultura, crianza de mascotas, carros, trenes, aviaciones, cocina.
700	Arte y deporte	Pintura, fotografía, artesanías, manualidades, dibujo, música, deportes, juegos.
800	Literatura	Poesía, cuentos, novelas y teatro.
900	Geografía e Historia	Atlas, mapas banderos, castillos, bibliografías, países, grupos étnicos.

Nota: Obtenida de Shewale (2021).

- **Precisión**

La precisión se refiere al número de elementos correctamente clasificados en la lista de recomendaciones. Esta métrica evalúa la proporción de elementos en la lista de recomendaciones que resultan de interés o satisfacción para el usuario. Su cálculo implica dividir el número

de elementos recomendados que son relevantes para el usuario entre el número total de elementos recomendados, según se indica en la ecuación descritas por (Silveira *et al*., 2019).

$$precisión = util(R_u) = \frac{|C_u \cap R_{u|}}{|R_u|}$$

- **Recall**

Se define como la proporción entre los artículos recomendados que un usuario ha consumido y el número total de artículos consumidos. En términos más simples, este indicador mide la efectividad del sistema al sugerir artículos que el usuario ya había consumido previamente (Silveira *et al*., 2019).

$$recall = \ util(R_u) = \frac{|C_u \cap R_{u|}}{|C_u|}$$

- **Exactitud**

Es una métrica empleada para evaluar la capacidad del modelo de recomendación en acertar las preferencias del usuario de manera precisa. Su cálculo se realiza al dividir el número de predicciones correctas entre el número total de predicciones realizadas. Esta métrica reviste gran importancia al evaluar la calidad de los modelos de recomendación basados en contenido, ya que dichos modelos se fundamentan en la similitud existente entre los elementos recomendados y aquellos que el usuario ha consumido en el pasado (Gil & Seguro, 2022).

$$exactitud = \frac{TP + TN}{Total}$$

- **Diversidad**

Es una medida que se utiliza para evaluar la variedad de elementos recomendados por el modelo de recomendación. Un modelo con una alta diversidad clasificará correctamente un mayor número de instancias de diferentes. La métrica propuesta para calcular la diversidad considera una disimilitud entre los elementos de la lista de recomendaciones, basándose en la definición dada. La función $d(i,j)$ se utiliza para determinar la distancia entre los elementos i y j de la lista de recomendaciones R_u (Silveira *et al*., 2019).

$$Div(R_u) = \sum_{i \in R_u} \sum_{j \in R_u \; i \neq j} d(i,j)$$

Intra-List Diversity

Por otro, Aróstegui Martín, (2020) define a la ILD como una medida que calcula la distancia media entre los elementos recomendados por pares, y estas distancias se calculan en función a las características de los ítems. Para calcular el ILD, es necesario establecer una medida de distancia $d(i,j)$. Esta medida de distancia es personalizable y forma parte integrante de la métrica (Castells et al., 2015). ILD de define como:

$$ILD = \frac{1}{|R|(|R|-1)} \sum_{i \in R_u} \sum_{j \in R_u} d(i,j)$$

- **Cobertura**

La cobertura mide la proporción de productos o servicios disponibles que están cubiertos por las recomendaciones. Entonces, un modelo con una alta cobertura recomendará un mayor número de

productos o servicios disponibles respecto al total. La ecuación de cobertura se calcula como el número de productos o servicios recomendados (C) dividido por el número total de productos (U) disponibles (Silveira *et al*., 2019).

$$cobertura = \frac{C}{U}$$

- **Serendipia**

La serendipia es una medida empleada para evaluar la capacidad del modelo de recomendación en proporcionar recomendaciones de elementos que resulten sorprendentes y valiosos para el usuario. La serendipia se sustenta en tres conceptos fundamentales: novedad, inesperabilidad y utilidad. La novedad se relaciona con la habilidad del modelo de recomendación de sugerir elementos que el usuario no ha consumido previamente. En consecuencia, la serendipia se refiere a la probabilidad de que el usuario descubra un producto o servicio que sea de su agrado (Castells *et al*., 2022).

- **Novedad**

Es un indicador empleado para valorar la eficacia del sistema de recomendación en la tarea de sugerir elementos que el usuario aún no ha explorado previamente. Se obtiene al dividir la cantidad de sugerencias que presentan elementos inexplorados por parte del usuario entre el total de sugerencias realizadas. La relevancia de la novedad radica en su capacidad para aumentar la satisfacción del usuario con el sistema de recomendación al introducirle opciones novedosas que puedan despertar su interés (Silveira *et al*., 2019).

- **Inesperabilidad**

Se trata de un indicador empleado para evaluar la habilidad de un modelo de recomendación en ofrecer sugerencias que resulten inesperadas o difíciles de predecir por parte del usuario. Su cálculo se basa en la utilización de medidas de distancia o similitud entre los elementos propuestos. (Silveira *et al*., 2019).

2.1.2. Modelo de datos orientado a grafos

2.1.2.1. Modelo de datos

Un modelo de datos es una representación abstracta de la estructura de los datos que se empleará en una base de datos. Este modelo describe la forma en que los datos se organizan, cómo se establecen las relaciones entre ellos y la manera en que se accede a la información almacenada (Jyothi, 2020). Es decir, define la estructura, restricciones y reglas para organizar y manipular datos en una base de datos u otro sistema de información esta se puede representar mediante diagramas, como diagramas entidad-relación o diagramas UML, que proporcionan una representación visual de los datos y sus relaciones *(Zhou et al*., 2020).

Un modelo de datos es una representación abstracta de los datos que se almacenan en un sistema informático y estos son utilizados para organizar los datos de manera que sean fáciles de entender y acceder (Khalil & Belaissaoui, 2022). A continuación, se mencionan algunos de los modelos de datos más reconocidos:

- Modelo de Datos Jerárquico

- Modelo de Datos de Red
- Modelo de Datos Relacional
- Modelo de Datos Orientado a Objetos
- Modelo de Datos Orientado a Documentos
- Modelo de Datos orientado a Grafos

2.1.2.2. Grafos

Un grafo es una estructura de datos que consta de un conjunto de nodos y un conjunto de relaciones que conectan pares de nodos, estas relaciones representan trayectorias o conexiones entre los nodos, así mismo, ambos conjuntos deben ser finitos y cualquiera de los dos puede estar vacío, Koffman & Wolfgang (2008). Los grafos en su forma más simple, se pueden resumir en forma de una matriz de conexión o de adyacencia, el conjunto completo de todas las conexiones por pares define la topología del grafo, proporcionando un mapa completo de todas las relaciones entre nodos y conexiones (Sporns, 2018).

Adicionalmente, un grafo es una estructura matemática que se compone de un conjunto de nodos y un conjunto de aristas que establecen conexiones entre estos nodos. Los nodos representan objetos o entidades, mientras que las aristas denotan relaciones entre dichos objetos (Khalil & Belaissaoui, 2022). Los grafos se utilizan para representar los datos y sus relaciones y cada nodo representa un objeto o entidad, y las relaciones entre los nodos representan las conexiones o interacciones entre ellos (Jyothi, 2020).

Se emplea para visualizar las conexiones entre objetos y se estructura mediante un conjunto de nodos, también conocidos como vértices, y un conjunto de aristas o bordes que conectan estos nodos. Los nodos representan objetos, mientras que las aristas reflejan las relaciones entre ellos. Estas relaciones pueden ser tanto dirigidas, con un sentido específico, como no dirigidas, sin una orientación definida (Zhou *et al*., 2020).

- **Grafos dirigidos**

Un grafo dirigido se configura como una estructura en la cual las aristas muestran una dirección específica. Este atributo implica que una arista vincula un nodo de origen con un nodo de destino, estableciendo, por lo tanto, la orientación del flujo de información o la relación entre los nodos (Zhou *et al*., 2020). Las relaciones en un grafo dirigido se representan mediante flechas que indican la dirección de la conexión. Por ejemplo, en una red social, la relación de "amistad" entre dos personas podría representarse como un grafo dirigido, ya que la amistad puede ser unilateral y no necesariamente recíproca (Jyothi, 2020). Es crucial recalcar que en un grafo dirigido, las conexiones entre nodos poseen un sentido definido, en contraste con un grafo no dirigido, donde las relaciones son bidireccionales (Khalil & Belaissaoui, 2022).

- **Grafos no dirigidos**

Es una representación en la que las aristas carecen de orientación. Esto implica que una arista puede enlazar un nodo con otro, y viceversa, sin una dirección definida (Khalil & Belaissaoui, 2022). En este tipo de

grafo, las aristas no poseen una dirección inherente, de manera que cada arista conecta dos nodos sin especificar una orientación particular. En consecuencia, la relación entre dos nodos se caracteriza por su simetría, lo que significa que la arista que une los nodos no alberga una dirección específica (Zhou *et al*., 2020).

La relación entre dos nodos es recíproca y no tiene un sentido específico. Además, las relaciones se representan mediante líneas que conectan los nodos. Por ejemplo, en una red de transporte, una relación de "conexión" entre dos ciudades podría ser un grafo no dirigido, ya que la conexión es recíproca y no tiene un sentido específico (Jyothi, 2020).

- **Algoritmos de grafos**

Los algoritmos de grafos son procedimientos matemáticos que operan sobre grafos y se utilizan con el propósito de abordar una amplia diversidad de problemas. Estos algoritmos permiten realizar diversas tareas en grafos, como encontrar la ruta más corta entre dos nodos, detectar ciclos o determinar si un grafo es conexo (Khalil & Belaissaoui, 2022). En resumen, los algoritmos de grafos constituyen un conjunto de técnicas y métodos utilizados en disciplinas como la informática, la ingeniería, la física y las ciencias sociales para analizar y resolver problemas relacionados con grafos (Zhou *et al*., 2020: Jyothi, 2020).

- **Aplicaciones de los grafos**

Algunos ejemplos de aplicaciones de grafos incluyen motores de recomendación, motores de búsqueda, análisis de redes sociales, análisis de fraude, análisis de datos de IoT, análisis de datos de bioinformática,

entre otros. Entonces, la tecnología de grafos es especialmente útil para estas aplicaciones porque permite modelar y analizar datos complejos y relaciones entre ellos de manera más eficiente y efectiva que otros enfoques de modelado de datos (Graph everywhere, 2020).

2.1.2.3. Modelo de bases de datos orientado a grafos

Se refiere a un modelo de datos que hace referencia a una representación en forma de grafo, en la que los nodos simbolizan entidades y las aristas representan las relaciones entre ellas. Este enfoque resulta especialmente beneficioso para datos que involucran relaciones y dependencias complejas, como es el caso de las redes sociales, los grafos de conocimiento y los sistemas de recomendación (Zhou *et al.*, 2020).

En este modelo, se representa la información a través de nodos y las relaciones que existen entre ellos, lo que proporciona una manera más intuitiva de expresar las conexiones complejas entre los datos. En contraposición a almacenar los datos en tablas, como en las bases de datos relacionales tradicionales, aquí se almacenan en forma de grafos, lo que brinda una mayor flexibilidad y capacidad de escalabilidad en la administración de datos (Jyothi, 2020).

Es un tipo de base de datos que utiliza estructuras de grafos para almacenar, mapear y consultar datos estos datos se representan como nodos, aristas y propiedades, que pueden usarse para modelar relaciones complejas entre entidades, se utilizan a menudo junto con redes neuronales de grafos para realizar tareas de aprendizaje automático (Zhou *et al*., 2020).

Son sistemas de almacenamiento NoSQL destinados a gestionar datos en forma de grafos, donde se representan los datos mediante nodos y sus relaciones, y se aplican algoritmos de grafos para el procesamiento y análisis de la información. Estas bases de datos son especialmente útiles para modelar y analizar datos que tienen relaciones complejas y no estructuradas (Jyothi, 2020).

2.1.3. Herramientas de desarrollo

La selección de un Sistema de Gestión de Bases de Datos (DBMS) adecuado resulta crítica en la implementación de un motor de recomendación. Dentro de este contexto, se encuentran disponibles diversos sistemas de bases de datos orientados a grafos, con enfoques tanto nativos como relacionales para el almacenamiento de datos. Es fundamental evaluar la disponibilidad y el respaldo de controladores oficiales que faciliten su integración con lenguajes de programación actuales. Asimismo, se torna esencial considerar la versatilidad y apoyo brindado por los lenguajes de programación. En esta investigación, se optó por Neo4j, y se ha utilizado del lenguaje de programación orientado a objetos, Python. La implementación de un servidor HTTP se llevó a cabo a través del Framework Flask.

2.1.3.1. Python

Según Guido van Rossum, creador de Python, la historia del lenguaje comienza en diciembre de 1989, cuando estaba trabajando en el Centro para la Investigación de Álgebra Computacional (CWI) en los Países Bajos. En ese momento, estaba buscando un proyecto de pasatiempo personal y decidió crear un nuevo lenguaje de programación,

Van Rossum quería crear un lenguaje que fuera fácil de aprender y usar, pero que también fuera potente y flexible. Estaba inspirado en otros lenguajes de programación, como ABC y Modula-2.

El primer lanzamiento de Python fue en febrero de 1991, con la versión 0.9.0. La versión 1.0 se lanzó en enero de 1994, y el desarrollo del lenguaje continuó desde entonces. Van Rossum fue el único desarrollador de Python hasta 1995, cuando se unió a la Corporación Nacional de Investigación de Internet (CNRI). En 2001, dejó CNRI para trabajar en Google, donde sigue siendo el principal desarrollador del lenguaje.

Es un lenguaje de programación de alto nivel, interpretado y de propósito general, este es conocido por su sintaxis clara y legible, lo que lo hace fácil de aprender y usar, Python es utilizado para procesar los datos y aplicar los algoritmos de similitud de Jaccard y coseno en el sistema de recomendación (Sukestiyarno *et al*., 2023). La importancia del lenguaje Python por Bárbaro *et al*., (2017), menciona que:

- Python se puede utilizar para una amplia gama de tareas, incluyendo desarrollo web, ciencia de datos, aprendizaje automático e inteligencia artificial.
- Es fácil de aprender.
- Python tiene una sintaxis simple y concisa, lo que lo hace fácil de aprender para los principiantes.
- Es un lenguaje gratuito y de código abierto: Python es gratuito y de código abierto, lo que significa que cualquiera puede utilizarlo y contribuir a su desarrollo.

- Tiene una gran comunidad de usuarios: Python tiene una gran comunidad de usuarios que proporciona apoyo y recursos para los desarrolladores.
- En el libro "*Python Crash Course*" de Eric Matthes,
- Es un lenguaje de programación interpretado: Python no requiere ser compilado antes de ser ejecutado, lo que lo hace más rápido y fácil de desarrollar.
- Tiene una gran biblioteca estándar: Python viene con una gran biblioteca estándar que incluye funciones y módulos para una amplia gama de tareas.
- Es compatible con una amplia gama de plataformas: Python se puede ejecutar en una amplia gama de plataformas, incluyendo Windows, macOS y Linux.
- Es un lenguaje general, fácil de aprender, gratuito y de código abierto, con una gran comunidad de usuarios.

2.1.3.2. Flask

Es un framewok web minimalista para Python que se destaca por ser simple, extensible y fácil de aprender. A diferencia de los frameworks web monolíticos, Flask es un conjunto de herramientas que permiten crear aplicaciones web de manera ligera. Aunque es menos completo que otros como Django, su flexibilidad lo hace adecuado tanto para proyectos simples como complejos. Además, se caracteriza por su sintaxis simple y concisa, lo que lo hace fácil de aprender. Puede utilizarse para crear desde páginas web de inicio y blogs hasta aplicaciones de comercio electrónico

o aplicaciones personalizadas que se conecten a bases de datos o utilicen servicios web (Bárbaro *et al*., 2017).

2.1.3.3. Neo4j

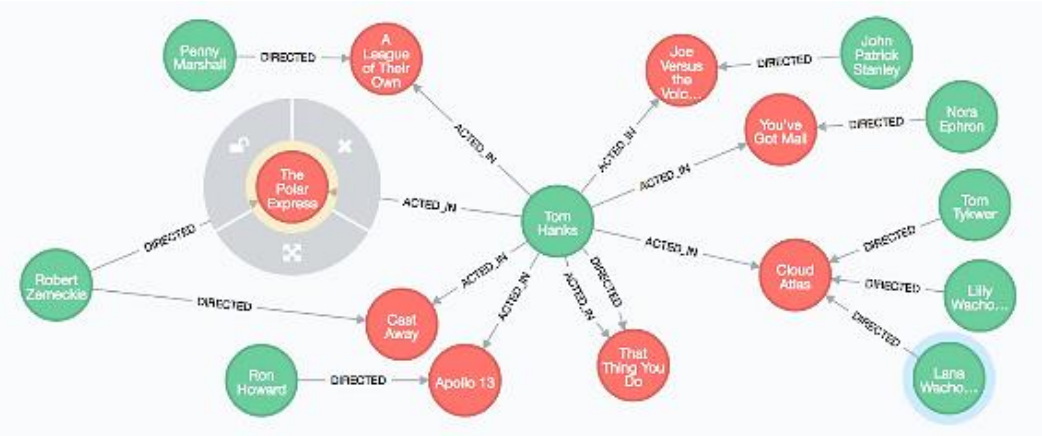
Se trata de una base de datos de grafos de código abierto que se utiliza para el almacenamiento, gestión y consulta de datos altamente interconectados, es capaz almacenar datos en nodos y relaciones, lo que permite una representación más natural y eficiente de datos altamente enlazada (Sukestiyarno *et al*., 2023)**.**

Este sistema hace uso de un lenguaje de consultas propio llamado $Cypher$, basado en patrones de grafos, para llevar a cabo la consulta y análisis de datos. Además, se caracteriza por ser escalable y adecuado para entornos distribuidos, lo que lo convierte en una opción viable para la gestión de grandes volúmenes de dato (Jyothi, 2020).

En resumen, se trata de una base de datos de grafos de código abierto, distribuida y escalable que almacena datos en forma de grafos, que son estructuras de datos que consisten en nodos y bordes. Los nodos representan entidades, como personas o cosas, y los bordes representan las relaciones entre esas entidades como lo indica en el artículo de (Besta *et al*., 2023).

Figura 3

Grafos en una base de datos en Neo4j



Nota: Imagen extraída de Rendón (2022)**.**

Características de Neo4J

- Es una base de datos orientado a grafos NoSQL.
- Los nodos son registros en la base de datos de grafos y las relaciones contienen información, lo que permite vincular nodos en redes semánticamente ricas.
- Proporciona un lenguaje de consulta de grafos llamado "$Cypher$" que se basa en muchas fuentes y se parece a SQL, pero con una representación icónica de patrones en el grafo.
- Es altamente flexible y permite a los usuarios actualizar nuevas relaciones y nuevos datos.
- Tiene módulos de transacciones que incluyen registro de transacciones y gestión de transacciones para garantizar el ACID de la transacción.

2.2. ANTECEDENTES

2.2.1. Antecedentes internaciones

Jethva *et al*., (2022) proponen un método de recomendación de libros basados en intereses de usuarios, para ello emplean tres metodologías. El filtrado colaborativo que se basó en la similitud entre los usuarios y sus preferencias de lectura. Para el análisis de datos, han utilizado el algoritmo de los K vecinos más cercanos (KNN). Por otro lado, el filtrado basado en contenido que sugiere ítems en función de la relación entre los perfiles de usuario y las características de los libros. Para analizar el contenido de los libros, han utilizado los algoritmos TF-IDF y Count-Vectorizer. El enfoque de filtrado basado en híbridos que combina ambas metodologías. Los datos se han obtenido desde un archivo separado por comas. El principal desafío fue lidiar con un conjunto de datos extenso, lo que dificultó la tarea de filtrar información y reducir los datos para que fueran más manejables. Los autores han concluido que el enfoque híbrido ha obtenido mejores resultados en términos de recomendación de libros basados en los intereses del usuario, quedando aún desafíos relacionados con el arranque en frío y la dispersión de datos.

Mutalib *et al*. (2020), implementaron un motor de recomendación impulsado por grafos para películas. Su objetivo fue el desarrollo de un motor de recomendación para un portal de películas. Para lograrlo, el motor de recomendación ha empleado un enfoque de filtrado colaborativo basado en la similitud de las calificaciones otorgadas a las películas. Además, ha utilizado un enfoque de filtrado basado en contenido para ofrecer recomendaciones fundamentadas en los géneros de las películas. Para el almacenamiento y

procesamiento de los datos han utilizado una base de datos de grafos cuyos datos fueron modelados como $(User) - [HAS\_RATED] \rightarrow (Movie) - [HAS\_KEYWORD] \rightarrow (Keyword), (Person) - [WRITER\_OF|PRODUCED|DIRECTED\_IN|ACTED\_ID] \rightarrow (Movie) - [HAS\_GENRE] \rightarrow (Genre)$. En sus conclusiones, los autores han resaltado la utilidad de las bases de datos de grafos para gestionar datos extensos y voluminosos en el proceso de recomendaciones. Asimismo, han sugerido que el sistema podría perfeccionarse aún más a través de la implementación de análisis de sentimientos para evaluar las reseñas, además de la exploración de enfoques híbridos y sistemas de recomendación basados en conocimientos.

Qassimi *et al*., (2021) plantearon una técnica de recomendación de libros basado en grafos de folksonomía. El objetivo fue mejorar el rendimiento y la efectividad de las recomendaciones. Para lograr esto, han incorporado un algoritmo de clustering espectral en la etapa de preprocesamiento para agrupar los datos según su similitud. La estructura de representación de los datos se ha organizado de la siguiente manera: $(User) - [HAS\_RATED] \rightarrow (Book) - [HAS\_TAG] \rightarrow (Tag)$. La técnica ha empleado información contextual del dominio de la aplicación y ha analizado la relación de folksonomía o clasificación colaborativa para construir grafos y etiquetas que han conformado la base de conocimiento del sistema. Para generar las recomendaciones, el algoritmo ha utilizado el grafo semántico para encontrar similitudes entre los libros y las preferencias de los usuarios. El data set utilizado fue el Goodbooks-10k, y las métricas de evaluación incluyeron precisión y recall. Como resultado, los autores han concluido que su técnica supera a otros sistemas híbridos basados en contenido y filtrado colaborativo en términos de precisión y efectividad. No

obstante, han reconocido ciertas limitaciones, como la necesidad de contar con un mayor volumen de datos etiquetados, así como la disponibilidad y confiabilidad de dichos datos.

Sen *et al*. (2021) presentaron un modelo de minería de reglas de asociación en una base de datos de grafos con el propósito de generar recomendaciones. Su enfoque principal consistió en desarrollar un modelo de recomendación que pudiera identificar el conjunto de productos que ejercían influencia en las ventas de otros productos. El proceso metodológico se dividió en tres etapas: la adquisición de datos, el modelado y la importación de estos en la base de datos, y el análisis de datos para la generación de recomendaciones. Se empleó un data set del sector de belleza de Amazon, que incluía 167,709 usuarios, 29,004 productos y 252,073 reseñas. Estos datos se sometieron a un proceso de preprocesamiento, convirtiéndolos de un formato de texto plano a uno estructurado, utilizando scripts de Python para su importación en el modelo de la base de datos. Los datos se modelaron como nodos y relaciones como $(User) - [WROTE] \rightarrow (Review) - [REVIEW\_OF] \rightarrow (Product)$. Finalmente, se propuso la técnica de minería de reglas de asociación para identificar el conjunto de productos influyentes en las ventas de otros productos. Además, se planteó una combinación de técnicas de filtrado y agrupamiento para reducir la complejidad y mejorar la eficiencia del proceso de análisis. Como resultado, los autores concluyeron que la base de datos de grafos proporcionó una mayor flexibilidad en el modelado y análisis de datos, y que la técnica de recomendación propuesta demostró ser más eficiente en términos de complejidad computacional y tiempo requerido en comparación con el algoritmo de Apriori.

Tian *et al.*, (2019) propusieron un método de recomendación personalizada para bibliotecas universitarias basado en un algoritmo de recomendación híbrido que fusionó el filtrado colaborativo y el enfoque basado en contenido. El filtrado colaborativo utilizó la similitud entre usuarios y la matriz de puntuación usuario-elemento para predecir preferencias. El enfoque basado en contenido se basó en las características de los elementos y la matriz de características para predecir preferencias. Ambos métodos se combinaron para obtener resultados precisos. Se utilizaron registros de 43,153 usuarios, 200,593 préstamos y 76,638 libros, y la plataforma Big Data Spark para implementar el sistema. La métrica de precisión evaluó el rendimiento del sistema, y se concluyó que los métodos híbridos brindan recomendaciones más precisas que enfoques puramente colaborativos o basados en contenido. Además, se destacó que la dispersión de la matriz usuario-libro fue del 99.99%, indicando que la mayoría de los usuarios había calificado solo una fracción de los libros disponibles.

2.2.2. Antecedentes nacionales

Amézquita Llerena (2020), utilizó un gestor de base de datos orientado a grafos en el procesamiento de información relacionada a la medicina. Representó los nodos y relaciones sobre el modelo de datos como $(Enfermedades) - [causadoPor] \rightarrow (Causas)$ conectándolos entre ellas a partir de síntomas, enfermedades y sus relaciones, la información procesada fue de 6221 artículos, el cual fue obtenida de dos repositorios médicos reconocidos mediante la técnica de Web Crawling. El autor concluyó que un gestor de base de datos es ideal para modelos estructuradas con un alto nivel de relación de datos, logrando resultados complejos con consultas simples.

Huamán Acuña (2019) diseñó e implementó un sistema de recomendación de libros acorde a los conocimientos previos del usuario. Las recomendaciones fueron construidas a partir de los conceptos y temas descritos en los libros ordenándolos por ontologías, estos temas fueron estructurados jerárquicamente creando perfiles de usuario para enlazar los conocimientos previos y la disponibilidad de temas nuevos, utilizó un modelo de datos RDF, consultado a través del lenguaje OWL, así mismo, redujo la amplia variedad de libros disponibles por recomendar, usando puntajes de valoración obtenidas de los cometarios. Se enfocó en generar una mayor precisión de la recomendación. Manejó la técnica de recomendación basada en ontologías asociada a tesauros. Con las pruebas echas concluyó que, ayudó a los usuarios a seleccionar el libro más adecuado a sus requerimientos, el 95.38% de las recomendaciones ayudó a comprender los temas o conceptos que los usuarios necesitaban, así mismo, permitió la retroalimentación a partir de los comentarios calificados por los otros usuarios.

Mamani Chile, (2022) presentó un modelo de recomendación de libros basado en algoritmos de similitud con el objetivo de optimizar el rendimiento del sistema. Para ello, implementó la técnica de filtrado colaborativo, que permitió identificar las preferencias de los usuarios y de otros usuarios con características similares. El conjunto de datos consistió en 1466 calificaciones de 989 libros otorgadas por 661 usuarios. Los algoritmos utilizados incluyeron el método de los vecinos más cercanos (KNN) y el algoritmo de similitud del coseno. La métrica de evaluación aplicada fue la matriz de confusión, que arrojó promedios esperados de precisión (0,91) y sensibilidad (0.91). El autor también discutió las limitaciones del sistema, como la dependencia de la retroalimentación explícita de los usuarios

y la falta de consideración de la disponibilidad de los libros en la biblioteca, lo que limitó la diversidad de los resultados. Concluyó que se puede lograr la optimización del rendimiento del modelo de recomendación de libros basados en algoritmos de similitud sin requerir una gran capacidad de cómputo, dependiendo del tamaño de la muestra del conjunto de datos.

2.3. MARCO CONCEPTUAL

Sistema de Recomendación: Es una herramienta que define criterios y valoraciones sobre los datos de los usuarios con el fin de predecir recomendaciones de valor (Graph everywhere, 2019).

Grafo: Es un modelo que está representado por nodos que se relacionan con otros nodos mediante aristas (Graph everywhere, 2019).

Python: Es un lenguaje multiplataforma de código abierto que permite una lectura y escritura clara con aplicaciones en inteligencia artificial, Big Data, machine learning y data science, entre otros temas de apogeo (Santander, 2021).

Neo4J: Programa de base de datos orientados a grafos que extrae valor adicional de los datos de alto rendimiento, ágil y flexible. Su uso está en temas de detección de fraude, recomendaciones en tiempo real y gestión de centros de datos (BBVA, 2018).

Grafo: es un conjunto no vacío de objetos o entes físicos que tienen relación entre ellos. Está formado por vértices, muchas veces también llamados nodos (siendo estos los que representan a los objetos), y un conjunto de arcos que representan las relaciones entre los vértices, también llamadas aristas (INAOE, 2010).

Normalizador Léxico: Al igual que el analizador léxico, es una herramienta que se utiliza para convertir el texto de un lenguaje natural a un formato estándar. Esto se hace para facilitar el procesamiento del texto por parte de los ordenadores (JAUME, 2012).

Machine Learning: es un campo de la inteligencia artificial que se centra en la creación de sistemas que pueden aprender y mejorar sin ser programados explícitamente. Los sistemas de aprendizaje automático se entrenan en grandes cantidades de datos, y son capaces de identificar patrones y tendencias que los humanos no pueden (Hinestroza, 2018).

Etiquetado Colaborativo: Consiste en el proceso en el que los usuarios trabajan juntos para asignar etiquetas a los datos. Esto se hace a menudo para crear grandes conjuntos de datos etiquetados que se pueden utilizar para entrenar modelos de aprendizaje automático. Estos pueden ser realizados por personas o máquinas. Si son personas se aprovecha la experiencia de muchas personas y ayuda a reducir sesgos humanos (Hinestroza, 2018).

Minería de Datos: Es el proceso de extraer patrones, tendencias de grandes conjuntos de datos. Estos patrones y tendencias pueden ser utilizados para tomar decisiones, mejorar procesos. Esa tecnología puede usar técnicas como el análisis de regresión, clasificación, asociación, agrupamiento y detección de anomalías (Rojas, 2019).

CAPÍTULO III

MATERIALES Y MÉTODOS

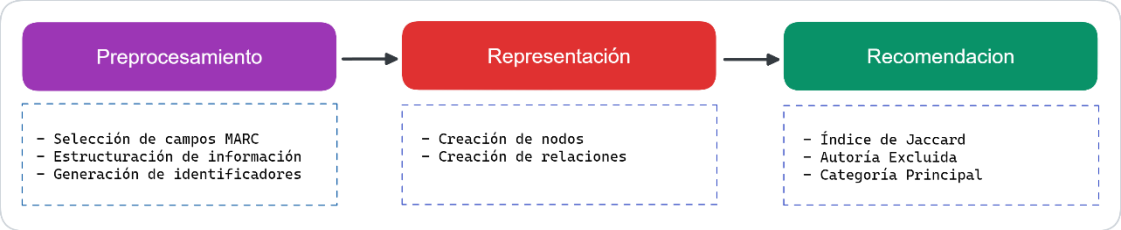
En este capítulo, se introduce el enfoque de modelo de datos basado en grafos que se emplea para afrontar la tarea de procesar recomendaciones de libros. La primera sección se detalla la organización de la información y la preparación de los datos, mientras que la segunda parte se concentra en la representación de la información de la biblioteca a través de la estructura de nodos y relaciones en un grafo. Para concluir, en la tercera sección, se detallan los algoritmos de similitud propuestos.

Esta investigación tiene un enfoque cuantitativo, de nivel aplicativo, utilizando algoritmos de similitud para generar recomendaciones de libros. Así mismo, se opta por un diseño preexperimental de una sola medición $(G - X - O)$ debido a la carencia de motores de recomendación basados en el estándar MARC para su comparación. La investigación es de corte transversal, debido que se lleva a cabo en un único punto temporal. Involucra como muestra la información de 2,265 libros procedentes de dos catálogos en línea. Las recomendaciones son posteriormente evaluadas utilizando la métrica de diversidad.

La *Figura 4* detalla las etapas seguidas para llevar a cabo la implementación del modelo de base de datos orientado a grafos con el propósito de generar recomendaciones de libros que produzcan diversidad en los resultados. En el primer paso, se procedió a la selección y estandarización de la información necesaria. Luego, se transformó la información preprocesada en una estructura de nodos y relaciones en un modelo de datos de grafo. Finalmente, se aplicaron los métodos de recomendación utilizando este modelo de datos, como se ilustra en la figura siguiente.

Figura 4

Implementación del modelo de datos para procesar recomendaciones



Elaboración propia.

3.1. PREPROCESAMIENTO DE LA INFORMACIÓN

La información primordial necesaria para la organización de la base de datos se centra en los campos MARC que representan la información de un libro, excluyendo cualquier dato redundante o de poca utilidad. Estos campos son los encargados de identificar un libro, tales como el título, el autor, el ISBN, el lugar y año de publicación, entre otros. La extracción de esta información se realiza directamente a partir del formato de catalogación del libro, según se describe en detalle en la *Tabla 2*.

Después de identificar los datos fundamentales del libro, se inicia el proceso de estructuración de los campos MARC en cinco categorías principales: Título, Clasificación, Responsabilidad, Editorial y Títulos Seriados, con el fin de lograr una descomposición minuciosa de la información que el libro contiene. El alcance de esta organización se puede consultar en la *Tabla 3*.

Tabla 2

Campos seleccionados del estándar MARC

Código	Campo	Subcampo
008	Códigos de información de longitud fija	-
020	Número internacional normalizado para libros (ISBN)	-
082	Clasificación decimal Dewey	$a: Numero de clasificación decimal Dewey. $m: Notación interna.
100	Encabezamiento de autor personal	$b: Nombre de persona. $c: Títulos y palabras asociadas al nombre. $d: Fechas asociadas al nombre.
245	Título y declaración de responsabilidad	$a: Título principal. $b: Parte restante del título.
250	Edición	$a: Mención de edición.
260	Publicación, distribución, etc.	$a: Lugar de publicación. $b: Nombre de la editorial. $c: Fecha de publicación.
300	Descripción física	$a: Extensión. $b: Otros detalles físicos. $c: Dimensiones.
490	Serie	$a: Título de la serie. $v: Designación numérica/secuencial del volumen.
504	Nota de bibliografía	$a: Nota de bibliografía, etc.
505	Nota de contenido	$a: Nota de contenido con formato preestablecido.
650	Encabezamiento de materia de asignatura	$a: Encabezamiento temático.

Elaboración propia.

Tabla 3

Organización de campos MARC en entidades principales

N	Entidad	Campos MARC
1	Datos del título	008, caracteres en posiciones específicas para indicar el tipo de material. 020, ISBN. 245 $a, título principal del documento. 245 $b, parte adicional del título que sigue al título principal. 250 $a, descripción de la edición del material. 260 $c, fecha de publicación del material. 504 $a, nota de bibliografía. 505 $a, nota de contenido principal. 650 $a, término principal de encabezamiento de materia que describe el contenido temático del material.
2	Datos de la clasificación	082 $a, Número principal de clasificación Dewey Decimal. 650 $a, Término de encabezamiento de materia.
3	Datos de responsabilidad	100 $b, Nombre personal del autor. 100 $c, Títulos, prefijos, sufijos u otras palabras asociadas al nombre del autor. 100 $d, fecha de nacimiento o de fallecimiento del autor.
4	Datos de editorial	260 $a, lugar de publicación del material. 260 $b, nombre de la editorial responsable de la publicación.
5	Datos de títulos Seriados	490 $a, título de la serie a la que pertenece el material. 490 $v, número o volumen específico de la serie al que pertenece el material.

Elaboración propia.

La estructuración de la entrada para el modelo de base de datos se basa en las entidades principales, utilizando el formato de datos estándar denominado Notación de Objetos JavaScript (JSON), el cual es legible para los seres humanos y de fácil accesibilidad para los sistemas informáticos. La *Figura 5* detalla la estructura.

Figura 5

Estructura de los datos de entrada para el modelo de base de datos

```
{
    "title": {...},
    "classification": {...},
    "person": [...],
    "publisher": [...],
    "serialTitle": {..},
    "copy": [...]
}
```

Elaboración propia.

De esta manera, cada elemento del formato JSON se relaciona con las entidades principales que agrupan los campos MARC descritos en la Tabla 3. Cada libro incluye información correspondiente a sus datos específicos, clasificación decimal Dewey, responsables, editoriales, título en serie y, además, los ejemplares. Estos detalles se describen a continuación:

- ***Title***: Es un objeto que contiene información detallada y específica sobre el libro, como el título, la edición, la fecha de publicación, el número ISBN y otros atributos relevantes. Este objeto se caracteriza por la atomización de la información, ya que todos los datos se representan en un solo objeto y se estructuran de acuerdo con los campos definidos en la norma MARC.
- ***Classification*:** Contiene información sobre la clasificación decimal Dewey, el cual se asigna a cada libro en función del tema que aborda. Esta información se relaciona con varios libros, pero cada libro tiene asignado un único número de clasificación. Su representación se realiza mediante un objeto que incluye el código y la descripción correspondientes a la clasificación.
- ***Person*:** Contiene información acerca de las entidades que tienen una relación de responsabilidad con el libro, tales como el autor, coautor, editor, ilustrador,

entre otros. Cada una de estas entidades puede estar relacionada con varios libros. Por tanto, la relación es $N:1$, y su representación se realiza a través de una matriz de objetos.

- ***Publisher*:** Contiene información de las entidades encargadas de la producción, publicación y distribución de los libros. Esta información se relaciona con múltiples libros, lo que implica una relación de muchos a muchos. Su representación se realiza a través de un arreglo de objetos.
- ***Serial Title*:** Contiene información del título en común que relaciona a una serie de libros, como el tomo, volumen, etc. Esta información es relevante para la identificación y organización de los libros que conforman una serie. Su representación se hace a través de un objeto conteniendo el número y el título.
- ***Copy*:** Contiene información de la localización física de un documento, como la signatura y la ubicación de sus ejemplares. Esta información es única para cada copia del libro, por lo que su relación con el título es de muchos a uno. Su representación se realiza a través de un arreglo de objetos.

Con la finalidad de elevar la calidad de la representación de la información en el modelo de datos basado en grafos, se ejecuta un procedimiento de preparación previa de los datos. Para este propósito, se ha desarrollado un normalizador léxico que se encarga principalmente de la limpieza de datos, generando identificadores únicos (Hash) e índices para eliminar posibles duplicaciones de entidades en el grafo. Una de las funciones clave del normalizador léxico es la generación de identificadores únicos para cada objeto descrito en el formato JSON, lo que permite una identificación inequívoca de cada entidad en el modelo de datos. Las tareas específicas del normalizador léxico son las siguientes:

- **Eliminar signos de puntuación, espacios en blanco extra y caracteres especiales:** Esto implica eliminar signos de puntuación como comas, puntos, espacios en blanco y caracteres especiales como símbolos innecesarios entre palabras en un enunciado.
- **Reemplazar caracteres Unicode compuestos:** Implica descomponer caracteres acentuados en su componente individual para ser tratados como secuencia de caracteres simples.
- **Transformar texto a minúsculas:** Esto implica convertir todas las palabras a minúsculas a fin de asegurar la eficiencia en las búsquedas.
- **Eliminar palabras vacías o Stopwords:** Esto implica eliminar palabras que no agregan significado al texto, como artículos, preposiciones y pronombres. Acorde a los títulos de libros, las palabras vacías más usadas son:

Tabla 4

Lista de StopWords

StopWords
a, e, o
además, así, mas
cómo, con, com
de, del, desde
el, ella, ello, al
en, entre, es, esta, estas, este, esto, estos
ha, han, he
la, las, les, lo, los
me, mi, muy, no
para, pe, por, que
sé, si, sido, son, su, sus
tal, te, tus, tu
un, una
y, ya, yo

Elaboración propia.

3.1.1. Creación de identificadores únicos

Implica generar un valor único a partir de una cadena de caracteres con el fin de identificar de manera inequívoca un elemento. A continuación, se describe el algoritmo utilizado para el preprocesamiento:

Algoritmo 1: Preprocesamiento de identificador único.

Entrada: Texto no procesado

Salida: Identificador de índice

```
para i ← letras en texto no procesado hacer
    si i es mayúscula entonces
        cambiar a minúscula
    fin si
    si i es carácter Unicode compuesto entonces
        descomponer i
    fin si
    si i es carácter especial entonces
        borrar i
    fin si
fin para
para i ← palabras en texto no procesado hacer
    si i es StopWord o espacio entonces
        borrar i
    fin si
fin para
devolver texto no procesado
```

Elaboración propia.

3.1.2. Creación de identificadores de índice

Indica generar un valor de índice que agrupa las palabras clave desde una cadena de caracteres para que un motor de búsqueda pueda rastrear y encontrar resultados, Tal como se describe en el *algoritmo 2*.

Algoritmo 2: Preprocesamiento de identificador de índice.

Entrada: Texto no procesado

Salida: Identificador de índice

```
1:  para i ← letras en texto no procesado hacer
2:      si i es mayúscula entonces
3:          cambiar a minúscula
4:      fin si
5:      si i es carácter Unicode compuesto entonces
6:          descomponer i
7:      fin si
8:      si i es carácter especial entonces
9:          borrar i
10:     fin si
11: fin para
12: para i ← palabras en texto no procesado hacer
13:     si i es StopWord o espacio extra entonces
14:         borrar i
15:     fin si
16: fin para
17: Identificador de índice ← ordenar palabras en texto no procesado
18: devolver identificador de índice
```

Elaboración propia.

3.1.3. Normalización de datos

Implica limpiar cadenas de caracteres para asegurar que los caracteres sean interpretados correctamente y no causen problemas con la codificación, el cual se detalla a continuación:

Algoritmo 3: Preprocesamiento de datos.

Entrada: Texto no procesado
Salida: Valor normalizado
1: **para** *i* ← *letras en texto no procesado* **hacer**
2: **si** *i es comilla simple o espacio extra* **entonces**
3: *borrar i*
4: **fin si**
5: **fin para**
6: **devolver** *texto no procesado*

Elaboración propia.

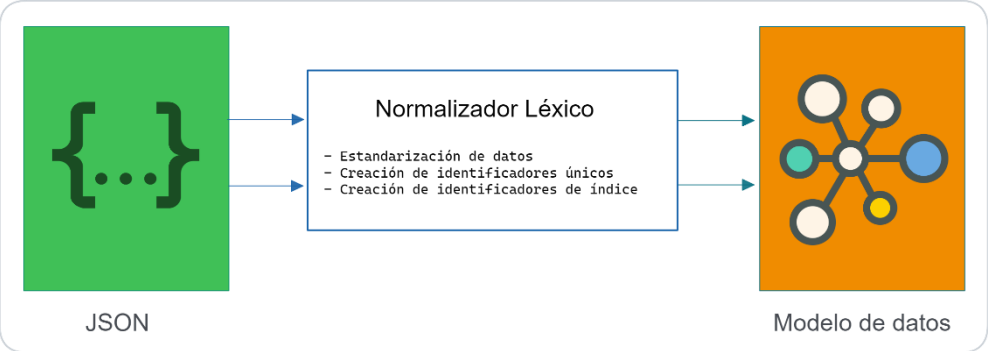
3.2. REPRESENTACIÓN DE INFORMACIÓN

La información bibliográfica se representa en el modelo de base de datos a partir de la entrada JSON derivada de los campos MARC, seguida del proceso de preprocesamiento efectuado por el normalizador léxico. El proceso de representación de la información bibliográfica en un modelo de base de datos orientado a grafos se visualiza en la *Figura 6*.

Las seis propiedades definidas dentro del JSON representan un nodo en el modelo de base de datos orientado a grafos. El nombre de la etiqueta del nodo concuerda con el nombre de la propiedad correspondiente en el JSON. Asimismo, las conexiones entre estos elementos representan las relaciones y se tiene en cuenta la dirección de estas. La *Figura 7* ilustra los elementos del JSON representados como nodos y relaciones de un grafo.

Figura 6

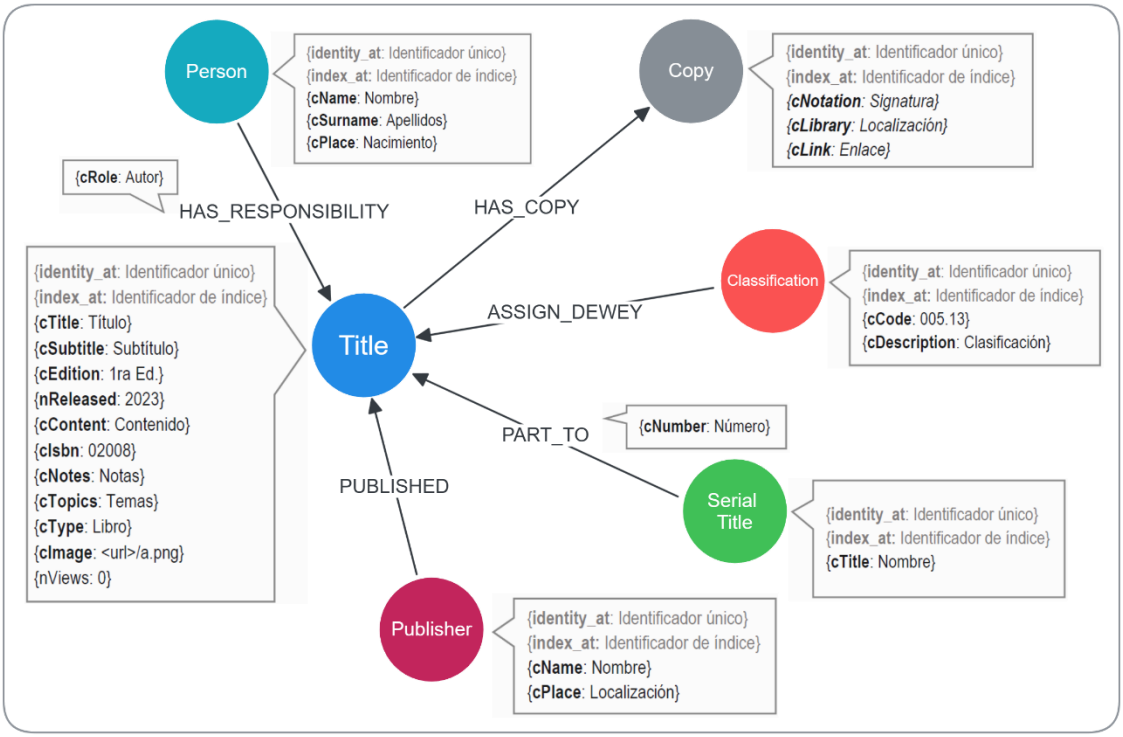
Proceso de representación de información bibliográfica



Elaboración propia.

Figura 7

Representación de información bibliográfica en nodos y relaciones



Elaboración propia.

Así mismo, cada nodo representado en el modelo de datos almacena un conjunto de propiedades cuya equivalencia corresponde a los campos MARC en la entrada JSON. Además, las relaciones entre nodos también se identifican mediante un nombre y un

sentido, teniendo en cuenta que algunas relaciones tienen propiedades adicionales, como en los casos de $PART\_TO$ y $HAS\_RESPONSIBILITY$. Por último, el normalizador léxico agrega las propiedades de identificador único y de identificador de índice para cada nodo, y realiza la normalización de las demás propiedades de los nodos y relaciones.

3.3. MÉTODOS DE RECOMENDACIÓN BASADAS EN SIMILITUD

En esta sección se presentan tres algoritmos diferentes para abordar el desafío de generar recomendaciones de libros basadas en similitud, con el objetivo de lograr diversidad en los resultados. Se considera el enfoque de similitud, en el cual se analizan las diversas conexiones entre los nodos para calcular la semejanza entre los libros.

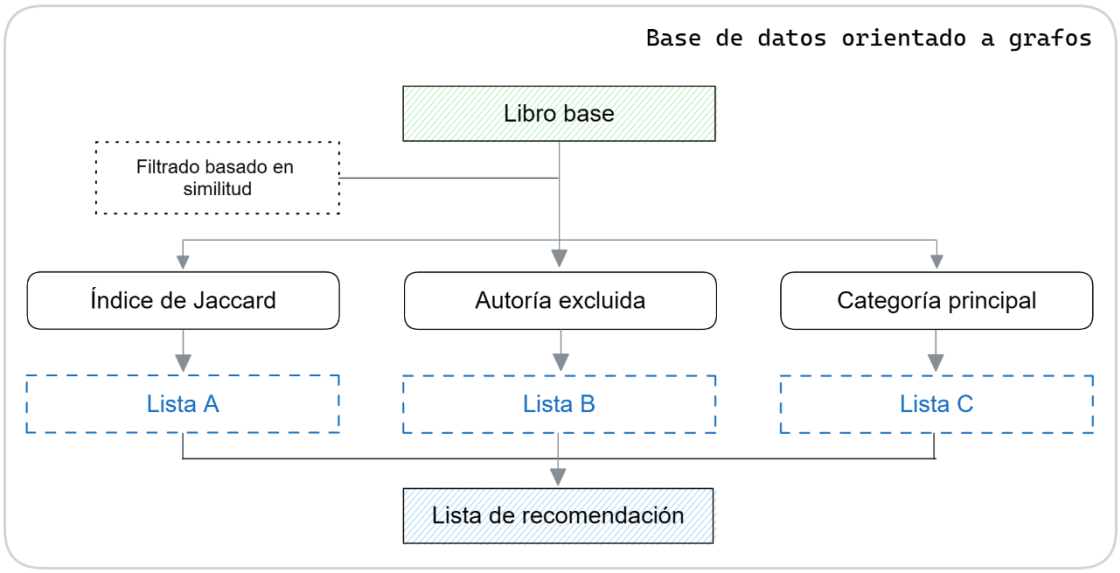
Así mismo, se utilizan los términos "título" y "libro" de manera equivalente. Esto se debe a que, en el modelo de datos utilizado, un libro se representa mediante un conjunto de nodos relacionados, como los nodos de título, autores o editoriales. Sin embargo, es importante resaltar que, en el modelo de datos, un nodo de tipo $Title$ representa un único registro irrepetible, este tipo de nodo se centra en la información específica relacionada con el título del libro, sin considerar automáticamente todos los demás nodos relacionados que son parte del libro. Esto significa que, en el contexto de la recomendación de libros, se focaliza en el nodo de tipo $Title$ para generar recomendaciones, teniendo en cuenta su información particular.

La *Figura 8* muestra la arquitectura para el proceso de recomendación de libros. En este proceso, la entrada que desencadena las recomendaciones es un libro, el cual, a nivel del modelo de datos, se representa mediante su título. Los métodos de recomendación propuestos: índice de Jaccard, autoría excluida y por categoría principal del Sistema Decimal Dewey asignada al título, utiliza la técnica de filtrado basado en similitud. Estos métodos generan una lista de recomendaciones independientes, con libros

similares al mismo libro de entrada. Estas listas parciales se combinan para proporcionar una lista de recomendación final que incluye resultados diversos. Es decir, se toman en cuenta los resultados de los tres métodos y se combinan para ofrecer una recomendación final más completa y variada.

Figura 8

Arquitectura del procesamiento de las recomendaciones



Elaboración propia.

3.3.1. Similitud por índice de Jaccard

Propuesto y desarrollado por Jaccard (1901), define el índice como una medida de similitud entre dos conjuntos, que reduce como la proporción entre el tamaño de la intersección de los conjuntos y el tamaño de la unión de estos. Costa (2021) expresa el índice de Jaccard en notación de teoría de conjuntos como:

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$$

Donde A y B son dos conjuntos, |A| número de elementos del conjunto A y |B| el número de elementos del conjunto B.

El índice de Jaccard tiene un valor entre *0* y *1*, donde *0* indica que los conjuntos son completamente diferentes y *1* indica que los conjuntos son idénticos. Un valor intermedio indica que los conjuntos tienen algún grado de similitud.

En esta investigación, se emplea el índice de Jaccard aplicado a grafos para evaluar la similitud entre conjuntos de nodos. En este contexto, el índice de Jaccard se describe como la relación entre el número de nodos vecinos compartidos por dos nodos y el número total de nodos vecinos diferentes que tienen esos dos nodos. En términos matemáticos, el índice de Jaccard entre dos nodos A y B se formula como:

$$J(A,B) = \frac{|n(A) \cap n(B)|}{|n(A)| + n(B) - |n(A) \cap n(B)|}$$

Donde:

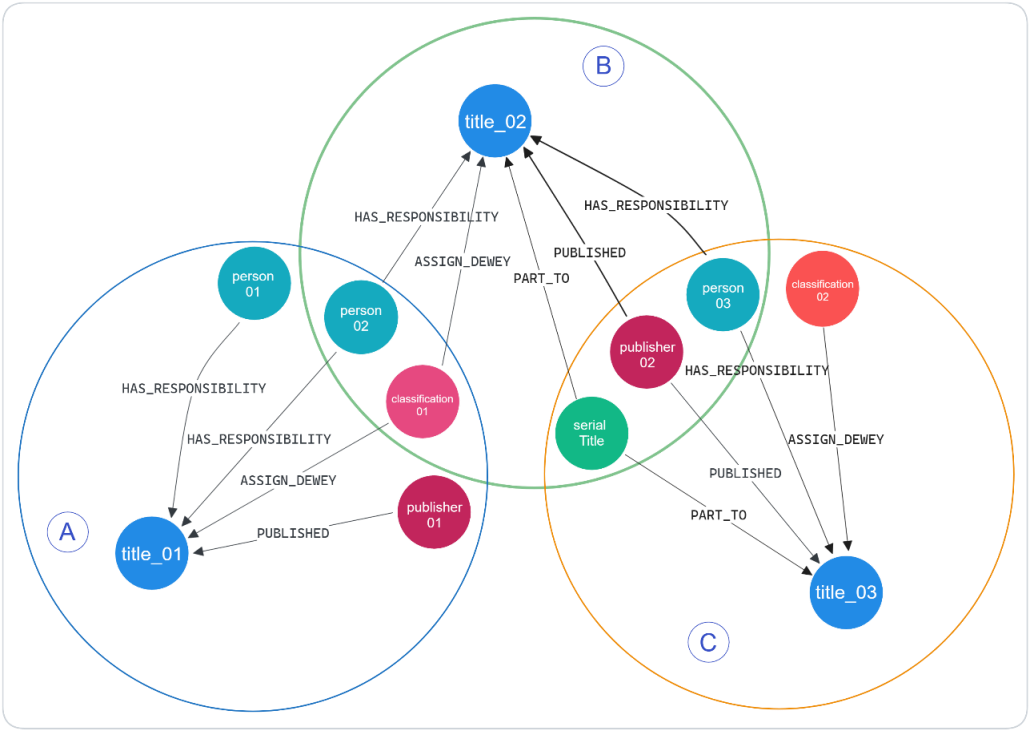
$n(A)$ y $n(B)$ son los conjuntos de vecinos de los nodos A y B, respectivamente.

La *Figura 9* muestra tres conjuntos diferentes de nodos, cada uno con información bibliográfica independiente. Sin embargo, estos conjuntos comparten nodos cuya disposición sugiere una situación ideal para el estudio de la métrica de similitud por índice de Jaccard.

Los nodos con la etiqueta $Title$ constituyen el fundamento de las recomendaciones. Es decir, se toman nodos de este tipo como entrada y, tras aplicar la métrica de similitud, se obtiene una lista de nodos del mismo tipo como resultado. En seguida, se presentan dos casos en los que aplica la fórmula matemática del índice de Jaccard expuestas en la *Figura 9*.

Figura 9

Distribución de nodos para una recomendación por el índice de Jaccard



Elaboración propia.

- **Caso 1:** Nodo de entrada $title\_01$.

Se identifica que el nodo $title\_01$ pertenece al conjunto A, el cual comparte dos nodos con el conjunto B, cuyo nodo base es $title\_02$. Por lo tanto, la similitud se obtiene de la siguiente manera:

$$Similitud(A,B) = \frac{|n(title\_01) \cap n(title\_02)|}{|n(title\_01) \cup n(title\_02)|}$$

$$Similitud(A,B) = \frac{2}{7} = \mathbf{0.2857}$$

La similitud entre estos dos conjuntos es del 28,57%, lo cual indica una coincidencia parcial entre los nodos de los conjuntos. Es relevante destacar que el conjunto A no comparte nodos con otros conjuntos, lo que implica que no hay nodos similares o superpuestos en otros conjuntos de nodos. En base a esta

información, se considera el nodo $title\_02$ como parte de la lista de recomendación cuando el nodo $title\_01$ es la entrada.

- **Caso 2:** Nodo de entrada *title_02*

En este caso, se identifica que el nodo $title\_02$ pertenece al conjunto B, el cual comparte dos nodos con el conjunto A (evaluado anteriormente en el caso 1) y tres nodos con el conjunto C, cuyo nodo principal es $title\_03$. En consecuencia, en este caso se tiene lo siguiente:

$$Similitud(A,B) = \frac{2}{7} = \mathbf{0.2857}$$

$$Similitud(B,C) = \frac{|n(title\_02) \cap n(title\_03)|}{|n(title\_02) \cup n(title\_03)|}$$

$$Similitud(B,C) = \frac{3}{6} = \mathbf{0.5}$$

La similitud entre el conjunto B y C es del 50%, lo cual claramente supera de manera significativa la similitud entre el conjunto A y B. Por lo tanto, se genera una lista de recomendaciones donde el nodo $title\_03$ se coloca en primer lugar, seguido del nodo $title\_01$. Por último, se establece un umbral mínimo con la intención de descartar similitudes cercanas a 0%.

3.3.2. Similitud por autoría excluida

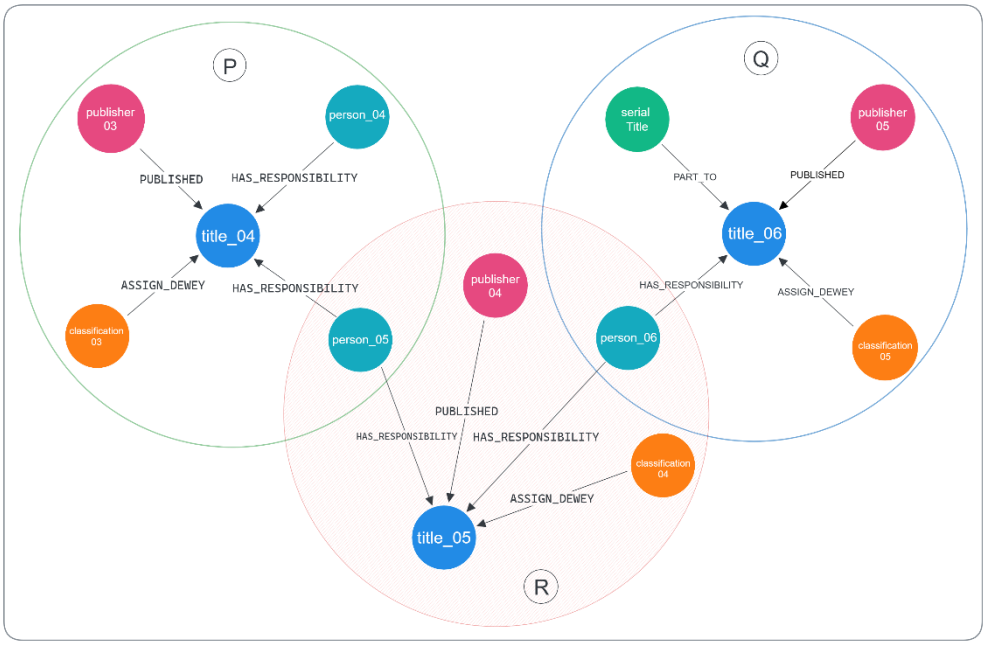
El índice de Jaccard se centra en el conjunto vecino de un nodo y no considera el contexto global del grafo en su totalidad. Esta limitación puede llevar a casos en los que no se encuentre ninguna similitud entre conjuntos de nodos, lo que resultaría en la falta de recomendaciones de libros.

Para abordar esta limitación, se emplea la estrategia de recomendación basada en coautoría excluida. El algoritmo consiste en identificar títulos a partir de un libro base de entrada, cuyos autores no hayan participado directamente, pero sus coautores sí. Esta estrategia tiene como objetivo ampliar el contexto de la recomendación al considerar los coautores del libro de entrada, más allá de su vecindad inmediata en el grafo. Esto implica buscar nodos de tipo $Title$ que sean similares y estén relacionados con los nodos de coautores del libro de interés, incluso si no están directamente conectados en el grafo.

La *Figura 10* muestra tres conjuntos de nodos independientes, dos de los cuales pertenecen a diferentes vecindades (P y Q), mientras que el tercero actúa como un puente para conectar estos dos conjuntos de nodos. Esta ilustración ejemplifica cómo la estrategia de autoría excluida permite descubrir libros similares a través de la relación de coautoría, superando las limitaciones de la vecindad inmediata en el grafo.

Figura 10

Distribución de nodos para una recomendación por autoría excluida



Elaboración propia.

Los nodos de tipo $Title$ a menudo están asociados con varios responsables que desempeñan roles específicos, como autor o director. Estos responsables son considerados elementos fundamentales para generar la recomendación basada en similitud. Es decir, similitud entre los nodos de tipo $Title$ se determina en función de los responsables asociados a cada título. A continuación, se proporciona una descripción detallada del caso expuesto en la *Figura 10*:

- **Caso:** Nodo de entrada $title\_04$

Siguiendo el algoritmo propuesto, se detalla los pasos para procesar la recomendación:

a) Obtener los autores asociados al nodo base de entrada, es decir, los autores que participaron en la creación de dicho libro. En este caso, el

libro base se identifica con el nodo $title\_04$ y los autores están representados por los nodos $person\_04$ y $person\_05$.

b) Recorrer todos los nodos de tipo $Title$ que tienen relación directa con los autores del nodo base de entrada. Estos nodos representan otros libros escritos por los mismos autores. En el ejemplo, se llega al nodo $title\_05$ a través del nodo autor $person\_05$.

c) Para cada nodo de tipo $Title$ identificado en el paso anterior, obtener los autores asociados a dicho nodo. En este caso, se obtiene el nodo $person\_06$.

d) Por cada autor obtenido en el paso anterior, verificar que no tengan relación directa con el nodo base de entrada. Esto implica que no hayan participado en la creación del libro $title\_04$.

e) Por cada uno de los autores identificados en el paso anterior, encontrar sus coautores, es decir, los autores que han colaborado con ellos en la creación de otros libros. En este caso, se encuentra el nodo $person\_06$.

f) Retornar los nodos de tipo $Title$ que tienen relación directa con los coautores encontrados en el paso anterior. Estos nodos representan otros libros en los que los coautores han trabajado juntos. En el caso específico, se encuentra el nodo $title\_06$.

De esta manera, se genera una lista de recomendación a partir del método de autoría excluida, que incluye los conjuntos de nodos P y Q.

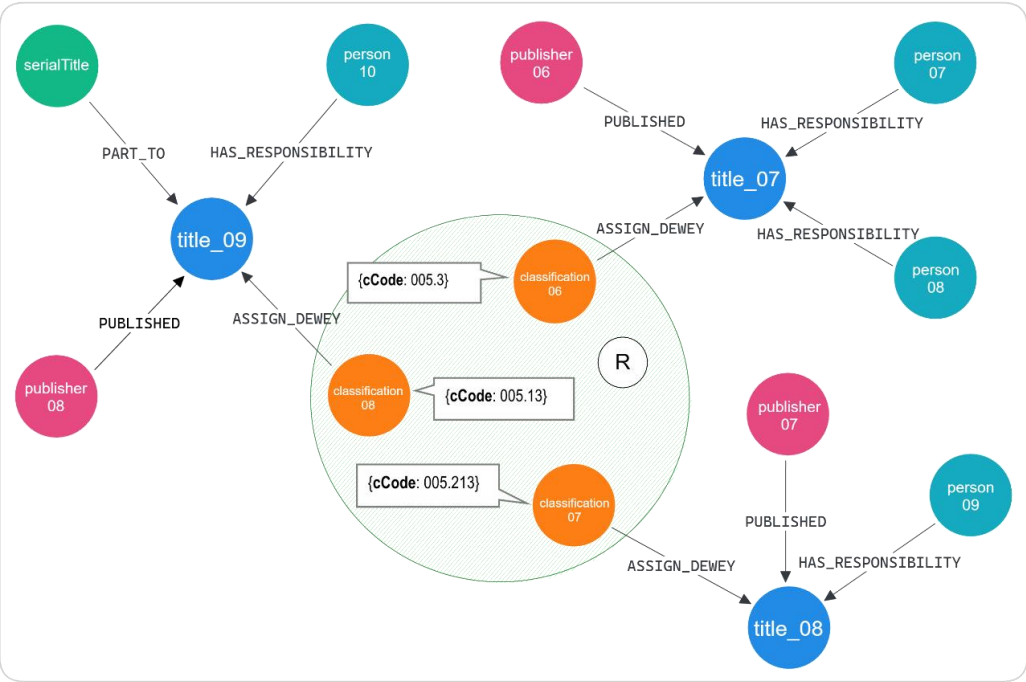
3.3.3. Similitud por categoría principal

La estrategia de recomendación basada en similitud por categoría principal utiliza el sistema de clasificación Dewey asociado a cada título de libro. Este

sistema organiza el conocimiento en diez categorías principales, las cuales se dividen en subcategorías más específicas. Cada subcategoría se representa mediante un número decimal único. En el modelo de datos, cada título de libro está siempre asociado a un número de clasificación Dewey, el cual pertenece a una categoría principal compartida por otros libros. En el proceso de recomendación, se considera la similitud entre los libros que comparten la misma categoría principal en su clasificación Dewey, lo que genera recomendaciones acordes a la afinidad temática general entre los libros. La *Figura 11* se presenta tres conjuntos de nodos que son completamente diferentes en términos de similitud entre sí. A pesar de esta falta de conexión directa, todos los nodos pertenecen a la misma categoría principal de conocimiento, la cual está clasificada por el sistema decimal Dewey. Esto significa que, aunque los nodos no estén vinculados entre sí, comparten una clasificación común, el cual se considera en el proceso de recomendación para identificar libros que pertenecen a la misma área temática.

Figura 11

Distribución de nodos para una recomendación por categoría principal.



Elaboración propia.

El conjunto R está compuesto por nodos de tipo $Classification$ que no tienen ninguna relación directa entre sí. Sin embargo, todos los nodos en el conjunto R pertenecen a la misma categoría principal de conocimiento. Esto indica que, a pesar de la falta de conexiones directas entre los nodos, comparten una categoría común basada en el sistema Dewey, lo cual se utiliza como técnica en el proceso de recomendación para identificar libros que pertenecen a la misma área temática principal. A continuación, se explica un caso práctico:

- **Caso:** Nodo de entrada $title\_09$

Se siguen los siguientes pasos para generar recomendaciones utilizando la técnica propuesta:

a) Identificar el nodo de clasificación que tiene una relación directa con el nodo base de entrada. En este caso, se trata del nodo $classification\_08$.

b) Obtener la categoría principal a partir del número de clasificación decimal Dewey asignado al nodo de clasificación.

c) Obtener todos los nodos de tipo clasificación identificada que pertenecen a la misma categoría principal.

d) Para cada nodo de tipo clasificación identificado en el paso anterior, retornar los nodos de tipo $Title$ que tienen una conexión directa con ese nodo. En este caso, los nodos $title\_07$ y $title\_09$ cumplen esta condición.

De esta manera, los nodos resultantes forman parte de la lista de recomendaciones generada utilizando la técnica de similitud por categoría principal.

3.4. CONSIDERACIONES DEL CAPÍTULO

En este capítulo, se ha explorado la extensión de cómo se representa la información bibliográfica en un modelo de datos basado en grafos, y se han expuesto los enfoques utilizados para el procesamiento de recomendaciones de libros, utilizando el estándar MARC como base. Se han introducido tres métodos de preprocesamiento diseñados para generar recomendaciones variadas. Asimismo, se han establecido tanto los fundamentos teóricos como las bases prácticas necesarias para llevar a cabo los experimentos que se abordarán en el próximo capítulo. En dicho capítulo, se ofrecerá un análisis detallado de los experimentos realizados y se presentarán los resultados obtenidos tras la aplicación de los métodos propuestos.

CAPÍTULO IV

RESULTADOS Y DISCUSIÓN

4.1. IMPLEMENTACIÓN DEL PREPROCESAMIENTO

En la etapa inicial, se creó la entrada JSON de cada libro para el modelo de base de datos, empleando información bibliográfica recopilada de dos catálogos en línea de bibliotecas públicas que siguen el formato de catalogación MARC. Este procedimiento resultó en la formación de seis secciones distintas que proporcionan una descripción detallada de las características de cada libro.

Estas secciones incluyen información como el título del libro, las personas involucradas en su creación, la clasificación decimal Dewey asignada, las editoriales asociadas a su publicación, el título de la serie, y las copias disponibles del libro. La *Figura 12* muestra el resultado obtenido en formato JSON, que contiene la información bibliográfica adquirida a través del estándar MARC del libro.

El fragmento de código que se presenta en la *Figura 13* corresponde a la implementación del normalizador léxico para el preprocesamiento de los datos. Este normalizador se encarga de eliminar espacios extra, generar identificadores únicos y campos de índice para los motores de búsqueda.

Para realizar la normalización de los datos se utiliza la librería "*normalize*" de Python que permite descomponer los caracteres acentuados en sus componentes individuales. Además, se utiliza la librería "*hashlib*" para generar un hash único que servirá como identificador único para cada uno de los objetos descritos en el formato JSON. Las funciones que realizan son las siguientes tareas:

Figura 12

JSON con información bibliográfica MARC


```json
{
    "title": {
        "cTitle": "Machine learning y deep learning",
        "cSubtitle": "Usando python, scikit y keras",
        "cEdition": "Primera edición",
        "nReleased": 2021,
        "cContent": "Introducción -- Datasets -- Regresión -- Clasificación ...",
        "cIsbn": "978-958-792-145-8",
        "cNotes": "Ilustraciones, gráficos",
        "cTopics": "Procesamiento de datos",
        "cType": "Libro",
        "cImage": "https://biblioteca.unap.edu.pe/opac_css/getimage.php?url_ ...",
    },
    "classification": {
        "cCode": "006.31",
        "cDescription": "Procesamiento de datos"
    },
    "person": [
        {
            "cName": "Jesús",
            "cSurname": "Bobadilla",
            "cPlace": "",
            "cRole": "Autor"
        }
    ],
    "publisher": [
        {
            "cName": "RA-MA Ediciones de la U",
            "cPlace": "Bogotá"
        }
    ],
    "serialTitle": {},
    "copy": [
        {
            "cNotation": "006.31 B66",
            "cLibrary": "Bib. Esp. Ing Sistemas",
            "cLink": "https://biblioteca.unap.edu.pe/opac_css/"
        }
    ]
}
```


Elaboración propia.

- La función "$GenerateIdentifier$" recibe una cadena de texto y genera un identificador único para ella, luego de eliminar los signos de puntuación y los caracteres especiales, descomponer los caracteres acentuados y eliminar las palabras vacías.
- La función "$GenerateIndex$" recibe una cadena de texto y genera un campo de índice para ella, luego de eliminar los signos de puntuación y los caracteres especiales, descomponer los caracteres acentuados y eliminar las palabras vacías.
- La función "$FormateText$" recibe una cadena de texto y elimina los espacios extra para su posterior normalización.

Las salidas del preprocesamiento de una oración aplicadas sobre el normalizador léxico se muestran en la *Figura 14*.

Figura 14

Fragmento de código del normalizador léxico

```python
def GenerateIdentifier(self, cString:str) -> str:
    cString = normalize("NFD", cString.lower())
    cString = "".join([char for char in cString if char.isalnum() or char.isspace()])
    cString = "".join([word for word in cString.split() if word not in self.__StopWords and len(word) > 1]).strip()
    arrBytes = cString.encode('utf-8')
    objHash = hashlib.sha256(arrBytes)
    return objHash.hexdigest()

def GenerateIndex(self, cString:str) -> str:
    cString = normalize("NFD", cString.lower())
    cString = "".join(char for char in cString if char.isalnum() or char.isspace())
    arrWords = cString.split()
    arrWords = [word for word in arrWords if word not in self.__StopWords and len(word) > 1]
    return " ".join(sorted(arrWords)).strip()

def FormateText(self, cString:str) -> str:
    if(cString.strip() == ""):
        return ""
    return " ".join(cString.split()).strip()
```

Elaboración propia.

Figura *13*

Preprocesamiento de una oración

```bash
Input: ' 978-958-792-145-8  Machine learning y deep learning  Usando python, scikit y keras Primera edición 2021 '
> 4fbde0249d558eea6e54da5881cff10cee6d998612343388f48383af9c5bd96a
> 2021 9789587921458 deep edicion keras learning learning machine primera python scikit usando
> 978-958-792-145-8 Machine learning y deep learning Usando python, scikit y keras Primera edición 2021
```

Elaboración propia.

4.2. IMPLEMENTACIÓN DE LA REPRESENTACIÓN

En esta sección, se detalla el método de representación de la información bibliográfica en un sistema de gestión de bases de datos orientado a grafos, en particular, en la herramienta Neo4j utilizada en este estudio. La entrada de datos se lleva a cabo a partir del formato JSON presentado en la *Figura 12.*

Cada propiedad del formato se representa como un nodo en el modelo de datos, y se le agregan dos nuevas propiedades: "$identity\_at$" y "$index\_at$". La primera corresponde al identificador único generado por el normalizador léxico, mientras que la segunda se refiere al campo de índice creado por la misma herramienta para facilitar las búsquedas. El fragmento de código que ilustra la representación de información bibliográfica en el modelo de datos basado en grafos se presenta en la *Figura 15*.

4.2.1. Nodos

Cada sección del JSON de entrada representa un nodo distinto en el modelo, lo que significa que dos registros idénticos pueden generar nodos diferentes. Para solucionar este problema, se utiliza la sentencia $MERGE$ el cual combina las funcionalidades de $CREATE$ y $MATCH$ del lenguaje $Cypher$, esto permite descartar nodos con propiedades idénticas y evitar duplicados. Además, se utiliza la propiedad "$identity\_at$", generada por el normalizador léxico, para identificar si el nodo se encuentra registrado en la base de datos.

4.2.2. Relaciones

Se sigue una estrategia que implica verificar la existencia de los nodos involucrados en la relación antes de crearlas. Para ello, se utiliza la sentencia $MERGE$ del lenguaje $Cypher$ para crear una nueva relación entre los nodos, si

aún no existe. Esta sentencia permite crear una nueva relación si no existe, o actualizar la relación existente, de esta manera se garantiza la integridad y unicidad de las relaciones en la base de datos orientada a grafos.

Figura 15

Fragmento de código para la carga de información bibliográfica

```
# ...
Node(
  ).Merge("Title", {"identity_at": cIdentity}
  ).OnCreate(
    {
      "cTitle" : cTitle,
      "...": ...,
      "index_at": cIndex
    }
  ).OnMatch(
    {
      "updated_at":datetime.now
    }
  )
# ...
Relationship(
  ).Merge(idClassification, "ASSIGN_DEWEY", idTitle)

```

Elaboración propia.

4.3. IMPLEMETACIÓN DE LOS MÉTODOS DE RECOMENDACIÓN

En esta sección se detalla la implementación de la recomendación mediante la construcción de consultas en el lenguaje *Cypher* de Neo4j, utilizando el patrón de diseño Builder en Python. El libro base de entrada para procesar las recomendaciones se selecciona a través de la propiedad $IdTitle$, que es un valor entero que representa el identificador único correspondiente al nodo $Title$. El motor de la base de datos orientada a grafos devuelve una lista de títulos recomendados que cumplen con los criterios de los métodos de recomendación propuestos. Este proceso se lleva a cabo de manera eficiente

y precisa gracias a la combinación de *Cypher* y *Python* en la construcción de las consultas.

4.3.1. Lista de recomendación por índice de Jaccard

En esta sección se presenta la implementación del primer método para generar la lista de recomendaciones.

La *Figura 16*, muestra un fragmento de código en *Python* que utiliza el lenguaje de consulta *Cypher* para construir consulta de recomendación. Esta consulta tiene como objetivo encontrar títulos que presenten una alta similitud con respecto a un título de entrada, utilizando el índice de Jaccard como métrica. La consulta se basa en las relaciones *HAS_RESPONSIBILITY*, *ASSIGN_DEWEY*, *PUBLISHED* y *PART_TO* para asociar nodos de tipo *Title* y encontrar los vecinos comunes. Es decir, la consulta busca nodos que compartan al menos una de estas relaciones con el nodo de entrada.

Se utiliza la cláusula *MATCH* para la localización de nodos de tipo *Title* con el alias *m*, los cuales representan el nodo de entrada. Se establece el alias *recom* para representar los títulos recomendados. La cláusula *WHERE* se utiliza para filtrar los nodos *m* basándose en su identificador *idTitle*. Luego, se aplica la cláusula *WITH* con el fin de transmitir los nodos *recom*, el cálculo de la intersección y la unión de los vecinos a la siguiente fase de la consulta. En esta fase posterior, se procede al cálculo del índice de Jaccard mediante la división del tamaño de la intersección entre el tamaño de la unión de los vecinos. Finalmente, los resultados se presentan en forma de una lista de títulos acompañados de sus respectivos índices de Jaccard.

Figura 16

Fragmento de código de la recomendación por índice de Jaccard

```
Match(
  ).Node("Title","m").LeftRelationship("HAS_RESPONSIBILITY | ASSIGN_DEWEY | PUBLISHED | PART_TO"
  ).Node("t").RightRelationship("HAS_RESPONSIBILITY | ASSIGN_DEWEY | PUBLISHED | PART_TO"
  ).Node("Title","recom"
).Where().Id("m", idTitle
).With(
  ).Node("m"
  ).And().Node("recom"
  ).And().Count("t").As("intersection"
  ).And().OnSet("m","HAS_RESPONSIBILITY | ASSIGN_DEWEY | PUBLISHED | PART_TO","mt").As("neighbors_m"
  ).And().OnSet("recom","HAS_RESPONSIBILITY | ASSIGN_DEWEY | PUBLISHED | PART_TO","recomt").As("neighbors_recom"
).With(
  ).Node("recom"
  ).And().Node("intersection"
  ).And().FromRaw("neighbors_m + [x IN neighbors_recom WHERE NOT x IN neighbors_m]").As("union"
).With(
  ).Node("recom"
  ).And().FromRaw("(1.0 * intersection) / SIZE(union)").As("jaccard"
).Select("recom.cTitle, jaccard")
```

Elaboración propia.

4.3.2. Lista de recomendación por autoría excluida

La implementación del método de recomendación basada por autoría excluida se muestra en la siguiente Figura:

Figura 17

Fragmento de código de la recomendación por autoría excluida

```
Match(
    ).Node("Title","m").LeftRelationship("HAS_RESPONSIBILITY"
    ).Node("person").RightRelationship("HAS_RESPONSIBILITY"
    ).Node("Title","other"
).And(
    ).Node("other").LeftRelationship("HAS_RESPONSIBILITY"
    ).Node("co").RightRelationship("HAS_RESPONSIBILITY"
    ).Node("Title","recom"
).Where(
    ).Id("m", idTitle
    ).And("m <> recom"
).Select("recom.cTitle")
```

Elaboración propia.

La *Figura 17* presenta un fragmento de código en $Python$ que construye una consulta en $Cypher$ para obtener títulos recomendados basados en la coautoría utilizando la relación $HAS\_RESPONSIBILITY$. La consulta comienza seleccionando el nodo de tipo $Title$ con un identificador único que representa el título de entrada. A continuación, se establece la relación con los nodos de tipo $Person$ y $Title$, lo que implica buscar libros recomendados cuyos coautores hayan colaborado en la creación del título de entrada.

Se utiliza la cláusula $MATCH$ para buscar nodos de tipo $Title$ con el alias m que representa el nodo de entrada, y se establece el alias $recom$ para representar los títulos recomendados. Se establece la relación $HAS\_RESPONSIBILITY$ entre los nodos $Title$ y $Person$ para identificar los autores del nodo de entrada. Luego,

se establecen una vez más la relación $HAS\_RESPONSIBILITY$ entre los nodos $other$ y co para identificar los coautores de otros títulos. Se utiliza la cláusula $WHERE$ para filtrar los nodos m según su identificador $idTitle$ y asegurar que el nodo de entrada no sea igual al nodo recomendado. Finalmente, se retorna una lista de recomendación con el campo $cTitle$ del nodo $recom$ como resultado.

4.3.3. Lista de recomendación por la categoría principal

La implementación del método de recomendación basada en la categoría principal del sistema de clasificación decimal Dewey se define en la *Figura 18*:

Figura 18

Fragmento de código para la recomendación por categoría principal

```
Match(
    ).Node("Title","m").LeftRelationship("ASSIGN_DEWEY"
    ).Node("Classification","cl1"
).Where(
    ).Id("m", idTitle
).Match(
    ).Node("Classification","cl2").RightRelationship("ASSIGN_DEWEY"
    ).Node("recom"
).Where("cl2.cCode").StartWith().Substring("cl1.cCode",3
    ).And("m <> recom"
).Select("recom.cTitle")
```

Elaboración propia.

La *Figura 19* muestra un fragmento de código que ejemplifica una consulta de recomendación de títulos basada en el sistema de clasificación decimal Dewey. En esta consulta, se inicia con el nodo identificado como m que representa el título de entrada, y se utiliza la relación $ASSIGN\_DEWEY$ para establecer una conexión con un nodo de tipo $Classification$ que posee un número Dewey especifico.

Se utiliza la cláusula $MATCH$ para buscar nodos de tipo $Title$ renombrados como m y se los relaciona con nodos de tipo $Classification$ mediante la relación $ASSIGN\_DEWEY$ para obtener el código Dewey asociado al título de entrada. A continuación, se aplica la cláusula $WHERE$ para filtrar los nodos de entrada según su identificador único $idTitle$. Luego, se realiza otro $MATCH$ para buscar nodos de tipo $Classification$ que están relacionados con los títulos recomendados mediante la relación $ASSIGN\_DEWEY$, y se aplica la condición de identificación de la categoría principal. Además, se asegura que el identificador del nodo m sea obtenido a través de la variable de entrada $idTitle$ y que no sea igual a $recom$ del nodo $Title$. Finalmente, se retorna una lista de recomendación que incluye el campo $cTitle$ del nodo de salida $recom$.

4.4. EVALUACIÓN DEL MODELO DE BASE DE DATOS

Para evaluar el modelo de base de datos orientado a grafos en el contexto de la generación de recomendaciones, es esencial definir de manera precisa los recursos empleados. En este capítulo, se presenta una descripción detallada de la data set empleado en el estudio, se realiza un análisis del rendimiento de la base de datos, y se efectúa el cálculo de la métrica de diversidad para la evaluación de la calidad de las recomendaciones.

4.4.1. Data Set

Para llevar a cabo las pruebas, se recolectó información bibliográfica de diversos catálogos en línea de bibliotecas públicas mediante la técnica de web scraping. Se seleccionaron aquella información bibliográfica cuya catalogación se basa en el estándar MARC. Se incluyeron libros de diferentes temáticas y áreas de interés, esta información se organizó acorde a las diez categorías principales

del sistema decimal Dewey. La distribución de los datos bibliográficos se encuentra detallada en la *Tabla 5*.

Tabla 5

Distribución de información bibliográfica según la categoría Dewey

Catalogo en línea	000-099	100-199	200-299	300-399	400-499	500-599	600-699	700-799	800-899	900-999	∑
Biblioteca Municipal Puno	34	40	9	186	171	97	121	104	170	271	1203
Universidad Nacional del Altiplano Puno	172	28	7	222	44	160	391	18	10	10	1062
Totales	206	68	16	408	215	257	512	122	180	281	**2265**

Elaboración propia.

Así, con la finalidad de obtener una perspectiva completa de la información, se ha logrado satisfactoriamente cargar en el modelo de datos basado en grafos un total de 17,137 elementos, entre los cuales se incluyen nodos y relaciones, como se detalla en la *Tabla 6*.

Tabla 6

Poblamiento del Modelo de Datos Orientado a Grafos

Elemento	Cantidad
Nodos	6454
Relaciones	10683

Elaboración propia.

4.4.2. Rendimiento de la base de datos orientado a grafos

Con el fin de evaluar el desempeño del modelo de base de datos, se toman en cuenta dos métricas fundamentales: el tiempo necesario para cargar los datos y el tiempo empleado en la ejecución de consultas. Estas mediciones se llevan a

cabo utilizando la plataforma *Neo4j Aura v*5.10 *Free*, que ofrece un entorno apropiado para la realización de pruebas y análisis.

En un primer paso, se evalúa el tiempo requerido para cargar los datos bibliográficos en la base de datos orientada a grafos, transformándolos en nodos y relaciones. Este proceso involucra la inserción de información completa de cada libro, lo que implica la asignación de propiedades pertinentes a cada nodo y relación. Se ha registrado el tiempo en milisegundos que transcurre desde el inicio de la operación hasta que el gestor empieza a enviar los datos, así como el tiempo total en milisegundos que toma la operación para completarse. Los resultados de esta evaluación se presentan en la *Tabla 7*.

Tabla 7

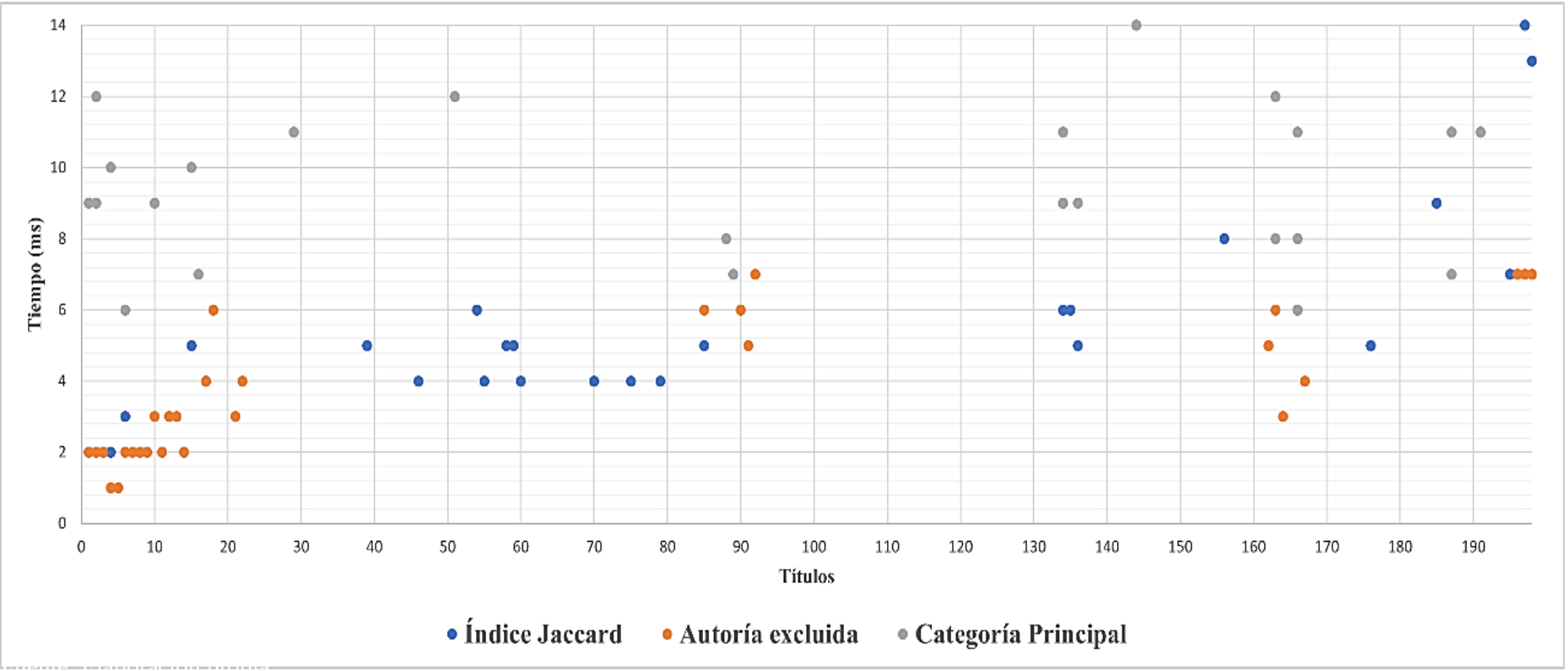
Resultados del tiempo de carga de datos.

N	Tipo	Recurso	Propiedades establecidas	Started streaming (ms)	Proceso completado (ms)
1	Nodo	*Person*	8	17	23
2	Nodo	*Publisher*	7	18	22
3	Nodo	*Classification*	7	16	21
4	Nodo	*Title*	15	20	26
5	Nodo	*Copy*	7	20	22
6	Relación	*HAS_RESPONSIBILITY*	1	2	15
7	Relación	*PUBLISHED*	0	1	14
8	Relación	*ASSIGN_DEWEY*	0	1	10
9	Relación	*HAS_COPY*	0	1	9
TOTAL					**160**

Elaboración propia.

Figura 19

Tiempos de procesamiento de las recomendaciones



Elaboración propia.

La *Figura 19* ilustra tres métodos de recomendación junto con sus ventajas y desventajas en términos de tiempo de procesamiento y cantidad de recomendaciones proporcionadas. Se realizaron diversas consultas de acuerdo con los métodos de recomendación propuestos, midiendo el tiempo de ejecución en cada caso.

Se emplearon diferentes títulos de entrada para generar listas de recomendaciones para cada método. Se registraron tanto el número de registros recuperados como el tiempo en milisegundos necesario para completar cada operación de procesamiento de recomendación, con tiempos que oscilan entre 1 y 14 milisegundos, según la cantidad de recomendaciones, como se presenta en la figura.

El método de índice de Jaccard presenta tiempos de procesamiento variables, con un máximo de 4 milisegundos para 20 recomendaciones y aumentando a 14 milisegundos para cerca de 199 resultados. El método de recomendación basado en la autoría excluida tuvo un tiempo máximo de procesamiento de 3 milisegundos para 20 recomendaciones y se mantuvo por debajo de 5 milisegundos para alrededor de 90 títulos recomendados. Por último, el método de recomendación por categoría principal tuvo un tiempo máximo de 12 milisegundos para 20 recomendaciones y se mantuvo en este umbral para cerca de 190 títulos recomendados.

4.4.3. Evaluación de la recomendación

Con el fin de evaluar la eficacia del modelo de datos propuesto para generar recomendaciones de libros, se han tenido en cuenta diversas métricas y se ha investigado bibliografía relevante sobre el tema. En esta investigación, la

métrica principal utilizada para evaluar las recomendaciones es la diversidad, concretamente la diversidad intra lista (ILD), que se refiere a la variabilidad de las recomendaciones ofrecidas por el modelo, abarcando diferentes temas y categorías de libros, al tiempo que se evita una concentración excesiva en un área.

De acuerdo con el trabajo de Aróstegui Martín, (2020) la métrica de ILD se calcula como la media de las distancias entre los ítems recomendados, utilizando una definición clara de la medida de distancia entre ítems, estas distancias se calculan en función a las características de los ítems. Así mismo, Silveira *et al*., (2019) explica que el cálculo de la diversidad se limita a la lista de recomendaciones en cuestión, abarcando su alcance y ámbito. De esta manera, la medida utilizada para determinar la distancia entre cada libro dentro de la lista de recomendaciones proporcionada por el modelo es el número asociado a la clasificación decimal Dewey del libro, La fórmula considera factores, como la cantidad de subcategorías diferentes presentes en la lista de recomendación, la variabilidad de la clasificación y la distribución equitativa de las temáticas abordadas. La fórmula utilizada se define como:

$$ILD = \frac{1}{|R|(|R|-1)} \sum_{i \in R} . \sum_{j \in R} d(i,j)$$

4.4.4. Disposición del modelo de datos

Dado que la lista de recomendaciones generadas por el modelo de datos orientado a grafos es obtenida a partir de tres métodos distintos, por lo que se centraliza el análisis de la diversidad global de las recomendaciones, considerando la variedad y amplitud de categorías o géneros literarios y temáticas abordadas en conjunto. Se busca evaluar la capacidad del modelo de generar recomendaciones

para ofrecer opciones diversas y evitar la concentración en una única clasificación o categoría del libro.

Tabla 8

Mejores resultados de diversidad de las recomendaciones

Prueba	Input		Elementos Recomendados |R|	Diversidad (ILD)
	Categoría	Título		
1	001.42	*Investigación científica*	24	0.736
2	079.8562	*La prensa en Puno*	16	0.816
3	101	*Filosofía*	14	0.901
4	133.5092	*Saint Germain*	11	0.563
5	246.85622	*Iglesia San Jerónimo de Asillo*	15	0.733
6	299.8562	*Magia en Chucuito*	23	0.806
7	301	*Sociología general*	23	0.889
8	371.914	*Cálculo*	14	0.879
9	401.4	*Funcionamiento del Lenguaje*	15	0.771
10	498.323	*Gramática del idioma quechua*	23	0.762
11	513	*Aritmética*	19	0.713
12	515.1	*Cálculo de varias variables*	14	0.539
13	624.171	*Análisis estructural*	20	0.952
14	637.13	*Estadística*	20	0.931
15	787.87	*Manual de Guitarra*	24	0.894
16	793.3198562	*Danzas y bailes del altiplano*	18	0.954
17	869.503	*Cuentos cortos*	25	0.810
18	869.56	*La novela puneña*	20	0.794
19	904.762	*Retorno a casa*	25	0.883
20	918.5	*Geografía general del Perú*	19	0.730

Elaboración propia.

Para evaluar la diversidad de las recomendaciones, se realizaron experimentos para cada una de las categorías principales de libros. Los resultados más destacados se presentan en la *Tabla 8*, donde cada prueba realizada se

corresponde con una lista de recomendación final y se incluyen los valores correspondientes a la métrica ILD.

4.5. DISCUSIÓN

- **Discusión al objetivo general**

Los resultados muestran que la combinación de los métodos de recomendación propuestos, basados en similitud y ejecutados en un modelo de base de datos orientado a grafos, logra generar diversidad en las recomendaciones, alcanzando un 95% según la métrica ILD. Así mismo, la combinación de diferentes técnicas de recomendación mejora la calidad de las recomendaciones, coincidiendo con el estudio de Tian *et al*. (2019) que demostró una mejora del 10% al combinar técnicas de recomendación. Sin embargo, se observa una diferencia importante en la manera de medir las recomendaciones, debido que, este estudio se centra en la diversidad de las recomendaciones, mientras que los estudios previos, como de Mutalib *et al*., (2020) priorizan la precisión y alcanza un 95% de precisión. Aunque ambos estudios obtuvieron resultados competitivos, utilizan diferentes métricas de evaluación.

- **Discusión al primer objetivo específico**

Se recolectaron información bibliográfica de dos catálogos en línea de bibliotecas reconocidas de la región, coincidiendo con el método de Amézquita Llerena (2020), que utiliza Web Scraping para obtener un total de 6221 artículos. Esto aseguró la adquisición de datos confiables. Además, para garantizar la calidad de los datos de entrada para el modelo de datos, se llevaron a cabo tareas de preprocesamiento que incluyeron la eliminación de datos duplicados, normalización y estructuración. Esto condujo a la creación de nodos de diversos tipos (Título, Persona, Título de serie, Copia, Clasificación

y Editorial), resultando en un total de 6,454 nodos. Este proceso se asemeja a la técnica empleada por Sen *et al.* (2021), quienes realizaron un preprocesamiento similar en un contexto diferente, lo que les permitió modelar más de 167,709 nodos en un modelo de base de datos de grafos. Estas similitudes subrayan la versatilidad de las técnicas de recolección y clasificación de información, incluso cuando se aplican en contextos y propósitos distintos.

- **Discusión al segundo objetivo específico**

Se representaron un total de 2,345 libros divididos en nodos y relaciones en un modelo de base de datos orientado a grafos. La distribución de los nodos se basa en las entidades principales identificadas en los datos de entrada, específicamente en el estándar MARC, que implica 6 tipos de nodos y 5 tipos de relaciones. Cada nodo y relación se caracteriza por sus propios atributos, además del sentido de las relaciones. Esto concuerda con los estudios de Amézquita Llerena (2020), Sen *et al.* (2021) y Mutalib *et al.* (2020), cuyos data sets influyeron en la estructura de representación en el modelo de datos, optando por utilizar $Cypher$ nativo para representar información bibliográfica y, agregar atributos a nodos y relaciones. Sin embargo, la distribución de nodos en el modelo de datos no está necesariamente predeterminada por las entidades presentes en los data sets, como se ilustra en el estudio de Qassimi *et al.* (2021) quienes emplearon la técnica de Clustering Espectral para representar nodos y relaciones en una colección de 10,000 libros. Además, a diferencia de las investigaciones anteriores, en este estudio se incorpora una capa de transformación de sentencias desde Python a $Cypher$.

- **Discusión al tercer objetivo específico**

Se implementaron algoritmos basados en el método de los k vecinos más cercanos, como el índice de Jaccard, lo que resultó en la generación de listas de recomendación con

diversidad en los resultados, tal como se evidencia a través de la métrica de diversidad. Este hallazgo encuentra respaldo en la investigación realizada por Mamani Chile (2022), quien observó una desventaja del 11% al evaluar el índice de Jaccard en función de la métrica de precisión. Por otro lado, los métodos de autoría excluida y la categoría principal se basan en la similitud del autor y la categoría, respectivamente. Este enfoque guarda similitudes con el trabajo de Mutalib *et al,*. (2020), quienes proporcionaron recomendaciones basadas en la similitud en los géneros de las películas disponibles. En ambos casos, la generación de recomendaciones se basa en el contenido de los elementos y no en la interacción usuario-libro.

- **Discusión al cuarto objetivo específico**

Se ha evaluado la calidad de las recomendaciones utilizando la métrica de diversidad ILD, y los experimentos indican que la combinación de los tres métodos que se basan en la similitud de las características del contenido ha logrado alcanzar un nivel impresionante del 95% de diversidad en las recomendaciones. La elección de esta métrica se fundamenta en la ausencia de perfiles de usuario e interacciones históricas usuario-libro. Este enfoque difiere significativamente con la investigación de Jethva *et al*. (2022), que se enfocó en la precisión de las recomendaciones, logrando un igualmente destacable 95% de precisión a pesar de enfrentar el desafío del arranque en frío. Una perspectiva similar fue compartida por Qassimi *et al*. (2021), quienes también priorizaron la precisión y lograron un 100% de precisión. No obstante, es importante subrayar que esta estrategia demanda una gran cantidad de datos para generar recomendaciones de calidad. Por otro lado, Mamani Chile (2022) centró su investigación en la precisión de las recomendaciones, logrando un nivel de precisión del 72%. Los experimentos realizados en este estudio demuestran que los enfoques propuestos lograron resultados competitivos y no se ven afectados por los desafíos mencionados anteriormente, debido que los

algoritmos desarrollados están específicamente diseñados para procesar recomendaciones teniendo en cuenta las entidades de similitud entre los libros.

- **Consideraciones finales**

El modelo propuesto en este estudio ha demostrado ser eficiente debido a la diversidad que ofrece en sus recomendaciones, al combinar tres métodos de recomendación: índice Jaccard, Autoría excluida y categoría principal. Este enfoque se caracteriza por su agilidad y bajo consumo de recursos computacionales, lo que lo hace adecuado para su implementación en diversas plataformas de recomendación, incluso más allá del ámbito de los libros.

Es importante destacar que algunos catálogos de libros se limitan a búsquedas en lugar de ofrecer recomendaciones, principalmente debido a su funcionamiento basado en el modelo relacional, que implica el uso de JOINS y puede ser computacionalmente costoso y complejo de programar. Por ejemplo, el cálculo de la distancia de Pearson involucra cálculos internos y demandantes en términos computacionales. Los sistemas de recomendación basados en modelos de datos en grafos han demostrado ser altamente efectivos en este sentido.

Este estudio se resalta la superioridad de los modelos de base de datos orientados a grafos en comparación con los modelos de recomendación diseñados en SQL. Los modelos de datos en grafos son particularmente efectivos para el procesamiento en tiempo real, como se observa en las redes sociales contemporáneas.

Sin embargo, es importante mencionar que, aunque este estudio se centró en la diversidad de las recomendaciones, se requieren evaluaciones adicionales que aborden métricas como precisión, recall, F1 score, exactitud, cobertura, serendipia, novedad y la inseparabilidad, lo que destaca la complejidad inherente a los modelos de recomendación.

Una limitación significativa fue la escasez de datos, debido que estos modelos suelen necesitar un conjunto de datos más extenso, idealmente con más de 5000 títulos de libros. Aunque en este estudio se contó con 2265 títulos, lo cual arrojó resultados positivos, se sugiere que futuros entrenamientos incluyan un mayor número de títulos disponibles.

Además, es importante tener en cuenta que el lenguaje Cypher, aunque poderoso, es relativamente nuevo desde su creación en 2016 y se actualiza constantemente, lo que puede generar desafíos en términos de adaptación y mantenimiento.

Por último, se destaca que la implementación de servidores de Python y Neo4J para modelos basados en grafos puede representar una limitación, ya que no es tan común como el despliegue de modelos SQL o NoSQL.

El uso de este tipo de modelos tiene un impacto significativo en bibliotecas y catálogos en línea, tanto gubernamentales como privados. Estos modelos permiten a los usuarios descubrir constantemente nuevas opciones de libros de acuerdo a sus intereses, incluso recomendando libros que nadie ha leído previamente. Estas tecnologías deben cumplir con los principios de privacidad y las normativas legales de cada país.

Además, es importante mencionar que la integración de modelos de base de datos orientados a grafos con modelos de inteligencia artificial, como ChatGPT y Llama-2, puede proporcionar un nivel excepcional de precisión en las recomendaciones. Esto representa un avance significativo en el campo de la recomendación de contenido.

V. CONCLUSIONES

- Con el propósito de obtener diversidad en los resultados, se han propuesto tres métodos de recomendación (Índice de Jaccard, Autoría excluida y Categoría principal) basados en la similitud de las características del contenido. Los experimentos han revelado que la combinación de estos métodos implementados en un modelo de base de datos orientado a grafos ha logrado una diversidad que oscila entre el 53% y el 95%, con un promedio del 80%, evaluada mediante la métrica ILD.

- Se ha logrado con éxito la implementación de un algoritmo de preprocesamiento de información bibliográfica, lo que ha contribuido a mejorar la calidad de la entrada garantizando la integridad y confiabilidad de los datos en el modelo de la base de datos. Este algoritmo lleva a cabo tres funciones esenciales. En primer lugar, se encarga de generar un identificador único para cada nodo; en segundo lugar, crea identificadores de índice; y finalmente, realiza la limpieza de datos. Gracias a esta implementación, se ha logrado preprocesar 2265 archivos JSON de entrada, cada uno correspondiente a libros basados en el estándar MARC. Durante este proceso, se han descartado nodos y relaciones duplicados, resultando en la generación de un total de 6454 identificadores únicos de nodos.

- Se ha logrado exitosamente la representación de datos bibliográficos basados en el estándar MARC mediante nodos y relaciones. En total, se han incorporado 6 tipos distintos de nodos y 5 tipos de relaciones, siguiendo la estructura $(Person) - [HAS\_RESPONSIBILITY] \rightarrow (Title) - [HAS\_COPY] \rightarrow (Copy), (Publisher) - [PUBLISHED] \rightarrow (Title), (Classification) - [ASSING\_DEWEY] \rightarrow (Title), (SerialTitle) - [PART\_TO] \rightarrow (Title)$. Lo que ha permitido poblar la base de datos con 6454 nodos y 10683 relaciones. Durante este proceso, se destacó la

eficacia y velocidad del modelo, y se observó que el tiempo promedio necesario para cargar el nodo de tipo $Title$ fue de aproximadamente 26 milisegundos. Siendo esta prueba uno de los últimos nodos agregados y con una configuración más compleja que incluyen 15 propiedades. Además, los tiempos de carga de los demás nodos oscilaron entre 21 y 23 milisegundos, respecto a la carga de las relaciones oscilaron entre 9 y 15 milisegundos. Esto sugiere que el modelo de base de datos gestiona de manera estable, consistente y ágil la representación de la información bibliográfica en forma de nodos y relaciones, incluso cuando se manejan diferentes configuraciones y tamaños de datos. Así mismo el modelo de base de datos demostró ser idóneo para representar relaciones de tipo $(1{:}\,1)$ y $(1{:}\,N)$ de manera nativa, al establecer conexiones directas entre los nodos involucrados lo que ha permitido eliminar la necesidad de elementos o índices intermedios adicionales.

- En este estudio, se han implementado con éxito tres métodos de recomendación basados en la similitud de características del contenido, debido que no se ha contado con la intervención de perfiles de usuarios: el Índice de Jaccard, la Autoría Excluida y la Categoría Principal. El primer método se basa en la relación entre el número de nodos vecinos compartidos por dos nodos y el número total de nodos vecinos diferentes que tienen esos dos nodos. El segundo método identifica libros a partir de un título de entrada cuyos autores no hayan participado directamente en la creación, pero sus coautores sí. El tercer método, recomienda libros que comparten la misma categoría principal en su clasificación decimal Dewey. Estos métodos han demostrado ser efectivos en la generación de recomendaciones diversas y relevantes, contribuyendo así al logro de nuestros objetivos de investigación.

- El método de Índice de Jaccard muestra tiempos de procesamiento variables. Para una lista de recomendación con un máximo de 20 resultados, el tiempo máximo es

de 4 milisegundos, manteniéndose por debajo de 8 milisegundos con aproximadamente 185 títulos recomendados. Sin embargo, este tiempo aumenta a un máximo de 14 milisegundos al procesar una lista con cerca de 199 resultados. El método de Recomendación basado en Autoría Excluida presenta un tiempo máximo de procesamiento de 3 milisegundos para una lista de 20 resultados y se mantiene por debajo de 5 milisegundos para alrededor de 90 títulos recomendados, disminuyendo aún más al procesar cerca de 165 resultados. El tiempo de procesamiento del método de Recomendación por Categoría Principal es de un máximo de 12 milisegundos para una lista de los primeros 20 resultados, manteniéndose en ese umbral para cerca de 190 títulos recomendados. En términos de tiempo de procesamiento, el método de Autoría Excluida es el más eficiente, seguido por el Índice de Jaccard y la Categoría Principal. No obstante, en cuanto al número de resultados obtenidos, el Índice de Jaccard ocupa el primer lugar, seguido de la Categoría Principal y, en última instancia, la Autoría Excluida. Finalmente, en relación con las listas de recomendación derivadas de la combinación de las recomendaciones individuales de los métodos propuestos, los resultados son satisfactorios según la evaluación realizada a través de la métrica de diversidad Intra-Lista. En el escenario más favorable, se logra alcanzar un nivel de diversidad del 95%, específicamente asociado a la categoría 793. Por otro lado, en el caso menos diverso, se obtiene un nivel de diversidad del 53%, correspondiente a la categoría 515, en promedio se obtiene un 80% de diversidad. Estos hallazgos señalan una diversidad aceptable en las recomendaciones generadas por la combinación de los diversos métodos propuestos.

VI. RECOMENDACIONES

- A las entidades públicas que proporcionan el servicio de catálogo en línea, se les recomienda enfocarse en la protección de la seguridad perimetral de sus servidores. Estos servidores se encuentran expuestos a ataques de denegación de servicio que pueden afectar negativamente el rendimiento del servicio.

- A las empresas de tecnología, bibliotecas y startups se les recomienda adoptar modelos de bases de datos orientados a grafos, ya que son altamente efectivos. Esto se aplica no solo a bibliotecas, sino también a otros sectores.

- Para los investigadores en gestión de datos bibliográficos, se sugiere explorar el uso de Web Scraping y procesamiento de lenguaje natural para automatizar la identificación de campos relevantes en estándares MARC.

- Los encargados de la gestión bibliográfica en bibliotecas deberían considerar agregar metadatos y etiquetas semánticas para enriquecer la información y personalizar recomendaciones.

- Los líderes de tecnología y desarrolladores de sistemas de recomendación deben explorar la inteligencia artificial y el aprendizaje automático para mejorar la precisión y personalización de las recomendaciones.

- Los futuros investigadores deben ampliar métricas de evaluación y realizar estudios de usuario para obtener retroalimentación sobre la utilidad y satisfacción de las recomendaciones.

VII. REFERENCIAS BIBLIOGRÁFICAS

Ahmadian, S., Joorabloo, N., Mahdi, J., & Milad, A. (2022). Alleviating data sparsity problem in time-aware recommender systems using a reliable rating profile enrichment approach. *Expert Systems with Applications*, *187*. https://doi.org/j.eswa.2021.115849

Alabdulrahman, R., & Viktor, H. (2021). Catering for unique tastes: Targeting grey-sheep users recommender systems through one-class machine learning. *Expert Systems with Applications*, *166*(September 2020), 114061. https://doi.org/10.1016/j.eswa.2020.114061

Amézquita Llerena, J. F. (2020). Software de recomendación médico basado en modelo de datos orientado a grafos con Neo4j [Universidad Católica de Santa María]. In *Universidad Católica de Santa María*. https://tesis.ucsm.edu.pe/repositorio/handle/UCSM/10290

Aróstegui Martín, J. (2020). Novedad y diversidad en recomendación con bandidos multi-brazo. In *Universidad Autónoma de Madrid*. https://zaguan.unizar.es/record/112622/files/TAZ-TFG-2022-641.pdf

Bárbaro, E., Sust, U., Javier, A., & Cuevas, S. (2017). Sistemas de recomendación semánticos: Una revisión del Estado del Arte Semantic recommendation systems : A State-of-the-Art Survey. *Revista Cubana de Ciencias Informáticas*, *11*(2), 189–206. http://scielo.sld.cu/pdf/rcci/v11n2/rcci14217.pdf

Besta, M., Gerstenberger, R., Fischer, M., Podstawski, M., Müller, J., Blach, N., Egeli, B., Mitenkov, G., Chlapek, W., Michalewicz, M., & Hoefler, T. (2023). *High-Performance Graph Databases That Are Portable, Programmable, and Scale to Hundreds of Thousands of Cores*. http://arxiv.org/abs/2305.11162

BNE. (2021). *MARC 21*. Servicios BNE. https://www.bne.es/es/servicios/servicios-para-bibliotecarios/normas-estandares-politicas-bne-procesos-tecnicos/marc21#:~:text=El formato MARC es una,recursos y servicios entre bibliotecas.

Breese, J. S., Heckerman, D., & Kadie, C. (1998). Empirical Analysis of Predictive Algorithms for Collaborative Filtering. *Microsoft Research*, 1–10. http://arxiv.org/abs/1301.7363

Castells, P., Hurley, N. J., & Vargas, S. (2015). Novelty and Diversity in Recommender Systems. In F. Ricci et al. (eds.), Recommender Systems Handbook (Ed.), *Springer Science+Business Media New York 2015* (Second Edi). https://doi.org/10.1007/978-1-4899-7637-6

Castells, P., Hurley, N., & Vargas, S. (2022). Value and Impact of Recommender

Systems. In *Recommender Systems Handbook: Third Edition*. https://doi.org/10.1007/978-1-0716-2197-4_14

Chicaiza, J., & Valdiviezo, P. (2021). A comprehensive survey of knowledge graph-based recommender systems: technologies, development, and contributions. In *Multidisciplinary Digital Publishing Institute* (Vol. 12, Issue 6). MDPI AG. https://doi.org/10.3390/info12060232

Costa, L. da F. C. (2021). Further Generalizations of the Jaccard Index. *Hal Open Science*.

Gao, C., Zheng, Y., Li, N., Li, Y., Qin, Y., Piao, J., Quan, Y., Chang, J., Jin, D., He, X., & Li, Y. (2023). A Survey of Graph Neural Networks for Recommender Systems: Challenges, Methods, and Directions. *ACM Transactions on Recommender Systems*, *1*(1), 1–51. https://doi.org/10.1145/3568022

Gavilán, C. M. (2008). *El formato MARC : variedades geográficas y*. 13. http://eprints.rclis.org/14525/1/marc.pdf

Gil, D., & Seguro, C. (2022). Machine Learning aplicado al Análisis del Rendimiento de Desarrollos de Sogtware. *Revista Politécnica*, 13. https://www.redalyc.org/journal/6078/607870799010/607870799010.pdf

Graph everywhere. (2020). *El papel de la tecnología de grafos en la era de la digitalización del negocio*.

GraphEverywhere. (2020). *Graphs Are Everywhere* (Ebook).

Hinestroza, D. (2018). EL MACHINE LEARNING A TRAVÉS DE LOS TIEMPOS, Y LOS APORTES A LA HUMANIDAD. *Universidad Libre Seccional Pereira*, *3*, 1–17. http://dx.doi.org/10.1186/s13662-017-1121-6%0Ahttps://doi.org/10.1007/s41980-018-0101-2%0Ahttps://doi.org/10.1016/j.cnsns.2018.04.019%0Ahttps://doi.org/10.1016/j.cam.2017.10.014%0Ahttp://dx.doi.org/10.1016/j.apm.2011.07.041%0Ahttp://arxiv.org/abs/1502.020

Hossain, I., Palash, M. A. H., Sejuty, A. T., Tanjim, N. A., Nasim, M. A. AL, Saif, S., Suraj, A. B., Haque, M. M. A., & Karim, N. (2022). A Survey of Recommender System Techniques and the Ecommerce Domain. *Department of Research and Development*, 1–22. https://doi.org/10.1093/comnet/xxx000

IFLA. (2001). *UNIMARC Manual Authorities Format 2nd revised and enlarged edition* (Vol. 22).

INAOE. (2010). Ciencias computacionales Proped ´ eutico : Programaci ´ on y Estructura de Datos Contents. *InDE*, 1–8. https://posgrados.inaoep.mx/archivos/PosCsComputacionales/Curso_Propedeutico/Programacion_Estructuras_Datos/Capitulo_10_Grafos.pdf

Isufi, E., Pocchiari, M., & Hanjalic, A. (2021). Accuracy-diversity trade-off in recommender systems via graph convolutions. *Information Processing and Management*, *58*(2), 102459. https://doi.org/10.1016/j.ipm.2020.102459

Jaccard, P. (1901). Étude comparative de la distribution florale dans une portion des Alpes et du Jura. *Bulletin de La Societe Vaudoise Des Sciences Naturelles*, *January 1901*. https://doi.org/10.5169/seals-266450

JAUME. (2012). Procesador Lenguaje - Analizador Léxico. *Revista : Publicación Periódica : Español (Spa)*, 1–38. https://repositori.uji.es/xmlui/bitstream/handle/10234/5877/lexico.apun.pdf?sequence=1

Jethva, D., Rule, A., & Ghag, K. (2022). Book Recommendation System. *International Journal for Innovative Research in Science and Technology Related*, *10*(5), 39–43. https://doi.org/10.35629/5252-040514801485

Joyanes Aguilar, L., & Zahonero Martínez, I. (2007). *Estructura de datos en C ++* (C. Sánchez González (ed.)). McGraw-Hill Interamericana.

Jyothi, D. N. (2020). Book Recommendation System using Neo4j Graph Database. In *The International journal of analytical and experimental modal analysis: Vol. XII* (Issue 0886).

Kavin, C. K. M., Priya, V., Priya, R. M., & Lakshmi, S. L. (2021). Book recommender system using improved collaborative filtering. *International Journal of Research in Engineering, Science and Management*, *4*(4), 51–56. https://journal.ijresm.com/index.php/ijresm/article/view/638/611

Kawai, M., Sato, H., & Shiohama, T. (2022). Topic model-based recommender systems and their applications to cold-start problems. *Expert Systems with Applications*, *202*. https://doi.org/10.1016/j.eswa.2022.117129

Khalil, A., & Belaissaoui, M. (2022). A Graph-oriented Framework for Online Analytical Processing. *International Journal of Advanced Computer Science and Applications*, *13*(5), 547–555. https://doi.org/10.14569/IJACSA.2022.0130564

Koffman, E. B., & Wolfgang, P. A. T. (2008). *Estructura de datos con C++: objetos, abstracciones y diseño* (Primera edicion). McGraw-Hill Interamericana.

Li, Y., Liu, K., Satapathy, R., Wang, S., & Cambria, E. (2023). *Recent Developments in Recommender Systems: A Survey* (Vol. 14, Issue 8). http://arxiv.org/abs/2306.12680

Liu, W., Xi, Y., Qin, J., Sun, F., Chen, B., Zhang, W., Zhang, R., & Tang, R. (2022). Neural Re-ranking in Multi-stage Recommender Systems: A Review. *IJCAI International Joint Conference on Artificial Intelligence*, 5512–5520. https://doi.org/10.24963/ijcai.2022/771

Mamani Chile, R. (2022). *Sistema de recomendación de libros basado en algoritmos de similaridad para el Centro de Recursos para el Aprendizaje y la Investigación de la Universidad Peruana Unión*. UNIVERSIDAD PERUANA UNIÓN.

Müllner, P. (2023). User Privacy in Recommender Systems. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *13982 LNCS*, 456–461. https://doi.org/10.1007/978-3-031-28241-6_52

Mutalib, R. D. A., Hassan, S., Chong, C. Y., Admodisastro, N., & Baharom, S. (2020). Graph-powered recommendation engine in movie recommender system. In *Journal of Critical Reviews* (Vol. 7, Issue 8). https://doi.org/10.31838/jcr.07.08.261

Patterson, J. (2020). Cataloguing remains an important skill at public libraries in the modern metadata landscape of Norway. *Evidence Based Library and Information Practice*, *15*(3), 187–188. https://doi.org/10.18438/eblip29788

PngWing. (2022). *Gráfico del vecino más cercano k-algoritmo*. https://www.pngwing.com/es/free-png-hwhco

Qassimi, S., Abdelwahed, E. H., Hafidi, M., & Qazdar, A. (2021). Towards a folksonomy graph-based context-aware recommender system of annotated books. In *Journal of Big Data* (Vol. 8, Issue 1). Departamento de Ciencias Computacionales FSTG. https://doi.org/10.1186/s40537-021-00457-3

Ramovecchi, J., & García, M. S. (2021). JoyMeter -Sistema de recomendación de actividades a usuarios de dispositivos móviles [Universidad del Centro de la Provincia de Buenos Aires]. In *Repositorio UNC*. http://www.ridaa.unicen.edu.ar/xmlui/bitstream/handle/123456789/1346/Ramovecchi%2C Hernán y García%2C María Sol.PDF?sequence=1&isAllowed=y

Rendón, A. (2022). *¿Conoces Neo4j o sabes de qué va?* Enmiloca. https://www.enmilocalfunciona.io/conoces-neo4j-o-sabes-de-que-va/

Rojas Rojas, R. C. (2019). *Mejora en la de búsqueda de contenidos sobre el catálogo de la Biblioteca Nacional mediante métodos usados en sistemas recomendadores* [Universidad de Chile]. https://repositorio.uchile.cl/bitstream/handle/2250/171800/Mejora-en-la-busqueda-de-contenidos-sobre-el-Catalogo-de-la-Biblioteca-Nacional.pdf?sequence=1&isAllowed=y

Salau, L., Hamada, M., Prasad, R., Hassan, M., Mahendran, A., & Watanobe, Y. (2022). State-of-the-Art Survey on Deep Learning-Based Recommender Systems for E-Learning. *Applied Sciences (Switzerland)*, *12*(23). https://doi.org/10.3390/app122311996

Sattar, A., & Bacciu, D. (2022). Graph Neural Network for Context-Aware Recommendation. In *Neural Processing Letters*. Università di Pisa. https://doi.org/10.1007/s11063-022-10917-3

Sen, S., Mehta, A., Ganguli, R., & Sen, S. (2021). Recommendation of Influenced Products Using Association Rule Mining: Neo4j as a Case Study. *SN Computer Science*, *2*(2), 1–17. https://doi.org/10.1007/s42979-021-00460-8

Shewale, N. (2021). *Unimarc 2021-06-21-CUHM-Metadata*.

Shtovba, S., & Petrychko, M. (2019). Jaccard index-based assessing the similarity of research fields in dimensions. *CEUR Workshop Proceedings*, *2533*, 117–128. https://ceur-ws.org/Vol-2533/paper11.pdf

Silveira, T., Zhang, M., Lin, X., Liu, Y., & Ma, S. (2019). How good your recommender system is? A survey on evaluations in recommendation. In *International Journal of Machine Learning and Cybernetics* (Vol. 10, Issue 5). Departamento de Ciencias Computacionales y Tecnología de Beijing - China. https://doi.org/10.1007/s13042-017-0762-9

Sporns, O. (2018). Graph theory methods: Applications in brain networks. *Dialogues in Clinical Neuroscience*, *20*(2), 111–120. https://doi.org/10.31887/DCNS.2018.20.2/OSPORNS

Stitini, O., Kaloun, S., & Bencharef, O. (2022). An Improved Recommender System Solution to Mitigate the Over-Specialization Problem Using Genetic Algorithms. *Electronics (Switzerland)*, *11*(2). https://doi.org/10.3390/electronics11020242

Sukestiyarno, Y. L., Sapolo, H. A., & Sofyan, H. (2023). Application of Recommendation Application of Recommendation System on E- Learning Platform Using Content-Based Filtering with Jaccard Similarity and Cosine Similarity Algorithms. *PrePrints*, 1–9. https://doi.org/10.20944/preprints202306.1672.v1

Tian, Y., Zheng, B., Wang, Y., Zhang, Y., & Wu, Q. (2019). College library personalized recommendation system based on hybrid recommendation algorithm. *Procedia CIRP*, *83*(March), 490–494. https://doi.org/10.1016/j.procir.2019.04.126

Wahyuningsih, T. (2021). Text Mining an Automatic Short Answer Grading (ASAG), Comparison of Three Methods of Cosine Similarity, Jaccard Similarity and Dice's Coefficient. *Journal of Applied Data Sciences*, *2*(2), 45–54. https://doi.org/10.47738/jads.v2i2.31

Wang, L., Shi, T., & Li, S. (2021). Research on the Application of User Recommendation Based on the Fusion Method of Spatially Complex Location Similarity. In *Hindawi Complexity* (Vol. 2021). https://doi.org/10.1155/2021/9998948

Zhang, S., Xiwei, X., Lina, Y., & Sen, W. (2020). Hybrid Collaborative Recommendation via Dual-Autoencoder. In *IEEE Access* (Vol. 8, Issue September).

https://doi.org/10.1109/ACCESS.2020.2979255

Zhang, S., Yao, L., Sun, A., & Tay, Y. (2019). Deep learning based recommender system: A survey and new perspectives. *ACM Computing Surveys*, *52*(1), 1–35. https://doi.org/10.1145/3285029

Zhiyuli, A., Chen, Y., Zhang, X., & Liang, X. (2023). *BookGPT: A General Framework for Book Recommendation Empowered by Large Language Model*. http://arxiv.org/abs/2305.15673

Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., & Sun, M. (2020). Graph neural networks: A review of methods and applications. *AI Open*, *1*(January), 57–81. https://doi.org/10.1016/j.aiopen.2021.01.001

ANEXOS

ANEXO 1: JSON con los datos de entrada para el modelo de datos

```
{
  "title":{
    "cTitle":"Machine learning y deep learning",
    "cSubtitle":"Usando python, scikit y keras",
    "cEdition":"Primera edición",
    "nReleased":2021,
    "cContent":"Introducción -- Datasets -- Regresión -- Clasificación -- Clustering -- Reducción de dimensiones -- Redes neuronales -- Clasificación usando redes neuronales -- Redes convolucionales -- Clasificacion usando redes convolucionales en datasets sencillos -- Generadores de datos -- Enriquecimiento de datos (dataaugmentation) -- Visualización de las capas ocultas -- Aprendizaje por transferencia (transfer learning) -- Autoencoders --Aprendizaje generativo.",
    "cIsbn":"978-958-792-145-8",
    "cNotes":"Ilustraciones, gráficos",
    "cTopics":"Procesamiento de datos",
    "cType":"Libro",
"cImage":"https://biblioteca.unap.edu.pe/opac_css/getimage.php?url_image=http%3A%2F%2Fimages-eu.amazon.com%2Fimages%2FP%2F%21%21isbn%21%21.08.MZZZZZZZ.jpg&noticecode=9789587921458&entity_id=&vigurl=https%3A%2F%2Fedicionesdelau.com%2Fwp-content%2Fuploads%2F2021%2F02%2FMachine-Learniing_DIG.jpg"
  },
  "classification":{
    "cCode":"006.31",
    "cDescription":"Procesamiento de datos"
  },
  "person":[
    {
      "cName":"Jesús",
      "cSurname":"Bobadilla",
      "cPlace":"",
      "cRole":"Autor"
    }
  ],
  "publisher":[
    {
      "cName":"RA-MA Ediciones de la U",
      "cPlace":"Bogotá"
    }
  ],
  "serialTitle":{
    },
  "copy":[
    {
      "cNotation":"006.31 B66",
      "cLibrary":"Bib. Esp. Ing Sistemas",
      "cLink":"https://biblioteca.unap.edu.pe/opac_css/"
    }
  ]
}
```

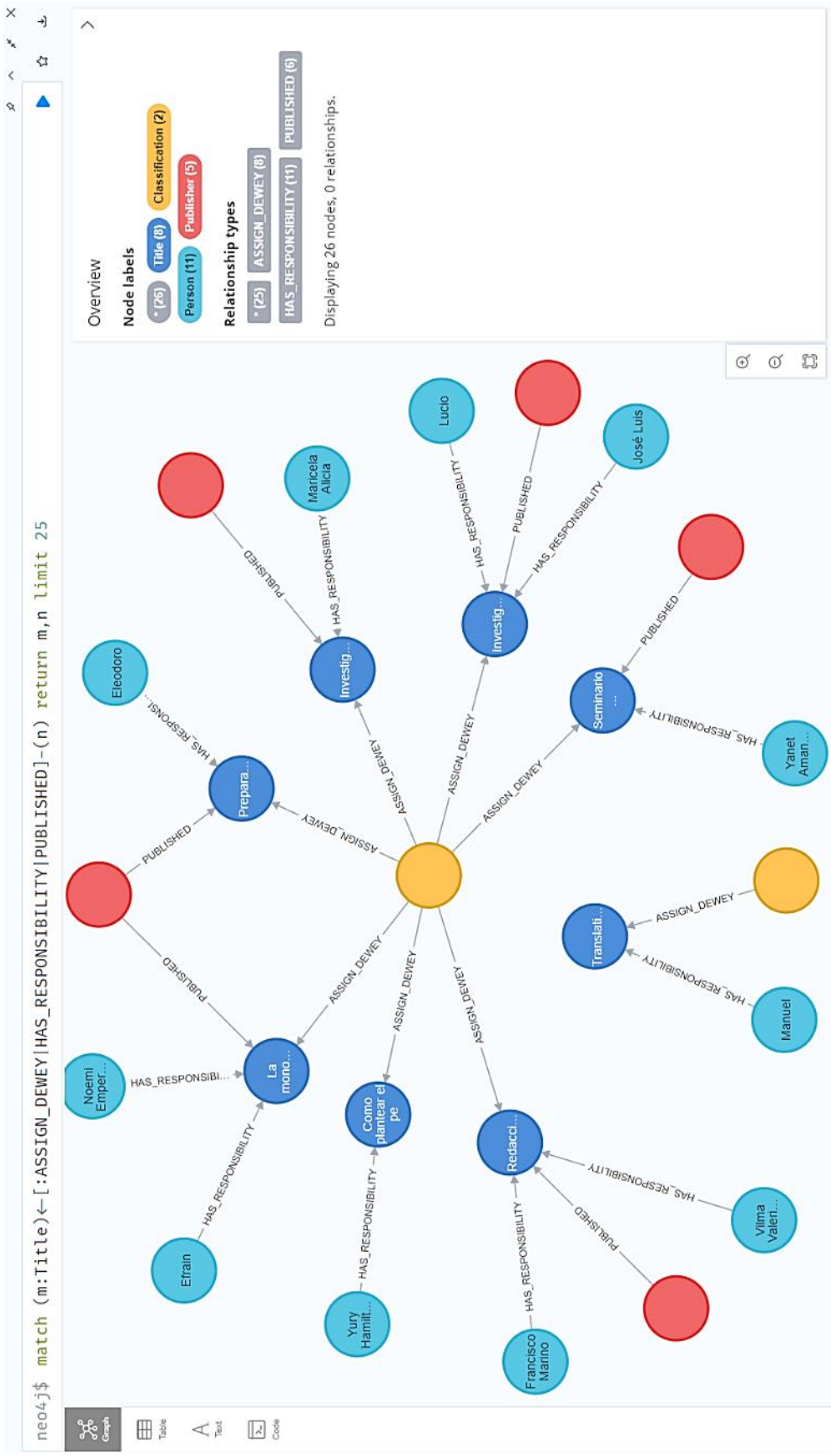
Elaboración propia.

Nota**.** Se muestra un JSON de ejemplo de un libro de la Biblioteca de Ing. de Sistemas como entrada al Analizador Léxico.

ANEXO 2: Información bibliográfica desde la fuente de datos

Nota: Sistema de Biblioteca - UNAP

ANEXO 3: Información bibliográfica en forma de nodos y relaciones en Neo4j



Elaboración propia

ANEXO 4: Representación de información bibliográfica

```
MERGE (i:Title{identity_at:'4fbde0249d558eea6e54da5881cff10cee6d998612343388f48383af9c5bd96a'})
    ON CREATE SET
        i.cTitle='Machine learning y deep learning',
        i.cSubtitle='Usando python, scikit y keras',
        i.cEdition='Primera edición',
        i.nReleased=2021,
        i.cContent='Introducción -- Datasets -- Regresión -- Clasificación -- Clustering -- Reducción de dimensiones -- Redes neuronales
-- Clasificación usando redes neuronales -- Redes convolucionales -- Clasificacion usando redes convolucionales en datasets sencillos --
Generadores de datos -- Enriquecimiento de datos (dataaugmentation) -- Visualización de las capas ocultas -- Aprendizaje por
transferencia (transfer learning) -- Autoencoders --Aprendizaje generativo.',
        i.cIsbn='978-958-792-145-8',
        i.cNotes='Ilustraciones, gráficos',
```

Table | Text | Warn | Code

Server version	Neo4j/5.10-aura
Server address	7ff90d04.databases.neo4j.io:7687
Query	MERGE (i:Title{identity_at:'4fbde0249d558eea6e54da5881cff10cee6d998612343388f48383af9c5bd96a'}) ON CREATE SET i.cTitle='Machine learning y deep learning',i.cSubtitle='Usando python, scikit y keras',i.cEdition='Primera edición',i.nReleased=2021,i.cContent='Introducción -- Datasets -- Regresión -- Clasificación -- Clustering -- Reducción de dimensiones -- Redes neuronales -- Clasificación usando redes neuronales -- Redes convolucionales -- Clasificacion usando redes convolucionales en datasets sencillos -- Generadores de datos -- Enriquecimiento de datos (dataaugmentation) -- Visualización de las capas ocultas -- Aprendizaje por transferencia (transfer learning) -- Autoencoders --Aprendizaje generativo.',i.cIsbn='978-958-792-145-8',i.cNotes='Ilustraciones, gráficos',i.cTopics='Procesamiento de datos',i.cType='Libro',i.cImage='https://biblioteca.unap.edu.pe/opac_css/getimage.php?url_image=http%3A%2F%2Fimages-eu.amazon.com%2Fimages%2FP%2F%21%21isbn%21%21.08.MZZZZZZZ.jpg&noticecode=9789587921458&entity_id=&vigurl=https%3A%2F%2Fedicionesdelau.com%2Fwp-content%2Fuploads%2F2021%2F02%2FMachine-Learniing_DIG.jpg',i.nViews=0,i.status_at=True,i.updated_at='2023-07-19 23:09:34.582959',i.created_at='2023-07-19 23:09:34.582959' ON MATCH SET i.updated_at='2023-07-19 23:09:34.582959' RETURN ID (i) AS idItem
Summary ▸	{, "query": {, "text": "MERGE (i:Title{identity_at:'4fbde0249d558eea6e54da5881cff10cee6d998612343388f48383af9c5bd96a'}) \r\n\t\tON CREATE SET i.cTitle='Machine learning y deep learning',i.cSubtitle='Usando python, scikit y keras',i.cEdition='Primera edición',i.nReleased=2021,i.cContent='Introducción -- Datasets -- Regresión -- Clasificación -- Clustering -- Reducción de dimensiones -- Redes neuronales -- Clasificación usando redes neuronales -- Redes convolucionales -- Clasificacion usando redes convolucionales en datasets sencillos -- Generadores de datos -- Enriquecimiento de datos (dataaugmentation) -- Visualización de las capas ocultas -- Aprendizaje por transferencia (transfer learning) -- Autoencoders --Aprendizaje generativo.',i.cIsbn='978-958-792-145-8',i.cNotes='Ilustraciones, gráficos',i.cTopics='Procesamiento de datos',i.cType='Libro',i.cImage='https://biblioteca.unap.edu.pe/opac_css/getimage.php?url_image=http%3A%2F%2Fimages-eu.amazon.com%2Fimages%2FP%2F%21%21isbn%21%21.08.MZZZZZZZ.jpg&noticecode=9789587921458&entity_id=&vigurl=https%3A%2F%2Fedicionesdelau.com%2Fwp-content%2Fuploads%2F2021%2F02%2FMachine-Learniing_DIG.jpg',i.nViews=0,i.status_at=True,i.updated_at='2023-07-19 23:09:34.582959',i.created_at='2023-07-19 23:09:34.582959' \r\n\t\tON MATCH SET i.updated_at='2023-07-19 23:09:34.582959' \r\nRETURN ID (i) AS idItem", ...
Response ▸	[, {, "keys": [...

Added 1 label, created 1 node, set 15 properties, started streaming 1 records after 20 ms and completed after 26 ms.

Elaboración propia.

ANEXO 5: Procesamiento de la recomendación por índice de Jaccard

```
neo4j$ MATCH (m:Title)←[:HAS_RESPONSIBILITY|ASSIGN_DEWEY|PUBLISHED|PART_TO]-(t)-[:HAS_RESPONSIBILITY|ASSIGN_DEWEY|PUBLISHED|PART_TO]→(recom:Title) WH…
```

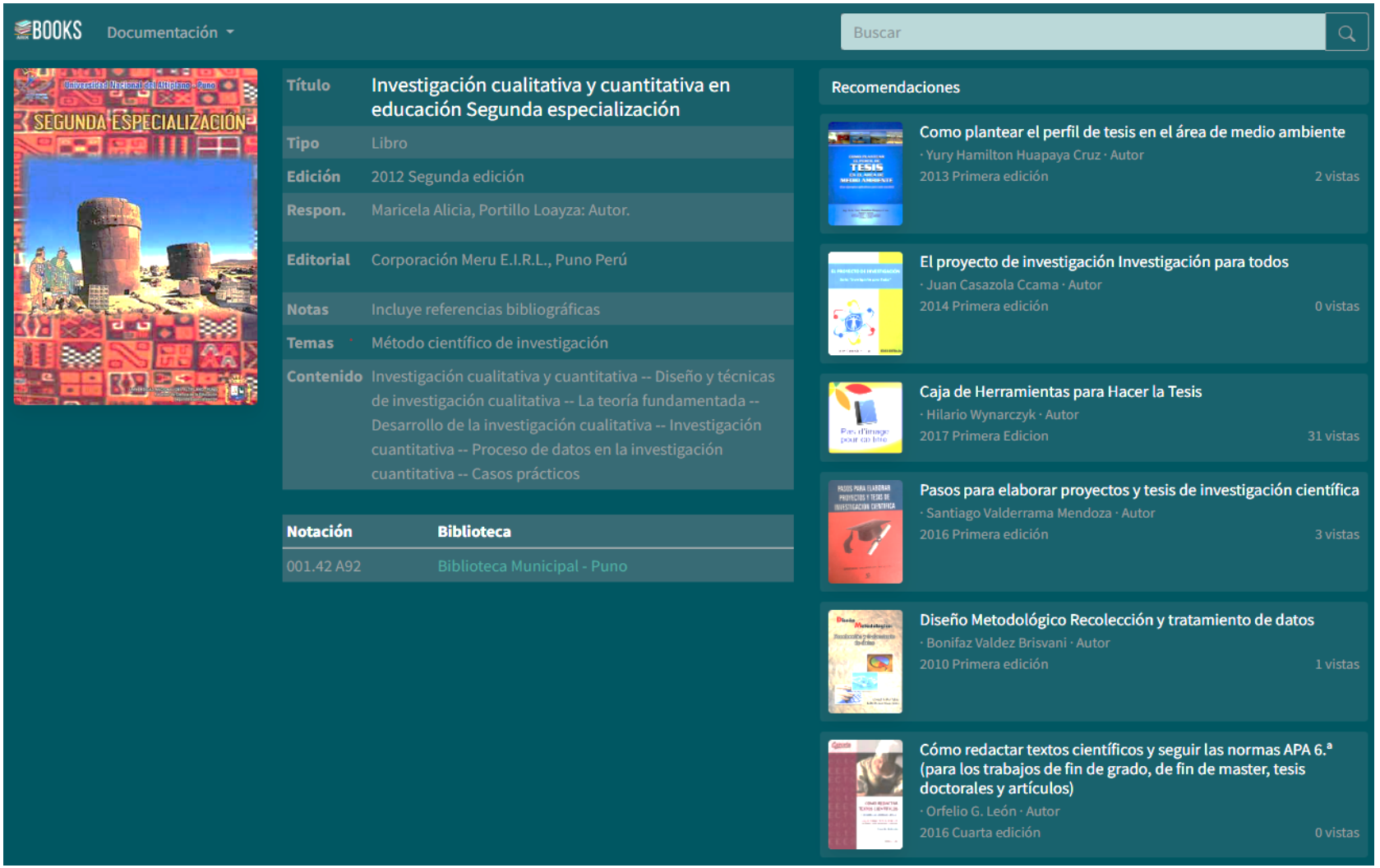
Table | Text | Warn | Code

	id	recom.cTitle	cla.cCode	code
1	3938	"Geografía general"	"910"	"910"
2	4114	"Geografía cartográfica"	"912.1"	"912"
3	4145	"Geografía"	"918.5"	"918"
4	45	"Pasos para elaborar proyectos y tesis de investigación científica"	"001.42"	"001"
5	2281	"Banco de matemáticas"	"510"	"510"
6	2351	"Aritmética"	"513"	"513"
7	5776	"Estadística aplicada a la investigación"	"519.5"	"519"
8	5778	"Estadística aplicada a la investigación"	"519.5"	"519"
9	7112	"El AEIOU del derecho"	"340"	"340"
10	5930	"Introducción a los Negocios de Exportación"	"621.3678"	"621"

Started streaming 10 records after 5 ms and completed after 13 ms.

Elaboración propia.

ANEXO 6: Modelo de recomendación en funcionamiento



Elaboración propia. Generado por el modelo de recomendación orientado a grafos.

DECLARACIÓN JURADA DE AUTENTICIDAD DE TESIS

Por el presente documento, Yo **Victor Jhampier Caxi Maquera** identificado con DNI N° 48207109 en mi condición de egresado de la **Escuela Profesional de Ingeniería de Sistemas**, informo que he elaborado la **Tesis** denominada: "**MODELO DE DATOS ORIENTADO A GRAFOS PARA EL PROCESAMIENTO DE RECOMENDACIONES DE LIBROS BASADOS EN EL ESTÁNDAR MACHINE READABLE CATALOGING**".

Es un tema original.

Declaro que el presente trabajo de tesis es elaborado por mi persona y **no existe plagio/copia** de ninguna naturaleza, en especial de otro documento de investigación (tesis, revista, texto, congreso, o similar) presentado por persona natural o jurídica alguna ante instituciones académicas, profesionales, de investigación o similares, en el país o en el extranjero.

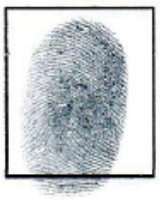
Dejo constancia que las citas de otros autores han sido debidamente identificadas en el trabajo de investigación, por lo que no asumiré como suyas las opiniones vertidas por terceros, ya sea de fuentes encontradas en medios escritos, digitales o Internet.

Asimismo, ratifico que soy plenamente consciente de todo el contenido de la tesis y asumo la responsabilidad de cualquier error u omisión en el documento, así como de las connotaciones éticas y legales involucradas.

En caso de incumplimiento de esta declaración, me someto a las disposiciones legales vigentes y a las sanciones correspondientes de igual forma me someto a las sanciones establecidas en las Directivas y otras normas internas, así como las que me alcancen del Código Civil y Normas Legales conexas por el incumplimiento del presente compromiso

Puno 28 de noviembre del 2023

Victor Jhampier Caxi Maquera
DNI N° 48207109



AUTORIZACIÓN PARA EL DEPÓSITO DE TESIS O TRABAJO DE INVESTIGACIÓN EN EL REPOSITORIO INSTITUCIONAL

Por el presente documento, Yo **Victor Jhampier Caxi Maquera**, identificado con DNI N° 48207109 en mi condición de egresado de **Escuela Profesional de Ingeniería de Sistemas**, informo que he elaborado la **Tesis** denominada: "MODELO DE DATOS ORIENTADO A GRAFOS PARA EL PROCESAMIENTO DE RECOMENDACIONES DE LIBROS BASADOS EN EL ESTÁNDAR MACHINE READABLE CATALOGING" para la obtención de **Título Profesional.**

Por medio del presente documento, afirmo y garantizo ser el legítimo, único y exclusivo titular de todos los derechos de propiedad intelectual sobre los documentos arriba mencionados, las obras, los contenidos, los productos y/o las creaciones en general (en adelante, los "Contenidos") que serán incluidos en el repositorio institucional de la Universidad Nacional del Altiplano de Puno.

También, doy seguridad de que los contenidos entregados se encuentran libres de toda contraseña, restricción o medida tecnológica de protección, con la finalidad de permitir que se puedan leer, descargar, reproducir, distribuir, imprimir, buscar y enlazar los textos completos, sin limitación alguna.

Autorizo a la Universidad Nacional del Altiplano de Puno a publicar los Contenidos en el Repositorio Institucional y, en consecuencia, en el Repositorio Nacional Digital de Ciencia, Tecnología e Innovación de Acceso Abierto, sobre la base de lo establecido en la Ley N° 30035, sus normas reglamentarias, modificatorias, sustitutorias y conexas, y de acuerdo con las políticas de acceso abierto que la Universidad aplique en relación con sus Repositorios Institucionales. Autorizo expresamente toda consulta y uso de los Contenidos, por parte de cualquier persona, por el tiempo de duración de los derechos patrimoniales de autor y derechos conexos, a título gratuito y a nivel mundial.

En consecuencia, la Universidad tendrá la posibilidad de divulgar y difundir los Contenidos, de manera total o parcial, sin limitación alguna y sin derecho a pago de contraprestación, remuneración ni regalía alguna a favor mío; en los medios, canales y plataformas que la Universidad y/o el Estado de la República del Perú determinen, a nivel mundial, sin restricción geográfica alguna y de manera indefinida, pudiendo crear y/o extraer los metadatos sobre los Contenidos, e incluir los Contenidos en los índices y buscadores que estimen necesarios para promover su difusión.

Autorizo que los Contenidos sean puestos a disposición del público a través de la siguiente licencia:

Creative Commons Reconocimiento-NoComercial-CompartirIgual 4.0 Internacional. Para ver una copia de esta licencia, visita: https://creativecommons.org/licenses/by-nc-sa/4.0/

En señal de conformidad, suscribo el presente documento.

Puno 28 de noviembre del 2023

Victor Jhampier Caxi Maquera
DNI N° 48207109