

# **Instituto Tecnológico y de Estudios Superiores de Occidente**

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

Departamento de Electrónica, Sistemas e Informática  
**Maestría en Sistemas Computacionales**



## **SISTEMA DE RECOMENDACIÓN BASADO EN GRAFOS: ESCENARIO DE RECOMENDACIÓN DE PRODUCTOS**

---

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

Presenta: **ING. SARAHI PARTIDA OCHOA**

Director: **MTRO. VÍCTOR HUGO ORTEGA**

Tlaquepaque, Jalisco. Junio de 2021.



# AGRADECIMIENTOS

Gracias al apoyo económico que me fue otorgado por el Consejo Nacional de Ciencias y Tecnología (CONACYT) con la beca número 744908 y por la empresa Oracle, ha sido posible que yo estudie en el Instituto Tecnológico y de Estudios Superiores de Occidente (ITESO).

El camino recorrido a través de estos dos años ha sido sembrado de dificultades con numerosas alegrías y experiencias las cuales forman a la persona que soy hoy en día. Hubo días los cuales fueron muy pesados pero que ahora quedan como parte de mi experiencia, de cómo sobrellevar los momentos de estrés para salir adelante.

En primer lugar, agradezco a mi asesor de tesis el Maestro Víctor Hugo Ortega por la confianza brindada, sus ideas, consejos, tiempo y apoyo dado durante la realización de este documento. No hubiera sido posible lograrlo sin su ayuda.

Sin duda alguna quiero reconocer el apoyo otorgado por mi Gerente Luis Rocholl haciendo posible la culminación de mi maestría; recuerdo cuando me uní a su equipo, mi primer día con *Database Testing*; la verdad fue una experiencia muy bonita sentirme parte de ellos sin siquiera conocerme, donde mis mentores fueron Juan Carlos o JC; la persona que me recomendó al equipo y Luis dándome la oportunidad de ser parte de su equipo confiando en mí. Gracias a Luis ahora soy líder de los proyectos en los cuales me encuentro laborando ahora, haciendo posible la financiación con Oracle. Mi mejor agradecimiento a Luis por seguir confiando en mí y ser mi mentor, enseñándome día a día como ser mejor en actividades laborales y levantarme de mis tropiezos.

Helena Kee la persona más bondadosa que he conocido. Sus consejos me han logrado entender dificultades presentadas con mis compañeros. Cuando la conocí recuerdo los nervios que sentí al hablar con ella, después me inspiró su tranquilidad. Después de su partida de la empresa, la sigo extrañando.

A mis compañeros de proyecto. A Jorge por su ayuda durante estos dos años. Sin su apoyo hubiese sido muy complicado terminar satisfactoriamente esta tesis. A Iñaqui, su paciencia para explicarme los algoritmos que yo no entendía compartiendo su experiencia y conocimiento conmigo.

A mi maestro Luis Fernando Gutiérrez, por su apoyo en la generación del sitio web que sirve de sustento a mi proyecto de tesis y por los conocimientos adquiridos en algoritmia. Realmente reitero mi profundo agradecimiento sin negarme nunca una consulta, donde incluso en fines de semana estuvo disponible para ayudarme.

A mi maestro José Francisco Cervantes Álvarez por ser mi guía en *machine learning*. Los modelos aprendidos para la complementación de mi tesis fueron de gran ayuda haciendo posible la culminación del presente trabajo.

A mi amigo casi hermano Oscar Guzmán, mi padrino, *cersei*, que desde que me conoció siempre me ha brindado su conocimiento, amistad y apoyo incondicional como ha sido en la salud y la enfermedad; enseñándome que en la vida existen personas que no sean parte de tu familia te hacen sentir parte de ella.

A mi Tía Hermelinda Ochoa por ser una segunda madre, dándome sus consejos y apoyo en los momentos difíciles de mi vida. A mis primos por su cariño, confianza y apoyo brindado a la familia.

A mis tíos: Alfonso Ochoa, en memoria de Rafael Ochoa y Antonio Ochoa por ser mi figura paterna dándome su apoyo y cariño incondicional.

Mi hermano ha sido mi ejemplo para continuar preparándome, me ha enseñado la bondad de la vida, donde no importa nada si no tenemos el amor de los que amamos; sin olvidar los momentos buenos y malos que hemos vivido juntos y seguiremos compartiendo.

Por último, le doy gracias a la persona más importante y valiosa que tengo en esta vida mi madre. Ella siempre ha estado luchando sin derrumbarse nunca por las adversidades que hemos vivido. Ha sido padre y madre desde que me llevaba en su vientre y enfrentar la muerte de mi padre cuando yo todavía no llegaba a este mundo. Me siento orgullosa de tener a mi madre conmigo. Los momentos donde yo estaba derrumbada ella siempre estaba ayudándome a superar mis caídas. Todo lo que he logrado al día de hoy se lo debo a ella por enseñarme que la vida es un continuo aprendizaje, donde debo luchar por lo que quiero lograr, donde existirán caídas, pero serán de aprendizaje para mejorar. Gracias de verdad....

En memoria de mi padre....

# DEDICATORIA

A mi madre y hermano por guiarme y estar a mi lado. La vida está inmersa en oportunidades, de nosotros depende si decidimos tomarlas.

# RESUMEN

NEO4J es un software de base de datos de grafos escalable y ejecutable en plataformas comunes; corriendo en modo autónomo en aplicaciones o modo manual con la intervención del usuario. En este documento, se describe el desarrollo de un sistema de recomendación de productos de compra utilizando un set de datos públicos de Amazon, el cual ha sido diseñado e implementado debido a las necesidades actuales presentadas en la sociedad por el uso masivo de plataformas virtuales. En dicho set de datos se obtiene la información cruda generando relaciones a las compras, reseñas, productos similares, clientes que hacen dichas compras y los grupos como tipo de libros.

La recomendación es de tipo híbrida debido al uso de una recomendación colaborativa y de grafos. Después de extraer la información, luego es mapeada a un grafo con nodos y relaciones en una base de datos de NEO4J; sus conexiones nos indican las relaciones entre los productos, clientes y grupos utilizándola para la generación de recomendaciones las cuales se generan automáticamente.

La información es almacenada en una base de datos de NEO4J donde se hizo uso de librerías disponibles como es APOC la cual contiene alrededor de 450 algoritmos incluyendo integración de datos, algoritmos de grafos o de conversión de datos. las consultas realizadas se basan en algoritmos de búsqueda, centralidad y comunidad, generando una recomendación más certera acorde al producto o artículo seleccionado. Es fundamental contar con la disponibilidad de recomendadores de este tipo debido al incremento masivo de compras en líneas en los últimos tiempos proveyendo recomendaciones a los usuarios.

# ABSTRACT

NEO4J is a scalable graph database software executable on common platforms running in standalone mode in applications or manual mode with user intervention. This document describes developing a shopping product recommendation system using an Amazon public data set; this system has been designed and implemented due to the needs presented in society by the massive use of virtual platforms. In this dataset, basic information is obtained by getting relationships to purchases, reviews, similar products, customers who make such purchases, and groups as a type of book.

The recommendation is of a hybrid type due to the use of collaborative recommendations and graphs. After extracting the information, it is then mapped to a graph with nodes and relationships in a NEO4J database; the connections indicate the relationships between products, customers, and groups, using it to generate automatically generated recommendations.

The information is stored in a NEO4J database, where use was made of available libraries such as APOC, which contains around 450 algorithms, including data integration, graphing, or data conversion algorithms. According to the selected product or article, the queries made are based on search algorithms, centrality, and community, generating a more accurate recommendation. It is essential to have recommenders of this type due to the online purchases massive increase in recent times providing recommendations to users.

# TABLA DE CONTENIDO

<b>1. INTRODUCCIÓN .....</b>	<b>15</b>
1.1. ANTECEDENTES .....	16
1.2. JUSTIFICACIÓN.....	16
1.3. PROBLEMA.....	16
1.4. OBJETIVOS.....	17
1.4.1. <i>Objetivo General:</i> .....	17
1.4.2. <i>Objetivos Específicos:</i> .....	17
1.5. NOVEDAD CIENTÍFICA, TECNOLÓGICA O APORTACIÓN .....	17
<b>2. ESTADO DEL ARTE O DE LA TÉCNICA.....</b>	<b>18</b>
2.1. XEROX PARC TAPESTRY (1992).....	20
2.2. MOVIE LENS (1996).....	20
2.3. FAB (1997) .....	21
2.4. E-COMMERCE (2004).....	21
2.5. CARACTERÍSTICAS DE POPULARIDAD DEL PRODUCTO (2006) .....	21
2.6. SERENDIPITY (2008).....	22
2.6.1. FILTRO BASADO EN CONTENIDO .....	22
2.6.2. PERFIL DEL ALUMNO .....	22
2.6.3. RECOMENDADOR.....	23
2.7. MEDIDA DE SIMILITUD DEL USUARIO BASADA EN LA PERSONALIDAD (2009).....	23
2.8. DATOS VINCULADOS (2010).....	23
2.9. RECOMENDADOR SOCIAL BASADO EN GRAFOS (2012) .....	24
2.10. ETIQUETADO EFECTIVO (2013).....	24
2.11. POPULARIDAD DEL USUARIO (2020) .....	25
2.12. COMPARATIVA DE RECOMENDADORES .....	25
<b>3. MARCO TEÓRICO/CONCEPTUAL.....</b>	<b>28</b>
3.1. SISTEMA DE RECOMENDACIÓN.....	29
3.2. MÉTODOS DE RECOMENDACIÓN.....	30
3.2.1. FILTRO BASADO EN CONTENIDO .....	30
3.2.1.1. MÉTODO DE REPRESENTACIÓN TF-IDF.....	31
3.2.1.2. ALGORITMO LDA .....	31
3.2.1.3. FILTRO COLABORATIVO.....	31
3.2.1.4. BASADO EN USUARIO .....	32
3.2.1.5. BASADO EN ÍTEMS .....	32
3.2.2. BASADO EN GRAFOS .....	32
3.2.3. HÍBRIDO .....	33
3.3. MÉTODOS DE EVALUACIÓN.....	34
3.3.1. PRECISIÓN .....	34
3.3.2. RECUPERACIÓN .....	34
3.3.3. MEDIDA F.....	35
3.4. MEDIDORES DE EFICIENCIA.....	35
3.4.1. ARRANQUE EN FRÍO.....	35
3.4.2. ESCASEZ.....	35
3.4.3. ESCALABILIDAD .....	36
3.4.4. HALLAZGO FORTUITO.....	36



3.5.	BASES DE DATOS NOSQL .....	36
3.5.1.	NEO4J .....	36
3.5.1.1.	CYPHER .....	37
3.5.1.2	NEOVYS.....	37
3.6.	ALGORITMOS DE CENTRALIDAD.....	38
3.6.1.1.	PAGE RANK.....	38
3.6.1.1.1.	INFLUENCIA .....	39
3.6.1.1.2	FÓRMULA DE PAGERANK.....	39
3.6.1.2	CENTRALIDAD DE INTERMEDIACIÓN.....	41
3.6.1.2.1	CALCULANDO CENTRALIDAD DE INTERMEDIACIÓN.....	41
3.7.	ALGORITMOS DE DETECCIÓN DE COMUNIDADES.....	43
3.7.1.1.	COMPONENTES FUERTEMENTE CONECTADOS .....	43
3.7.1.2.	LABEL PROPAGATION .....	44
3.7.1.2.1	APRENDIZAJE SEMI-SUPERVISADO Y ETIQUETAS DE SEMILLAS.....	46
3.7.1.3.	MODULARIDAD DE LOUVIAN .....	46
3.1.7.3.2	AGRUPACIÓN BASADA EN LA CALIDAD MEDIANTE MODULARIDAD .....	47
3.8.	MINERÍA DE TEXTO.....	49
3.9.	DESARROLLO WEB.....	50
3.9.1	SERVIDOR FRONTEND .....	50
3.9.2	SERVIDOR BACKEND.....	50
3.9.3	JSON .....	51
<b>4.</b>	<b>DESARROLLO METODOLÓGICO.....</b>	<b>52</b>
4.1	METODOLOGÍA DEL PROYECTO.....	53
4.2	OBTENCIÓN DE LA INFORMACIÓN CRUDA .....	58
4.2.1	AMAZON.....	60
4.3	MINERÍA DE TEXTOS .....	60
4.3.1	NORMALIZACIÓN.....	60
4.3.2	TOKENIZACIÓN .....	61
4.3.3	TRADUCCIÓN A UN FORMATO CSV .....	61
4.4	GENERACIÓN DE LA BASE DE DATOS.....	61
4.5	IMPLEMENTACIÓN DE ALGORITMOS DE CLUSTERING (LOUVAIN), RELEVANCIA (PAGERANK) .....	61
4.6	CREACIÓN DE SERVIDOR BACKEND.....	62
4.7	DESARROLLO DE LA INTERFAZ GRÁFICA UI.....	63
<b>5.</b>	<b>RESULTADOS Y DISCUSIÓN.....</b>	<b>65</b>
5.1.	RESULTADOS .....	66
5.2.	DISCUSIÓN.....	88
<b>6.</b>	<b>CONCLUSIONES .....</b>	<b>89</b>
6.1.	CONCLUSIONES.....	90
6.2.	TRABAJO FUTURO.....	90

# LISTA DE FIGURAS

Figura 1. Arquitectura general de <i>Serendipity</i> [7].....	22
Figura 2. Tipos de sistemas de recomendación [13]. ....	30
Figura 3. Proceso de un sistema de recomendación basado en contenido [13].....	31
Figura 4. Matrices de índices para un sistema de recomendación de filtro colaborativo [13]. ....	32
Figura 5. Grafo usado para un sistema de recomendación básico [13]. ....	33
Figura 6. Método híbrido (CF + CFB) paralelo [13]. ....	33
Figura 7. Precisión vs Recuperación [13]. ....	35
Figura 8. Entidades en <i>Cypher</i> .....	37
Figura 9. Visualización de un grafo usando Neovis [25]. ....	38
Figura 10. Visualización de un grafo usando <i>Neovis</i> [26]. ....	39
Figura 11. Iteraciones de <i>PageRank</i> [26].....	40
Figura 12. Los nodos fundamentales se encuentran en cada camino más corto entre dos nodos. La creación de rutas más cortas puede reducir la cantidad de nodos fundamentales para usos como la mitigación de riesgos [26].....	41
Figura 13. Conceptos básicos para calcular la centralidad de intermediación [26]. ....	42
Figura 14. Visualización de la centralidad de intermediación [26].....	43
Figura 15. Componentes Fuertemente Conectados [26]. ....	44
Figura 16. Propagación de Etiquetas método de empuje simple [26]. ....	45
Figura 17. Propagación de Etiquetas método de tracción [26]. ....	45
Figura 18. Cuatro puntuaciones de modularidad basadas en diferentes opciones de partición.[26] .....	48
Figura 19. Proceso de algoritmo de <i>Louvian</i> [26].....	48
Figura 20. Etapas del sistema de recomendación de productos de co-compra. ....	54
Figura 21. Estructura del grafo de sistema de recomendación.....	55
Figura 22. Primera fase del recomendador de artículos. ....	56
Figura 23. Segunda fase del recomendador de artículos.....	57
Figura 24. Tercera fase del recomendador de artículos. ....	58
Figura 25. Set de datos de los productos de co-compra de Amazon [33]. ....	59
Figura 26. Formato de productos de Amazon [33]. ....	59
Figura 27. Salida de un <i>query</i> de fuertemente conectados.....	62
Figura 28. Salida de un <i>query</i> de débilmente conectados. ....	62
Figura 29. Página principal de la interfaz gráfica del recomendador de productos de co-compra. ....	63
Figura 30. Fragmento de código del <i>router</i> de cliente. ....	64
Figura 31. Directorio <i>import</i> en NEO4J.....	66
Figura 32. Comando de importación de CSVs a NEO4J. ....	67
Figura 33. Número de nodos y relaciones de la base de datos de NEO4J. ....	67
Figura 34. <i>Plugin</i> de APOC de NEO4J.....	67
Figura 35. Página principal del sistema de recomendación de productos de Amazon.....	68
Figura 36. Imágenes en el carrusel de la página principal .....	68

Figura 37. Menú de filtros disponibles en el sistema de recomendación.....	69
Figura 38. Menú de productos. ....	69
Figura 39. <i>Query</i> de todos los productos. ....	69
Figura 40. <i>Query</i> de los productos relevantes.....	70
Figura 41. Búsqueda de todos los productos.....	70
Figura 42. Búsqueda de productos relevantes.....	70
Figura 43. Búsqueda por listas.....	71
Figura 44. Búsqueda por palabras.....	71
Figura 45. Búsqueda por etiquetas.....	71
Figura 46. Remover etiquetas. ....	72
Figura 47. Filtrado por <i>Product ID</i> . ....	72
Figura 48. Filtrado por <i>Product ID</i> y <i>Title</i> .....	72
Figura 49. Filtrado por <i>Total Reviews</i> , <i>Categories</i> , <i>Average Rating</i> , <i>Salesrank</i> , <i>Downloaded</i> . ....	73
Figura 50. <i>Query</i> del algoritmo de <i>Lovian</i> . ....	73
Figura 51. <i>Query</i> del algoritmo de similaridad. ....	74
Figura 52. Recomendaciones por producto.....	74
Figura 53. Recomendaciones por grupo. ....	74
Figura 54. Recomendaciones por cliente. ....	74
Figura 55. Grafo de recomendaciones por producto.....	75
Figura 56. Menú de grupos. ....	75
Figura 57. <i>Query</i> de todos los grupos. ....	75
Figura 58. <i>Query</i> de grupos relevantes. ....	76
Figura 59. Búsqueda de todos los grupos. ....	76
Figura 60. Búsqueda de grupos relevantes.....	76
Figura 61. Filtrado de grupos con <i>Group ID</i> . ....	77
Figura 62. Filtrado de grupos con <i>Group ID</i> y <i>Group name</i> .....	77
Figura 63. <i>Query</i> de producto usando el algoritmo de <i>Louvian</i> .....	77
Figura 64. <i>Query</i> de producto usando el algoritmo de <i>Louvian</i> .....	78
Figura 65. Recomendaciones por producto.....	78
Figura 66. Recomendaciones por grupo. ....	78
Figura 67. Recomendaciones por cliente. ....	79
Figura 68. Grafo de recomendaciones por grupo.....	79
Figura 69. Menú de clientes.....	80
Figura 70. <i>Query</i> de todos los clientes. ....	80
Figura 71. <i>Query</i> de clientes relevantes. ....	80
Figura 72. Búsqueda de <i>All Customers</i> .....	80
Figura 73. Búsqueda de <i>Relevant Customers</i> .....	81
Figura 74. Filtrado de <i>Customer ID</i> .....	81
Figura 75. <i>Query</i> de <i>Louvian</i> para clientes. ....	81
Figura 76. <i>Query</i> de similar para clientes. ....	82

Figura 77. Recomendaciones por producto.....	82
Figura 78. Recomendaciones por grupo. ....	82
Figura 79. Recomendaciones por cliente. ....	83
Figura 80. Grafo generado para una recomendación de cliente.....	83
Figura 81. Menú de reseñas. ....	84
Figura 82. <i>Query</i> de reseñas.....	84
Figura 83. Búsqueda de reseñas por defecto.....	84
Figura 84. Búsqueda de reseñas por cliente. ....	85
Figura 85. Búsqueda de reseñas por producto. ....	85
Figura 86. Búsqueda de reseñas por mes. ....	86
Figura 87. Búsqueda de reseñas por mes. ....	86
Figura 88. Recomendación por filtrado de reseñas. ....	87
Figura 89. Grafo de recomendación por reseñas.....	87
Figura 90. Comando de llamadas a NEO4J. ....	88
Figura 91. Descarga NEO4J [35].....	94
Figura 92. Clave de activación NEO4J [36]. ....	94
Figura 93. Panel principal Neo4j [36].....	95
Figura 94. Crear nuevo proyecto en NEO4J [36]. ....	96
Figura 95. Inicializar base de datos en NEO4J [36].....	97
Figura 96. Administrar base de datos en NEO4J [36]. ....	97
Figura 97. Instalar APOC a nivel proyecto [37]. ....	99
Figura 98. Instalar APOC a nivel base de datos paso 1 y 2 [37].....	100
Figura 99. Instalar APOC a nivel base de datos paso 3 y 4 [37].....	101
Figura 100. Visualización de un grafo usando <i>neovis</i> [37].....	105

## LISTA DE TABLAS

Tabla 1. Ventajas ofrecidas por el recomendador de productos de co-compra.....	25
Tabla 2. Ventajas y desventajas de los diferentes sistemas de recomendación [15].....	34
Tabla 3. Cálculo del Nodo D [26].....	42
Tabla 4. Nodos.....	55
Tabla 5. Aristas.....	55

# LISTA DE ACRÓNIMOS Y ABREVIATURAS

ACRÓNIMO	DESCRIPCIÓN
IDI	Investigación, desarrollo e innovación.
TOG	Trabajo de obtención de grado.
NDCG	<i>Normalized Discounted Cumulative Gain</i> . Métrica que evalúa la calidad de una lista de ítems recomendados.
MAP	<i>Mean Average Precision</i> . Métrica que mide la exactitud promedio de ítems relevantes para el usuario final.
RMSE	<i>Root Mean Square Error</i> . Se usa para identificar la diferencia entre el valor esperado y el valor calculado de las instancias en el modelo o grafo. Considerando la etapa de <i>training</i> y <i>testing</i> del mismo modelo.
TF-IDF	<i>Term frequency – Inverse document frequency</i> . Mide la importancia de una palabra dentro de un documento y dentro de un set de documentos.
CYPHER	Lenguaje de consultas que permite a los usuarios obtener información de una base de datos basada en grafos.
NEO4J	Es un software libre de Base de datos orientada a grafos, implementado en Java.
APOC	<i>Awesome Procedures On Cypher. A Package Of Components</i> . Paquete de algoritmos.
SNAP	<i>Stanford Network Analysis</i> . Análisis de Redes de Stanford
GroupLens	Estudio, comprensión y construcción de sistemas utilizados por personas
CF	<i>Collaborative Filter</i> . Filtro Colaborativo
CFB	<i>Content Filter Based</i> . Filtro Basado en Contenido
CBR	<i>Content Base Recommender</i> . Recomendador Basado en Contenidos
JSON	<i>JavaScript Object Notation</i> . Notación de Objetos en <i>JavaScript</i>
IDE	<i>Integrated Development Environment</i> . Entorno de Desarrollo Integrado
ES6	<i>ECMAScript 6</i> . Especificación de lenguaje de programación publicada por <i>ECMA International</i> .
LPA	<i>Label Propagation Algorithm</i> . Algoritmo Propagación de Etiquetas
CSV	<i>Comma Separated Values</i> . Valores Separados por Comas
CSS	<i>Cascading Style Sheets</i> . Hoja de Estilos en Cascada
HTML	<i>HyperText Markup Language</i> . Lenguaje de Marcas de Hipertexto
PHP	<i>Hypertext Preprocessor</i> . Preprocesador de Hipertexto
CMS	<i>Content Management System</i> . Sistema de Gestión de Contenidos
ASIN	<i>Amazon Standard Identification Number</i> . Número de Identificación Estándar de Amazon
LDA	<i>Latent Dirichlet Allocation</i> . Asignación de <i>Dirichlet</i> Latente
ACID	<i>Atomicity, Consistency, Isolation, Durability</i> . Atomicidad, Consistencia, Isolación, Durabilidad
SQL	<i>Structured Query Language</i> . Lenguaje de Consulta Estructurado
NoSQL	<i>Not only SQL</i> . No solo SQL
ADN	Ácido Desoxirribonucleico
DVD	<i>Digital Versatile Disc</i> . Disco Versátil Digital
VS	<i>Visual Studio</i>
API	<i>Application Programming Interface</i> . Interfaz de Programación de Aplicaciones
NPM	<i>Node Package Manager</i> . Administrador de Paquetes de Nodo
URL	<i>Uniform Resource Locator</i> . Localizador de Recursos Uniforme
UI	<i>User Interface</i> . Interfaz de Usuario

---

# 1. INTRODUCCIÓN

---

## 1.1. Antecedentes

Un sistema de recomendación usando la correlación elemento a elemento recomienda productos a un cliente basado en asociaciones entre el nuevo artículo. Al enfoque de artículos en los cuales el cliente expresa su interés es conocido como *Content-base approach*. El perfil de un artículo consiste en atributos del producto y las ocurrencias de elección.

Existen sistemas de recomendaciones que usan correlación de usuario a usuario, conocido como aproximación de filtros colaborativos. Estos sistemas ofrecen recomendaciones basadas en correlaciones entre usuarios buscando comportamientos en el sistema. Los enfoques basados en contenido y filtrado colaborativo no son mutuamente excluyentes, donde existen esfuerzos en integrarlos y obtener una recomendación más precisa.

## 1.2. Justificación

En el mundo actual la tecnología ha avanzado a pasos agigantados, la compra en línea se ha convertido en una necesidad. Es fundamental poseer un recomendador de productos, acorde a los gustos de compradores en línea desde la comodidad de su casa.

Con este proyecto se busca que los usuarios interesados en un tema en específico de artículos, tengan la opción de recomendaciones certeras, sin la necesidad de buscar una gran cantidad de artículos que no sean de su agrado. Hoy en día, existen diversos tipos de sistemas de recomendación como los basados en contenido, basados en filtro colaborativo, pero un sistema basado en grafos y colaborativo como el que se aplica en este proyecto no, dicho recomendador es más eficaz, ya que se puede aprovechar la topología del modelo y la relación entre las entidades para hacer un sistema híbrido de recomendaciones con mayor exactitud.

## 1.3. Problema

Existe un bombardeo de información en las páginas web de compras digitales, donde el usuario es abrumado y no encuentra artículos de su interés. Es necesario poseer una herramienta generadora de recomendaciones certeras al usuario y de rápida respuesta. Con nuestro recomendador de productos; al cliente se le recomendarán artículos acordes a su historial de búsquedas; sugerencias de productos comprados por usuarios que poseen perfiles similares. Las reseñas del usuario son de vital importancia para conocer más acerca de sus intereses de gustos o cuestiones no de su agrado respectivamente. Actualmente existen recomendadores de productos aislados, donde se basan en el historial de compras de los usuarios, pero no validan las reseñas o relación de productos en una sola herramienta.



## 1.4. Objetivos

### *1.4.1. Objetivo General:*

Desarrollar un sistema de recomendación de productos basado en grafos. El usuario tiene la opción de hacer búsquedas por producto, cliente, reseña y grupo; donde se despliegan una lista de recomendaciones, dicha información se muestra en una interfaz gráfica que apoya al usuario. El set de datos se guarda en una base de datos NEOJ4. Nuestro sistema es de propósito específico para el manejo de artículos de compra en línea.

### *1.4.2. Objetivos Específicos:*

- Obtener búsquedas certeras por cliente, producto, y grupo
- Tener la flexibilidad de generar filtros por reseñas, productos similares, productos de co-compra
- Proveer recomendaciones por cliente, producto, y grupo
- Generación de un grafo visual de la recomendación

## 1.5. Novedad científica, tecnológica o aportación

En la recomendación de productos acorde a las reseñas de clientes, es evitar que se sature con información que el usuario no esté interesado ya que está buscando comprar productos de calidad, utilidad que realmente sean de su interés y no se desvíe con información que no desea. Este documento presenta la creación de un sistema de recomendación híbrido para la recomendación de productos. En el siguiente capítulo se hablará sobre estudios previos de sistemas de recomendación, sus tipos y un resumen general de las diferencias con el presente trabajo.

---

## 2. ESTADO DEL ARTE O DE LA TÉCNICA

---

El objetivo fundamental del estado del arte es buscar conocimientos paralelos al proyecto y establecer comparativas, generando diferentes opciones de comprender el problema actual. Se busca obtener un plus al recomendador de productos y ventajas ofrecidas sobre los recomendadores existentes.

La extensa información en tiendas digitales ha creado un inconveniente a los usuarios en la búsqueda de recursos en línea de compras. Usuarios con ciertos intereses se le puede proveer la misma información en respuesta a las consultas de búsqueda que ellos realizan. Historias individuales de compras no son tomadas en cuenta para sugerir productos a los usuarios. Para recuperar productos relevantes con una búsqueda en específico, existen tiendas digitales con la información necesaria para generar una recomendación. Se tiene que tener en cuenta la votación, intereses.

Una situación similar aparece en los anuncios mundiales, donde existe extensa información del producto que están ofreciendo las tiendas en línea. Algunos sitios de venta en línea emplean varios enfoques y fuentes de información, los cuales sugieren artículos relevantes a los clientes acorde a las búsquedas que realizan cuando navegan por sus páginas. Los sistemas de recomendaciones generan anuncios basados en compras pasadas y características de los productos. Para cualquier sistema de recomendación es fundamental la elección de la información para elaborar dicha recomendación. A continuación se listan 6 importantes categorías para realizar un sistema de recomendación exitosa [1]:

- Recuperación en bruto
- Selección manual
- Resumen estadístico
- Basado en atributos
- Correlación de elemento a elemento
- Correlación de usuario a usuario.

Existen diversos tipos de sistemas de recomendación que se adaptan a las diferentes necesidades que la plataforma y usuario final tengan. Desde la fuente de información que buscará almacenar todo el contenido para tener acceso desde los algoritmos como la base de datos. Además, existen tipos de sistemas de recomendación basado en sus tipos de motor de búsqueda; los más relevantes que han surgido a lo largo de los años son [1]:

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

Un sistema de recomendación usando la correlación elemento a elemento recomienda productos a un cliente basado en asociaciones entre el nuevo artículo. El enfoque de artículos en los cuales el cliente expresa su interés. Este método es conocido como *Content-base approach* [2]. Un perfil de un artículo usualmente consiste en algunos atributos del producto y el perfil del usuario es creado desde los perfiles involucrados de los productos que cuenta como historial.

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

Una manera simple de combinar ambos es ejecutar recomendaciones usando dos esfuerzos por separado y combinar los resultados. Algunos sistemas son híbridos, los cuales combinan dos esfuerzos por un nivel más bajo generados por artículos o usuarios. Un esfuerzo puro de filtros colaborativos usualmente confía en la información de transacciones, mientras un contenido basado en el enfoque usualmente utiliza datos factuales del artículo [1].

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

## 2.1. Xerox PARC Tapestry (1992)

*Information Tapestry* fue un sistema experimental que se consideraba híbrido debido a que utilizaba un filtrado colaborativo y un filtrado basado en el contenido, así como la valoración y el resaltado automático, para adaptar la entrega y presentación de la información a los intereses personales los usuarios de *NetNews*[3].

Este sistema permitía almacenar el conocimiento de los usuarios sobre los artículos o noticias que éstos habían leído y posteriormente esta información era utilizada por otros usuarios que aún no habían leído el artículo o noticia, para establecer si la información del documento era relevante o no. En sus inicios este tipo de sistemas fue adoptado con el nombre de filtro colaborativo dado que permite que los usuarios creen filtros a través de sus ítems de interés, en el caso de *Tapestry* artículos o noticias, y colaborativo pues los usuarios añadían las anotaciones con las opiniones sobre los documentos. Las opiniones añadidas eran utilizadas para la búsqueda personalizada de otros usuarios [3].

## 2.2. Movie Lens (1996)

*MovieLens* es de los primeros sistemas de recomendación elaborados por el grupo Lens (*GroupLens*) y basa sus recomendaciones en la información proporcionada por los usuarios del sitio web, como la clasificación de películas. El sitio utiliza una variedad de algoritmos de recomendación, incluidos algoritmos de filtrado colaborativo [4].

En el año de 1996, *GroupLens* formó una empresa comercial llamada *Net Perceptions*, que atendía clientes que incluían las empresas *EOnline* y *Amazon.com*. *EOnline* utilizó los servicios de *Net Perceptions* para crear el sistema de recomendaciones para *Moviefinder.com*, mientras que *Amazon.com* utilizó la tecnología de la compañía para formar su motor de recomendación inicial para las compras de los consumidores. *GroupLens* utiliza también metodologías de análisis de información masiva dentro de un grupo de estudiantes científicos en Minnesota [4].

### 2.3. Fab (1997)

FAB es un sistema de recomendación híbrido diseñado para ayudar a los usuarios a examinar la enorme cantidad de información disponible en la *World Wide Web* y creado en el año de 1997 por Balavanovic y Shoham [2].

Se considera como un sistema de recomendación híbrido debido a que es basado en contenido y filtrado colaborativo. El sistema trabaja modelando el perfil del usuario basado en el contenido de los análisis cuando un usuario otorga calificación a una página web y compara estos perfiles para determinar similitudes entre usuarios para una recomendación colaborativa [2].

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

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

### 2.4. E-commerce (2004)

Es un modelo de grafos que proporciona una representación de datos genéricos y puede admitir diferentes métodos de recomendación. Para demostrar su utilidad y flexibilidad, desarrollaron tres métodos de recomendación: recuperación directa, extracción de asociaciones y recuperación de asociaciones de alto grado. Usan un conjunto de datos de una librería en línea como banco de pruebas de investigación. Los resultados de la evaluación mostraron que la combinación de la información del contenido del producto y la información histórica de las transacciones de los clientes lograba predicciones más precisas y recomendaciones relevantes que usar solo información colaborativa. Sin embargo, las comparaciones entre diferentes métodos mostraron que la recuperación de asociaciones de alto grado no tuvo un desempeño significativamente mejor que el método de extracción de asociación, o el método de recuperación directa en su banco de pruebas [5].

### 2.5. Características de popularidad del producto (2006)

Presenta un enfoque novedoso para la recomendación de productos automatizada basado en las características de popularidad de los productos. La popularidad juega un papel importante en el proceso de compra del consumidor, pero no se le ha prestado mucha atención en la investigación de recomendaciones. Se utiliza un modelo tridimensional de popularidad para desarrollar clases de popularidad de productos. Estos se combinan con el conjunto de datos *Movie Lens* para crear un sistema de recomendación de películas híbrido que combina información de género y popularidad [6].

## 2.6. Serendipity (2008)

Es un sistema de recomendación basado en contenido, desarrollado en la Universidad de Bari. El sistema es capaz de proporcionar recomendaciones para elementos en varios dominios (por ejemplo, películas, música, libros), siempre que las descripciones de los elementos estén disponibles como documentos de texto (por ejemplo, resúmenes de tramas, reseñas, resúmenes breves). A continuación, nos referiremos a los documentos como descripciones textuales de los elementos que se recomendarán. La Figura 1 muestra la arquitectura general del sistema. El proceso de recomendación se realiza en tres pasos, cada uno de los cuales es manejado por un componente separado [7].

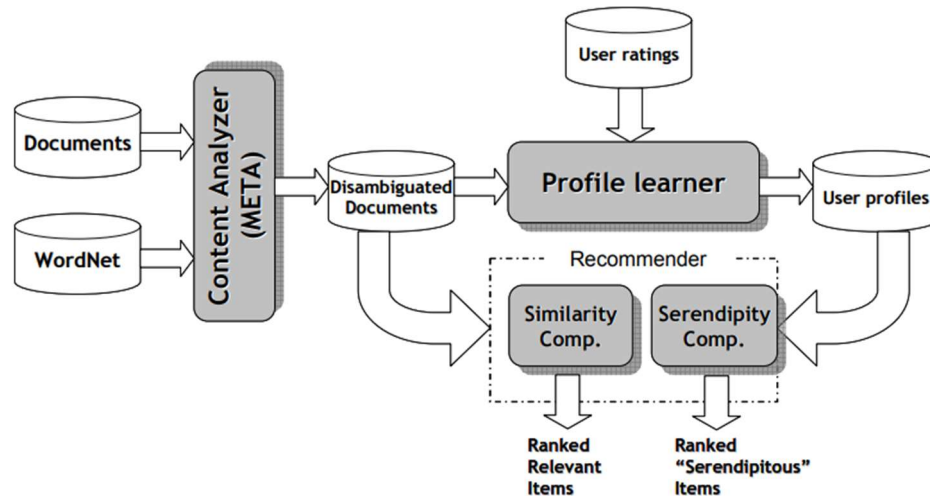


Figura 1. Arquitectura general de *Serendipity* [7].

### 2.6.1. Filtro basado en contenido

Permite introducir la semántica en el proceso de recomendación mediante el análisis de documentos para identificar conceptos relevantes que representen el contenido. Este proceso selecciona, entre todos los posibles significados (sentidos) de cada palabra polisémica, el correcto según el contexto en el que se produce la palabra. De esta forma, los documentos se representan utilizando conceptos en lugar de palabras clave, en un intento de superar los problemas debidos a la ambigüedad del lenguaje natural. El resultado final del paso de preprocesamiento es un repositorio de documentos sin ambigüedad. Esta indexación semántica se basa en gran medida en técnicas de procesamiento del lenguaje natural y se basa en gran medida en el conocimiento lingüístico almacenado en la ontología léxica de *WordNet* [7].

### 2.6.2. Perfil del alumno

Implementa una técnica de aprendizaje supervisado para aprender un modelo probabilístico de los intereses del usuario activo a partir de documentos desambiguados calificados según sus intereses. Este modelo representa el perfil semántico, que incluye aquellos conceptos que resultan más indicativos de las preferencias del usuario. Consideramos el problema del aprendizaje de los perfiles de usuario como una tarea de categorización de texto binario, ya que cada documento tiene que ser clasificado como interesante o no con respecto a las preferencias del usuario. Por lo tanto, el conjunto de categorías está restringido a

POS, que representa la clase positiva (me gusta al usuario) y NEG la negativa (no le gusta al usuario). El modelo probabilístico inducido se utiliza para estimar la probabilidad a posteriori,  $P(X | d)$ , del documento  $d$  perteneciente a la clase [7].

### 2.6.3. Recomendador

Explora el perfil de usuario para sugerir documentos relevantes, comparando los conceptos contenidos en el perfil semántico con los contenidos en los documentos a recomendar. En este componente se ha incluido el módulo dedicado a descubrir ítems potencialmente fortuitos, además del módulo que se encarga del cálculo de similitud entre ítems y perfiles. Para integrar la "pobre similitud" de Toms dentro del recomendador, se ha incluido un conjunto de heurísticas en el módulo para el cálculo de serendipia. El módulo dedicado a comparar ítems con perfiles (Similitud Computación) produce una lista de ítems clasificados según la probabilidad a posteriori [7].

## 2.7. Medida de similitud del usuario basada en la personalidad (2009)

Propone un enfoque novedoso para calcular la similitud del usuario para sistemas de recomendación de filtrado colaborativo que se basa en el modelo de las cinco grandes personalidades. Los resultados experimentales mostraron que el desempeño de las medidas propuestas: "es igual o mejor" (dependiendo de la medida bajo evaluación) son medidas basadas en calificaciones utilizadas en los sistemas de recomendación colaborativa de última generación. Esto hace del enfoque propuesto posea beneficios en términos de complejidad computacional, para calcular las similitudes del usuario haciéndolo prometedor para futuros sistemas colaborativos de recomendación que estará más orientado al afecto [8].

Se centra en los sistemas de filtros colaborativos basados en memoria. Los recomendadores de filtros colaborativos se basan en la presunción de que cuando la similitud entre dos usuarios es alta, a ambos les gustarán artículos similares. La medida de similitud es, por tanto, una parte crucial de cualquier sistema de filtros colaborativos. El estricto enfoque técnico en los sistemas de recomendación que ignora las experiencias afectivas de los usuarios durante el consumo de contenido multimedia es extraño porque la industria del entretenimiento se basa en dar emociones a las personas. De hecho, tratan la satisfacción del usuario al consumir contenido multimedia como un estado afectivo [8].

## 2.8. Datos Vinculados (2010)

Es un sistema de recomendación colaborativo. El término datos vinculados se refiere a un conjunto de mejores prácticas para publicar y conectar datos estructurados en la web. Tomados en conjunto, todos vinculados, los datos constituyen la web de datos. Mientras que el mundo web proporciona los medios para crear una red de documentos legibles por humanos. El sistema de recomendación colaborativo agrega las valoraciones de los usuarios descubriendo similitudes entre los productos. Las altas barreras de entrada proveen buenas recomendaciones caracterizada por tres desafíos proveyendo recomendaciones de:

- Nuevos artículos
- Nuevos usuarios, donde el desafío se centra la inexistencia de datos disponibles sobre el producto o el usuario

Se centra en el uso de filtrado colaborativo para explotar conexiones binarias simples entre usuarios y elementos. Estos datos se pueden encontrar en cualquier fuente de datos; sigue el modelo del principio de sociabilidad centrada en el objeto. Además, mediante el uso de datos tan abundantemente disponibles, se pueden mitigar los problemas de nuevos elementos, nuevos usuarios y escasez al explotar la escala web de datos vinculados [9].

## 2.9. Recomendador social basado en grafos (2012)

Los conceptos se segmentan en la base de datos ontológica en grupos de conceptos que son compartidos por varios perfiles de usuario. Se define un grafo social en el que los nodos representan a los usuarios dentro de la red social y las aristas representan las conexiones explícitas entre los usuarios. Se define un grafo de vecindad para cada grupo de conceptos. Se definen grafos de afinidad social multicapa. Se identifican los elementos de datos sobre los que actúan los usuarios dentro de la red social en un intervalo de tiempo determinado. Se determinan los usuarios dentro de la red social que han actuado sobre los elementos de datos identificados. Se seleccionan una o más capas de los grafos de afinidad social para cada elemento identificado. Los valores de aprobación inicial en los nodos se inyectan para cada artículo identificado. Los valores de aprobación se propagan a través de las capas seleccionadas de los grafos de afinidad social para cada elemento identificado hasta que se cumplen algunos criterios de detención. Se genera una lista ordenada de elementos sobre los que actúan otros usuarios para cada usuario dentro de la red social [10].

## 2.10. Etiquetado efectivo (2013)

El etiquetado afectivo de contenido multimedia ha demostrado ser útil en sistemas de recomendación. Es una metodología para la adquisición implícita de etiquetas afectivas para imágenes. Se basa en una técnica de detección de emociones que toma como entrada las secuencias de video de las expresiones faciales de los usuarios. Extrae características de bajo nivel de *Gabor* de los fotogramas de video y emplea una técnica de aprendizaje automático de *k* vecinos más cercanos para generar etiquetas afectivas en el espacio de valencia-excitación-dominio. Se realizó un estudio comparativo del desempeño de un sistema de recomendación basado en contenido (CBR) para imágenes que utiliza tres tipos de metadatos para modelar a los usuarios y los elementos [11]:

- Metadatos genéricos
- Etiquetas afectivas adquiridas explícitamente
- Implícitamente adquirieron etiquetas afectivas con la metodología propuesta

Los resultados muestran que el CBR funciona mejor cuando se utilizan etiquetas explícitas. Sin embargo, las etiquetas adquiridas implícitamente producen un rendimiento significativamente mejor del CBR que los metadatos genéricos, al tiempo que son una herramienta de retroalimentación discreta [11].



## 2.11. Popularidad del usuario (2020)

Es un sistema de recomendación basado en la popularidad social de una publicación (es decir, la cantidad de visitas o me gusta); en los servicios de redes sociales es importante tanto para los usuarios comunes como para las empresas que desean promocionarse. Se ha implementado un sistema de soporte de etiquetado en línea; para lograr esto utilizando un algoritmo que recomienda los *hashtags* apropiados considerando no solo la popularidad del contenido sino también la popularidad del usuario.

La eficacia de esta tecnología se ha verificado subiendo fotos con *hashtags* recomendados en un servicio de red social real. Los servicios para compartir en línea como Flickr, Instagram y Facebook se han convertido en una parte importante de la vida moderna. En estos servicios de redes sociales, el número de visitas, comentarios y favoritos recibidos después de la carga indican la popularidad del contenido, que puede denominarse “popularidad social” o “puntuaciones de popularidad social”. Aunque la popularidad social es fundamental para muchas personas y corporaciones que desean obtener más atención, hay hechos de que solo una pequeña cantidad de contenido social se hizo popular, mientras que la gran mayoría del contenido solo puede atraer una atención limitada [12].

## 2.12. Comparativa de recomendadores

A continuación, se muestran las ventajas que ofrece el recomendador de productos de co-compra basado en grafos con los recomendadores mencionados previamente. Tabla 1.

Tabla 1. Ventajas ofrecidas por el recomendador de productos de co-compra.

Tipo de recomendador	Recomendador	Ventajas de la actual propuesta
Híbrido	<i>Xerox PARC Tapestry</i> (1992)	<b>Xerox</b> se centra en la recomendación basada en artículos, donde la preocupación principal es valorar las búsquedas de artículos y noticias. Las reseñas se usan para dar recomendaciones a otros usuarios [3].  <b>Nuestro recomendador de productos</b> se centra en lo que el usuario quiere comprar, se basa principalmente en la utilidad y si realmente el producto es lo que el usuario necesita.
Colaborativo	<i>Movie Lens</i> (1996)	<b>Movie Lens</b> utiliza un modelo semejante a la propuesta de productos basados en co-compra; valora las calificaciones de los usuarios de películas y acorde a dicho valor le asigna un valor, para luego poder dar recomendaciones de nuevas películas [4].  Los algoritmos utilizados en <b>nuestra solución de productos de co-compra</b> , se hace uso de algoritmos de centralidad, <i>pageRank</i> para valorar la votación de artículos y ofrecer al usuario productos con mayor precisión.
Híbrido	<i>Fab</i> (1997)	<b>Fab</b> surgió por la necesidad de ofrecer a los usuarios respuestas de búsquedas acorde a la consulta realizada. Ofrece una combinación de filtros tales como es de contenido y colaborativo

		<p>encontrando similitudes entre los usuarios generando un resultado acorde a las búsquedas realizadas previamente [2].</p> <p><b>Nuestro recomendador de co-compra</b> ofrece una respuesta rápida a las consultas realizadas por el usuario.</p>
Basado en grafos	E-commerce (2004)	<p><i>E-commerce</i> brinda 3 métodos de recomendación de recuperación directa, extracción de asociaciones y recuperación de asociaciones, donde se logran predicciones más precisas, pero solo a alto nivel [12].</p> <p><b>Nuestro recomendador de co-compra</b> cubre los 3 métodos de recomendación que ofrece <i>E-commerce</i> agregando un filtro colaborativo, generando consultas más precisas al usuario.</p>
Sistemas de popularidad	Características de popularidad del producto (2006)	<p><b>Características de popularidad del producto</b> como su nombre lo dice se basa en la popularidad en compra de productos, acorde a al género o frecuencias de compra, se le recomienda al usuario el producto más comprado [6].</p> <p><b>Nuestro recomendador de co-compra</b> se basa en las valoraciones, reseñas, las ventas de dicho producto haciendo uso de algoritmos de comunidad generando una respuesta próxima a lo que el usuario necesita.</p>
Basado en contenido	Serendipity (2008)	<p><i>Serendipity</i> se basa principalmente en las descripciones de los documentos textuales para generar una recomendación al usuario. Toma en cuenta el perfil de alumno donde se aplica un aprendizaje supervisado generando luego la recomendación [7].</p> <p><b>Nuestro recomendador de co-compra</b> hace la valoración de un conjunto de datos tales como: el número total de reseñas, tipo de categoría, rango de ventas generando una recomendación más certera.</p>
Colaborativo	Medida de similitud del usuario basada en la personalidad (2009)	<p><b>Medida de similitud del usuario basada en la personalidad</b> mide las emociones y las reacciones de los usuarios [8].</p> <p><b>Nuestro recomendador de co-compra</b> no toma en base la personalidad del usuario tal como cual, si no de aspectos de comportamiento como gustos u orientaciones.</p>
Colaborativo	Datos Vinculados (2010)	<p><b>Datos vinculados</b> se centra en las relaciones existentes en la web de usuarios y elementos y en base a dicha red generar recomendaciones [9].</p> <p><b>Nuestro recomendador de co-compra</b> ofrece un apartado de grupos, donde se muestra la vinculación de datos entre libros, música, productos.</p>

Basado en grafos	Recomendador social basado en grafos (2012)	<p><b>El recomendador social basado en grafos</b> ofrece una recomendación acorde a los seguidores de una persona en particular y posibles conexiones entre personas [10].</p> <p><b>Nuestro recomendador de co-compra</b> no está orientado a las redes sociales tal cual, pero si en la interacción del usuario con la web, donde se realizan todas las consultas generando una recomendación.</p>
Basado en contenido	Etiquetado Efectivo (2013)	<p><b>El recomendador de etiquetado efectivo</b> obtiene las características de bajo nivel y emplea una técnica de aprendizaje automático para generar el etiquetado de manera explícita [11].</p> <p>El etiquetado en <b>nuestro recomendador de co-compra</b> no es realizado de manera explícita tal como lo realiza el etiquetado efectivo. El etiquetado son las categorías ofrecidas al usuario divididas por producto, grupos y clientes</p>
Sistemas de popularidad	Popularidad del usuario (2020)	<p><b>El recomendador de popularidad de usuario</b> se basa en los <i>hashtags</i> posteados en Flickr, Instagram y Facebook se han convertido en una parte importante de la vida moderna [12].</p> <p>En <b>nuestro recomendador</b> cada producto toma categorías de su id, el número total de reseñas, tipo de categoría, rango de ventas generando una recomendación más certera. Si solo se toma de base la popularidad para hacer recomendaciones, se puede recomendar productos que no sean del agrado del usuario.</p>

En el presente capítulo se habló de recomendadores similares al recomendador de productos; se muestra en la Tabla 1 las ventajas de nuestra solución en base a los demás recomendadores. Nuestra solución ofrece la opción de generar recomendaciones por categorías: producto, grupo y cliente, donde el usuario podrá elegir su preferida acorde a los algoritmos de búsqueda utilizados en las consultas. El recomendador con mayor similitud es el de “Características de popularidad del producto” basado en el género y la popularidad; nuestro recomendador ofrece un plus en poseer recomendación de co-compra aparte de las reseñas de los clientes. En el siguiente capítulo se aborda el marco teórico donde hablaremos de los tipos de algoritmos utilizados en el recomendador de co-compra.

---

## 3. MARCO TEÓRICO/CONCEPTUAL

---

En la actualidad el uso de internet se ha vuelto una necesidad para cada una de las personas que habitamos el planeta. Debido a los efectos de la pandemia la mayoría de los trabajadores de empresas transnacionales están de manera remota realizando sus actividades laborales; los maestros y alumnos de las escuelas cambiaron a la modalidad en línea, donde la comunicación se hace a través de video llamadas. Las compras en línea han tenido un crecimiento exponencial en los últimos 2 años incrementando la demanda de servidores y de internet en casa respectivamente. De las compras en línea surge la necesidad de tener recomendadores de artículos en línea a los usuarios evitando salir de casa, mejorando la comodidad y seguridad de los usuarios.

Los sistemas de recomendación tienen aplicación para diversas áreas del mundo actual como es la economía, educación, arte, ocio y artículos de compra en línea, haciendo mención de solo algunos. El último caso es el tema que se abordará en el documento.

Un sistema de recomendación debe ser capaz de ofrecer a sus usuarios opciones válidas a los artículos que necesite. La compra de productos en línea ha ayudado a las personas en la eficiencia de sus elecciones, debido que si no se adapta a sus necesidades puede regresarlo o cambiarlo por uno de su agrado desde la comodidad de su casa [13].

### 3.1. Sistema de recomendación

“Los sistemas de recomendación son herramientas de software o técnicas que ayudan a proveer sugerencias de diferentes artículos a diferentes usuarios” [1]. Los sistemas de recomendación han sido del agrado de la sociedad, debido a que, desde antes de sus primeras apariciones como software, se ha utilizado a manera de sugerencia entre amigos, personas relacionadas a cada uno reflejándolo en los productos y tecnologías consumidas, generándose como parte de nuestra vida diaria.

Existen diversos algoritmos o métodos de recomendación que toman en cuenta diversas características, al momento de buscar artículos relevantes para los usuarios; por ejemplo, unos consideran más importante la relación que pueda existir entre usuarios categorizados como semejantes y hay sistemas de recomendación que te sugieren artículos dependiendo del contenido relacionado a las preferencias de cada usuario. En la Figura 2 se puede observar los diferentes tipos de métodos o algoritmos de recomendación, que pueden ser aplicados a artículos de compra u otros tópicos.

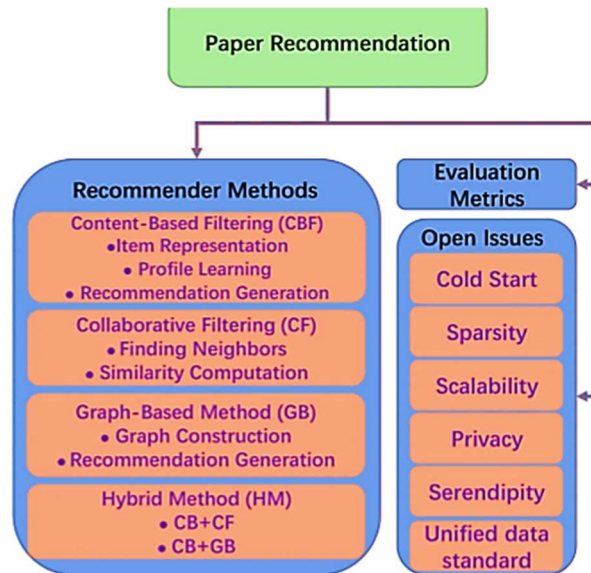


Figura 2. Tipos de sistemas de recomendación [13].

## 3.2. Métodos de recomendación

Para que pueda existir una recomendación, dicho sistema deberá aprender los gustos del usuario o verificar las características de cada artículo atado a un tema en específico, dependerá de las acciones que se realicen, se pueden dividir las técnicas o métodos; los más utilizados pueden categorizarse en cuatro tipos, basado en contenido, filtro colaborativo, híbrido y basado en grafos [14].

### 3.2.1. Filtro basado en contenido

La técnica de filtro basado en contenido se centra en el artículo para generar la predicción y luego su recomendación se basará en el usuario, donde se validan el historial de preferencias, construyendo un perfil y buscando similitudes con las características de diferentes artículos [13].

Existe ventajas de la técnica como los intereses de cada individuo, y no de los demás. Así mismo, si los intereses llegan a cambiar en un futuro, también lo harán sus recomendaciones, donde se ofrece recomendaciones fortuitas para que el usuario pueda obtener información que no esté actualmente en sus intereses, pero pueda estarlo [13].

Se debe tomar en cuenta que el proceso se divide principalmente en 3 etapas: *Item representation* (Tiene el objetivo de estructurar la información no específica, representar las características de los artículos como vectores para posteriormente computar la similitud con el *profile* del usuario, se llegan a utilizar métodos de representación como el TF-IDF en el caso de documentos), *Profile learning* (El objetivo de este paso es construir el perfil del usuario utilizando algoritmos de extracción de tópicos como el LDA o *Latent Dirichlet Allocation* y *Recommendation generation* (Obtiene la lista de posibles recomendaciones relevantes para el usuario, evaluando la similitud de los vectores resultantes de los procesos previos). Figura 3 [13].

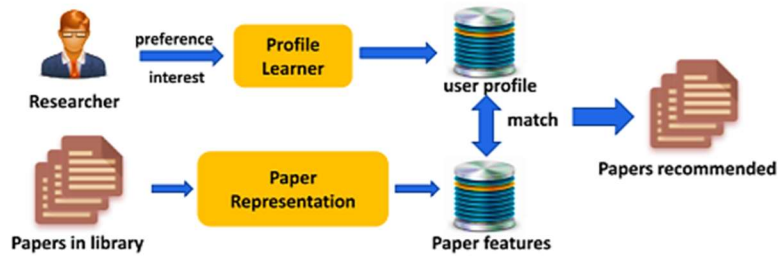


Figura 3. Proceso de un sistema de recomendación basado en contenido [13].

### 3.2.1.1. Método de representación TF-IDF

El término TF-IDF (*Term Frequency – Inverse document frequency*) es una medida estadística utilizada para evaluar la importancia de una palabra en un documento y en una colección de documentos. TF-IDF es el producto de dos factores, la frecuencia de término y la frecuencia inversa de documento [15].

La frecuencia de término es el número de veces que un término  $t$  ocurre en un documento  $D$ , y la frecuencia inversa de documento es una medida que expresa si el término es común o no en la colección de documentos [15].

### 3.2.1.2. Algoritmo LDA

El algoritmo LDA (*Latent Dirichlet Allocation*) es un modelado de tópicos o temas, y es utilizado para clasificar texto de un documento en un tópico particular. El algoritmo construye un tópico por cada modelado de cada documento y las palabras generadas en cada uno de estos son modeladas como distribuciones de *Dirichlet*. Se utiliza algoritmo para la separación e interpretación de temas de un documento [16].

### 3.2.1.3. Filtro colaborativo

EL filtro colaborativo se basa en las acciones, *ratings* o calificaciones hechas en los artículos de parte de otros usuarios que tengan un perfil similar (usuarios vecinos). Al igual que la técnica basada en contenido, este método necesita conocer los intereses del usuario; en otras palabras, el sistema de recomendación de filtro colaborativo es el proceso de recomendar artículos usando la opinión de otros usuarios vecinos [17].

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

### 3.2.1.4. Basado en usuario

Los sistemas de recomendación usan el perfil de otros usuarios similares para hacer la predicción. Para este caso en particular, los usuarios se dividen en categorías o grupos, donde los usuarios en el mismo grupo comparten intereses similares.

### 3.2.1.5. Basado en ítems

Se enfoca principalmente en la relación existente entre los ítems, en vez de las relaciones entre los usuarios. Si los usuarios otorgan calificaciones positivas sobre algunos ítems, el sistema podrá recolectar los ítems candidatos a ser elegidos basándose en el *rating* histórico de calificaciones del usuario.

Unas de las desventajas de este recomendador se tienen el arranque en frío, es decir, para los ítems nuevos que no tienen una calificación asignada, no pueden ser recomendados.

En general, como se muestra en la Figura 4, la técnica de recomendación de filtro colaborativo se basa en la construcción de una matriz de *ratings*, donde se debe considerar que no se tienen matrices densas, es decir, no todos los usuarios dan su calificación a todos los ítems, por lo tanto es más óptimo el basado en ítems al manejar matrices más pequeñas [13].

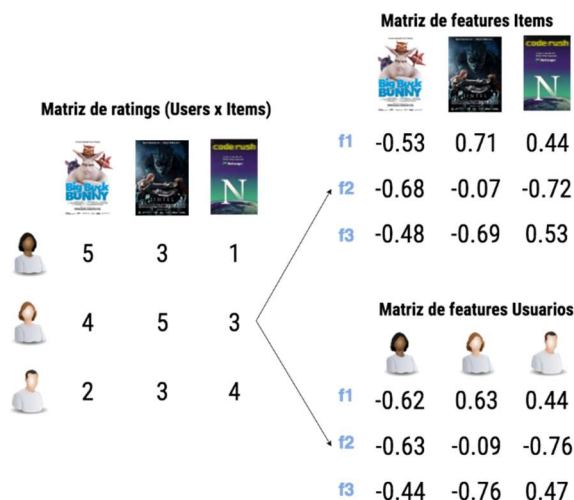


Figura 4. Matrices de índices para un sistema de recomendación de filtro colaborativo [13].

### 3.2.2. Basado en grafos

La técnica de recomendación basada en grafos se centra en la construcción de un grafo para generar las predicciones y recomendaciones donde cada artículo/usuario puede verse como entidad o nodo y las posibles relaciones que existen entre ellos se pueden representar como aristas [18]. La Figura 5 se muestra a un grafo utilizado como sistema de recomendación de artículos, donde los usuarios y los productos representan los nodos.



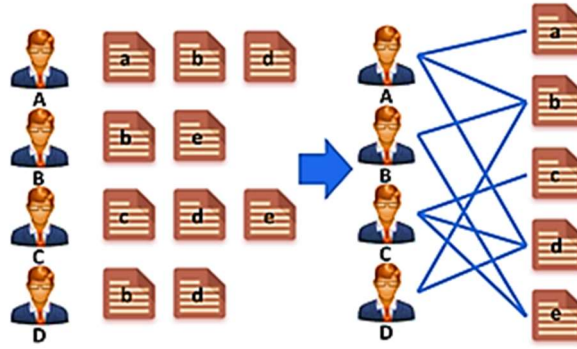


Figura 5. Grafo usado para un sistema de recomendación básico [13].

Para generar la recomendación se suele utilizar algoritmos de recorrido o búsqueda como el de caminata aleatoria con reinicio, el cual evalúa la probabilidad de estando en un vértice inicial  $i$ , llegar a un vértice final  $j$  pasando por nodos adyacentes y se debe considerar el caso de que se pueda llegar al nodo inicial  $i$ , por eso el nombre de reinicio [19].

### 3.2.3. Híbrido

De los métodos explicados previamente, se tienen pros/contras debido que no tienen la capacidad de cubrir un número extenso de opciones; con el método híbrido se busca unir el comportamiento de 2 o más métodos para complementar sus debilidades. La ventaja principal del método híbrido es que se puede hacer la combinación de diferentes técnicas de recomendación y obtener información de diferentes fuentes [13].

Como se puede observar en la Figura 6, un tipo de sistema híbrido utilizado es el paralelo de los sistemas basado en contenido y basado en filtro colaborativo, de esta manera se toma en cuenta los intereses del usuario a través de un perfil creado y el *rating* hecho por el usuario a diferentes artículos. Al final, se terminan recomendando la coincidencia de las 2 listas diferentes de artículos posibles que resulten relevantes para el usuario [20].

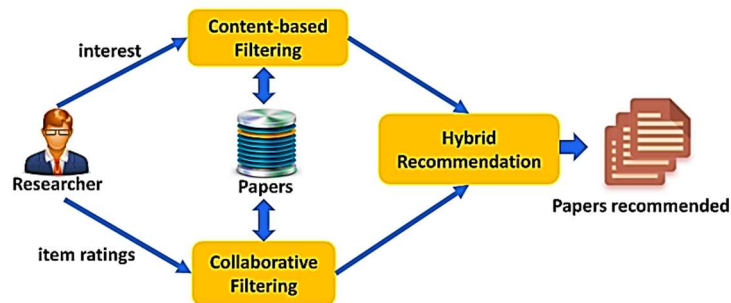


Figura 6. Método híbrido (CF + CFB) paralelo [13].

Se tiene diversas técnicas de recomendación, las más utilizadas son la basada en contenido y la de filtro colaborativo. En la Tabla 2 se puede observar las principales ventajas y desventajas de la utilización de las mismas.

Tabla 2. Ventajas y desventajas de los diferentes sistemas de recomendación [15].

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

### 3.3. Métodos de evaluación

Los sistemas de recomendación tienen como finalidad ser modelos de predicción, donde poseen algoritmos que buscan minimizar el error de dicha función. Por lo tanto, es importante medir y determinar si la técnica de recomendación aplicada para dicho sistema de recomendación es efectiva. Los métodos de evaluación más utilizados son Precisión, Recuperación y Medida F (F1) [13] [21].

#### 3.3.1. Precisión

El método de precisión es usado para medir la exactitud de recomendar artículos relevantes para los usuarios en un sistema de recomendación; es decir, dentro de los artículos recomendados al usuario, cuales fueron relevantes para él. Su fórmula viene representada en la Figura 7 [13].

#### 3.3.2. Recuperación

El método de recuperación cuantifica la fracción de artículos relevantes dentro de un conjunto de artículos recomendados al usuario final. Es decir, de los artículos que resultaron relevantes para el usuario, cuantos fueron seleccionados. Su fórmula viene representada en la Figura 7 (*recall*) [13].



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

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

Figura 7. Precisión vs Recuperación [13].

### 3.3.3. Medida F

El método de medida F representa el promedio armónico entre precisión y recuperación. Puede existir el caso en el que se pueden contradecirse los resultados de dichos factores; caso dado cuando el número de artículos recomendados se incrementa, entonces la recuperación tiende a crecer y la precisión tiende a disminuir [13].

## 3.4. Medidores de eficiencia

Todos los sistemas de recomendación tienen problemas que se han tratado de combatir a lo largo del tiempo y como han ido evolucionando. Claro ejemplo del modelo híbrido que busca combinar 2 o más técnicas de recomendación para anular los siguientes problemas muy comunes [20].

### 3.4.1. Arranque en frío

El arranque en frío es el estado inicial de las entidades de un sistema de recomendación, es decir cuando un usuario o artículo son recién creados, es complicado modelar un perfil (método basado en contenido) o relacionarlo con usuarios que parecen ser similares (método de filtro colaborativo), debido a la escasez de información [20].

### 3.4.2. Escasez

EL número de usuarios es mayor al número de artículos en su mayoría, logrando predicciones efectivas; existe el caso en donde el número de usuarios es menor al número de artículos o incluso que el número de artículos sea muy poco calificado por los usuarios; es decir, que no tengan muchas relaciones.

Dicho problema provoca que se tenga poca información para poder realizar una recomendación con calidad, en especial para la técnica de filtro colaborativo que utiliza matrices de calificación de elementos [13].

### 3.4.3. Escalabilidad

Esta definición se refiere a la capacidad de un sistema de recomendación de trabajar efectivamente en diferentes ambientes donde existe una gran cantidad de usuarios y de artículos [13].

### 3.4.4. Hallazgo fortuito.

El comportamiento común de un sistema de recomendación, es predecir artículos candidatos a los intereses de los usuarios, pero de vez en cuando es útil tener recomendaciones fortuitas para incrementar el área de conocimiento. El número de estas recomendaciones fortuitas no puede ser elevado, ya que el sistema de recomendación pierde credibilidad [13].

## 3.5. Bases de datos NOSQL

Las bases de datos NoSQL (*NOT Only SQL*) aparecieron con WEB2.0 cuando empresas como Facebook, Twitter, y Youtube tenían gran demanda en sus páginas de millones de usuarios teniendo que dar respuestas a millones de consultas, provocando que las bases de datos relacionales empezaran a fallar [22].

Las bases de datos NoSQL utilizan una variedad de modelos para acceder y administrar grandes cantidades de información. Su principal objetivo es la alta escalabilidad de los sistemas, no llegan a suplir una base de datos relacionales, pero si mejoran su comportamiento cuando se maneja gran volumen de información. Existen diversos tipos como las basadas en documentos, llave-valor y columnas; sin embargo, para el desarrollo de este proyecto, nos enfocaremos en las basadas en grafos [23].

Grafos: el propósito de una base de datos de grafos es facilitar la creación y la ejecución de aplicaciones que funcionan con conjuntos de datos altamente conectados. Los casos de uso típicos para una base de datos de grafos incluyen redes sociales, sistemas de recomendaciones, detección de fraude y grafos de conocimiento [23].

### 3.5.1. NEO4J

NEO4J es una base de datos basada en grafos. La información se representa como nodos de un grafo, y sus relaciones con las aristas del mismo modelo. Se suelen utilizar grafos multicapa o grafos heterogéneos donde cada capa contiene un tipo de entidad y existen relaciones entre ellas [20].

NEO4J es un *software open source* de base de datos orientado a grafos implementado en Java y desarrollado por *Neo Technology*. Entre las principales características de esta base de datos se encuentran [20]:

- Intuitivo. Usa un modelo de datos gráfico para la representación del modelo.
- Confiable. Soporta transacciones ACID. Buscando en todo momento la atomicidad, consistencia y persistencia del modelo.
- Altamente escalable. Soporte para miles de nodos, relaciones y propiedades.
- Expresivo. Potente lenguaje de consulta (CYPHER).
- Simple. Accesible a través de una API orientada a objetos (JAVA) [24].

### 3.5.1.1. CYPHER

CYPHER es un lenguaje declarativo basado en el lenguaje de bases de datos relacionales SQL, que permite manipular la información en NEO4J. CYPHER es el lenguaje propio de NEO4J, su sintaxis utiliza un estilo *ascii-art*, lo que lo hace muy intuitivo. Los nodos se representan con círculos y las relaciones con flechas; por ejemplo, su representación consiste en poner nodos entre paréntesis y relaciones como flechas encerradas entre corchetes. Las propiedades de las entidades se indican con una estructura similar a los diccionarios, Figura 8 [24].

```
1  (nodo) - [:RELACION] -> (nodo)
2  (nodo {nombre:'Oscar', apellido:'Garcia'})
```

Figura 8. Entidades en *Cypher*.

### 3.5.1.2 NEOVYS

*Neovys* es una librería de *JavaScript* para la visualización de nodos en una aplicación de UI. Existen diferentes motivaciones y herramientas para crear visualizaciones de grafos. Esto incluye herramientas para explorar el grafo, el tipo de visualizaciones interactivas que puede ver en el navegador NEO4J, o visualizaciones para mostrar los resultados de algún análisis. Estos pueden ser interactivos (algo que se incrustará en una aplicación web o incluso en una aplicación independiente) o estáticos, destinados a transmitir un significado específico que podría usarse en forma impresa o en una publicación de blog [25].

En la Figura 9 podemos ver personajes que están conectados y con la ayuda del diseño podemos comenzar a ver grupos en el grafo. Sin embargo, es necesario visualizar las centralidades (*PageRank*) y los resultados de detección de comunidades [25].

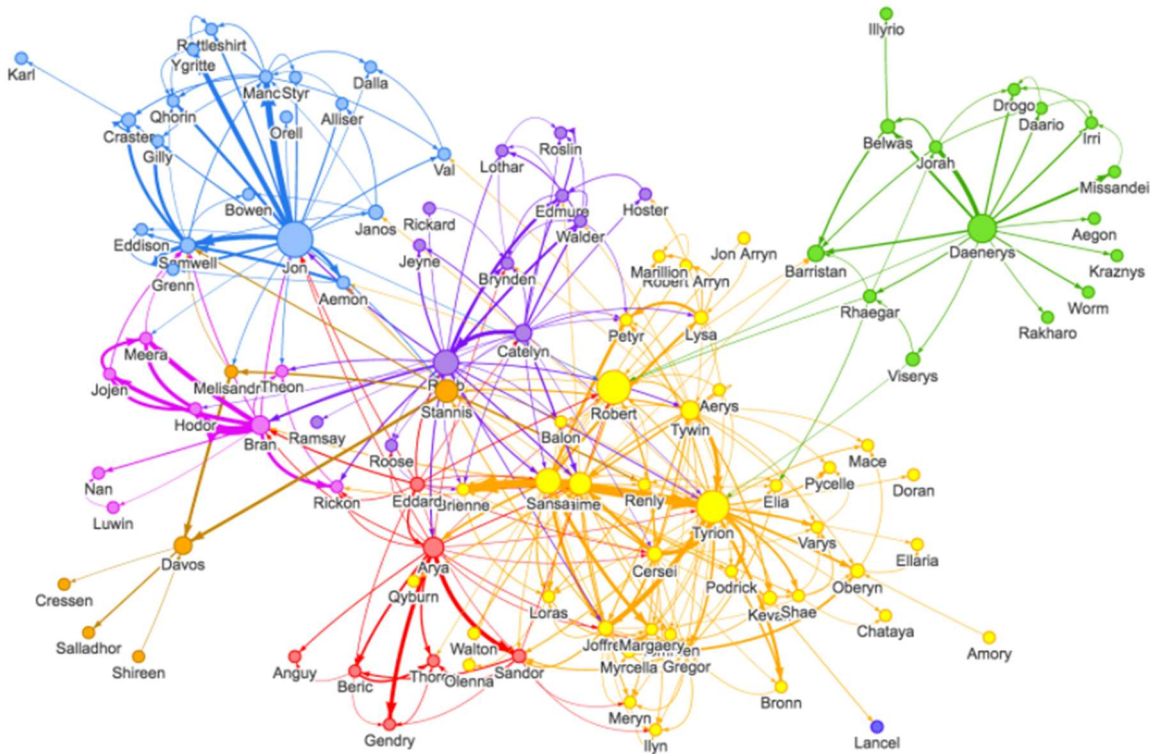


Figura 9. Visualización de un grafo usando Neovis [25].

Específicamente nos gustaría analizar con Neovis.js [25]:

- El tamaño del nodo debe ser proporcional a la puntuación de *PageRank* del personaje. Esto nos permitirá identificar rápidamente nodos importantes en la red [25].
- El color del nodo lo determina la propiedad comunitaria. Esto nos permitirá visualizar clústeres.
- El grosor de la relación debe ser proporcional a la propiedad de peso en la relación [25].
- *Neovis.js*, al combinar el controlador *JavaScript* para NEO4J y la biblioteca de visualización *vis.js*, nos permitirá construir esta visualización [25].

### 3.6. Algoritmos de centralidad

Los algoritmos de centralidad se utilizan para comprender los roles de nodos particulares en un grafo y su impacto en cierta red. Son útiles porque identifican los nodos más importantes y nos ayudan a comprender la dinámica del grupo, como la credibilidad, la accesibilidad, la velocidad a la que se propagan las cosas y las relaciones entre los grupos. Aunque muchos de estos algoritmos se inventaron para el análisis de redes sociales, desde entonces han encontrado usos en una variedad de industrias y campos [26].

#### 3.6.1.1. Page Rank

*PageRank* es el más conocido de los algoritmos de centralidad. Mide la influencia transitiva (o direccional) de los nodos. *PageRank* considera la influencia de los nodos vecinos de un nodo. Por ejemplo,

tener algunos amigos importantes puede hacerte más influyente que tener muchos amigos menos poderosos. *PageRank* se calcula distribuyendo iterativamente el rango de un nodo entre sus vecinos o atravesando aleatoriamente el grafo y contando la frecuencia con la que se golpea cada nodo durante estos recorridos [26].

*PageRank* lleva el nombre del cofundador de Google, Larry Page, quien lo creó para clasificar los sitios web en los resultados de búsqueda de Google. La suposición básica es que una página con más enlaces entrantes y más influyentes es más probable una fuente creíble. *PageRank* mide el número y la calidad de las relaciones entrantes a un nodo para determinar una estimación de la importancia de ese nodo. Se presume que los nodos con más influencia sobre una red tienen más relaciones entrantes de otros nodos influyentes [26].

### 3.6.1.1.1. Influencia

La intuición detrás de la influencia es que las relaciones con los nodos más importantes contribuyen más a la influencia del nodo en cuestión que las conexiones equivalentes a los nodos menos importantes. La medición de la influencia generalmente implica puntuar nodos, a menudo con relaciones ponderadas, y luego actualizar las puntuaciones en muchas iteraciones. A veces, se puntúan todos los nodos y en otras se utiliza una selección aleatoria como distribución representativa [26].

### 3.6.1.1.2 Fórmula de PageRank

*PageRank* se define en el documento original de *Google* acorde a la Figura 10.

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

Figura 10. Visualización de un grafo usando *Neovis* [26].

De la Figura 10 se tiene las siguientes afirmaciones:

- Asumimos que una página  $u$  tiene citas de las páginas  $T1$  a  $Tn$ .
- $d$  es un factor de amortiguación que se establece entre 0 y 1. Por lo general, se establece en 0,85. Puede pensar en esto como la probabilidad de que un usuario continúe haciendo clic. Esto ayuda a minimizar el descenso de rango, que se explica en la siguiente sección.
- $1-d$  es la probabilidad de que se llegue a un nodo directamente sin seguir ninguna relación.
- $C(Tn)$  se define como el grado de salida de un nodo  $T$  [26].

En la Figura 11 muestra un pequeño ejemplo de cómo *PageRank* continuará actualizando el rango de un nodo hasta que converja o cumpla con el número establecido de iteraciones.

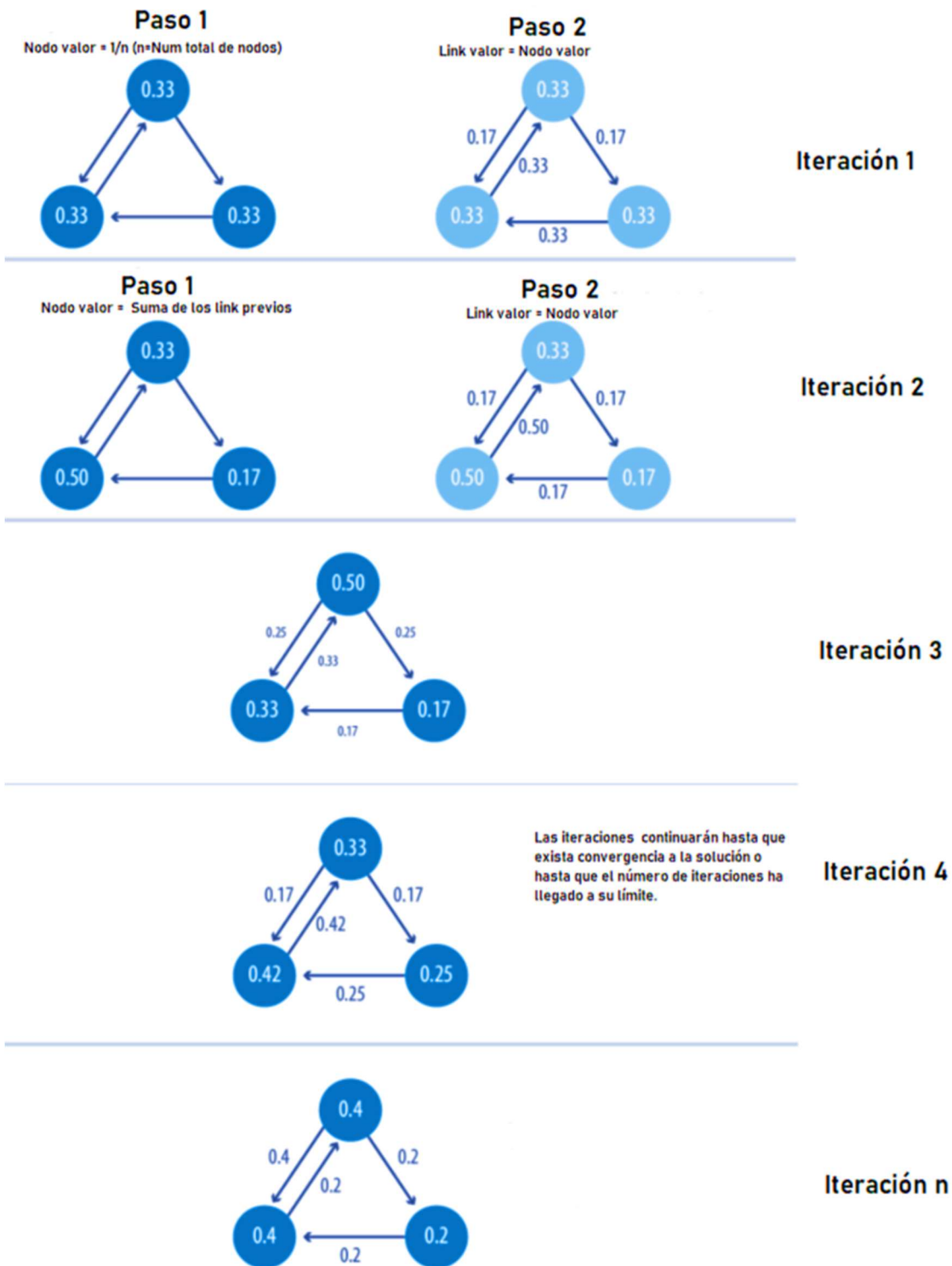


Figura 11. Iteraciones de PageRank [26].



### 3.6.1.2 Centralidad de intermediación

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

Intermediación La centralidad se consideró una de las “tres concepciones intuitivas distintas de centralidad” cuando fue presentada por Linton C. Freeman en su artículo de 1971, “Un conjunto de medidas de centralidad basadas en la intermediación” [26].

Un puente en una red puede ser un nodo o una relación. En un grafo muy simple, puede encontrarlos buscando el nodo o la relación que, si se elimina, haría que una sección del grafo se desconectara. Sin embargo, como eso no es práctico en un grafo típico, usamos un algoritmo de centralidad de intermediación. También podemos medir la intermediación de un clúster tratando al grupo como un nodo [26].

Un nodo se considera fundamental para otros dos nodos si se encuentra en cada ruta más corta entre esos nodos, como se muestra en la Figura 12.



Figura 12. Los nodos fundamentales se encuentran en cada camino más corto entre dos nodos. La creación de rutas más cortas puede reducir la cantidad de nodos fundamentales para usos como la mitigación de riesgos [26].

#### 3.6.1.2.1 Calculando centralidad de intermediación

La centralidad de intermediación de un nodo se calcula sumando el número de nodos entre el número de caminos más cortos entre dos nodos.

La Figura 13 y Tabla 3 se ilustran los pasos para resolver la centralidad de intermediación.

Tabla 3. Cálculo del Nodo D [26].

Pares de Nodos con el camino más corto a través de D	Total de caminos más cortos a través de los nodos	% del total a través de D (1/Total)
A, E	1	1
B, E	1	1
C, E	1	1
B, C	2 ( a través de A y D)	0.5
<b>Calificación de Centralidad</b>		<b>3.5</b>

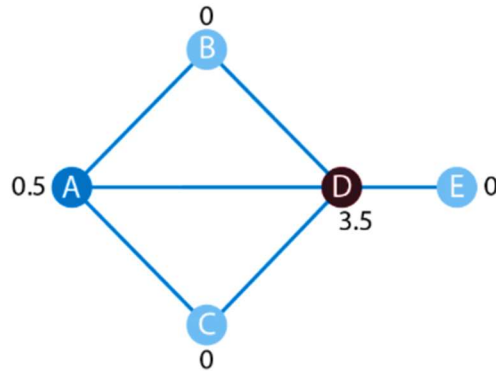


Figura 13. Conceptos básicos para calcular la centralidad de intermediación [26].

Este es el procedimiento:

- Para cada nodo, se buscan los caminos más cortos que lo atraviesen.
- B, C, E no tienen rutas más cortas y se les asigna un valor de 0.
- Para cada ruta más corta en el paso 1, se calcula su porcentaje del total de rutas más cortas posibles para ese par.
- Se suman luego todos los valores del paso 2 para encontrar la puntuación de centralidad de intermediación de un nodo. La tabla 3 ilustra los pasos 2 y 3 para el nodo D.
- Se repite el proceso para cada nodo [26].

En la Figura 14 se muestra un ejemplo de centralidad de intermediación; Alice es el agente principal en esta red, pero Mark y Doug no se quedan atrás. En el subgrafo más pequeño, todos los caminos más cortos pasan por David, por lo que es importante para el flujo de información entre esos nodos [26].

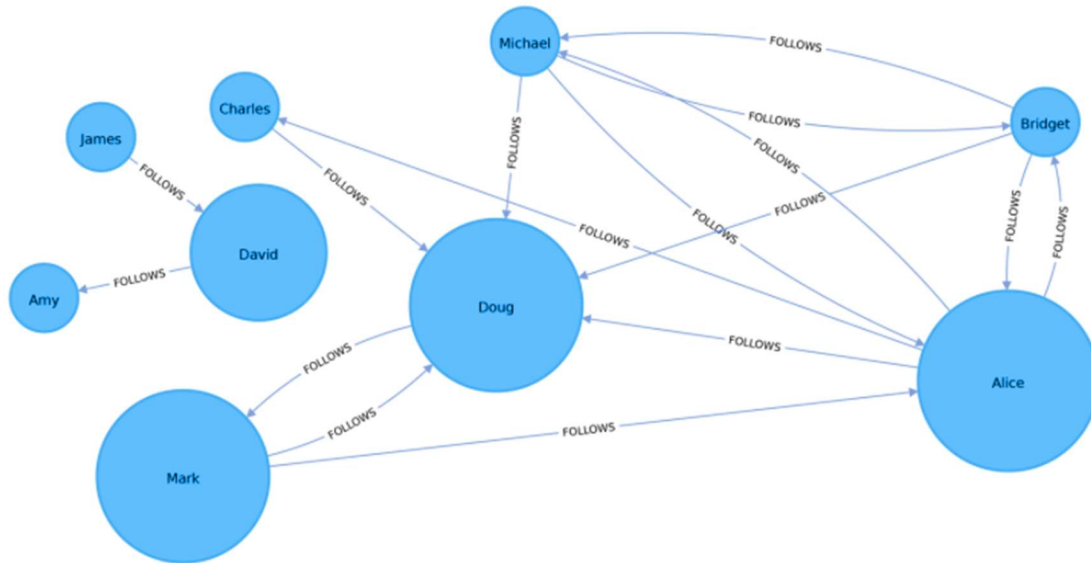


Figura 14. Visualización de la centralidad de intermediación [26].

### 3.7. Algoritmos detección de comunidades

La formación de comunidades es común en todo tipo de redes, e identificarlas es fundamental para evaluar el comportamiento grupal y los fenómenos emergentes. El principio general para encontrar comunidades es que sus miembros tendrán más relaciones dentro del grupo que con nodos fuera de su grupo. La identificación de estos conjuntos relacionados revela agrupaciones de nodos, grupos aislados y estructura de red. Esta información ayuda a inferir comportamientos o preferencias similares de grupos de pares, estimar la resiliencia, encontrar relaciones anidadas y preparar datos para otros análisis. Los algoritmos de detección de comunidades también se utilizan comúnmente para producir visualización de redes para inspección general [26].

#### 3.7.1.1. Componentes Fuertemente Conectados

El algoritmo Componentes Fuertemente Conectados o mejor conocido en inglés *Strongly Connected Components* (SCC), es uno de los algoritmos más nuevos de grafos. SCC encuentra conjuntos de nodos conectados en un grafo dirigido donde cada nodo es accesible en ambas direcciones desde cualquier otro nodo en el mismo conjunto. Sus operaciones en tiempo de ejecución escalan bien, proporcionalmente al número de nodos. En la Figura 15, puede ver que los nodos de un grupo SCC no necesitan ser vecinos inmediatos, pero debe haber rutas direccionales entre todos los nodos del conjunto [26].

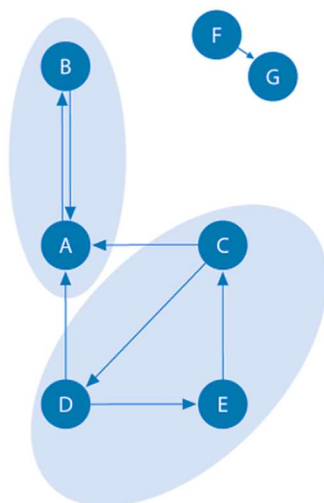


Figura 15. Componentes Fuertemente Conectados [26].

Utilizar Componentes Fuertemente Conectados como un paso inicial en el análisis de grafos para ver cómo está estructurado o para identificar grupos estrechos que pueden justificar una investigación independiente. Un componente que está fuertemente conectado se puede utilizar para perfilar comportamientos o inclinaciones similares en un grupo para aplicaciones como motores de recomendación [26].

Muchos algoritmos de detección de comunidades, como SCC, se utilizan para encontrar y colapsar clústeres en nodos individuales para un mayor análisis entre clústeres. También se puede usar SCC para visualizar ciclos para análisis, como encontrar procesos que pueden bloquearse porque cada subprocesso está esperando que otro miembro tome acción [26].

### 3.7.1.2. Label propagation

El algoritmo de Propagación de Etiquetas o mejor conocido en inglés como *Label Propagation* (LPA) es un algoritmo rápido para encontrar comunidades en un grafo. En LPA, los nodos seleccionan su grupo en función de sus vecinos directos. Este proceso se adapta bien a las redes donde las agrupaciones son menos claras y las ponderaciones se pueden usar para ayudar a un nodo a determinar en qué comunidad ubicarse. También se presta bien al aprendizaje semisupervisado porque puede sembrar el proceso con etiquetas de nodo indicativas asignadas previamente [26].

La intuición detrás de este algoritmo es que una sola etiqueta puede convertirse rápidamente en dominante en un grupo de nodos densamente conectados, pero tendrá problemas para cruzar una región escasamente conectada. Las etiquetas quedan atrapadas dentro de un grupo de nodos densamente conectados y los nodos que terminan con la misma etiqueta cuando finaliza el algoritmo se consideran parte de la misma comunidad. El algoritmo resuelve superposiciones, donde los nodos son potencialmente parte de múltiples clústeres, asignando membresía al vecindario de etiquetas con la relación combinada y el peso de nodo más altos. LPA es un algoritmo relativamente nuevo propuesto en 2007 por U. N. Raghavan, R. Albert y S. Kumara, en un artículo titulado “Algoritmo de tiempo casi lineal para detectar estructuras comunitarias en redes a gran escala” [26].

La Figura 16 y la Figura 17 muestran dos variaciones de la Propagación de Etiquetas, un método de empuje simple y el método de tracción más típico que se basa en pesos de relación. El método de extracción se presta bien a la paralización [26].

### Propagación de etiquetas-Empuje

Envía etiquetas a los vecinos para encontrar clústeres



Ejemplo de 2 nodos de una semilla dada de etiquetas



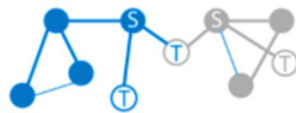
Ellos buscan vecinos inmediatos como objetivos para propagar etiquetas



Donde no existe conflicto se propaga la etiqueta



Los nodos recientemente etiquetados ahora se activan como nuevas semillas



Los conflictos se resuelven en función de una medida establecida, como los pesos de las relaciones.



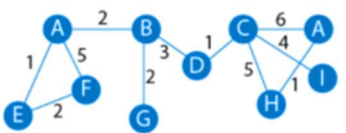
El proceso continúa hasta que se actualizan todos los nodos. Se identifican 2 grupos

LPA se puede ejecutar con etiquetas semilla más nodos sin etiquetar o cada nodo comenzando con una etiqueta única. Cuantas más etiquetas únicas, más resolución de conflictos se utiliza.

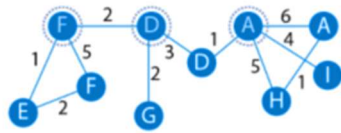
Figura 16. Propagación de Etiquetas método de empuje simple [26].

### Propagación de etiquetas-Tracción

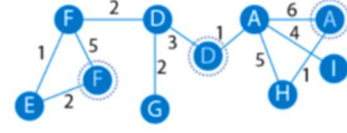
Extrae etiquetas de vecinos en función de los pesos de las relaciones para encontrar clústeres



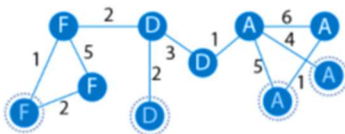
Ejemplo con 2 nodos con la misma etiqueta, "A". Todos los demás son únicos. El valor predeterminado de los pesos de nodo es 1 y en este ejemplo, se ignoran.



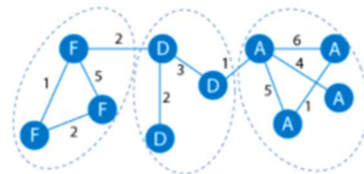
Los nodos se barajan para procesar el orden y cada nodo considera las etiquetas de sus vecinos directos (3 están resaltadas). Los nodos adquieren la etiqueta que coincide con los pesos de relación totales más altos.



Hay que tener en cuenta que estos 3 nodos no cambian las etiquetas porque sus relaciones de mayor peso en este paso tienen la misma etiqueta.



Esto continúa hasta que todos los nodos hayan actualizado sus etiquetas.



Se identifican 3 grupos. Las etiquetas en sí mismas no tienen significado.

Figura 17. Propagación de Etiquetas método de tracción [26].

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

- Cada nodo se inicializa con una etiqueta única (un identificador) y, opcionalmente, se pueden utilizar etiquetas de "semilla" preliminares.
- Las etiquetas se propagan a través de la red.
- En cada iteración de propagación, cada nodo actualiza su etiqueta para que coincida con la que tiene el peso máximo, que se calcula en función de los pesos de los nodos vecinos y sus relaciones. Los lazos se rompen de manera uniforme y aleatoria.
- LPA alcanza la convergencia cuando cada nodo tiene la etiqueta mayoritaria de sus vecinos [26].

A medida que las etiquetas se propagan, los grupos de nodos densamente conectados llegan rápidamente a un consenso sobre una etiqueta única. Al final de la propagación, solo quedarán unas pocas etiquetas, y los nodos que tienen la misma etiqueta pertenecen a la misma comunidad [26].

### 3.7.1.2.1 Aprendizaje semi-supervisado y etiquetas de semillas

A diferencia de otros algoritmos, Propagación de Etiquetas puede devolver diferentes estructuras de comunidad cuando se ejecuta varias veces en el mismo grafo. El orden en el que LPA evalúa los nodos puede influir en las comunidades finales que devuelve [26].

El rango de soluciones se reduce cuando algunos nodos reciben etiquetas preliminares (es decir, etiquetas de semillas), mientras que otros no están etiquetados. Es más probable que los nodos sin etiqueta adopten las etiquetas preliminares [26].

Este uso de la Propagación de Etiquetas puede considerarse un método de aprendizaje semi-supervisado para encontrar comunidades. El aprendizaje semi-supervisado es una clase de tareas y técnicas de aprendizaje automático que operan en una pequeña cantidad de datos etiquetados, junto con una mayor cantidad de datos sin etiquetar. También podemos ejecutar el algoritmo repetidamente en grafos a medida que evolucionan [26].

Finalmente, LPA a veces no converge en una única solución. En esta situación, los resultados de nuestra comunidad cambiarán continuamente entre algunas comunidades notablemente similares y el algoritmo nunca se completará. Las etiquetas de semillas ayudan a guiarlo hacia una solución. *Spark* y *NEO4J* usan un número máximo establecido de iteraciones para evitar una ejecución interminable [26].

### 3.7.1.3. Modularidad de Louvian

El algoritmo de Modularidad de *Louvain* encuentra clústeres comparando la densidad de la comunidad a medida que asigna nodos a diferentes grupos. Se puede pensar en esto como un análisis de "qué pasaría si" para probar varias agrupaciones con el objetivo de alcanzar un óptimo global [29].

Propuesto en 2008, el algoritmo *Louvain* es uno de los algoritmos basados en modularidad más rápidos. Además de detectar comunidades, también revela una jerarquía de comunidades a diferentes escalas. Esto es útil para comprender la estructura de una red en diferentes niveles de granularidad [26].

*Louvain* cuantifica qué tan bien se asigna un nodo a un grupo al observar la densidad de conexiones dentro de un grupo en comparación con una muestra promedio o aleatoria. Esta medida de asignación comunitaria se llama modularidad. es el más conocido de los algoritmos de centralidad. Mide la influencia transitiva (o direccional) de los nodos. *PageRank* considera la influencia de los nodos vecinos de un nodo. Por ejemplo, tener algunos amigos muy poderosos puede hacerte más influyente que tener muchos amigos menos poderosos. *PageRank* se calcula distribuyendo iterativamente el rango de un nodo entre sus vecinos o atravesando aleatoriamente el grafo y contando la frecuencia con la que se golpea cada nodo durante estos recorridos [26].

### 3.1.7.3.2 Agrupación basada en la calidad mediante modularidad

Modularidad, es una técnica para descubrir comunidades al dividir un grafo en módulos (o grupos) más generales y luego medir la fuerza de las agrupaciones. A diferencia de simplemente observar la concentración de conexiones dentro de un grupo, este método compara las densidades de relación en grupos determinados con las densidades entre grupos. La medida de la calidad de esas agrupaciones se llama modularidad [26].

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

- Fusionan comunidades más pequeñas en comunidades más grandes.
- Puede ocurrir una meseta donde varias opciones de partición están presentes con modularidad similar, formando máximos locales e impidiendo el progreso. [26]

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

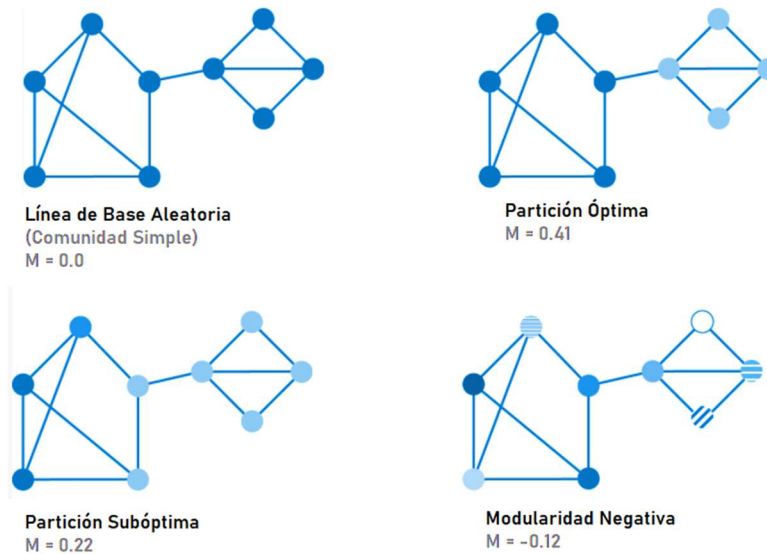


Figura 18. Cuatro puntuaciones de modularidad basadas en diferentes opciones de partición.[26]

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

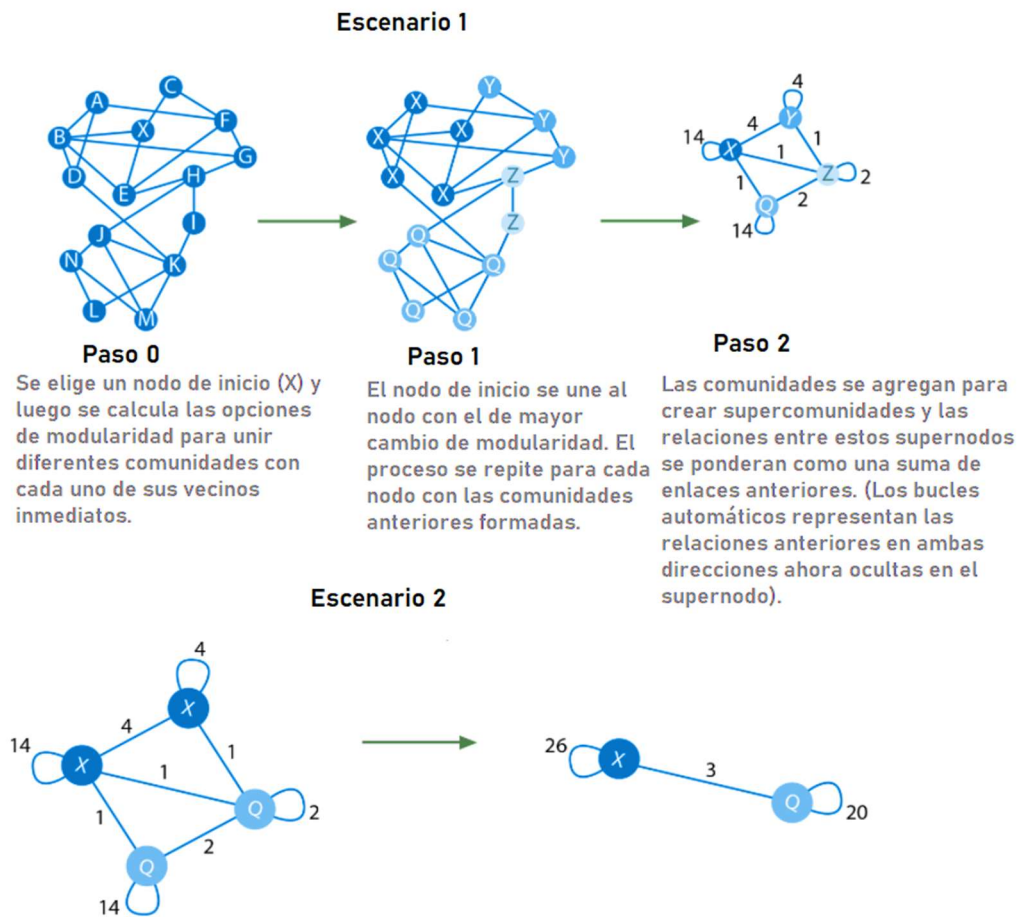


Figura 19. Proceso de algoritmo de *Louvian* [26].



## 3.8. Minería de texto

La minería de textos es el conjunto de procesos necesarios para transformar documentos o recursos de texto no estructurados en información relevante y estructurada. La información puede utilizarse para descubrir automáticamente patrones ocultos y predecir resultados futuros utilizando una combinación de técnicas estadísticas, lingüísticas y de reconocimiento de patrones. Un proceso típico de minería de textos consta de los siguientes pasos [27]:

- Identificar el texto y someterlo a pretratamiento: limpieza del texto para eliminar información innecesaria, división en *tokens* individuales (componentes más pequeños) e identificación de las categorías gramaticales según la gramática del idioma utilizado [27] .
- Extraer información pertinente y transformarla en datos estructurados. La información se recupera buscando en todo el texto convertido en *tokens* y almacenando los resultados de una manera más estructurada y organizada que permita el análisis del mismo [27] .
- Seleccionar características importantes para crear modelos y categorías conceptuales. El número de conceptos presentes en los datos no estructurados suele ser muy grande. La clave es identificar las características más relevantes y utilizarlas para construir modelos significativos basados en categorías y relaciones de datos [27] .

### 3.8.1 Tokenización

La tokenización es una parte interesante del análisis de texto. Un *token* en términos de lenguaje natural es "una instancia de una secuencia de caracteres en un documento en particular que se agrupan como una unidad semántica útil para su procesamiento". Como las raíces y las ramas de un árbol, todo el lenguaje humano es un lío de excrecencias naturales: divididas, en descomposición, vibrantes y florecientes. La tokenización es parte de la metodología que usamos cuando enseñamos a las máquinas sobre palabras [28].

En el caso de nuestro proyecto, es necesario este procedimiento para detectar la información relevante de nuestra fuente principal de recursos, y de esta manera utilizando expresiones regulares, recopilar fragmentos de la información que será guardada en la base de datos no relacional [28].

### 3.8.2 Normalización.

La normalización es un proceso que convierte una lista de palabras en una secuencia más uniforme. Dicho proceso es útil para preparar texto para su posterior uso. Al transformar las palabras a un formato estándar, otras operaciones pueden trabajar con los datos y no tendrán que lidiar con problemas que puedan comprometer el proceso. Por ejemplo, convertir todas las palabras a minúsculas simplificará el proceso de búsqueda [29].

Este proceso está relacionado con nuestro proyecto porque es necesaria la eliminación de caracteres especiales y palabras sin contexto para reducir así la dimensionalidad del texto crudo inicial y optimizar el rendimiento temporal de nuestros algoritmos de búsqueda y análisis [29].

## 3.9. Desarrollo WEB

El desarrollo web es un término que define la creación de sitios web para Internet o una intranet. Para lograr esto, la tecnología de software se usa en el lado del servidor y del lado del cliente, lo que implica una combinación de procesos de base de datos utilizando un navegador para realizar ciertas tareas o mostrar información. Este tipo de desarrollo implica la creación de un sitio web; por lo general, se refiere al lado de codificación y programación de la producción del sitio web en lugar del lado del diseño web [30].

En este proceso se incluye desde una simple página de texto HTML hasta aplicaciones complejas y extensas en características diseñadas para acceder desde varios dispositivos conectados a Internet. Ejemplos de desarrollo web en funciones incluyen sitios web de comercio electrónico, sistemas de gestión de contenido (CMS) y redes sociales. Los lenguajes y software de programación de desarrollo web más comunes incluyen lenguaje de marcado de hipertexto (HTML), hojas de estilo en cascada (CSS), *JavaScript*, PHP, *Drupal* y *MySQL* [30].

Para el desarrollo de nuestro sistema de recomendación se hizo un sistema web local con la capacidad de atender las peticiones de los usuarios (*Backend*) y devolver en formato JSON la información resultante de las consultas realizadas [30].

### 3.9.1 Servidor Frontend

El servidor de *Frontend* es el cliente y es el que se encarga de toda la lógica de este cuando desea realizar alguna petición; podemos decir que este concepto surge en el 2008 debido a las herramientas que surgieron a partir de ese momento: HTML5, CSS3 (2008), JSON (2013 - 2015), *AngularJS* (2010), *Ember*, *Backbone*, *Rest* (2000) y *Nodejs* [31].

El *Frontend* ha ido evolucionando con nuevas tecnológicas como lo son: ES6 (2015), *React* (2013), *Vue* (2014), *Angular* (2016), *GraphQL* (2015; en el panorama actual entra en el juego *WebAssembly* que es un nuevo tipo de código que se ejecuta en los navegadores web modernos y proporciona nuevas funciones y grandes ganancias en el rendimiento [31].

### 3.9.2 Servidor Backend

El servidor de *Backend* es la capa de acceso a datos de un software o cualquier dispositivo, que no es directamente accesible por los usuarios, además contiene la lógica de la aplicación que maneja dichos datos. El *Backend* es una aplicación especializada que entiende la forma como el navegador solicita cosas [31].

Algunos de los lenguajes de programación para *Backend* son Python, Node.js, PHP, Go, Ruby y C#. Todos los lenguajes mencionados tienen diferentes frameworks que nos permiten trabajar mejor según el proyecto que se esté desarrollando, como *Django*, *Flask*, *Express.js*, *Laravel*, *Symphony Framework*, *Ruby on Rails* y *ASP.Net* [31].

Existen diversas formas de configuración de servidores para hacer la conexión entre el *Frontend* y el *Backend*, donde la mayoría de las veces se trabajan en servidores separados con diferentes capacidades de almacenamiento; para el recomendador de productos, se decidió utilizar un solo servidor local donde se encuentren *Backend* y *Frontend* pero trabajando de la misma manera en cuanto a envío y recepción de peticiones [31].

### 3.9.3 JSON

JSON y sus siglas *JavaScript Object Notation* o Notación de Objetos de JavaScript, es un formato ligero de intercambio de datos, que resulta sencillo de leer y escribir para los programadores. Al ser JSON un formato que es independiente de cualquier lenguaje de programación, es donde los servicios comparten información por este método sin necesidad de hablar el mismo idioma; es decir, el emisor puede ser Java y el receptor Python, pues cada uno tiene su propia librería para codificar y decodificar cadenas en este formato [32].

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

En el siguiente Capítulo se abordará el desarrollo metodológico, explicando las etapas de manera general.

---

## 4. DESARROLLO METODOLÓGICO

---

En el presente capítulo se explica a detalle la solución de nuestra propuesta, la cual es una herramienta recomendador de productos en línea. Los pasos seguidos para la generación de dicha herramienta se enlistan en la metodología del proyecto.

## 4.1 Metodología del proyecto

Se aplicó una metodología en cascada de 4 etapas, una por semestre (IDI). En la primera etapa se hizo el diseño de los requerimientos del proyecto y especificaciones generales del mismo. En la etapa dos, se empezó con el desarrollo del modelo basado en el set de datos específicos, en la etapa tres se realizó la interfaz gráfica y la implementación del mismo. En la etapa final se elaboró la verificación final y corrección de posibles errores presentes en el sistema, así mismo se concluyó la escritura de la documentación donde cada IDI se conectaron consecutivamente.

Nuestro sistema de recomendación provee información importante acerca de productos o servicios a los usuarios interesados con base en la información que se obtiene de las búsquedas que realizan a través de internet.

En la Figura 20 se muestra un resumen de las etapas englobadas del sistema; el usuario puede generar búsquedas acordes a los productos, grupos o clientes donde se introduce la parte de *Frontend* en la interfaz gráfica, luego tenemos el servidor de *Backend* manteniendo la comunicación de la parte de *Frontend* y la base de datos NEO4J para la extracción de datos, por otra parte; los set de datos de Amazon se convirtieron en archivos CSV los cuales se les aplicó una limpieza y normalización de datos para hacer posible su envío a la base de datos.

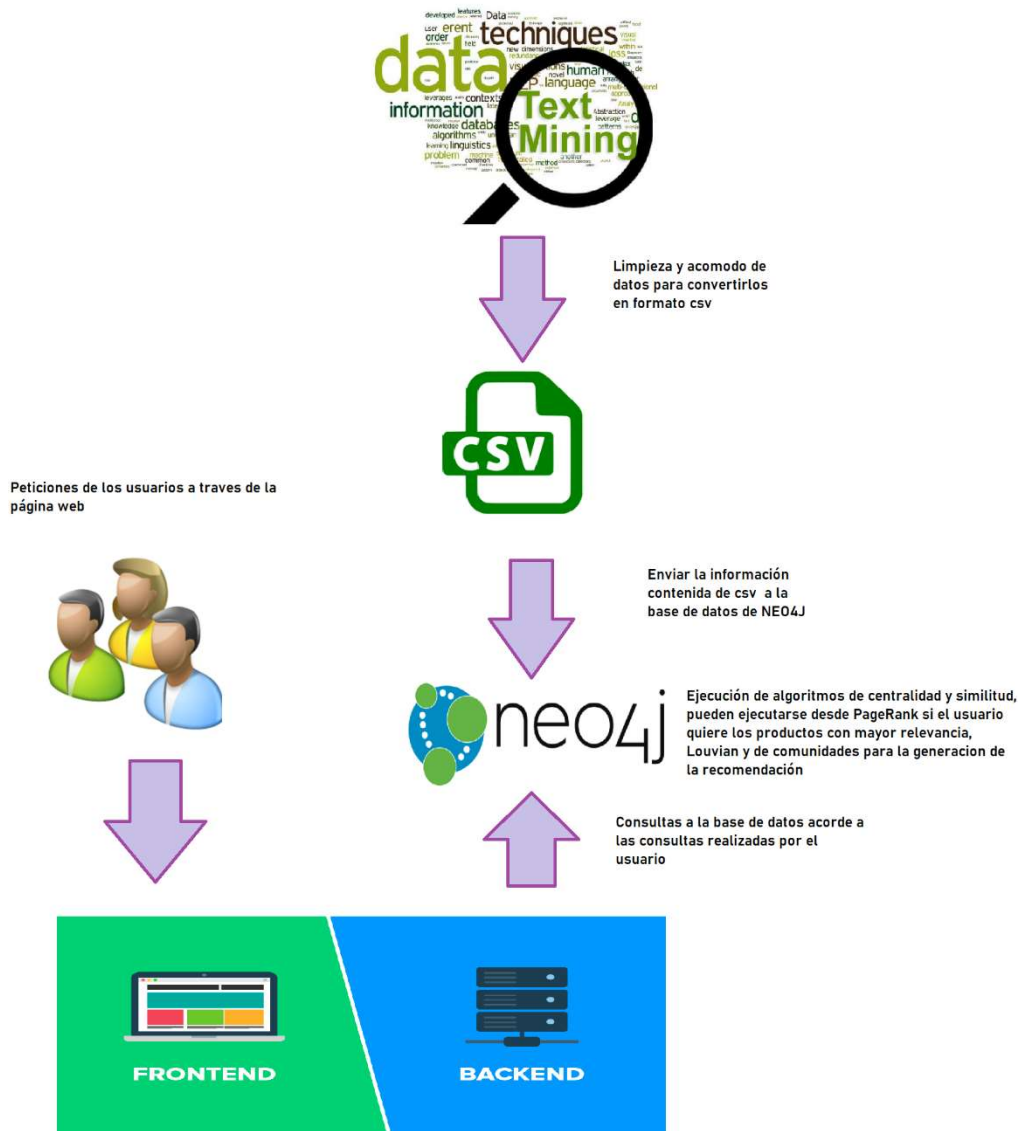


Figura 20. Etapas del sistema de recomendación de productos de co-compra.

El esquema del grafo generado de los productos de co-compra de Amazon incluyeron 3 nodos como se enlistan en la Tabla 4 y 6 relaciones enlistadas en la Tabla 5. Los nodos involucrados son clientes con id único, productos con llaves de propiedades tales como id único, título, reseñas totales, categorías, calificación promedio, ASIN, rango de ventas y número de descargas y grupos con un id único y nombre. En la Figura 21 se muestra el esquema del recomendador de productos de co-compra.

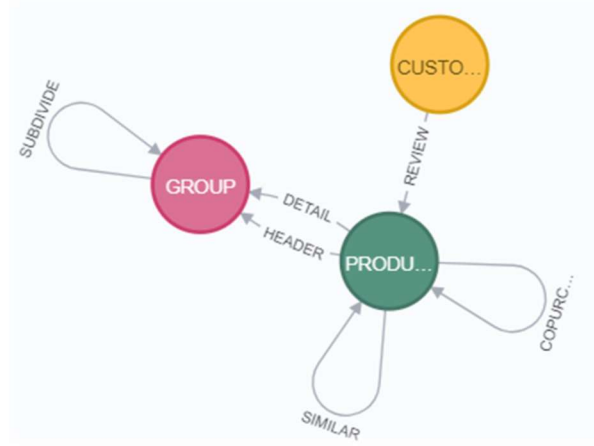


Figura 21. Estructura del grafo de sistema de recomendación.

Tabla 4. Nodos.

<i>Product</i>	<i>Id</i>
	<i>Title</i>
	<i>Total Reviews</i>
	<i>Categories</i>
	<i>Average Rating</i>
	<i>Asin</i>
	<i>Salesrank</i>
	<i>Downloaded</i>
<i>Customer</i>	<i>Id</i>
<i>Group</i>	<i>Id</i>
	<i>Name</i>

Tabla 5. Aristas.

<i>Detail</i>	
<i>Header</i>	
<i>Review</i>	<i>Ratings</i>
	<i>Votes</i>
	<i>Helpful</i>
	<i>Day</i>
	<i>Month</i>
	<i>Year</i>
<i>Similar</i>	
<i>Subdivide</i>	
<i>Copurchasing</i>	

En las Figuras 22, 23 y 24 se muestran con mayor profundidad las etapas que engloban nuestro recomendador de productos. La primera etapa es la obtención de la información cruda, la segunda etapa

es la parte de la conexión de la base de datos de la parte de *Backend* y la llamada de algoritmos, la tercera etapa es el despliegue del grafo generado de la recomendación.

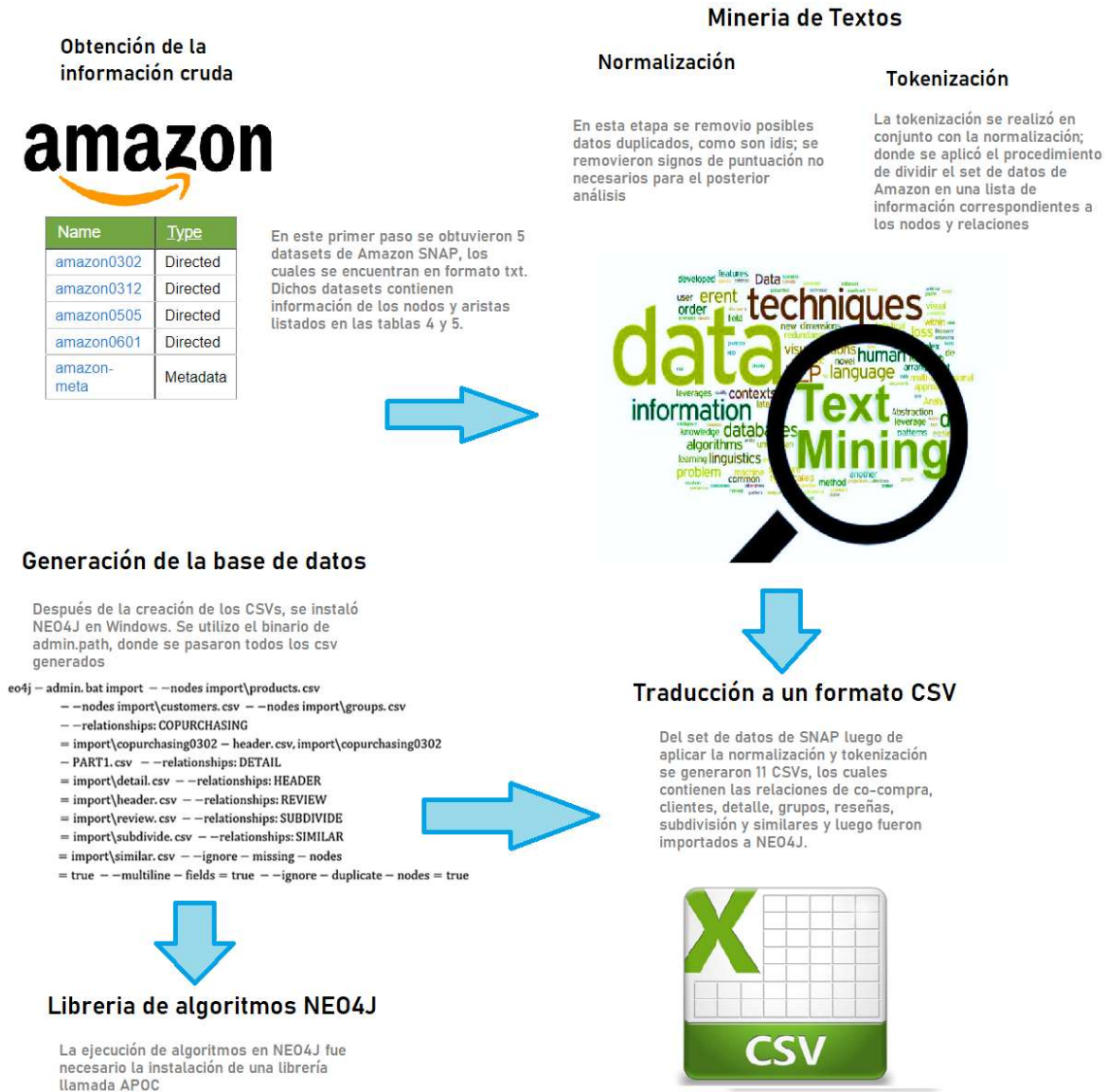


Figura 22. Primera fase del recomendador de artículos.



APOC ✓ Installed

3.5.0.4

The APOC library consists of many (about 450) procedures and functions to help with many different tasks in areas like data integration, graph algorithms or data conversion.



### Conexión DB NEO4J Con el código de backend

Se hizo la conexión de la base de datos de NEO4J haciendo uso de la librería de "neo4j-driver" de nodejs generando un módulo exclusivo para dicha configuración.

### Búsquedas para hacer recomendaciones de artículos en específico

asin	avg_rating	categories_n	downloaded	id	reviews_total	salesrank	titulo	Recomendador
027104445	4	2	2	11	8	15	23809	Living With Spikes

Cuando se despliega la ocurrencia de búsquedas, el usuario puede elegir entre 3 opciones de recomendaciones ya sea por producto, cliente, grupo. Los recomendadores actuales te despliegan una o dos opciones como máximo, nuestro recomendador tiene 3 opciones y al final te despliega un grafo recomendador

asin	avg_rating	categories_n	downloaded	id	reviews_total	salesrank	titulo	Recomendación
072104445	4	2	2	11	8	15	23809	Living With Spikes
027104445	4	2	2	11	8	15	23809	Living With Spikes
027222934	5	2	2	1	2	396585	Patterns of Preaching: A Sermon Sampler	★★★★★
07387202797	4	2	12	2	12	168596	Candlemas: Feast of Flames	★★★★★
048628785	5	1	1	3	1	1270652	World War II Allied Fighter Planes Trading Cards	★★★★★
084232827	4	5	1	4	1	631289	Life Application Bible Commentary: 1 and 2 Timothy and Titus	★★★★★
157794202	0	2	0	5	0	455160	Prayers That Avail Much for Business: Executive	★★★★★
048622025	4	5	17	6	17	168784	How the Other Half Lives: Studies Among the Tenements of New York	★★★★★
B00005AU3R	4	3	3	7	3	5392	Butik	★★★★★

8 rows visible

### Módulo de algoritmos

En este módulo se tiene la clase principal de NEO4J donde se crea la sesión de la base de datos para ser capaces de ejecutar los queries de los algoritmos de:

```

asin0231118597
avg_rating4
categories_n4
downloaded15
id8
reviews_total15
salesrank277409
tituloLosing Matt Shepard
Recommendation★★★★★
asin0439240751
avg_rating5
categories_n6
downloaded4
id33
reviews_total4
salesrank97166
tituloDouble Jeopardy (T*Witches, 6)
Recommendation★★★★★

```

### PageRank

Si el usuario hace click en "Relevant Products/Groups/Customers", PageRank se llamará para desplegar una lista de artículos, donde luego el usuario podrá correr la recomendación

Select Category

- All Products
- All Groups
- All Customers
- By Reviews
- Relevant Products
- Relevant Groups
- Relevant Customers

Detail C11 Product

Product id: 10

Title: Title

Total Reviews: Total Reviews

Average Rating: Average Rating

Categories: Categories

asin: asin

Average Rating: Average Rating

asin: asin

Salesrank: Salesrank

Downloaded: Downloaded

### Producto

Recommendations

asin	avg_rating	categories_n	downloaded	id	reviews_total	salesrank	titulo
053278223	4	4	258	1212	208	3016	The Martian Chronicles
019330814	4	4	3	1213	3	151807	The Tactics of Small Boat Racing
047129576	4	10	11	1214	11	63874	New Sales Speak
B00004AT44	5	2	6	1215	6	17031	Vier Geestret
0801862515	5	5	2	1494	2	190163	Walker's Promoters of the World
0373122179	5	3	2	1495	2	434711	Rafael's Mistress (Harlequin Presents, No. 2217)
0395901294	4	2	2	1496	2	882768	The Store of a Million Items: Stories
0303973704	4	1	5	2134	5	30499	Voices of the Dumbek

8 rows visible

### Grupo

Recommendations


id	nombre
283155	Butik(283155)
139452	(139452)

### Cliente

Recommendations

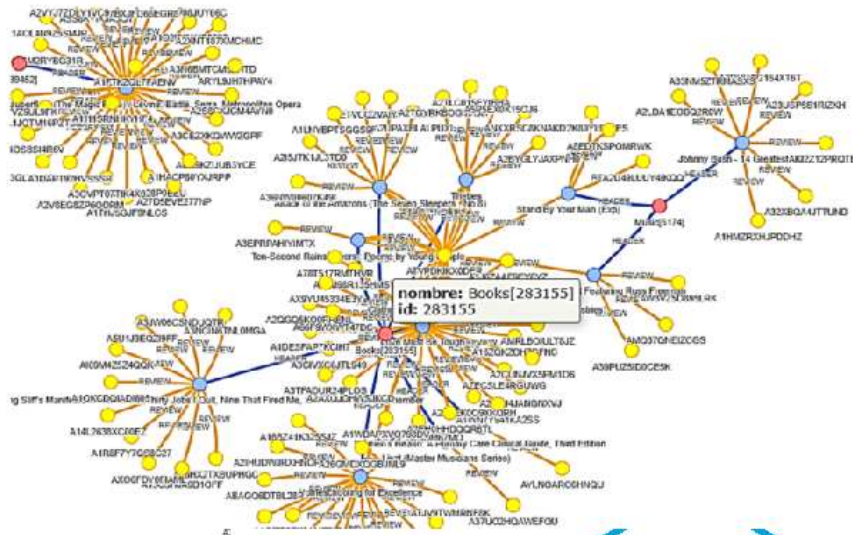
id
A7V0K8002R
A2N0K9Y64D9H
A3F188F9G21H
A102QZAK215M
A3G5ZNA8R37

Figura 23. Segunda fase del recomendador de artículos.

**Tipos de recomendaciones**  **Louvian & Elementos similares**

Dichos algoritmos se ejecutan si el usuario quiere un recomendación específica acorde al artículo seleccionado en la página web

 Network Recommendation Graph



Nuestra solución provee una gamma de categorías, donde el usuario puede elegir artículos por producto, grupos y clientes; si necesita un artículo en específico puede ejecutar la búsqueda en específico ya sea por id, nombre u otra categoría.

Figura 24. Tercera fase del recomendador de artículos.

## 4.2 Obtención de la información cruda

El proceso de la información cruda consiste en la elección de un set de datos, el cual contenga datos relevantes para las consultas que se realicen.

El paso inicial fue la elección del set de datos a utilizar para su posterior análisis [33].

Los datos se recopilaron rastreando el sitio web de Amazon y contienen metadatos de productos e información de revisión sobre 548,552 productos diferentes (libros, CD de música, DVD y cintas de video VHS).

Para cada producto se tiene la siguiente información [33]:

- Título
- Rango de ventas

- Lista de productos similares (que se compran conjuntamente con el producto actual)
- Categorización detallada de productos
- Reseñas de productos: tiempo, cliente, calificación, número de votos, cantidad de personas que encontraron útil la reseña
- Los datos se recopilaron en el verano de 2006

En la Figura 25 se enlistan el set de datos utilizados para el recomendador de productos de co-compra.

### Product co-purchasing networks

Name	Type	Nodes	Edges	Description
amazon0302	Directed	262,111	1,234,877	Amazon product co-purchasing network from March 2 2003
amazon0312	Directed	400,727	3,200,440	Amazon product co-purchasing network from March 12 2003
amazon0505	Directed	410,236	3,356,824	Amazon product co-purchasing network from May 5 2003
amazon0601	Directed	403,394	3,387,388	Amazon product co-purchasing network from June 1 2003
amazon-meta	Metadata	548,552	1,788,725	Amazon product metadata: product info and all reviews on around 548,552 products.

Figura 25. Set de datos de los productos de co-compra de Amazon [33].

El formato de cada producto se muestra en la Figura 26.

```

Id: 15
ASIN: 1559362022
title: Wake Up and Smell the Coffee
group: Book
salesrank: 518927
similar: 5 1559360968 1559361247 1559360828 1559361018 0743214552
categories: 3
|Books[283155]|Subjects[1000]|Literature & Fiction[17]|Drama[2159]|United States[2160]
|Books[283155]|Subjects[1000]|Arts & Photography[1]|Performing Arts[521000]|Theater[215
|Books[283155]|Subjects[1000]|Literature & Fiction[17]|Authors, A-Z[70021]|( B ) [70023]
reviews: total: 8 downloaded: 8 avg rating: 4
2002-5-13  cutomer: A2IGOA66Y6O8TQ  rating: 5  votes: 3  helpful: 2
2002-6-17  cutomer: A2OIN4AUH84KNE  rating: 5  votes: 2  helpful: 1
2003-1-2   cutomer: A2HN382JNT1CIU  rating: 1  votes: 6  helpful: 1
2003-6-7   cutomer: A2FDJ79LDU4O18  rating: 4  votes: 1  helpful: 1
2003-6-27  cutomer: A39QMV9ZKRJXO5  rating: 4  votes: 1  helpful: 1
2004-2-17  cutomer: AUUVMSTQ1TXDI  rating: 1  votes: 2  helpful: 0
2004-2-24  cutomer: A2C5K0QTL9UAT  rating: 5  votes: 2  helpful: 2
2004-10-13 cutomer: A5XYF0Z3UH4HB  rating: 5  votes: 1  helpful: 1

```

Figura 26. Formato de productos de Amazon [33].

Formato de datos [33]:

- Id: Identificación del producto (número 0, ..., 548551)
- ASIN: Número de identificación estándar de Amazon
- *title*: Nombre / título del producto
- grupo: grupo de productos (libro, DVD, vídeo o música)
- *salesrank*: Amazon *Salesrank*
- similar: ASIN de productos comprados conjuntamente (las personas que compran X también compran Y)
- categorías: ubicación en la jerarquía de categoría de producto a la que pertenece el producto (separada por |, ID de categoría en [])

- revisiones: información de la revisión del producto: tiempo, identificación del usuario, calificación, número total de votos en la revisión, número total de votos útiles (cuántas personas encontraron la revisión útil)

## 4.2.1 Amazon

Amazon es una de las 500 mayores empresas de EE.UU. La compañía, con sede en Seattle (Washington) es un líder global en el comercio electrónico. Desde que Jeff Bezos lanzó Amazon.com en 1995, se ha hecho un progreso significativo en la oferta, en los sitios web y en la red internacional de distribución y servicio al cliente. En la actualidad, Amazon ofrece gran variedad de productos, desde libros o productos electrónicos, hasta raquetas de tenis o diamantes. Tiene presencia directa en Estados Unidos, Reino Unido, Alemania, Francia, Italia, España, Japón, Canadá y China, pero además podemos servir a los clientes en la mayoría de los países del mundo [34].

La innovación tecnológica es la base para la expansión de Amazon y permite a los clientes tener más y más categorías de productos, adaptados a sus necesidades y a un precio más bajo. Ofrecemos experiencia de compra personalizada: por ejemplo, nuestros clientes pueden comprar rápidamente, con el "Pedido 1-Clic" y utilizar diversas herramientas, tales como consejos para descubrir nuevos productos [34].

En 2000, Amazon comenzó a ofrecer su plataforma de comercio a otros minoristas y proveedores. Hoy en día, muchas empresas están trabajando con *Amazon Services* para desarrollar su oferta de comercio electrónico, de servicios técnicos y de servicio al cliente y la gestión y el envío de los pedidos. Las principales marcas también utilizan Amazon como canal de ventas, ofreciendo sus productos directamente a los sitios de Amazon. Por último, los desarrolladores externos de software utilizan *Amazon Web Services* para crear aplicaciones y servicios que se proporcionan a los clientes y vendedores de Amazon [34].

La evolución de la página web de Amazon está determinada por nuestro deseo de innovar constantemente, que siempre ha formado parte del ADN de nuestra compañía. El comercio electrónico se encuentra todavía en una etapa temprana. En los próximos años, se verá cómo Amazon crea nuevas tecnologías, se expande a más regiones geográficas y continua mejorando la vida de los consumidores y vendedores de todo el mundo [34].

## 4.3 Minería de textos

### 4.3.1 Normalización

La normalización se realizó generando scripts de Python para la extracción de la información, donde se mapearán el set de datos en CSVs para su futuro análisis en NEO4J. En este proceso se removieron *Stop Words* las cuales son palabras sin un significado o importancia en el análisis de nuestro set de datos. Se colocó la información en minúsculas para evitar posibles títulos de productos con mayúsculas intermedias y se validó si existía posible duplicidad de ids de productos, clientes o grupos donde se pudo validar que dicho set de datos no contenía datos duplicados.

### 4.3.2 Tokenización

La tokenización se realizó en conjunto con la normalización; donde se aplicó el procedimiento de dividir el set de datos de Amazon en una lista de información correspondientes a los nodos de clientes, grupos y productos; de igual manera con las aristas de co-compra, detalle, encabezado, reseñas, similares y subdivisión, de las cuales se mostrará más a fondo en el Capítulo 5.

### 4.3.3 Traducción a un formato CSV

Del set de datos de SNAP, se generaron 11 CSVs, los cuales contienen las relaciones de co-compra, clientes, detalle, grupos, reseñas, subdivisión y similares y luego fueron importados a NEO4J.

SNAP es un set de datos de la universidad de Stanford (*Stanford Large Network Dataset Collection*), donde se albergan más de 50 grandes conjuntos de datos de red, grafos y redes de comunicación por solo mencionar algunos. En el Apéndice C se habla más detalladamente de los conjuntos de datos disponibles en esta comunidad.

La relación de los CSV se hablará más a detalle en el Capítulo 5.

## 4.4 Generación de la base de datos

Después de la creación de los CSVs, se instaló NEO4J en Windows. NEO4J ofrece una llave de activación en tu primera descarga, donde es posible hacer el despliegue de bases de datos de grafos. Acerca de la instalación y configuración de NEO4J se encuentra en el Apéndice A donde se explica cómo crear y eliminar un proyecto, configuraciones al proyecto, creación de una base de datos, su administración, entre otras utilidades.

## 4.5 Implementación de algoritmos de clustering (Louvain), relevancia (PageRank)

Antes de la elección de los algoritmos se realizó un análisis, donde se ejecutaron *queries* de algoritmos fuertemente conectados Figura 27 y débilmente conectados Figura 28 por mencionar solo un par.

```
CALL algo.scc.stream("GROUP", "SUBDIVIDE") YIELD nodeId, partition RETURN partition,
collect(algo.getNodeById(nodeId)) AS libraries ORDER BY size(libraries) DESC LIMIT 1;
```

"partition"	"libraries"
0	[{"nombre":"Health, Mind & Body[10]","id":"10"}, {"nombre":"Personal Health[282800]","id":"282800"}, {"nombre":"Aging[282840]","id":"282840"}, {"nombre":"General[4655]","id":"4655"}, {"nombre":"Books[283155]","id":"283155"}, {"nombre":"Subjects[1000]","id":"1000"}, {"nombre":"Children's Books[4]","id":"4"}, {"nombre":"Animals[2787]","id":"2787"}, {"nombre":"Dinosaurs[2808]","id":"2808"}, {"nombre":"Nonfiction[2810]","id":"2810"}, {"nombre":"Literature & Fiction[17]","id":"17"}, {"nombre":"Genre Fiction[10134]","id":"10134"}, {"nombre":"Historical[10177]","id":"10177"}, {"nombre":"Travel[27]","id":"27"}, {"nombre":"Asia[16772]","id":"16772"}, {"nombre":"China[16785]","id":"16785"}, {"nombre":"General[67588]","id":"67588"}, {"nombre":"Nonfiction[53]","id":"53"}, {"nombre":"Social Sciences[11232]","id":"11232"}, {"nombre":"Sociology[11288]","id":"11288"}, {"nombre":"General[11289]","id":"11289"}, {"nombre":"War[10195]","id":"10195"}, {"nombre":"Professional & Technical[173507]","id":"173507"}, {"nombre":"Architecture[173508]","id":"173508"}, {"nombre":"General[1005]","id":"1005"}, {"nombre":"Self-Help[4736]","id":"4736"}, {"nombre":"General[4738]","id":"4738"}, {"nombre":"Entertainment[86]","id":"86"}, {"nombre":"Music[4507]","id":"4507"}, {"nombre":"Theory, Composition & Performance[4539]","id":"4539"}, {"nombre":"Appreciation[4540]","id":"4540"}, {"nombre":"Relationships[4729]","id":"4729"}, {"nombre":"Interpersonal Relations[4732]","id":"4732"}, {"nombre":"Specialty Stores[504360]","id":"504360"}, {"nombre":"New & Used Textbooks[465600]","id":"465600"}]

Figura 27. Salida de un *query* de fuertemente conectados.

Algoritmo débilmente conectado, donde se nos retorna el número de caminos posibles de un nodo y una relación Figura 28.

```
CALL algo.unionFind("GROUP", "SUBDIVIDE", {write: True, writeProperty: "MINST"})YIELD nodes,
setCount, loadMillis, computeMillis, writeMillis;
```

"nodes"	"setCount"	"loadMillis"	"computeMillis"	"writeMillis"
3422	1	321	27	382

Figura 28. Salida de un *query* de débilmente conectados.

Para la implementación de los algoritmos previamente mencionados sin olvidar tampoco *PageRank* y *Louvian* NEO4J ofrece una biblioteca llamada APOC, definida como: “Procedimientos Impresionantes en *Cypher*” (*Awesome Procedures On Cypher*); cuenta como más de 450 procedimientos estándar. Es necesario su instalación debido que no viene por defecto en NEO4J; en el Apéndice B se muestran los pasos para su configuración.

## 4.6 Creación de servidor Backend

Para la implementación de *PageRank* y *Louvian* del lado de *Backend* se utilizó un driver creado en *JavaScript* para generar las peticiones a la base de datos de NEO4J. Para un uso más amigable del lenguaje se hizo uso de IDE de *Visual Studio Code*; posee un amplio set de *plugins* útiles como *prettier* utilizado

para el formateo de código, *vscode-icons* para mostrar los documentos con icono acorde a la extensión del archivo y *JavaScript (ES6) code snippets*, esta extensión contiene fragmentos de código para JavaScript en la sintaxis de ES6 para el editor de código VS (admite *JavaScript* y *TypeScript*). En el Apéndice D se enlistan unos primeros pasos para el uso de *Visual Studio Code*.

Node.js es parte de *JavaScript*, donde se crean aplicaciones de servidores, como es el caso de la plataforma de recomendador de productos de co-compra, donde se necesita de un sistema para las peticiones de back. En el Apéndice E se explica la instalación de Node.js

## 4.7 Desarrollo de la interfaz gráfica UI

La interfaz gráfica se despliega en un servidor localmente haciendo uso de *node* para correr la aplicación. En la figura 29 se muestra la página principal de la interfaz gráfica. En el Capítulo 5 se hablará a más detalle de las categorías mostradas en la parte izquierda de la página de inicio.



Figura 29. Página principal de la interfaz gráfica del recomendador de productos de co-compra.

## 4.8 UI Conexión entre Frontend y Backend

La unión del *Backend* y el *Frontend* de la aplicación es la interacción con el usuario final; se hizo uso de *routers* generados en *JavaScript*, divididos por peticiones de cliente, grupo, producto y reseñas. En la Figura 30 se muestra un pequeño fragmento del *router* de Clientes.

```

1  const router = require("express").Router();
2  const amazon = require("../neo4jdb/Neo4jDriverClassAmazon");
3
4  router.route("/").get(async (req, res) => {
5    let page = false;
6    let limit = "25";
7    let from = "1990-05-20";
8    let to = "2020-05-22";
9    let author = "";
10   let value = "";
11   let field = "id";
12
13   if (req.query.pageRank) {
14     page = true;
15   }
16
17   if (req.query.limit) {
18     if(req.query.limit >= 0 && req.query.limit <=100000){
19       limit = req.query.limit;
20     }else{
21       res.status(400).send(`{error : Limite ${req.query.limit} esta fuera del
22         rango permitido, pruebe entre 0-100}`);
23       return;
24     }
25   }
26 }

```

Figura 30. Fragmento de código del *router* de cliente.

## 4.9 Visualización de las recomendaciones (subgrafo) usando Neovis.js

Se hizo uso de una librería externa para la visualización del grafo resultante de la recomendación llamada *neovis.js*. En el Apéndice F se explica detalladamente las características que ofrece dicha librería como es la obtención grafica de datos en vivo, la manipulación y visualización de información específica de los nodos y relaciones mostradas respectivamente. Se explicará más a detalle en el Capítulo 5 la implementación de la librería.



---

## 5. RESULTADOS Y DISCUSIÓN

---

## 5.1. Resultados

En esta sección se mencionan los resultados obtenidos del sistema de recomendación de productos de co-compra. Después de la elección del set de datos de Amazon obtenidos de SNAP y del proceso de limpieza se generaron los CSVs para ser cargados en NEO4J.

Los CSVs generados fueron los siguientes:

- Clientes
- Producto
- Grupos
- Reseñas
- Co-compras
- Detalles
- Encabezados
- Similares
- Subdivisiones

Los CSVs de co-compra contienen las relaciones de productos a grupos. El CSV de clientes es referente al nodo de *customer* contiene el valor de ID y la etiqueta correspondiente. El CSV de productos es referente al nodo de *product* contiene los valores de id, ASIN, título, *salesrank*, número de categorías, reseñas totales, número de descargas totales, clasificación promedio y etiqueta correspondiente. El CSV de grupos es referente al nodo *group*, contiene los valores de ID, nombre y la etiqueta correspondiente. El CSV de detalles contienen las relaciones de productos a grupos, se les agrego un peso de 1 para la ejecución de algoritmos. El CSV de encabezado contiene las relaciones de productos a grupos, se les agrego un peso de 1 para la ejecución de algoritmos. El CSV de similar contiene las relaciones de productos a productos, se les agrego un peso de 1 para la ejecución de algoritmos. El CSV de subdivisión contiene las relaciones de grupos a grupos, se les agrego un peso de 1 para la ejecución de algoritmos. El CSV de reseñas contiene las relaciones de clientes a productos, se les agrego un peso de 1 para la ejecución de algoritmos, se tienen los valores votos, evaluaciones.

Después de la creación de los CSVs, se creó una base de datos vacía en NEO4J donde posteriormente se cargaron los CSVs previamente descritos. Los CSVs deben estar localizados en el directorio de *import* dentro de la base de datos creada. Para localizar dicho directorio se hizo *click* en “*Open Folder*” en “*My Project*”. Figura 31.

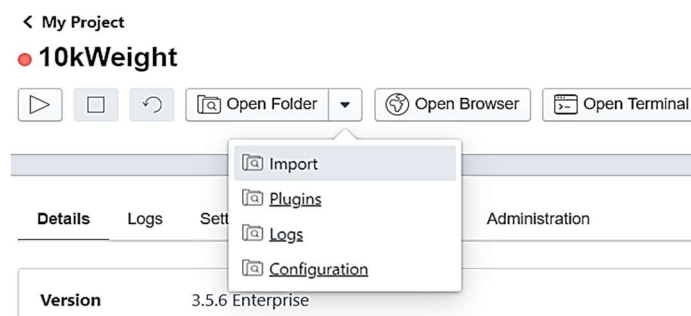


Figura 31. Directorio *import* en NEO4J.

En una terminal de línea de comandos se ejecutó el comando mostrado en la Figura 32.

```
bin\neo4j - admin.bat import --nodes import\products.csv
--nodes import\customers.csv --nodes import\groups.csv
--relationships: COPURCHASING
= import\copurchasing0302 - header.csv, import\copurchasing0302
PART1.csv --relationships: DETAIL
= import\detail.csv --relationships: HEADER
= import\header.csv --relationships: REVIEW
= import\review.csv --relationships: SUBDIVIDE
= import\subdivide.csv --relationships: SIMILAR
= import\similar.csv --ignore - missing - nodes
= true --multiline - fields = true --ignore - duplicate - nodes = true
```

Figura 32. Comando de importación de CSVs a NEO4J.

Para cargar los csv en la base de datos de NEO4J previamente creada. Se utilizó el binario *neo4j-admin.bat* para cargarlos y *-nodes* con los valores de los nodos: grupos, clientes y productos y con *-relationships* las relaciones de los nodos como son, detalle, título, reseña, subdivisión y similar.

Se cargaron 105,944 nodos y 1'004,223 relaciones. La versión de NEO4J es 3.5.6 *Enterprise*.  
Figura 33.

<b>Version</b>	3.5.6 Enterprise
<b>Status</b>	STOPPED
<b>Nodes</b>	105944
<b>Labels</b>	3
<b>Relationships</b>	1004223
<b>Relationship types</b>	6

Figura 33. Número de nodos y relaciones de la base de datos de NEO4J.

La ejecución de algoritmos en NEO4J fue necesario la instalación de una librería llamada APOC, en la opción de *plugins* se instaló APOC tal como se muestra en la Figura 34.

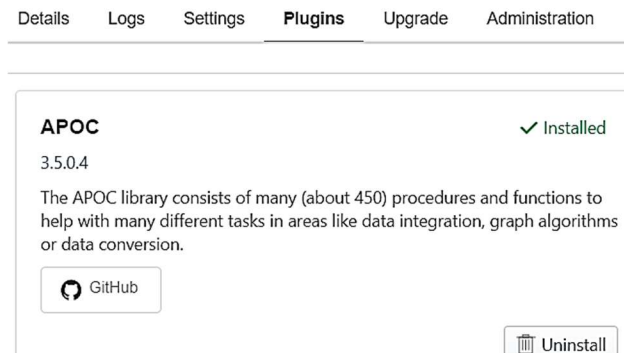


Figura 34. Plugin de APOC de NEO4J.

Los algoritmos utilizados para generar las recomendaciones acordes a los productos fueron los siguientes:

- *PageRank*-Centralidad
- *Louvian*
- Similaridad

Se hizo uso de librerías disponibles en neo4j como es APOC la cual contiene alrededor de 450 algoritmos incluyendo are como integración de datos, algoritmos de grafos o de conversión de datos. Solo se implementaron los algoritmos de *PageRank*, *Louvian* y de similaridad.

La página principal del sistema de recomendación de productos se muestra en la Figura 35.



Figura 35. Página principal del sistema de recomendación de productos de Amazon.

Se agregó un carrusel de imágenes en la página, mostrando imágenes: log del ITESO, Amazon, NEO4J logo. Figura 36.



Figura 36. Imágenes en el carrusel de la página principal

En el panel de la parte izquierda de la página principal tenemos el menú de filtros por productos, grupos, clientes y por reseñas; si se necesitan los datos relevantes se aplica el algoritmo de *PageRank* para los filtros previamente listados. Figura 37.

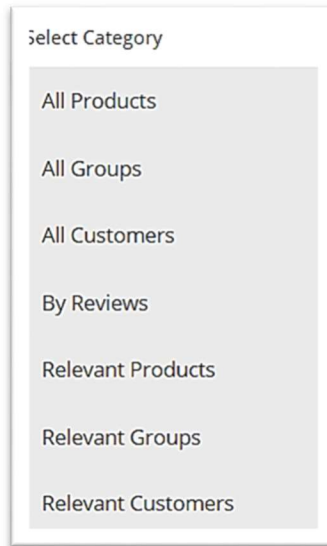


Figura 37. Menú de filtros disponibles en el sistema de recomendación.

Si se selecciona *All Products* o *Relevant Products* se desplegará el menú con entradas de búsqueda. Figura 38. Se tiene la opción de hacer búsquedas por *Product id*, *Title*, *Total Reviews*, *Categories*, *Average Rating*, *Asin* y *Salesrank*.

Figura 38. Menú de productos.

En el caso de *All Products* se ejecuta el *query* de la Figura 39.

```
MATCH (n:PRODUCT) WHERE n.titulo CONTAINS "" RETURN n
```

Figura 39. *Query* de todos los productos.

En el caso de *Relevant Products* se ejecuta el *query* de la Figura 40.

```
CALL algo.pageRank.stream(PRODUCT,null,{iterations: 20,dampingFactor: 0.85})YIELD nodeId,score
WHERE algo.getNodeById(nodeId).titulo CONTAINS
RETURN algo.getNodeById(nodeId) AS page ORDER BY score DESC LIMIT 500;
```

Figura 40. *Query* de los productos relevantes.

Se puede observar la aplicación del algoritmo de *PageRank* en el caso de productos relevantes. En la Figura 41 tenemos la salida del *query* de *All Products* y en la Figura 42 *Relevant Products*.

asin	avg_rating	categories_n	downloaded	id	reviews_total	salesrank	titulo	Recommendation
0771044445	0	0	0	0	0	0		
0827229534	5	2	2	1	2	396585	Patterns of Preaching: A Sermon Sampler	
0738700797	4	2	12	2	12	168596	Candlemas: Feast of Flames	
0486287785	5	1	1	3	1	1270652	World War II Allied Fighter Planes Trading Cards	
0842328327	4	5	1	4	1	631289	Life Application Bible Commentary: 1 and 2 Timothy and Titus	
1577943082	0	2	0	5	0	455160	Prayers That Avail Much for Business: Executive	
0486220125	4	5	17	6	17	188784	How the Other Half Lives: Studies Among the Tenements of New York	
B00000AU3R	4	3	3	7	3	5392	Batik	

8 rows visible

1 2 3 4 5 ... 63

Figura 41. Búsqueda de todos los productos.

asin	avg_rating	categories_n	downloaded	id	reviews_total	salesrank	titulo	Recommendation
0231118597	4	4	15	8	15	277409	Losing Matt Shepard	
0439240751	5	6	4	33	4	97166	Double Jeopardy (T*Witches, 6)	
0965740056	0	4	0	93	0	1485593	BizPricer¿ Business Valuation Manual w/Software	
0393049388	3	3	3	23	3	1277867	Strange Fire: A Novel	
039572774X	4	4	15	94	15	166549	The American Heritage Dictionary of Idioms	
1550416243	0	2	0	56	0	2167012	Three Bricks Shy of a Load: A Collection of True Stories About Ordinary People Doing Really Dumb Stuff!	
0887402631	0	1	0	95	0	1369585	The Heavy Flak Guns, 1933-1945: 88Mm, 105Mm, 128Mm, 150Mm, and Ballistic Directional Equipment	
B00000IC82	3	16	13	151	13	44671	Laurel & Hardy - Flying Deuces/Utopia	

8 rows visible

1 2 3 4 5 ... 63

Figura 42. Búsqueda de productos relevantes.

La ventana de búsqueda puede mostrar los detalles en formato de lista, tal como se muestra en la Figura 43.

asin	avg_rating	categories_n	downloaded	id	reviews_total	salesrank	titulo	Recommendation
asin0231118597	avg_rating4	categories_n4	downloaded15	id8	reviews_total15	salesrank277409	tituloLosing Matt Shepard	Recommendation
asin0439240751	avg_rating5	categories_n6	downloaded4	id33	reviews_total4	salesrank97166	tituloDouble Jeopardy (T*Witches, 6)	Recommendation

Figura 43. Búsqueda por listas.

Se puede hacer búsqueda por palabras en específicos para hacer filtros, por ejemplo, se busca la palabra *Losing* regresando un artículo con dicha ocurrencia, tal como se muestra en la Figura 44.

asin	avg_rating	categories_n	downloaded	id	reviews_total	salesrank	titulo	Recommendation
0231118597	4	4	15	8	15	277409	Losing Matt Shepard	

Figura 44. Búsqueda por palabras.

Se tiene filtrado por categoría Figura 45; por defecto tenemos todas las filas referentes a cada producto, si no se necesita desplegar algún valor se puede remover de la caja.

esrank	titulo	Recommendation
7409	Losing Matt Shepard	
66	Double Jeopardy (T*Witches, 6)	
35593	BizPricer; Business Valuation Manual w/Software	
7867	Strange Fire: A Novel	
549	The American Heritage Dictionary of Idioms	
7012	Three Bricks Shy of a Load: A Collection of True Stories About Ordinary People Doing Really Dumb Stuff!	
9585	The Heavy Flak Guns, 1933-1945: 88Mm, 105Mm, 128Mm, 150Mm, and Ballistic Directional Equipment	
71	Laurel & Hardy - Flying Deuces/Utopia	

Figura 45. Búsqueda por etiquetas.

En la Figura 46 se muestra como se remueven las filas de ASIN, *avr\_rating*, *categories*, *ID*, *reviews\_totals*, *salesrank* de la búsqueda y solo se deja el título y la recomendación. Dichos filtros aplican para clientes, productos y grupos.

titulo	Recommendation
Losing Matt Shepard	
Double Jeopardy (T*Witches, 6)	
BizPricer; Business Valuation Manual w/Software	
Strange Fire: A Novel	
The American Heritage Dictionary of Idioms	
Three Bricks Shy of a Load: A Collection of True Stories About Ordinary People Doing Really Dumb Stuff!	
The Heavy Flak Guns, 1933-1945: 88Mm, 105Mm, 128Mm, 150Mm, and Ballistic Directional Equipment	
Laurel & Hardy - Flying Deuces/Utopia	

8 rows visible

1 2 3 4 5 ... 63

Figura 46. Remover etiquetas.

Como se mostró en la Figura 46, es posible generar filtros por *Product id*, *Title*, *Total Reviews*, *Categories*, *Average Rating*, *ASIN* y *Salesrank*; se generó una búsqueda de *product ID* igual a 777 obteniendo una ocurrencia de 20 productos Figura 47.

Product Id

777

asin	avg_rating	categories_n	downloaded	id	reviews_total	salesrank	titulo	Recommendation
B00005YTRI	1	2	1	777	1	76594	American Troubadour	
089255228X	4	2	2	2777	2	142680	Starting With "I": Personal Essays by Teenagers	
1569373329				1777				
096729570X	0	11	0	3777	0	613636	Discovering Acadia: An Introduction to the Park & Its Environment (Discovering Ser) (Discovering Ser)	
6305771111	2	19	7	5777	7	28637	Las Vegas Hillbillies	
188938612X	4	3	7	6777	7	750958	The Appalachian Trail Workbook for Planning Thru-Hikes	
1586850083	4	4	2	7770	2	461035	Girlfriend, You Are the Best: A Fable for Our Times	
0471956031	0	7	0	7771	0	1358760	An Introduction to Management for Engineers	
1586480065	5	1	6	7772	6	897088	American Greats	
0679449876	4	1	85	7773	85	232035	Tender at the Bone : Growing Up at the Table	

Figura 47. Filtrado por *Product ID*.

Si se quiere un filtrado de *Product ID* y *Title* es posible, dando un resultado de una ocurrencia. Dicha concurrencia se muestra en la Figura 48.

Product

Product Id

777

Title

Vegas

asin	avg_rating	categories_n	downloaded	id	reviews_total	salesrank	titulo	Recommendation
6305771111	2	19	7	5777	7	28637	Las Vegas Hillbillies	

Figura 48. Filtrado por *Product ID* y *Title*.

En la Figura 49 se hace un filtrado por *Total Reviews* 10, *Categories* 5, *Average Rating* 4, *Salesrank* 2, *Downloaded* 10.



Total Reviews  
10

Categories  
5

Average Rating  
4

Asin  
Asin

Salesrank  
4

Downloaded  
10

---

Search

asin	avg_rating	categories_n	downloaded	id	reviews_total	salesrank	titulo	Recommendation
0441555217	4	5	10	3365	10	46622	Myth Conceptions (Myth Books)	
B0000026NC	4	5	102	3504	102	6412	Delicate Sound of Thunder	
0671034014	4	5	104	4625	104	264418	Mercy	
B00004592N	4	5	10	6141	10	38102	Kronos Caravan	
039573097X	4	5	10	6190	10	182078	Indira: The Life of Indira Nehru Gandhi	
0836251830	4	5	10	6273	10	321986	Welcome to Jatorassic Park: A FoxTrot Collection	
B000002GK1	4	5	110	7702	110	3925	Badmotorfinger	

Figura 49. Filtrado por *Total Reviews*, *Categories*, *Average Rating*, *Salesrank*, *Downloaded*.

En las opciones de filtrado se tiene la opción de hacer recomendaciones de cierto producto por productos/grupos/clientes recomendados; de la tabla de la Figura 49 se eligió *Welcome to Jatorassic Park: A FoxTrot Collection*, generándose una recomendación de productos como se muestra en la Figura 52, de grupos Figura 53 y clientes Figura 54. En la Figura 50 y Figura 51 se enlistan los *queries* ejecutados generando las recomendaciones acordes al producto seleccionado.

```
CALL algo.louvain.stream(PRODUCT,null)YIELD nodeId,community WHERE algo.getNodeById(nodeId).titulo
CONTAINS Welcome to Jatorassic Park: A FoxTrot Collection
RETURN DISTINCT algo.getNodeById(nodeId) AS node,community LIMIT 200
```

Figura 50. *Query* del algoritmo de *Lovian*.

El *query* de similar toma la salida de *Louvian*, dependiendo si se selecciona producto, grupo o cliente, se ejecutará el *return* correspondiente. Figura 51.

MATCH (customer: CUSTOMER) – [review: REVIEW] → (product: PRODUCT) – [header: HEADER]  
 → (group: GROUP) WHERE product.titulo IN ["Invitation to Theology", "No One Can Hear You",

*Top Percussion, 15 Grandes Exitos de Nicola Di Bari en Espanol, Hug Me,*

*Reflexology: The Timeless Art of Healing, Real Ghostbusters vol. 6: Revenge of Murray the Mantis,*

*High Performance Cluster Computing: Architectures and Systems, Black Lightning,*

*Pumpkinhead, Every Eye Beholds You: A World Treasury of Prayer,*

*Bees of the World (Of the World Series), Learn Library of Congress Classification,*

*Victorian Dwellings for Village and Country (1885) (Dover Books on Architecture),*

*Financial Success : Harnessing the Power of Creative Thought, Fatality, The Ultimate George Gershwin, Vol. 1,*

*He Who Lets Us Be: A Theology of Love, The Little Book Of Happiness,*

*"BET Journeys in Black: Louis Farrakhan"] AND product.titulo CONTAINS "" RETURN DISTINCT product LIMIT 200*

Figura 51. Query del algoritmo de similaridad.

Recommendations

asin	avg_rating	categories_n	downloaded	id	reviews_total	salesrank	titulo
0553278223	4	4	268	1212	268	3206	The Martian Chronicles
0393308014	4	4	3	1213	3	151367	The Tactics of Small Boat Racing
0471395706	4	10	11	1214	11	63874	New Sales Speak
B00000ATA4	5	2	6	1215	6	17031	Vier Gewinn
0801862515	5	5	2	1494	2	190163	Walker's Primates of the World
0373122179	5	3	2	1495	2	434711	Rafaello'S Mistress (Harlequin Presents. No. 2217)
0395901294	4	2	2	1496	2	882768	The Store of a Million Items: Stories
6303975704	4	1	5	2134	5	30499	Voices of the Doumbek

8 - rows visible

Figura 52. Recomendaciones por producto.

Recommendations

id	nombre
283155	Books(283155)
139452	[139452]

Figura 53. Recomendaciones por grupo.

Recommendations

id
ATVPDKIKX0DER
A2NJO6YE9S4DBH
A2F1X6YFQJ21FH
A1OZQCZAK21S6M
AL5DS2NABF67F
AVFBIM1W41DXO
A316SOXDIE0M8R
A3559TE3P9RRNL

8 - rows visible

Figura 54. Recomendaciones por cliente.

Se tiene la opción de mostrar el grafo formado de recomendaciones, se usó la librería de *neovis* para su creación. Figura 55. Como podemos observar pasando el cursor sobre los nodos se muestra información, en el ejemplo se muestra el nodo de un producto recomendado que es: *The Martian Chronicles*; dicho producto se encuentra listado en la previa Figura 52 “recomendaciones por producto”.

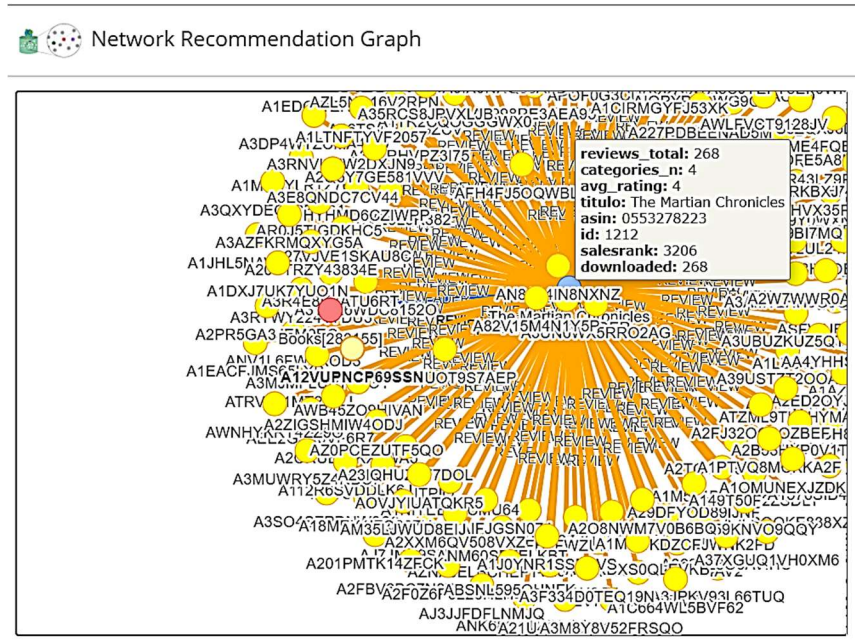


Figura 55. Grafo de recomendaciones por producto.

Si se selecciona *All Groups* o *Relevant Groups* se desplegará el menú con entradas de búsqueda. Figura 56. Se tiene la opción de hacer búsquedas por *Group ID* y *Group Name*.

Search

Group

---

Group Id

Group Name

Figura 56. Menú de grupos.

En el caso de *All Groups* se ejecuta el *query* de la Figura 57.

```
MATCH (n:GROUPS) WHERE n.titulo CONTAINS "" RETURN n
```

Figura 57. Query de todos los grupos.

Para *Relevant Groups* se ejecuta el *query* de la Figura 58.

```
CALL algo.pageRank.stream(GROUP,null,{iterations:20,dampingFactor:0.85})YIELD nodeId,score
WHERE algo.getNodeById(nodeId).titulo CONTAINS
RETURN algo.getNodeById(nodeId) AS page ORDER BY score DESC LIMIT 500;
```

Figura 58. *Query* de grupos relevantes.

Se puede observar la aplicación del algoritmo de *PageRank* en el caso de grupos relevantes. En la Figura 59 tenemos la salida del *query* de *All Groups* y en la Figura 60 *Relevant Groups*.

id	nombre	Recommendation
10	Health, Mind & Body[10]	★★★★★
282800	Personal Health[282800]	★★★★★
282840	Aging[282840]	★★★★★
4655	General[4655]	★★★★★
283155	Books[283155]	★★★★★
1000	Subjects[1000]	★★★★★
4	Children's Books[4]	★★★★★
2787	Animals[2787]	★★★★★

8 rows visible

1 2 3 4 5 ... 69

Figura 59. Búsqueda de todos los grupos.

id	nombre	Recommendation
283155	Books[283155]	★★★★★
1000	Subjects[1000]	★★★★★
504358	Formats[504358]	★★★★★
139452	[139452]	★★★★★
504360	Specialty Stores[504360]	★★★★★
5174	Music[5174]	★★★★★
130	DVD[130]	★★★★★
404272	VHS[404272]	★★★★★

8 rows visible

1 2 3 4 5 ... 69

Figura 60. Búsqueda de grupos relevantes.

Se filtró grupos que contengan ID “130” donde se obtuvo un filtro tal como se muestra en la Figura 61.

Group

Group Id

Group Name

Search

id	nombre	Recommendation
130	DVD[130]	
513060	Traditional Vocal Pop[513060]	
513062	General[513062]	
281300	Anime & Manga[281300]	
11300	African-American Studies[11300]	
11302	Ethnic Studies[11302]	
17130	Central America[17130]	
4130	Windows - General[4130]	

8 rows visible

Figura 61. Filtrado de grupos con *Group ID*.

Se pueden hacer filtros simultáneos con *Group ID* y *Group name*; se generó un filtro de id contenga 130 y nombre *African-American Studies*. Figura 62.

Group

Group Id

Group Name

Search

id	nombre	Recommendation
11300	African-American Studies[11300]	

Figura 62. Filtrado de grupos con *Group ID* y *Group name*.

En el apartado de filtrado se tiene la opción de hacer recomendaciones de cierto grupo por productos/grupos/clientes recomendados; de la tabla de la Figura 61 se eligió “DVD”, generándose una recomendación de productos como se muestra en la Figura 65, de grupos Figura 66 y clientes Figura 67. A continuación, se enlistan los *queries* ejecutados generando las recomendaciones acordes al producto seleccionado se muestran en la Figura 63 y Figura 64.

```
CALL algo.louvain.stream(PRODUCT,null)YIELD nodeId,community
WHERE algo.getNodeById(nodeId).titulo
CONTAINS The Wounded Buzzard on Christmas Eve (Hank the Cowdog, 13)
RETURN DISTINCT algo.getNodeById(nodeId) AS node,community LIMIT 200
```

Figura 63. *Query* de producto usando el algoritmo de *Louvian*.

El query de similar toma la salida de *Louvian*, dependiendo si se selecciona producto, grupo o cliente, se ejecutará el *return* correspondiente. Figura 64.

```

MATCH (customer: CUSTOMER) – [review: REVIEW] → (product: PRODUCT) – [header: HEADER]
→ (group: GROUP) WHERE product.titulo IN
["DC Talk: Free at Last – The Movie", "The Whimsical Tarot: A Deck for Children and the Young at Heart",
Help! I'm Laughing And I Can't Get Up Fall-down Funny Stories To Fill Your Heart And Lift Your Spirit,
The Book of Shaker Furniture, Polio (Epidemics),
Modern Oral And Tibetan: A Language Primer (Studies in Linguistics and Semiotics, 5),
New Haven Negroes: A Social History, Manners By Michele, Volume 4; How to Set the Table,
The Fearless Executive, The Shih Tzu : An Owner's Guide to a Happy Healthy Pet (Happy Healthy Pet),
The Art & Craft of Handmade Books: New Ideas and Innovative Techniques,
Working the Angles: The Shape of Pastoral Integrity,
French Chefs Cooking: Recipes and Stories from the Great Chefs of France,
1000 Years of Famous People, Faure: Violin Sonatas Op.13 & Op.108/Franck: Violin Sonata in a,
End of Time: A Meditation on the Philosophy of History, Whistle Stop,
Run To Cadence W/ The U.S. Army Airborne Vol. 2, The Assassin's Handbook,
"The Sun Is Always Shining Somewhere (Rookie Read – About Science Series)"]
AND product.titulo CONTAINS "" RETURN DISTINCT product LIMIT 200

```

Figura 64. Query de producto usando el algoritmo de *Louvian*.

## Recommendations

asin	avg_rating	categories_n	downloaded	id	reviews_total	salesrank	titulo
B00007L4N1	5	8	6	211	6	46276	DC Talk: Free at Last - The Movie
1572813156	5	3	5	1340	5	419516	The Whimsical Tarot: A Deck for Children and the Young at Heart
0785276149	5	3	2	2054	2	59924	Help! I'm Laughing And I Can't Get Up Fall-down Funny Stories To Fill Your Heart And Lift Your Spirit
0870232754	4	3	4	2055	4	148486	The Book of Shaker Furniture
0814405657	4	6	8	858	8	186729	The Fearless Executive
0876053886	4	2	5	859	5	140352	The Shih Tzu : An Owner's Guide to a Happy Healthy Pet (Happy Healthy Pet)
1579904386	5	3	5	6058	5	129709	The Art & Craft of Handmade Books: New Ideas and Innovative Techniques
0802802656	4	1	10	6059	10	34748	Working the Angles: The Shape of Pastoral Integrity

8 rows visible

Figura 65. Recomendaciones por producto.

## Recommendations

id	nombre
283155	Books[283155]
5174	Music[5174]
139452	[139452]

Figura 66. Recomendaciones por grupo.



Si se selecciona *All Customers* o *Relevant Customers* se desplegará el menú con entradas de búsqueda. Figura 69. Se tiene la opción de hacer búsquedas por *Customer ID*.

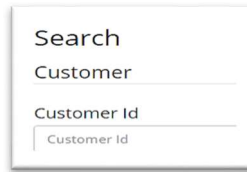


Figura 69. Menú de clientes.

En el caso de *All Customers* se ejecuta el *query* de la Figura 70.

```
MATCH (n: CUSTOMER) WHERE n.titulo CONTAINS "" RETURN n
```

Figura 70. *Query* de todos los clientes.

En el caso de *Relevant Customers* se ejecuta el *query* de la Figura 71.

```
CALL algo.pageRank.stream(CUSTOMER, null, {iterations: 20, dampingFactor: 0.85})YIELD nodeId, score
WHERE algo.getNodeById(nodeId).titulo CONTAINS
RETURN algo.getNodeById(nodeId) AS page ORDER BY score DESC LIMIT 500;

CALL algo.pageRank.stream(CUSTOMER, null, {iterations: 20, dampingFactor: 0.85})YIELD nodeId, score
WHERE algo.getNodeById(nodeId).titulo CONTAINS
RETURN algo.getNodeById(nodeId) AS page ORDER BY score DESC LIMIT 500;
```

Figura 71. *Query* de clientes relevantes.

Se puede observar la aplicación del algoritmo de *PageRank* en el caso de grupos relevantes. En la Figura 72 tenemos la salida del *query* de *All Customers* y en la Figura 73 *Relevant Customers*.

id	Recommendation
A2JW67OY8U6HHK	
A2VE83MZP98ITY	
A11NCO6YTE4BTJ	
A9CQ3PLRNIR83	
A13SG9ACZ9O5IM	
A1BDAI6VEYMAZA	
A2P6KAWXJ16234	
AMACWC3M7PQFR	

8 rows visible

Figura 72. Búsqueda de *All Customers*.



id	Recommendation
A2VE83MZFP98ITY	
A11NCO6YTE4BTJ	
A9CQ3PLRNIR83	
A13SG9ACZ9O5IM	
A1BDAI6VEYMAZA	
A2P6KAWXJ16234	
AMACWC3M7PQFR	
A3GO7UV9XX14D8	

8 rows visible

1 2 3 4 5 ... 63

Figura 73. Búsqueda de *Relevant Customers*.

Si se necesita hacer un filtrado por *Customer ID*, se puede lograr con frases del ID y nos arrojará las ocurrencias con dicha frase. En la figura 74 se hizo una búsqueda con “MAZA” arrojándonos solo una ocurrencia.

Search  
Customer

Customer Id  
MAZA

id	Recommendation
A1BDAI6VEYMAZA	

Figura 74. Filtrado de *Customer ID*.

En la tabla de filtrado se tiene la opción de hacer recomendaciones de cierto grupo por productos/grupos/clientes recomendados; de la tabla de la Figura 83 se eligió “A1HPJKECRYBG6V”, generándose una recomendación de productos como se muestra en la Figura 77, de grupos Figura 78 y clientes Figura 79. A continuación, se enlistan los *queries* ejecutados generando las recomendaciones acordes al producto seleccionado en la Figura 75 y Figura 76.

```
CALL algo.louvain.stream(PRODUCT,null)YIELD nodeId,community
WHERE algo.getNodeById(nodeId).titulo CONTAINS Welcome to Jatorassic Park: A FoxTrot Collection
RETURN DISTINCT algo.getNodeById(nodeId) AS node,community LIMIT 200
```

Figura 75. *Query* de *Louvian* para clientes.

El query de similar toma la salida de *Louvian*, dependiendo si se selecciona producto, grupo o cliente, se ejecutará el *return* correspondiente. Figura 76.

```
MATCH (customer: CUSTOMER) – [review: REVIEW] → (product: PRODUCT) –
[header: HEADER] → (group: GROUP) WHERE product.titulo IN ["Invitation to Theology",
No One Can Hear You, Knowledge Management in Education: Enhancing Learning & Education,
DK Nature Encyclopedia, DNA and the I Ching: The Tao of Life, Top Percussion,
15 Grandes Exitos de Nicola Di Bari en Espanol, Hug Me, Reflexology: The Timeless Art of Healing,
Real Ghostbusters vol. 6: Revenge of Murray the Mantis,
High Performance Cluster Computing: Architectures and Systems, Black Lightning, Pumpkinhead,
Every Eye Beholds You: A World Treasury of Prayer, Bees of the World (Of the World Series),
The I-5 Killer (Signet True Crime S.), Coastal Fish Identification: California to Alaska,
Learn Library of Congress Classification,
Victorian Dwellings for Village and Country (1885) (Dover Books on Architecture),
"Financial Success : Harnessing the Power of Creative Thought"] AND product.titulo
CONTAINS "" RETURN DISTINCT product LIMIT 200
```

Figura 76. Query de similar para clientes.

#### Recommendations

asin	avg_rating	categories_n	downloaded	id	reviews_total	salesrank	titulo
0830815627	5	1	1	1000	1	497795	Invitation to Theology
B000063JZW	2	15	2	1002	2	40410	No One Can Hear You
0789434113	5	5	3	1505	3	24627	DK Nature Encyclopedia
B000005LFE	4	3	8	2091	8	61430	Top Percussion
B000005LFG	4	2	4	2092	4	19519	15 Grandes Exitos de Nicola Di Bari en Espanol
0060293179	5	6	2	2297	2	910298	Hug Me
6304945175	5	2	4	2334	4	2499	Reflexology: The Timeless Art of Healing
6302824990	5	6	7	2335	7	27017	Real Ghostbusters vol. 6: Revenge of Murray the Mantis

8 rows visible

Figura 77. Recomendaciones por producto.

#### Recommendations

id	nombre
283155	Books[283155]
5174	Music[5174]
139452	[139452]

Figura 78. Recomendaciones por grupo.

## Recommendations

id
ATVPDKIKX0DER
A3UN6WX5RRO2AG
AS29VX05GNSWH
A2S166WSCFIFP5
A2HWD9PTM7R8XN
A23IOP10D1OG7V
A1CZICCP2M5PX
ADLN37B0153N

8 rows visible

Figura 79. Recomendaciones por cliente.

El grafo de recomendaciones generado para cliente es mostrado en la Figura 80.

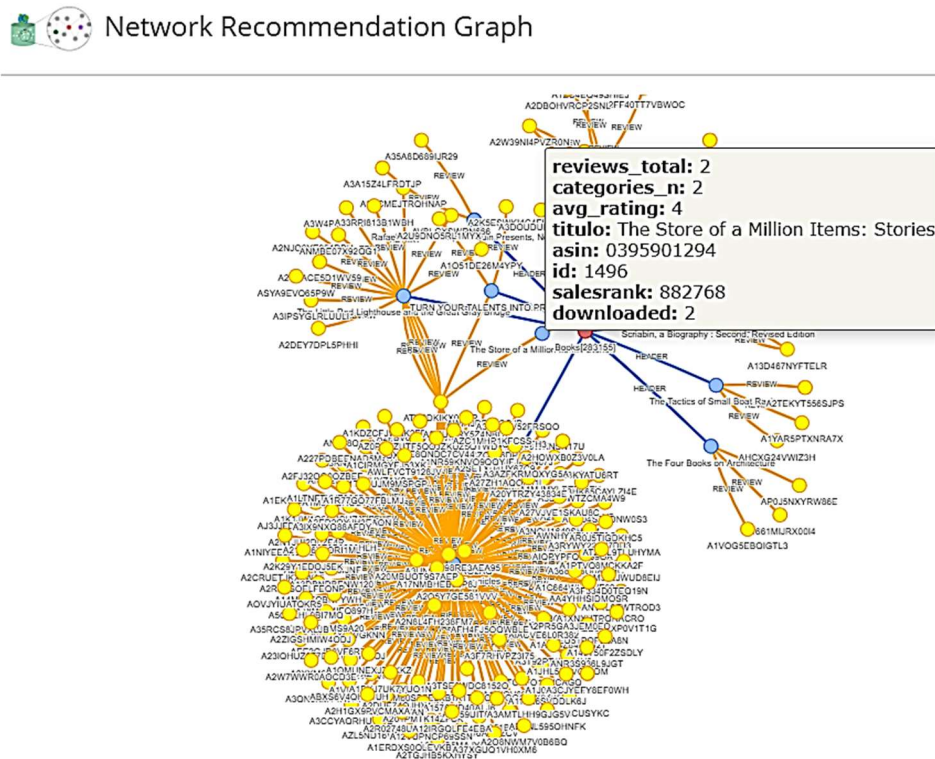


Figura 80. Grafo generado para una recomendación de cliente.

Si se selecciona *By Reviews* se desplegará el menú con entradas de búsqueda. Figura 81. Se tiene las opciones de hacer búsquedas por *Ratings*, *Votes*, *Helpful*; elección por: *Product*, *Customer* y *Group*; elección por año, mes y día.

Figura 81. Menú de reseñas.

Query ejecutado de la ejecución del filtro de reseñas de la Figura 82.

```
MATCH (customer: CUSTOMER) - [r: REVIEW] -> (product: PRODUCT) -
[: HEADER] -> (group: GROUP) where (toInteger(r.year) = 2005)AND
(toInteger(r.helpful) = 5)AND (toInteger(r.votes) = 5)AND
(toInteger(r.ratings) = 5)AND product.titulo CONTAINS
RETURN DISTINCT product, r.year as date, r.helpful, r.votes, r.ratings LIMIT 500
```

Figura 82. Query de reseñas.

La búsqueda por defecto de todos los productos nos enlista información tal como se muestra en la Figura 83. Se tiene predeterminado los valores de *Ratings*, *Votes*, *Helpful* a 5, búsqueda por producto y año 2005.

asin	avg_rating	categories_n	downloaded	helpful	id	ratings	reviews_total	salesrank	titulo	votes	year	Recommendation
0689834705	4	6	36	5	1165	5	36	110574	Counting Kisses	5	2005	★★★★★
B00005LL26	4	18	260	5	1312	5	261	443	Willy Wonka & The Chocolate Factory (30th Anniversary Edition)	5	2005	★★★★★
1573229571	4	3	290	5	1318	5	292	494999	About a Boy (Movie Tie-In)	5	2005	★★★★★
B00004RF55	4	22	38	5	2149	5	38	39382	The Drive	5	2005	★★★★★
0393308278	5	3	6	5	2274	5	6	139450	Wars of Watergate: The Last Crisis of Richard Nixon	5	2005	★★★★★
0374199698	4	2	510	5	2341	5	517	14033	Middlesex: A Novel	5	2005	★★★★★
0688163173	4	5	324	5	3011	5	324	47991	Shutter Island: A Novel	5	2005	★★★★★
6302799139	4	20	225	5	3111	5	225	9534	1776	5	2005	★★★★★

Figura 83. Búsqueda de reseñas por defecto.

En la Figura 84 se hace una búsqueda por cliente con *ratings*, *votes*, *helpful* 5 y del año 2005, seleccionando un caso de uso genérico. Se puede observar una lista de clientes recomendados por los *ratings*, *votes*, *helpful* esperados, tenemos en la columna de recomendación por producto (icono azul), por clientes (icono amarillo) y por grupos (icono rojo). Tenemos la opción de elegir una recomendación acorde a nuestra necesidad.

Review

Choose option: CUSTOMER ▾

Ratings

5

Votes

5

Helpful

5

Choose date: year ▾

2005

Search

























helpful	id	ratings	votes	year	Recommendation
5	A3MSB482DIB95L	5	5	2005	  
5	A1Y87E22UVYCE6	5	5	2005	  
5	ASZVJ9A6H1YL1	5	5	2005	  
5	A1YHHM3QFKCBCK	5	5	2005	  
5	A3QZCA4LTTVGAD	5	5	2005	  
5	AJQ1539GZBKUG	5	5	2005	  
5	AXQ8T2D9IT6UG	5	5	2005	  
5	A170QV8HVJ3CMT	5	5	2005	  

Figura 84. Búsqueda de reseñas por cliente.

En la Figura 85 se hace la búsqueda con *ratings* 5, *votes* y *helpful* 6 y año 2004 con productos.

Choose option: PRODUCT ▾

Ratings

5

Votes

6

Helpful

6

Choose date: year ▾

2004

Search

























asin	avg_rating	categories_n	downloaded	helpful	id	ratings	reviews_total	salesrank	titulo	votes	year	Recommendation
B000056PNB	4	15	24	6	42	5	24	19685	Pixote	6	2004	  
0140445684	4	3	40	6	120	5	40	17693	Capital : A Critique of Political Economy (Penguin Classics)	6	2004	  
0883680947	4	2	39	6	198	5	39	933676	Imitation of Christ	6	2004	  
0875421210	4	6	47	6	265	5	47	100261	Earth Power: Techniques of Natural Magic (Llewellyn's Practical Magick)	6	2004	  
6304702329	4	6	178	6	306	5	178	69930	Before Sunrise	6	2004	  
B00006LU75	3	9	5	6	312	5	5	18879	The Cat - Gogen Yamaguchi, 10th Dan	6	2004	  
0840733208	4	5	14	6	335	5	14	7013	Getting Ready for Marriage Workbook : How to Really Get to Know the Person You're Going to Marry	6	2004	  
B0000296JC	4	6	544	6	459	5	545	301940	Make Yourself	6	2004	  

Figura 85. Búsqueda de reseñas por producto.

En la Figura 86 se muestra una búsqueda por mes igual a 12, dejando los demás valores por default.

Choose date: month ▾

12

Search

asin	avg_rating	categories_n	downloaded	helpful	id	month	ratings	reviews_total	salesrank	titulo	votes	Recommendation
0385504209	3	4	3049	6	296	12	5	3049	19	The Da Vinci Code	6	👍👍👍
0195039343	5	6	16	6	307	12	5	16	44292	Manic-Depressive Illness	6	👍👍👍
0586041559	5	2	2	6	400	12	5	2	708756	Wellington: Pillar of State	6	👍👍👍
0486284999	4	6	62	6	439	12	5	62	105471	Narrative of the Life of Frederick Douglass (Dover Thrift Editions)	6	👍👍👍
0679444653	4	2	424	6	441	12	5	424	225347	One Hundred Years of Solitude (Everyman's Library (Cloth))	6	👍👍👍
B0000296JO	4	5	16	6	456	12	5	16	76162	Made in Japan	6	👍👍👍
0970430353	4	3	8	6	1087	12	5	8	533414	Illustration For Designers - Second Edition	6	👍👍👍
0893145467	5	4	41	6	1113	12	5	41	148588	The Secret Teachings of All Ages : An Encyclopedic Outline of Masonic, Hermetic, Qabbalistic, and Rosicrucian Symbolical Philosophy	6	👍👍👍

Figura 86. Búsqueda de reseñas por mes.

En la Figura 87 se muestra una búsqueda por día igual a 30.

Choose date: day ▾

30

Search

asin	avg_rating	categories_n	day	downloaded	helpful	id	ratings	reviews_total	salesrank	titulo	votes	Recommendation
B000056PNB	4	15	30	24	6	42	5	24	19685	Pixote	6	👍👍👍
0895296586	3	5	30	12	6	164	5	12	224993	What's Really Wrong With You: A Revolutionary Look at How Muscles Affect Your Health	6	👍👍👍
0875421210	4	6	30	47	6	265	5	47	100261	Earth Power: Techniques of Natural Magic (Llewellyn's Practical Magick)	6	👍👍👍
B000002KYA	4	8	30	10	6	1174	5	10	30784	Keep On Doing	6	👍👍👍
0671047558	4	8	30	11	6	1336	5	11	630807	Tenderheaded : A Comb-Bending Collection of Hair Stories	6	👍👍👍
B00000IP34	4	10	30	18	6	2036	5	18	37644	Bridges	6	👍👍👍
B000005Q4JH	4	1	30	826	6	3510	5	826	6402	Invincible	6	👍👍👍
B00000INB1	4	4	30	33	6	865	5	33	4424	Karen Voight - Energy Sprint	6	👍👍👍

Figura 87. Búsqueda de reseñas por día.

Recomendaciones por el producto *Bridges* listado en la Figura 86 el cual cuenta con *votes*, *helpful* igual a 6 y un *salesrank* igual a 37644, nos arroja recomendaciones mostradas en la Figura 88. Tenemos una lista de productos los cuales se muestra interconectados en la Figura 88. Con esta demostración nos damos cuenta la utilidad de nuestro recomendador de productos acorde a la elección de productos, grupos o clientes.

asin	avg_rating	categories_n	downloaded	id	reviews_total	salesrank	titulo
0788822721	5	19	15	2439	15	4444	Winnie the Pooh and the Blustery Day
074344602X	4	3	7	2441	7	55094	Killing Blow: Errand of Vengeance Book Two (Star Trek The Original Series)
0786867558	5	1	28	3231	28	18512	The Naked Chef Takes Off
0820316636	3	2	14	3232	14	183838	God's Little Acre
087628876X	5	4	2	3233	2	184301	The Special Education Teacher's Book of Lists
1558604561	5	9	2	3234	2	205516	SQL: 1999 - Understanding Relational Language Components (Morgan Kaufmann Series in Data Management Systems)
1584650788	4	5	1	3316	1	181488	Hands-On Nature: Information and Activities for Exploring the Environment With Children
0375800050	4	4	1	3318	1	710560	Learning Word Sounds: Kindergarten (Star Wars Fun-To-Learn Books)
0804724954	4	1	1	3338	1	1280986	Free Exchange
0804713855	5	3	3	3340	3	159530	Fishes of the Pacific Coast: Alaska to Peru, Including the Gulf of California and the Galapagos Islands
0863152376	5	3	2	3341	2	641065	Goethe on Science: A Selection of Goethe's Writings
1558195815	4	2	14	3350	14	797735	The Holman Pocket-Size Bible: Niv : Black Bonded Snap (International Version)

25 rows visible

Figura 88. Recomendación por filtrado de reseñas.

Se puede visualizar el grafo Figura 89, donde se tiene la opción de desplegar la descripción de cada nodo pasando por encima el ratón id de 3340 el cual es *Fishes of the Pacific Coast* listado en la Figura 88 como opción de recomendación; de la misma forma los demás productos forman parte del grafo generado.

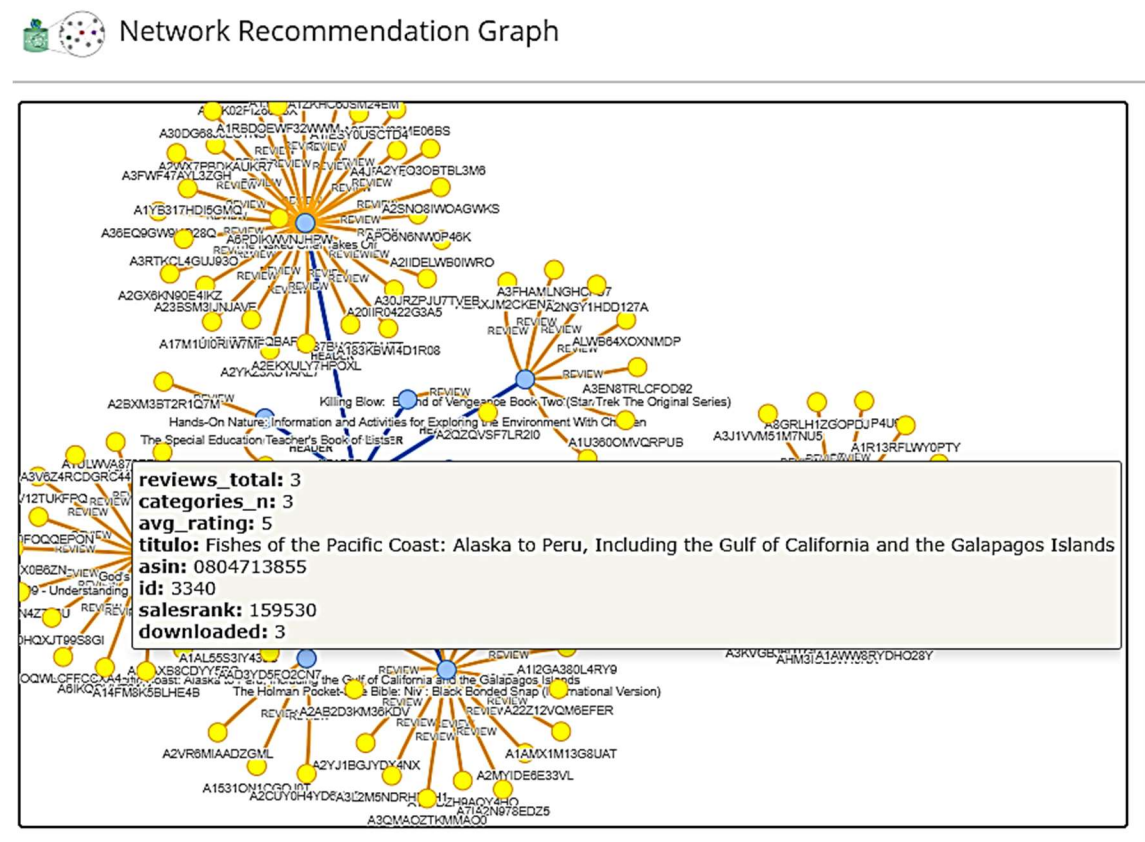


Figura 89. Grafo de recomendación por reseñas.

Del lado del *Backend* se utilizó *JavaScript* para generar las llamadas a la base de datos de NEO4J. Se descargó un paquete llamado *neo4j-driver* y se logró la conexión a la base de datos de NEO4J. En la Figura 90 se muestra el comando utilizado.

```
let driver = neo4j.driver(
  `bolt://${server}:${port}`, neo4j.auth.basic(user, password));
// Register a callback to know if driver creation was successful:
driver.onCompleted = function () {
  console.log('Connection Successful!');
};

// Register a callback to know if driver creation failed.
// This could happen due to wrong credentials or database unavailability:
driver.onError = function (error) {
  console.log('Driver instantiation failed', error);
};
```

Figura 90. Comando de llamadas a NEO4J.

## 5.2. Discusión

Existe un número indefinido de recomendadores, pero no con los objetivos planteados en este proyecto. Nuestro recomendador es una solución híbrida utilizando recomendaciones colaborativas y de grafos basándose principalmente en los productos que el usuario quiere comprar y eliminando de su alcance los poco probables, generando una agilidad en tiempo. Se hace uso de algoritmos de centralidad y similaridad valorando la votación de artículos ofreciendo una mayor precisión en el resultado.

Acorde a los recomendadores existentes con Xerox solo son cubiertas las reseñas de los usuarios. Con el sistema recomendador de productos se cubre también el caso de varios datos tales como: el número total de reseñas, tipo de categoría, el rango de ventas con algoritmos de comunidad. Con *Movie Lens* el usuario solo cubre género y popularidad. Con *Fab* se cubren filtros de búsqueda de usuarios. En el caso de *E-commerce*, nuestra solución brinda un filtro colaborativo como plus a dicho recomendador; con los sistemas de popularidad no tenemos la valoración del número total de reseñas tipos de categorías, y las ventas del producto. Con el recomendador de co-compra se abarca un extenso número de categorías previamente mencionadas en los capítulos 4 y 5 como son filtrado de productos, por cliente y grupo. Ninguna de las propuestas listadas en la Tabla 1 cumple con lo que hace nuestra herramienta de manera conjunta; las ventajas que posee nuestro recomendador se listaron en la Tabla 1 versus cada recomendador.



---

## 6. CONCLUSIONES

---

## 6.1. Conclusiones

La contribución principal del presente trabajo de fin de maestría es el desarrollo y verificación de un nuevo sistema de recomendación de productos de compra basado en grafos. Este sistema de recomendación toma en cuenta las reseñas de los clientes, los productos más comprados, para hacer recomendaciones.

Se pudo adquirir satisfactoriamente los sets de datos necesarios para la implementación del proyecto; tomados del repositorio de SNAP, donde luego se les aplicó un tratamiento para convertirlos en formato CSV y cargarlos en la base de datos de NEO4J.

En nuestro recomendador se buscar mejorar los siguientes aspectos:

- Generación de recomendaciones certeras para el usuario
- Uso amigable con la interfaz gráfica del sistema de recomendador
- Ofrecer una amplia gama de opciones de búsquedas para la obtención de una recomendación
- Proporcionar una respuesta rápida en la generación de consultas a la base de datos de NEO4J
- Aplicar algoritmos de búsqueda, ranking, comunidad

El usuario puede generar consultas por 3 categorías principales: por producto, cliente y grupo; de las cuales se pueden aplicar filtros exhaustivos para una recomendación óptima.

En resumen, se cumplió con el objetivo general y los objetivos particulares descritos en la introducción.

## 6.2. Trabajo Futuro

En el presente trabajo se enfocó el análisis en productos de Amazon; como trabajo a futuro se puede analizar el comportamiento con productos de compra de eBay u otras empresas de compra en línea. Comparar tiempos de respuesta entre los sets de datos a prueba y si existe alguna pérdida, mejorar la ejecución en tiempo y respuesta de dicho proyecto. Hacer una valoración experimental y robustez de la aplicación.

# BIBLIOGRAFÍA

- [1] Ms.Uma.N, "A Technical Review On Recommendation Systems In A Framework Of Job Recommender System Using Naive Bayes Algorithm," *Global Journal Of Engineering Science And Researches*, vol. 5, no. 6, pp. 253–257, Jun. 2018, doi: 10.5281/zenodo.1302263.
- [2] M. Balabanovic and Y. Shoham, "Fab: Content-Based, Collaborative Recommendation," *Communications of the ACM*, vol. 40, no. 1, pp. 66–72, Mar. 1997, doi: 10.1145/245108.245124.
- [3] "Recommender system and method - Xerox Corporation." <https://www.freepatentsonline.com/y2004/0254911.html> (accessed Jan. 16, 2021).
- [4] "MovieLens." <https://movielens.org/> (accessed Jan. 16, 2021).
- [5] Z. Huang, W. Chung, and H. Chen, "A graph model for E-commerce recommender systems," *Journal of the American Society for Information Science and Technology*, vol. 55, no. 3, pp. 259–274, Jun. 2004, doi: <https://doi.org/10.1002/asi.10372>.
- [6] H. Ahn, "Utilizing Popularity Characteristics for Product Recommendation," *Int. J. Electron. Commerce*, vol. 11, no. 2, pp. 59–80, Dec. 2006, doi: 10.2753/JEC1086-4415110203.
- [7] L. Iaquinta, M. de Gemmis, P. Lops, G. Semeraro, M. Filannino, and P. Molino, "Introducing Serendipity in a Content-Based Recommender System," in *2008 Eighth International Conference on Hybrid Intelligent Systems*, vol. 2, no. 4, pp. 168–173, Sep. 2008, doi: 10.1109/HIS.2008.25.
- [8] M. Tkalčić, M. Kunaver, J. Tasic, and A. Kosir, "Personality based user similarity measure for a collaborative recommender system," *5th Workshop on Emotion in Human-Computer Interaction-Real World Challenges*, vol. 5, no. 5, pp.20-49, Jan. 2009, doi: 10.1567/hcic,2008.11.
- [9] B. Heitmann and C. Hayes, "Using Linked Data to Build Open, Collaborative Recommender System," *AAAI Spring Symposium*, vol. 10, no. 3, pp.20-33, Oct.2010, doi: 10.2053/pbus.2009.10.
- [10] E. Lumer, "Social Graph Based Recommender," US20120001919A1, Jan. 05, 2012. Available: <https://patents.google.com/patent/US20120001919A1/en>
- [11] M. Tkalčić, A. Odic, A. Kosir, and J. Tasic, "Affective Labeling in a Content-Based Recommender System for Images," *IEEE Transactions on Multimedia*, vol. 15, no. 2, pp. 391–400, Feb. 2013, doi: 10.1109/TMM.2012.2229970.
- [12] X. Wang, Y. Zhang, and T. Yamasaki, "Earn More Social Attention: User Popularity Based Tag Recommendation System," in *Companion Proceedings of the Web Conference 2020*, New York, NY, USA, vol. 10, no. 3, pp. 212–216, Apr. 2020, doi: 10.1145/3366424.3383543.
- [13] X. Bai, M. Wang, I. Lee, Z. Yang, X. Kong, and F. Xia, "Scientific Paper Recommendation: A Survey," *IEEE Access*, vol. 7, no. 33, pp. 9324–9339, May. 2019, doi: 10.1109/ACCESS.2018.2890388.
- [14] A. Pujahari and V. Padmanabhan, "Group Recommender Systems: Combining User-User and Item-Item Collaborative Filtering Techniques," in *2015 International Conference on Information Technology (ICIT)*, vol. 10, no. 22, pp. 148–152, Dec. 2015, doi: 10.1109/ICIT.2015.36.
- [15] A. Mishra and S. Vishwakarma, "Analysis of TF-IDF Model and its Variant for Document Retrieval," in *2015 International Conference on Computational Intelligence and Communication Networks (CICN)*, vol. 10, no. 9, pp. 772–776, Dec. 2015, doi: 10.1109/CICN.2015.157.

- [16] Y. Wang, Z. Hong, and M. Shi, "Research on LDA Model Algorithm of News-oriented Web Crawler," in 2018 IEEE/ACIS 17th International Conference on Computer and Information Science (ICIS), vol. 2, no. 3, pp. 748–753, Jun. 2018, doi: 10.1109/ICIS.2018.8466502.
- [17] Y. Park, S. Park, S. Lee, and W. Jung, "Fast Collaborative Filtering with a k-nearest neighbor graph," in 2014 International Conference on Big Data and Smart Computing (BIGCOMP), vol. 3, no. 5, pp. 92–95, Jan. 2014, doi: 10.1109/BIGCOMP.2014.6741414.
- [18] I. Popescu, K. Portelli, C. Anagnostopoulos, and N. Ntarmos, "The case for graph-based recommendations," in 2017 IEEE International Conference on Big Data (Big Data), vol. 4, no. 3, pp. 4819–4821, Dec. 2017, doi: 10.1109/BigData.2017.8258553.
- [19] D. Le, "Random walk with restart: A powerful network propagation algorithm in Bioinformatics field," in 2017 4th NAFOSTED Conference on Information and Computer Science, vol. 10, no. 2, pp. 242–247, Nov. 2017, doi: 10.1109/NAFOSTED.2017.8108071.
- [20] A. Wijonarko, D. Nurjanah, and D. S. Kusumo, "Hybrid recommender system using random walk with restart for social tagging system," in 2017 International Conference on Data and Software Engineering (ICoDSE), vol. 11, no. 3, pp. 1–6, Nov. 2017, doi: 10.1109/ICoDSE.2017.8285875.
- [21] A. Tharwat, "Classification assessment methods," Applied Computing and Informatics, vol. 5, no. 11, pp. 55–60, Aug. 2018, doi: 10.1016/j.aci.2018.08.003.
- [22] W. K. Hauger and M. S. Olivier, "NoSQL databases: forensic attribution implications," SAIEE Africa Research Journal, vol. 109, no. 2, pp. 119–132, Jun. 2018, doi: 10.1019/Ha.re.2018.02.005.
- [23] "Bases de datos no relacionales | Bases de datos de gráficos | AWS," Amazon Web Services, Inc. <https://aws.amazon.com/es/nosql/> (accessed Oct. 26, 2019).
- [24] "1. Getting Started with Neo4j - Neo4j Cookbook." <https://learning.oreilly.com/library/view/neo4j-cookbook/9781783287253/ch01.html> (accessed May 07, 2020).
- [25] "neo4j-contrib/neo4j.js," Neo4j Contrib, 2020. <https://github.com/neo4j-contrib/neo4j.js> (accessed Dec. 31, 2020)
- [26] "Graph Algorithms." <https://learning.oreilly.com/library/view/graph-algorithms/9781492047674/> (accessed Dec. 31, 2020).
- [27] "Chapter 1 Introduction to Text Analytics - Text Mining and Analysis." [https://learning.oreilly.com/library/view/text-mining-and/9781612907871/xhtml/Chapter\\_1.xhtml](https://learning.oreilly.com/library/view/text-mining-and/9781612907871/xhtml/Chapter_1.xhtml) (accessed Jan. 21, 2021).
- [28] "What is Tokenization | Tokenization In NLP," Analytics Vidhya, <https://www.analyticsvidhya.com/blog/2020/05/what-is-tokenization-nlp/> (accessed Jan. 17, 2021).
- [29] "Text Normalization. Why, what and how. | by Tiago Duque | Towards Data Science." <https://towardsdatascience.com/text-normalization-7ecc8e084e31> (accessed Jan. 17, 2021).
- [30] "¿Que es la programación web?," Open Analytics - SEO y Programación, <https://openanalytics.es/programacion-web/> (accessed Jan. 17, 2021).
- [31] H. M. Abdullah and A. M. Zeki, "Frontend and Backend Web Technologies in Social Networking Sites: Facebook as an Example," in 2014 3rd International Conference on Advanced Computer Science Applications and Technologies, vol. 12, no. 23, pp. 85–89, Dec. 2014, doi: 10.1109/ACSAT.2014.22.
- [32] "JSON: ¿Qué es y para qué sirve?," NextU LATAM, <https://www.nextu.com/blog/que-es-json/> (accessed Jan. 17, 2021).
- [33] "Stanford Large Network Dataset Collection." <https://snap.stanford.edu/data/#web> (accessed Jan. 02, 2021).

- [34] "Amazon.es: Acerca de Amazon." <https://www.amazon.es/Acerca-Amazon-Descubre-Nuestra-Empresa-Nuestra-Tecnologia/b?ie=UTF8&node=1323175031> (accessed Jan. 01, 2021).
- [35] "Neo4j Desktop Download - Launch and Manage Neo4j Databases," Neo4j Graph Database Platform. <https://neo4j.com/download/> (accessed Jan. 03, 2021).
- [36] "Neo4j Desktop User Interface Guide - Developer Guides," Neo4j Graph Database Platform. <https://neo4j.com/developer/neo4j-desktop/> (accessed Jan. 03, 2021).
- [37] "Neo4j APOC Library - Developer Guides," Neo4j Graph Database Platform. <https://neo4j.com/developer/neo4j-apoc/> (accessed Jan. 03, 2021).
- [38] "SNAP: Network datasets: Amazon co-purchasing network metadata." <https://snap.stanford.edu/data/amazon-meta.html> (accessed Jan. 02, 2021).
- [39] "Documentation for Visual Studio Code." <https://code.visualstudio.com/docs> (accessed Jan. 03, 2021).
- [40] "Node.js Complete Reference Guide - Node.js Complete Reference Guide." <https://learning.oreilly.com/library/view/nodejs-complete-reference/9781789952117/cover.xhtml> (accessed Mar. 15, 2020).

## APÉNDICE A.

### Instalación NEO4J

Descargar la última versión disponible en la página oficial de NEO4J. Figura 91 [35].

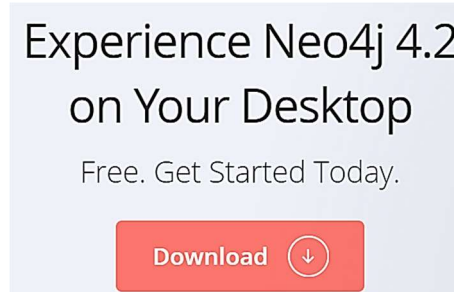


Figura 91. Descarga NEO4J [35].

Si abre NEO4J *Desktop* por primera vez, debe solicitarle que registre el software con una clave de activación. Esta clave de activación se genera cuando descarga por primera vez NEO4J *Desktop* y se mostrará en la página de confirmación de descarga. Guárdelo en un lugar seguro [36].

Copie esta clave de activación en el formulario de registro del software y haga clic en activar para continuar Figura 92 [36].

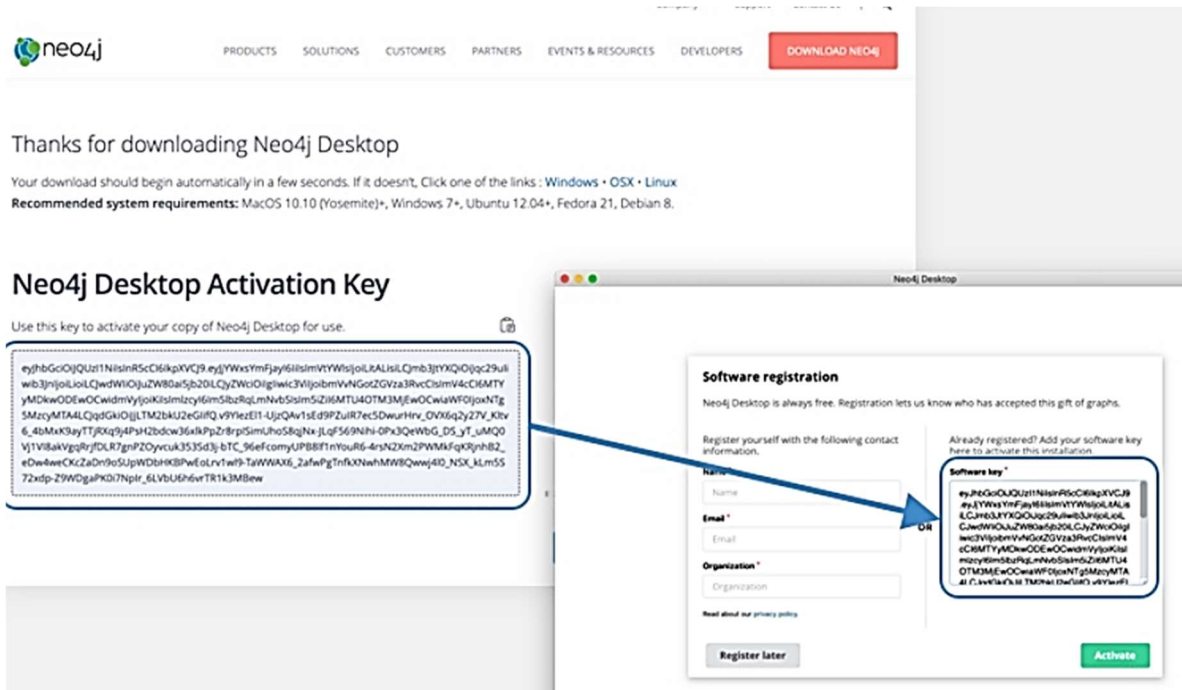


Figura 92. Clave de activación NEO4J [36].

## Navegación

El menú a lo largo del lado izquierdo de NEO4J *Desktop* muestra íconos de alto nivel para proyectos, aplicaciones de grafos, configuraciones, claves de activación e información de NEO4J. En la parte superior, la carpeta con un marcador (icono del proyecto de escritorio de NEO4J) muestra la lista de proyectos que ha creado. Figura 93 [36].

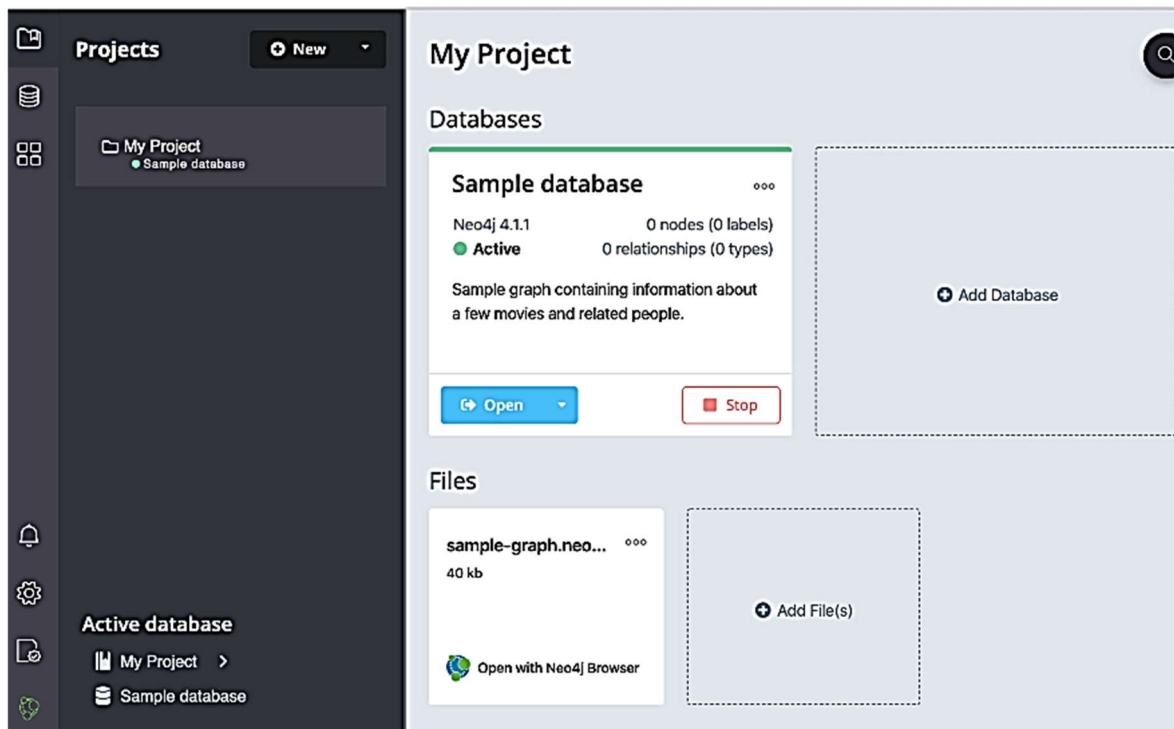


Figura 93. Panel principal Neo4j [36].

## Crear y eliminar un proyecto

Para crear un nuevo proyecto, simplemente haga clic en “nuevo” a la derecha del menú proyectos. La aplicación agregará una fila en el menú para un proyecto llamado proyecto. Puede cambiar el nombre haciendo clic en el icono de lápiz que aparece cuando se desplaza sobre el nombre del proyecto en el panel derecho. Para eliminar el proyecto, coloque el cursor sobre la fila proyecto en el menú de proyectos y haga clic en el icono de la papelera que aparece en el lado derecho (icono de eliminación del escritorio de NEO4J) [36].

## Configuraciones

La configuración de la aplicación se puede modificar o ver haciendo clic en el engranaje hacia la parte inferior del menú del lado izquierdo en NEO4J *Desktop* (icono de configuración de escritorio de NEO4J). Puede modificar cualquiera de estos ajustes en cualquier momento. Figura 94 [36].

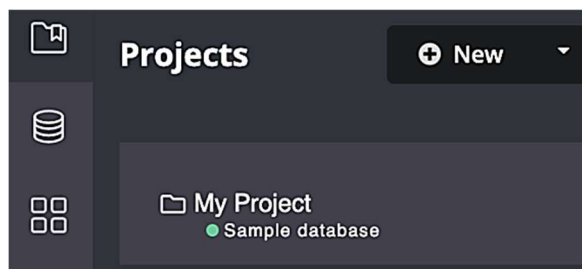


Figura 94. Crear nuevo proyecto en NEO4J [36].

El primer bloque es para la configuración de privacidad donde puede ajustar qué datos se envían a NEO4J para obtener asistencia y futuras mejoras. Las siguientes dos secciones son donde puede configurar un proxy o ejecutar NEO4J *Desktop* en modo fuera de línea [36, p. 4].

No se recomienda ejecutar NEO4J *Desktop* en modo fuera de línea a menos que lo requiera su empresa. El modo sin conexión restringe algunas capacidades de uso. En la sección distribuciones de NEO4J de la configuración, muestra qué versiones de la base de datos se utilizan en sus proyectos (si corresponde) y enumera los proyectos y bases de datos configurados actualmente. La última sección sobre herramientas de desarrollo es agregar e integrar aplicaciones de grafos con NEO4J. También puede acceder a la consola de una aplicación para depurar e informar de errores a través de las herramientas de desarrollo [36].

## Perfil y Acerca de

El icono de certificado (icono de claves de activación) muestra las licencias que se adjuntan a esta instalación en NEO4J *Desktop*, que incluye la activación de *desktop* junto con las licencias de la aplicación gráfica. El icono de la parte inferior (icono de NEO4J) es información sobre el propio NEO4J e incluye enlaces al sitio web principal de la empresa, así como información sobre la versión y la licencia [36].

## Crear una base de datos de grafos

Desde el icono proyectos en el menú del lado izquierdo, elija (o cree) un proyecto. Aparecerá un panel a la derecha para mostrar las bases de datos de su proyecto y los complementos disponibles o instalados. Para crear una base de datos, haga clic en el cuadrado punteado que dice agregar grafo. Luego haga clic en el botón crear un grafo local. Escriba un nombre para su base de datos y establezca una contraseña. También puede elegir una versión específica de NEO4J (la predeterminada es la última versión). Un pequeño icono junto a cada número de versión en el menú desplegable mostrará qué versiones ya se han descargado. Una vez que haya completado la información, haga clic en crear. Puede tomar unos minutos descargar la versión y crear la base de datos. Si olvida su contraseña, puede restablecerla haciendo clic en los tres puntos en la parte superior derecha de la tarjeta Base de datos, haciendo clic en administrar y navegando a la pestaña administración [36].





Figura 95. Inicializar base de datos en NEO4J [36].

### Instancia de sección db

El botón Iniciar en la esquina inferior derecha de cada tarjeta de base de datos iniciará la base de datos. Una vez iniciado, habrá un botón en la esquina inferior derecha para detener la base de datos. Una vez que se ha iniciado una base de datos, se habilitará el botón Abrir en la esquina inferior izquierda de la tarjeta. Al hacer clic en Abrir, se abrirá el navegador NEO4J y se conectará a la base de datos activa. Junto al botón Abrir hay un menú desplegable que le permitirá abrir una lista de aplicaciones gráficas [36].

### Administrar una base de datos

Para administrar una base de datos, haga clic en el icono de tres puntos en la parte superior derecha de la tarjeta y haga clic en administrar.

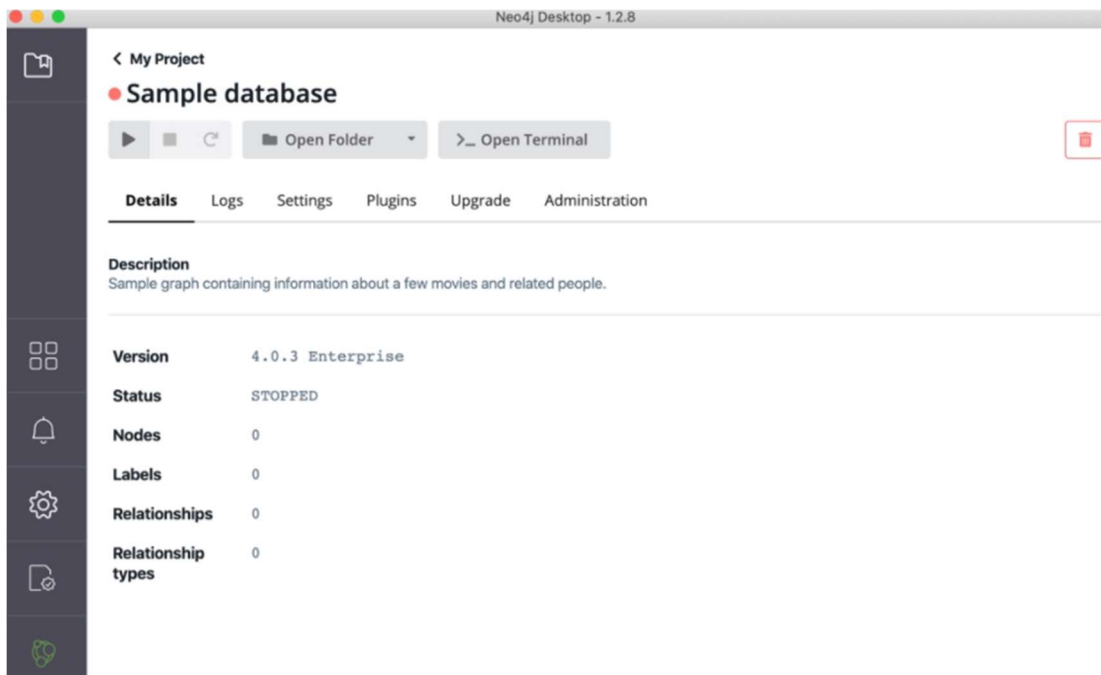


Figura 96. Administrar base de datos en NEO4J [36].

## **Administrar el panel de base de datos**

En la parte superior de la pantalla, verá el nombre de la base de datos y un icono para indicar su estado. Directamente debajo hay tres botones para iniciar, detener o reiniciar la base de datos [36].

### **Botones de estado**

El botón abrir carpeta abrirá la carpeta raíz de esta base de datos. El icono de flecha a la derecha de este botón le permitirá abrir carpetas adicionales relacionadas con esta base de datos; por ejemplo, las carpetas de complementos o registros [36].

### **Carpeta abierta**

El botón Abrir terminal abrirá una nueva ventana de terminal en la carpeta raíz de la base de datos. Debajo de estos botones hay varias pestañas para obtener información y configuraciones. La pestaña detalles muestra la descripción, la versión y el estado de su base de datos junto con algunas estadísticas de la base de datos. Cuando la base de datos se está ejecutando, también mostrará los números de puerto y las direcciones necesarias para interactuar con la base de datos [36].

### **Detalles de la base de datos**

La pestaña registros mostrará toda la salida del registro de transmisión de la base de datos. La pestaña configuración muestra los valores de configuración de la base de datos. Estos se pueden cambiar, si es necesario. Una vez realizados los cambios, puede aplicarlos y *desktop* le ofrecerá reiniciar la base de datos (necesario para que los cambios surtan efecto) [36].

## APÉNDICE B.

### Instalación APOC

Se cree que la biblioteca APOC es la biblioteca de extensión más grande y más utilizada para Neo4j. Incluye más de 450 procedimientos estándar que brindan funcionalidad para utilidades, conversiones, actualizaciones de grafo y más. Están bien soportados y son muy fáciles de ejecutar como funciones independientes o de incluir en consultas *Cypher*. Antes de comenzar a escribir funciones ad hoc para su aplicación, asegúrese de verificar APOC primero para ver si existe [37].

Hay dos formas de instalar APOC.

- 1) A nivel de proyecto: instale el complemento APOC para todas las bases de datos existentes en el proyecto elegido. Figura 97.

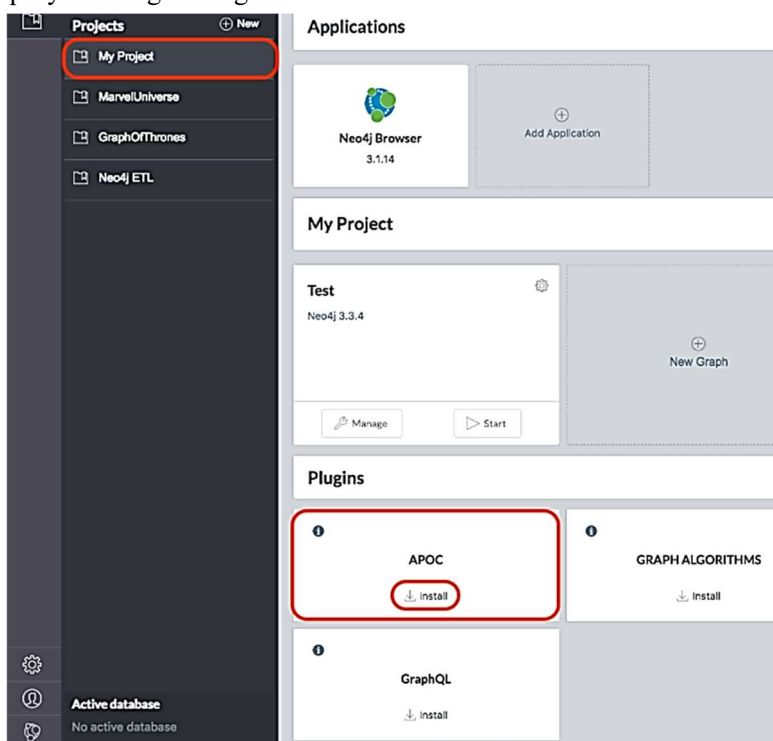


Figura 97. Instalar APOC a nivel proyecto [37].

- 2) En el nivel de la base de datos: instale el complemento APOC para una sola base de datos elegida. Figura 98.

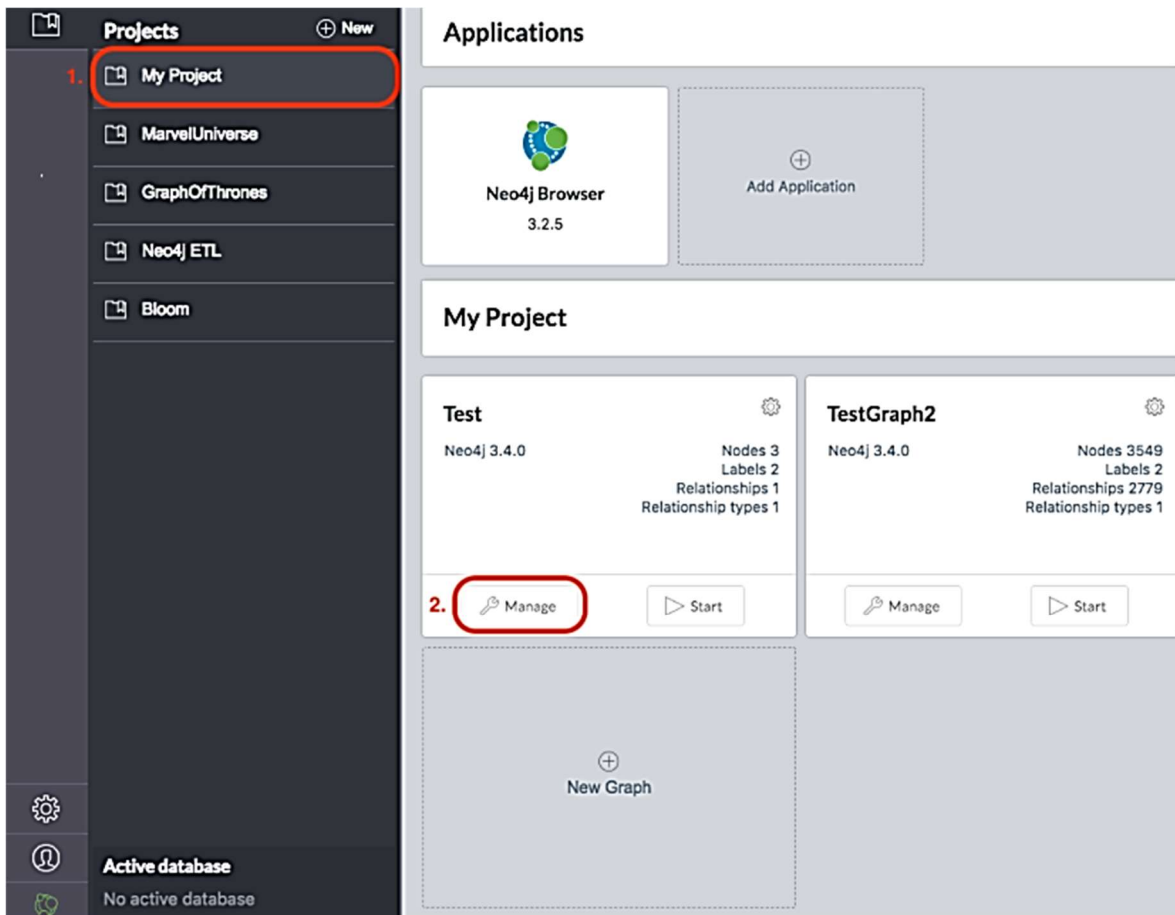


Figura 98. Instalar APOC a nivel base de datos paso 1 y 2 [37].

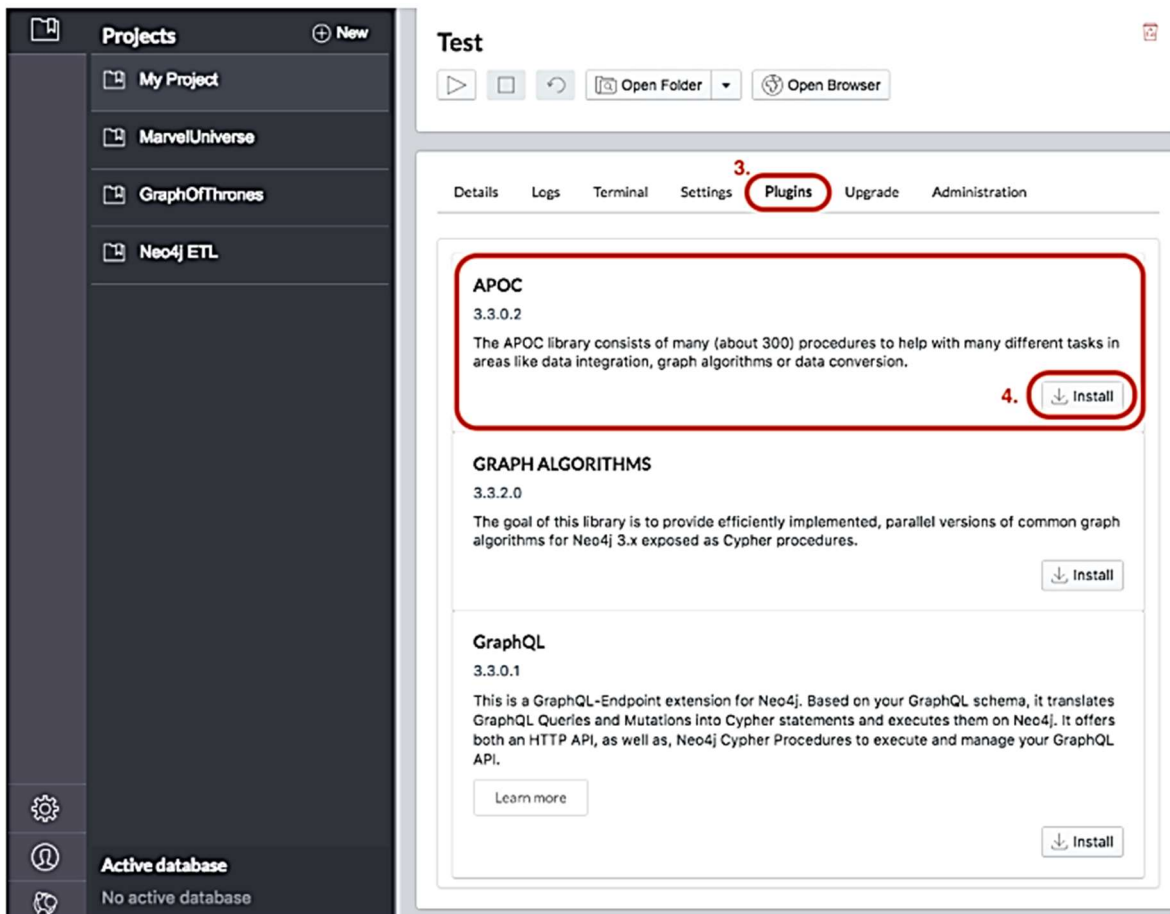


Figura 99. Instalar APOC a nivel base de datos paso 3 y 4 [37].

## APÉNDICE C.

### ***SNAP: Stanford Large Network Dataset Collection***

Una colección de más de 50 grandes conjuntos de datos de red desde decenas de miles de nodos y bordes hasta decenas de millones de nodos y bordes. Incluye redes sociales, grafos de web, redes de carreteras, redes de Internet, redes de citas, redes de colaboración y redes de comunicación [38].

- Redes sociales: redes sociales online, los bordes representan interacciones entre personas
- Redes con comunidades de verdad fundamental: comunidades de redes de verdad fundamental en redes sociales y de información
- Redes de comunicación: redes de comunicación por correo electrónico con bordes que representan la comunicación.
- Redes de citas: los nodos representan artículos, los bordes representan citas
- Redes de colaboración: los nodos representan científicos, los bordes representan colaboraciones (coautoría de un artículo)
- Grafos web: los nodos representan páginas web y los bordes son hipervínculos.
- Redes de Amazon: los nodos representan productos y los bordes vinculan productos comúnmente co-compra.
- Redes de Internet: los nodos representan computadoras y comunicación de bordes.
- Redes de carreteras: los nodos representan intersecciones y bordes de carreteras que conectan las intersecciones.
- Sistemas autónomos: grafos de internet
- Redes firmadas: redes con bordes positivos y negativos (amigo / enemigo, confianza / desconfianza)
- Redes sociales en línea basadas en la ubicación: redes sociales con registros geográficos
- Redes, artículos y metadatos de Wikipedia: charlas, edición, votación y datos de artículos de Wikipedia
- Redes temporales: redes en las que los bordes tienen marcas de tiempo
- *Twitter* y *Memetracker*: frases de *memetracker*, enlaces y 467 millones de tweets
- Comunidades en línea: datos de comunidades en línea como *Reddit* y *Flickr*
- Revisiones en línea: datos de sistemas de revisión en línea como *BeerAdvocate* y *Amazon*
- Acciones de usuario: acciones de usuarios en plataformas sociales.
- Redes de comunicación cara a cara: redes de interacciones cara a cara (no online)
- Conjuntos de datos de clasificación de grafos: grafos disjuntos de diferentes clases

Las redes SNAP también están disponibles en *Suite Sparse Matrix Collection* de Tim Davis.

## APÉNDICE D.

### Visual Studio Code

*Visual Studio Code* es un editor de código fuente ligero pero potente que se ejecuta en su escritorio y está disponible para Windows, *macOS* y Linux. Viene con soporte incorporado para *JavaScript*, *TypeScript* y Node.js y tiene un rico ecosistema de extensiones para otros lenguajes (como C ++, C #, Java, Python, PHP, Go) y tiempos de ejecución (como .NET y Unity) [39].

#### Primeros pasos

Para aprovechar al máximo *Visual Studio Code*, siga los siguientes pasos:

- Vídeos de introducción: comience su viaje con *VS Code* a través de estos vídeos de introducción.
- Configuración: instale *VS Code* para su plataforma y configure el conjunto de herramientas para sus necesidades de desarrollo.
- Interfaz de usuario: introducción a la interfaz de usuario básica, los comandos y las funciones del editor de *VS Code*.
- Configuración: personalice el código *VS Code* según cómo le gusta trabajar.
- Idiomas: conozca la compatibilidad de *VS Code* con sus lenguajes de programación favoritos.
- Node.js: este tutorial le permite ejecutar y depurar rápidamente una aplicación web Node.js.
- Consejos y trucos: ingrese directamente con consejos y trucos para convertirse en un usuario avanzado de *VS Code*
- *Azure*: *VS Code* es excelente para implementar sus aplicaciones web en la nube.
- API de extensión: aprenda a escribir una extensión de código *VS Code*.
- Atajos de teclado
- Aumente su productividad con los atajos de teclado de *VS Code*.
- Hoja de referencia de métodos abreviados de teclado: aprenda los métodos abreviados de teclado más utilizados.
- Extensiones de mapa de teclas: cambie los atajos de teclado de *VS Code* para que coincidan con otro editor.
- Personalizar atajos de teclado: modifique los atajos de teclado predeterminados.

## APÉNDICE E.

### Node.js

Node.js es una plataforma para crear aplicaciones de servidor rápidas y escalables utilizando JavaScript. Node.js es el tiempo de ejecución y npm es el administrador de paquetes para los módulos de Node.js. *Visual Studio Code* tiene soporte para los lenguajes *JavaScript* y *TypeScript* listos para usar, así como para la depuración de Node.js. Sin embargo, para ejecutar una aplicación Node.js, necesitará instalar el tiempo de ejecución de Node.js en su máquina [40].

*Node Package Manager* se incluye en la distribución de Node.js. Deberá abrir una nueva terminal (símbolo del sistema) para que el nodo y las herramientas de línea de comandos npm estén en su PATH [40].

Linux: hay paquetes específicos de Node.js disponibles para los distintos tipos de Linux. Consulte Instalación de Node.js a través del administrador de paquetes para encontrar el paquete Node.js y las instrucciones de instalación adaptadas a su versión de Linux [40].

Consejo: para probar que tienes Node.js correctamente instalado en tu computadora, abre una nueva terminal y escribe `node --help` y deberías ver la documentación de uso [40].



