



UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE CUENCA
CARRERA DE INGENIERÍA DE SISTEMAS

**DISEÑAR E IMPLEMENTAR UN SISTEMA DE RECOMENDACIÓN DE
CONTENIDOS EDUCATIVOS PARA NIÑOS DE 5 A 7 AÑOS**

Trabajo de titulación previo a la obtención del
título de Ingeniero de Sistemas

AUTOR: JONNATHAN ALFREDO SIMBAÑA ABAD

TUTOR: ING. REMIGIO ISMAEL HURTADO ORTIZ, PHD.

Cuenca - Ecuador

2024

**CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE
TITULACIÓN**

Yo, Jonnathan Alfredo Simbaña Abad con documento de identificación N° 0301672085, manifiesto que:

Soy el autor y responsable del presente trabajo; y, autorizo a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo de titulación.

Cuenca, 08 de abril de 2024

Atentamente,



Jonnathan Alfredo Simbaña Abad

0301672085

**CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE
TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA**

Yo, Jonnathan Alfredo Simbaña Abad con documento de identificación N° 0301672085, expreso mi voluntad y por medio del presente documento cedo a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que soy autor del Proyecto técnico: “Diseñar e implementar un sistema de recomendación de contenidos educativos para niños de 5 a 7 años”, el cual ha sido desarrollado para optar por el título de: Ingeniero de Sistemas, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribo este documento en el momento que hago la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Cuenca, 08 de abril de 2024

Atentamente,



Jonnathan Alfredo Simbaña Abad

0301672085

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, Remigio Ismael Hurtado Ortiz con documento de identificación N° 0104621388, docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: DISEÑAR E IMPLEMENTAR UN SISTEMA DE RECOMENDACIÓN DE CONTENIDOS EDUCATIVOS PARA NIÑOS DE 5 A 7 AÑOS, realizado por Jonnathan Alfredo Simbaña Abad con documento de identificación N° 0301672085 obteniendo como resultado final el trabajo de titulación bajo la opción Proyecto técnico que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Cuenca, 08 de abril de 2024

Atentamente,



Ing. Remigio Ismael Hurtado Ortiz, PhD.

0104621388

Índice de Contenido

1. Introducción	11
2. Problema	11
2.1. Antecedentes	11
2.2. Importancia y alcance	13
3. Objetivos Generales y Específicos	14
3.1. Objetivos Generales	14
3.2. Objetivos Específicos	14
4. Revisión de la literatura o fundamentos teóricos	14
4.1. Sistemas de recomendación	14
4.2. Grafos	15
4.3. Flutter	17
4.4. FastAPI	18
4.5. Docker	19
4.6. Neo4j	21
4.7. MongoDB	23
4.8. Web Crawler	24
5. Marco metodológico	25
5.1. Metodología de desarrollo ágil Scrum	26
5.2. Descripción de las Actividades	26
5.3. Fases del scrum	28
5.4. Arquitectura	43
6. Resultados	45
6.1. Requerimientos	46
6.1.1. Neo4j para uso local	46
6.1.2. Neo4j para uso en la nube	46
6.2. Diseño	46
6.2.1. Diseño de la Aplicación	46
6.2.2. Diseño de la aplicación móvil	49
6.2.3. Diseño del modelo de recomendación	54
6.2.4. Desarrollo	57
7. Cronograma	65
8. Presupuesto	68

Conclusiones y trabajos futuros -----	69
Recomendaciones -----	70

Índice de figuras

Figura 1.Ejemplo de modelo de grafos (Needham y hodler,2019).....	15
Figura 2.- solución de problema de Königsberg, y la creación de los grafos por Euler. (Baton y Van Bruggen, 2018).....	15
Figura 3.Comparación entre un grafo directo e Indirecto. (Needham y Hodler,2019).....	17
Figura 4."Flutter framework", modelo para la creación de un nuevo proyecto(Rose,2022)	18
Figura 5.-Project scaffolding iniciado en una variable aplicable en Python. (Adeshina, 2022)	19
Figura 6.- Arquitectura de un container. (Kithulwatta et al., 2021).....	20
Figura 7.- Un desarrollo de un workflow por medio de Docker. (Kane y Matthias, 2023).....	20
Figura 8.- Un ejemplo de un diagrama prototipo de un Docker Compose. (Reis et al., 2022)	21
Figura 9.- Ciencia de grafos para tecnología Neo4j. (Bader, 2022)	22
Figura 10.- Propiedades de un grafo dentro de un contenedor. (Bader, 2022)	23
Figura 11.- Arquitectura de modelo de replicación ofrecido por MongoDB. (Schultz et al, 2019)	24
Figura 12.- Prototipo de un Web crawler. (Klymash et al, 2020).....	25
Figura 13.-Sprints del proyecto.....	29
Figura 14.-Entrenamiento de filtro online. (Falk,2019).....	31
Figura 15.-El recuperador de Ítems devuelve una recomendación en base al FILTRO. (Falk,2019).....	31
Figura 16.- Las dos formas diferentes en las que trabaja el sistema basado en vecinos ceranos. (Falk, 2019).....	32
Figura 17.-Página Wikinclusion.org (Montoya.R)	34
Figura 18.-. Estructura de un elemento HTML junto a sus propiedades (Ramos. G, 2021). ..	35
Figura 19.-Ejemplo de modo inspección para identificar los elementos HTML.....	35
Figura 20.-. Creación de la infraestructura de grafos con los modelos.....	36
Figura 21.-Lógica de negocio	37
Figura 22.-Creación del contenedor utilizando docker-compose	38
Figura 23.-Creación del contenedor utilizando docker-compose	39
Figura 24.-Creación del contenedor utilizando docker-compose	40
Figura 25.-Algoritmos de similaridad.....	40
Figura 26.-Instalación de plugins.....	41
Figura 27.-Creación del droplet	42
Figura 28.-Resultado de la creación del droplet	43
Figura 29.-Arquitectura del sistema.....	43
Figura 30.-Arquitectura Frontend	45
Figura 31.-Visualización de nodos creados	47
Figura 32.-Creación de las colecciones en MongoDB.....	49
Figura 33. Pantalla del login	50

Figura 34. Pantalla de registro	50
Figura 35. Pantalla home	51
Figura 36. Pantalla de favoritos	51
Figura 37. Pantalla del perfil del usuario	52
Figura 38. Pantalla de bloques	52
Figura 39. Pantalla de Competencia	53
Figura 40. Pantalla de habilidad.....	53
Figura 41. Pantalla detalle de habilidad	54
Figura 42.-Pantalla de los software disponibles	54
Figura 43.-Ejemplo del modelado en neo4j	55
Figura 44.-Ejemplo del modelo basado en grafos	55
Figura 45.-Ejemplo del sistema recomendador	56
Figura 46.-Funcionamiento del sistema recomendador basado en contenido	56
Figura 47.-Algoritmo de similitud de coseno, utilizando GDS	57
Figura 48.-Resultado de los vecinos más cercanos.....	57
Figura 49.-Algoritmo para listar las habilidades.....	57
Figura 50.-Conexión a MongoDB	57
Figura 51. Métodos usados en el sistema.....	58
Figura 52.-Conexión de neo4j con Python	59
Figura 53. Ejemplo de un método neo4j	59
Figura 54.-Estructura del servicio FastApi	60
Figura 55. Estructura de los routers	60
Figura 56. Creando un servicio GET con MongoDB y neo4j	61
Figura 57. Ejemplo del consumo de una api.....	61
Figura 58.-Inicio de un usuario nuevo	62
Figura 59.-Inicio del sistema recomendador.....	62
Figura 60.-Pantalla de encuestas.....	63
Figura 61.-Reporte de encuestas por habilidad.....	64
Figura 62.-Detalle de la encuesta por pregunta	64

Índice de Tablas

Tabla 1.-Actividades a desarrollar del OE1	26
Tabla 2.-Actividades a desarrollar del OE2	27
Tabla 3.-Actividades a desarrollar del OE3	27
Tabla 4.-Actividades a desarrollar del OE4	27
Tabla 5.-Actividades a desarrollar del OE5	28
Tabla 6.-Actividades a desarrollar del OE6	28
Tabla 7.-Roles del proyecto	28
Tabla 8.- Elementos de la función de la similaridad. (Falk, 2019)	32
Tabla 9.- Tipos de algoritmos utilizados en los sistemas de filtro colaborativo (Falk,2019) ...	33
Tabla 10.-Tabla de la cantidad de información recolectada	36
Tabla 11.-Tabla de valoraciones	42
Tabla 12.-Requerimientos mínimos para el uso de neo4j para uso personal	46
Tabla 13.-requerimiento para el uso de neo4j en la nube	46
Tabla 14.-Presupuesto del desarrollo de la aplicación	68

RESUMEN

En la actualidad, existe una cantidad enorme de niños que tiene accesos a dispositivos inteligentes en el cual es un peligro para ellos. En nuestros días la problemática sigue en pie, debido a la inmensidad de contenido en línea. No obstante, esto ha creado múltiples problemáticas sociales, podemos evidenciar esto después de la pandemia, donde los niños en específico llegaron a presentarse como un grupo vulnerable a contenido de carácter grotesco, hoy en día van desde recibir publicidad de sus streamers y youtubers favoritos, comerciales de productos, contenido para adultos hasta links maliciosos, e, incluso se pudo ver como herramientas cibernéticas han logrado ser utilizadas para filtrar información spam, o phishing, donde los web crawlers nos ayudan a contrarrestar de cierta manera estos crímenes virtuales producidos por la falta de control del contenido que visualizan nuestro infantes dentro de una etapa tan delicada como lo sería la escolar. Por lo tanto, con el fin de recomendar contenido adecuado y no malicioso para los niños, en este trabajo se ha desarrollado y se presenta un sistema de recomendación de contenidos educativos para niños de 5 a 7 años. Se implementan nuevas tecnologías junto a una base de datos basados en grafos, y esto permite realizar recomendaciones basadas en los gustos de los niños para filtrar la información relacionada. El sistema de recomendación es un tipo de sistema basado en contenidos. Se obtiene el grado de satisfacción de los usuarios con respecto a las funciones del sistema de recomendación mediante encuestas o preguntas de evaluación del servicio. Esto nos permite saber si el contenido o habilidad que recomienda el sistema es de utilidad o no para los niños.

La información de contenidos educativos que se ha recolectado para este proyecto se almacena en bases de datos de NEO4j donde se ha implementado el uso de sus librerías como es APOC y GDS ya que contiene una gran cantidad de algoritmos de búsqueda, generando así una recomendación más certera acorde a los gustos del niño, ya que sus conexiones nos indican las relaciones entre las competencias, clientes y habilidades para la generación de recomendaciones las cuales se generan de forma automática.

En trabajos futuros de este proyecto es importante mejorar el sistema de recomendación, se tiene que proveer al sistema más información referente al usuario y así generar una mejor precisión al momento de recomendar datos al usuario, generar los servicios para tener en diferentes dispositivos, en el repositorio de contenido educativo alimentar al repositorio con más información, es decir con archivos multimedia, software y material de apoyo.

ABSTRACT

Currently, there is a huge number of children who have access to smart devices in which it is a danger for them. Today the problem still exists, due to the immensity of online content. However, this has created multiple social problems, we can see this after the pandemic, where children specifically came to present themselves as a group vulnerable to content of a grotesque nature, today they range from receiving advertising from their favorite streamers and YouTubers, product commercials, adult content to malicious links, and it was even possible to see how cyber tools have managed to be used to filter spam information, or phishing, where web crawlers help us counteract in a certain way these virtual crimes produced by the lack of control of the content that our infants view within a stage as delicate as school. Therefore, to recommend appropriate and non-malicious content for children, in this work an educational content recommendation system for children aged 5 to 7 years has been developed and presented. New technologies are implemented together with a graph-based database, and this allows recommendations to be made based on children's tastes to filter related information. The recommendation system is a type of content-based system. The degree of user satisfaction with respect to the functions of the recommendation system is obtained through surveys or service evaluation questions. This allows us to know whether the content or skill that the system recommends is useful or not for children.

The educational content information that has been collected for this project is stored in NEO4j databases where the use of its libraries such as APOC and GDS has been implemented since it contains many search algorithms, thus generating a better recommendation. accurate according to the child's tastes, since its connections indicate the relationships between competencies, clients, and skills for the generation of recommendations which are generated automatically.

In future work of this project, it is important to improve the recommendation system, the system must be provided with more information regarding the user and thus generate better precision when recommending data to the user, generate services to have on different devices, in the educational content repository feed the repository with more information, that is, with multimedia files, software and support material.

1. Introducción

El poder filtrar la información que día a día consumimos, es algo que relativamente es nuevo, ya que los sistemas de recomendación al usarse exactamente para esto se han convertido en una necesidad hoy en día, incluso para las empresas, claro es el caso de Netflix, Amazon, Google, y así por nombrar empresas de distribución de contenido digital. Donde el problema radica en que estos programas no distinguen entre quienes ven el contenido, debido a esta problemática, han surgido campañas de concientización a manera de podcast, o también a manera de cursos, donde se capacitan a los padres para poder contrarrestar esta flaqueza de los sistemas de recomendación. Debido a que realmente existe muchos estudios que indican que los infantes están consumiendo contenido no adecuado para su edad, en realidad incluso según Estefanell, (2020) niños menores a 2 años, no deberían estar enfrente de una pantalla. Los eventos del Covid-19 han logrado que en gran medida esta problemática se intensifique, en una oportunidad para hacer que los niños se ha propuesto el proyecto para diseñar e implementar un sistema de recomendación educativo, gracias al uso de los sistemas de recomendación y a la base de datos NoSQL, se pude ver como este proyecto trata de contrarrestar el consumo indiscriminado del contenido que circula por la red.

2. Problema

2.1. Antecedentes

Los sistemas de recomendación surgen más o menos a partir de los 90 como un medio para resolver el problema de búsqueda de información ineficiente, como puede ser el ejemplo de la empresa Google que nace en 1998. (Alfonso Azcuy, 2021); Hoy por hoy llegan a verse muy frecuentemente en múltiples plataformas de *streaming*, o en su defecto en las redes sociales. Estos sistemas van adaptándose a las necesidades del usuario, o, dicho de otra manera.

Estos sistemas a partir de identificar las preferencias y sugerir ítems relevantes para cada usuario, almacena los perfiles creados por los usuarios, y, a su vez genera en el usuario un sentimiento de pertenencia, debido a que la IA (inteligencia artificial) recuerda quién es y produce un sentimiento de calidez en el usuario. (Alfonso Azcuy, 2021; Mendoza Olguín et al., 2019).

No obstante, el uso de estos sistemas además de ser utilizados a manera de herramienta facilitadora al momento de la realización de búsqueda, estos son utilizados como una herramienta de marketing para las empresas. Oscar Escamilla y Sergio Marcellin (2018); nos

relatan el caso de la plataforma Netflix, la cual en 2007 organizo un concurso con la finalidad de encontrar a una persona que mejorara las predicciones en al menos un 10% con respecto al sistema de Netflix operante para la época. A su vez se puede destacar el caso de Amazon, el cuál según palabras de Monideepa et al., (2023); quienes citando a la revista *IEEE Internet Computing* la cual publicó en 2003 una noticia titulada *Amazon.com Recommendations: Item-to-Item Collaboration filtering*. Donde, nos relata como Amazon fue uno de los pioneros en esta especie de sistema de recomendaciones con un filtro colaborativo de artículo inteligente, a comparación del ambiguo sistema de filtro por atención inteligente al cliente. Sin embargo, en nuestros días estas técnicas de *Machine Learning* han sido llevadas al campo de la educación. Los objetivos de desarrollo sostenible (ODS); han llevado a cabo múltiples políticas para “garantizar una educación inclusiva, equitativa y de calidad y promover oportunidades de aprendizaje durante la vida para todos desde el año 2000.” (UNDP, 2023) en específico la cuarta ODS hace alusión a estas políticas en pro de la educación por medios virtuales. En la actualidad existen múltiples problemáticas educativas debido al uso indebido de tics dentro y fuera del aula. Ya que, según Flores Vivar y García Peñalvo, (2023); “estas tecnologías han causado una disrupción en los procesos de enseñanza aprendizaje en los distintos niveles de educación.” Esto a su vez, puede tener una relación con el problema de las bases de datos del contenido digital consumido por los estudiantes. En particular por el uso indebido de estas mismas redes se han dado múltiples vulneraciones a menores por medios digitales, esto debido a que “los menores van desde recibir publicidad hasta la exposición de contenidos de violencia o pornográficos... en el peor de los casos puede llegar a presentarse como un tipo de adicción” (Melamud y Waisman, 2019).

A su vez, un número mayor de pruebas empíricas revelan que los niños acceden desde edades más tempranas; e incluso, en algunos países se evidencia que niños menores a 15 años tienen la misma probabilidad de acceder a sitios web que utilizan un adulto promedio de 25 años.(Salcedo Aparicio et al., 2021)

No obstante, también se puede destacar lo que nos dice Berlinerblau (2018) lo cual dice que en el caso de los abusos cibernéticos no solo los agresores se limitan a los adultos, el autor afirma que entre el 20% y el 40% de abusos cometidos son perpetrados por niños mayores, adolescentes y personas con menos de 21 años. Lo que nos da a entender que el contenido que consumen los niños es vital para su formación. De lo contrario puede llegar a ser parte de un problema de carácter macrosocial. En reacción a esta problemática se han realizado innumerables campañas de concienciación dirigida a los padres con la temática del correcto

uso de la tecnología. En contraste ante la problemática sobre la desinformación del monitoreo parental sobre el contenido que ven los infantes, tenemos el caso de Rosado y Ruiz, (2022) quienes implementaron un podcast sobre esta problemática, donde se concluyó que dentro de la navegación de los infantes, se encontraba mucho contenido inapropiado, e incluso, estos estaban expuestos a riesgos tales como el ciberbullying, extorsión, y el grooming; lo que en consecuencia podría generar daños psicosociales. La problemática social de la violencia sexual hoy en día se da a la indiscriminación del contenido que se consume en las pantallas, de cierta manera este problema se potencio en la pandemia, hemos abordado algunos casos los cuales debido a un uso inadecuado, falta de monitoreo e incluso una falta de un sistema de recomendación adecuado que filtre el contenido que los niños consumen, son algunos factores incidentes que mantienen el problema radicado en los niños y niñas en la etapa escolar (5-8); donde el libre acceso a sitios web diseñados para la interacción adulta causen que los contenidos visualizados por los infantes no sea el apropiado para su edad. Existen una diversidad de contenidos y herramientas educativos que intentan hacer buen uso de estos sistemas y plataformas, sin embargo, el desarrollo de estas tecnologías ha sido destinado de cierta manera más a plataformas de ocio o redes sociales, donde se dejó de lado el contenido que pueden visualizar los infantes.

2.2. Importancia y alcance

El sistema está enfocado para niños entre 5 a 7 años para que puedan acceder a todos los recursos necesarios sin que obtengan resultados de búsqueda inapropiados para su edad y a su vez que se obtengan resultados óptimos, los resultados se mostrar en un diseño infantil para que sea más llamativo para ellos, se recomienda a las familias y a los niños promover el juego, la lectura, el arte y la escritura, ya que estas actividades contribuyen positivamente al aprendizaje de los niños.

Se hace el uso de nuevas tecnologías para mejorar este proceso de desarrollo de software, ya que gracias a estos se ha optimizado el obtener resultados en menos tiempos.

Entre las tecnologías se utilizando fastAPI para la creación de Apis, Luego se tiene Docker para ejecutar contenedores de sistemas pequeños en diferentes sistemas para mejorar su funcionamiento.

3. Objetivos Generales y Específicos

En esta sección describimos los objetivos que hemos planteado para el desarrollo del proyecto. Se proporcionan más detalles en los objetivos generales y específicos.

3.1. Objetivos Generales

Diseñar e implementar un sistema de recomendación de contenidos educativos para niños de 5-7 años utilizando grafos y servicios web.

3.2. Objetivos Específicos

- **OE1.**Revisar el estado del arte sobre sistemas de recomendación basados en grafos.
- **OE2.**Diseñar e implementar un repositorio de contenidos educativos.
- **OE3.**Diseñar e implementar una arquitectura para el sistema de acopio y distribución
- **OE3.**Desarrollar un módulo inteligente para la recomendación de contenidos educativos.
- **OE4.**Probar y validar la recomendación basada en encuestas.

4. Revisión de la literatura o fundamentos teóricos

En esta sección estudiaremos y conoceremos los fundamentos principales para el desarrollo de nuestro proyecto, el estado del arte y las tecnologías que utilizaremos como los Frameworks de desarrollo, servidores web, también se estudiara el uso de grafos, el uso de sistemas de recomendación.

4.1. Sistemas de recomendación

Se pueden definir como un sistema creado con la finalidad de guiar a los usuarios durante su búsqueda mediante las diferentes plataformas. A su vez, se ve que estos sistemas de recomendación pretenden recomendar atreves de la cantidad de contenido y de usuarios que estas tengan, y por lo tanto estas pueden llegar a dividirse en sistemas colaborativos y los sistemas de contenido. (Riquelme y Botella, 2020).

4.2. Grafos

“Un grafo es un grupo de vértices y una acumulación de aristas, donde cada una se conecta a un par de vértices.” (Sedgewick y Kevin, 2018, pg515) estos grafos gracias a sus nodos etiquetadores de marcas que pueden relacionarse entre sí por una relación de tipo. (Needham y Hodler, 2019) este procesamiento de datos ayuda a que visualizar la información sea relativamente más fácil, tal y como se visualiza en la figura (5), hoy, en día han adquirido una diversidad de usos, donde podemos destacar el desarrollo de aplicaciones para “mapeo, contenido web, circuitos, calendarios, comercio, citas online, noticias online, softwares y redes sociales.” (Sedgewick y Kevin, 2018)

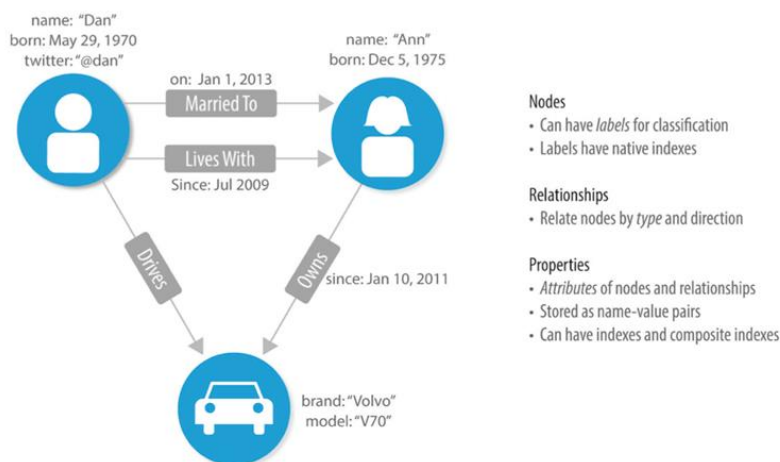


FIGURA 1.EJEMPLO DE MODELO DE GRAFOS (NEEDHAM Y HODLER,2019)

Podemos rescatar que el creador del grafo vendría siendo Euler, el cual tratando de resolver el problema de los 7 puentes de *Königsberg*, conceptualiza dentro de la teoría matemática lo que hoy conocemos como grafos.(Baton y Van Bruggen, 2018) tal y como se aprecia en la figura (6).



FIGURA 2.- SOLUCIÓN DE PROBLEMA DE KÖNIGSBERG, Y LA CREACIÓN DE LOS GRAFOS POR EULER. (BATON Y VAN BRUGGEN, 2018)

Después de haber conceptualizado los grafos, podemos destacar también qué gracias a esta base de datos de “prueba”, los cuales adquieren este nombre debido a que contienen datos significativos durante la fase de prueba de una aplicación. Lo que la hace que este sea uno de los pasos más significativos en el desarrollo de una base de datos orientada a grafos. (Riquelme y Botella, 2020) podemos ver que este sistema de organización de datos es una buena opción al momento de realizar el desarrollo de una base de datos, sin embargo, Riquelme y Botella (2020) destacan qué:

las tecnologías NoSQL no cuentan con una estandarización de los lenguajes de consulta, lo que provoca que cada tecnología proporcione sus propias Apis y lenguajes de consulta... también se puede ver como las propiedades ACID no se mantienen, lo que en consecuencia provoca que solo las propiedades BASE se mantengan, produciendo posibles pérdidas de fiabilidad y consistencia.

Una vez analizado los pros y contras que presentan los grafos, se podría sumar a todo esto sus subtipos, donde se destacaran los grafos directos e indirectos.

- **Grafo indirecto versus grafo directo**

El tipo de grafo indirecto vendría a ser uno de los más usuales, ya que debido a su fácil escritura de codificación y eficiente acceso a la misma gracias a que usa un matiz de índice. (Sedgewick & Kevin, 2018) Por otra parte, podríamos definir a los grafos directos, como aquellos los cuales tienen una dirección específica, donde se pueden apreciar conexiones de entrada o *in-links*, y también las conexiones de salida o *out-links*, las cuales se originan del nodo. (Needham & Hodler, 2019) podríamos asimilar que esta misma es la gran diferencia entre un grafo directo en indirecto, ya que, debido a su facilidad de apreciación, como se ve en la Figura (3); donde esta diferencia es clara.

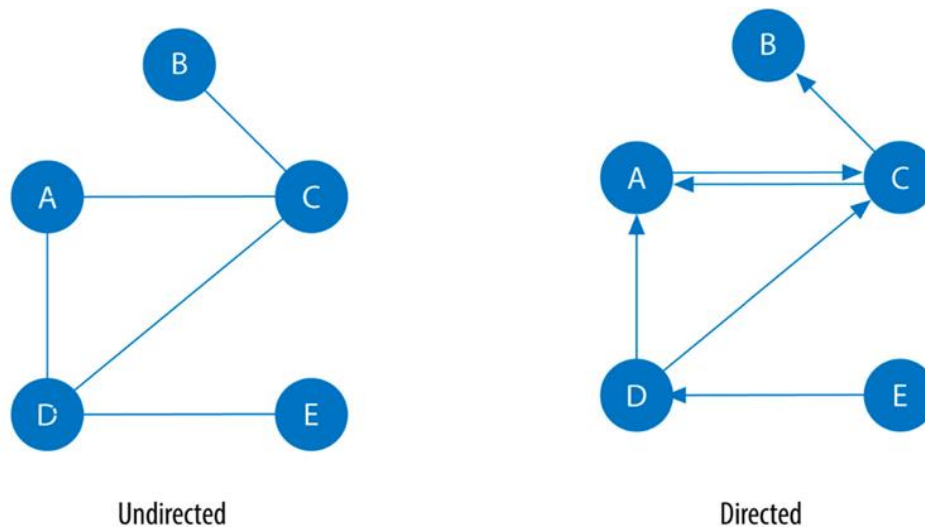


FIGURA 3.COMPARACIÓN ENTRE UN GRAFO DIRECTO E INDIRECTO. (NEEDHAM Y HODLER,2019)

4.3. Flutter

Un kit de desarrollo de software Flutter, es un software la cual cumple con la una compleja función, que no cualquier aplicación puede, la cual es la de ser reconocida por todos los sistemas operativos en móviles. Ya que Bailey et al., (2021) nos cuentan que desde el lanzamiento de la Apple App Store, y siguiéndole Google Play Store, marcan una nueva manera de compartir programas de una modo controlada, donde este debe seguir un modelo que lidie con el ecosistema web. Resaltando que el fruto de años y años de recolección de datos han formado el Flutter Framework. Muchos estudios han señalado que Flutter y Dart, al haber sido creados por Google, son muy dependientes de este. (Bailey et al., 2021) A su vez, se denota su libre y fácil acceso. Al momento de la realización de un proyecto con el modelo Flutter, se debe usar un modelo preexistente para el desarrollo de una nueva. (Rose, 2022) algunos de los modelos de Flutter Framework se muestran en la figura (8).

Type	Description
app	This is the default for flutter create and is used to generate a Flutter application.
module	This option will enable you to create a module that can be integrated with other applications.
package	This option will enable a sharable Flutter project
plugin	This option provides an api base for use with Android and iOS
skeleton	This option provides a best practice application based on a Detail View

FIGURA 4."FLUTER FRAMEWORK", MODELO PARA LA CREACIÓN DE UN NUEVO PROYECTO(ROSE,2022)

Por último, se debe de tener en cuenta la capacidad de tu máquina, donde los modelos serán una referencia para el modelaje del código de la documentación website API en la dirección <http://docs.flutter.dev/>. (Rose, 2022); A su vez, se pueden encontrar unos cuantos tipos de modelajes para basar el modelo de cualquier proyecto. Podemos ver como Flutter, es una muy buena opción al momento de realizar nuestro proyecto. Esto debido a tal y como se resalta, la basta variedad de plantillas que se pueden llegar con obtener en esta aplicación, la hacen que sea un formato muy fácil y simple para el desarrollo de aplicaciones destinadas a un microservicio.

4.4. FastAPI

Esta Framework website de Python, es una ligera herramienta moderna de API (Aplicación Programming Interfaces), estas también se caracterizan por conectar múltiples elementos dentro del desarrollo de sitios web, tales como HTML, JavaScript, e, imágenes. (Adeshina, 2022; Lubanovic, 2023) a su vez podemos destacar que los tipos de Apis seria las de protocolo o de control estructural y las de formato, o de control de contenido. Usualmente el uso de rutas por medio de solicitudes HTTP. Donde, una vez registrada y revisada la solicitud, en la ruta se define después del proceso de solicitud dentro de gestor de ruta. (Adeshina, 2022) Usualmente estas rutas son una vía fácil de utilizar dentro del lenguaje Python, podemos ver un ejemplo de programación dentro de una carpeta de proyectos gracias a que esta ruta sencilla de aplicación esa gestionada por FastAPI().(Adeshina, 2022; Lubanovic, 2023) así como apreciamos en la figura (5).

```
from fastapi import FastAPI

app = FastAPI()

@app.get("/")
async def welcome() -> dict:
    return { "message": "Hello World" }
```

FIGURA 5.-PROJECT SCAFFOLDING INICIADO EN UNA VARIABLE APLICABLE EN PYTHON.
(ADESHINA, 2022)

En cuanto al protocolo HTTP, Lubanovic, (2023) cita los tres componentes de la World Wide Web (www) planteadas por Berners-Lee, donde nos plantea que HTML, es un lenguaje para la visualización de datos; también que HTTP es un protocolo destinado al servicio al cliente, y, por último que las URLs son un esquema de direccionamiento de recursos web. (Adeshina, 2022) A su vez, esta capacidad de las Apis para poder conectarse y tener sincronización con el resto de la red, la hacen un factor determinante en la codificación del proyecto, el poder constituir el proyecto de esta manera, generara en sí el poder plantear nuevas actualizaciones sea una tarea menos tediosa, y que llega a cierto punto a ser confusa. Las Apis al tener una sincronización tan grande con las herramientas NoSQL, hacen que la consistencia y latencia de datos sea igual o mejor a la de las tecnologías SQL.

4.5. Docker

Este programa fue lanzado en el 2013 por Solomon Hykes, el cual puede definirse como una de las primeras herramientas que permitió tener un acceso más amplio para poder configurar nueva tecnología denominada “*containerization*”, o [Empaquetamiento]. (Kane y Matthias, 2023) A su vez, Docker es un popular ejemplo de que la tecnología de contenedores ha evolucionado a ser la tecnología estándar *defecto* de desarrollo de softwares. (Reis et al., 2022) Estas herramientas son de cierta manera una mejora a la productividad dentro de los marcos de trabajo de los proyectos destinados al desarrollo de aplicaciones y programas; Ya que estos, son basados en contenedores virtuales, los cuales usan un solo núcleo para iniciar múltiples fases dentro del sistema operativo, donde al virtualizar borradores de inicio como si fueran una aplicación con un sistema operativo. (EFE et al., 2020) Tal y como se muestra en la figura (6).

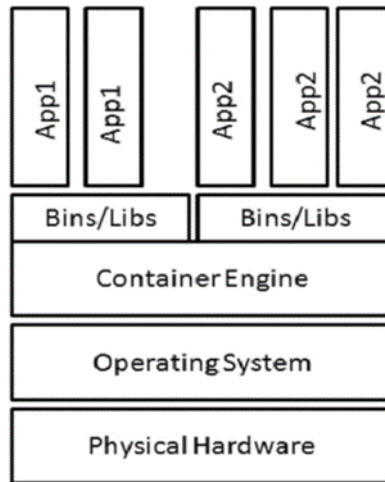


FIGURA 6.- ARQUITECTURA DE UN CONTAINER. (KITHULWATTA ET AL., 2021)

Por otro lado, tenemos al Docker File, el cual no es más que el archivo donde se almacenan los datos de los proyectos que empiezan en Docker. Sin embargo, como nos indica Reis et al., (2022) dentro de su estudio se encontró que muchos errores de calidad de almacenamiento se presentaban dentro de los montos etiquetados MAINTAINER dentro del conjunto fichado en la base de las imágenes y sus dependencias, además de muchos Docker files. Después, tenemos a Kane y Matthias, (2023) quienes nos afirman que Docker genera que las decisiones arquitectónicas sean más simples, lo que en consecuencia crea que la herramienta sea fácil de escribir y compartir entre las diferentes aplicaciones; además, que esta envuelve todos los requerimientos juntos en un solo formato llamado el standart Open Container Initiative (OCI). A su vez los autores destacan como Docker ayuda de una manera mucho más sintética al desarrollo de *workflow*, tal y como se muestra en la figura (11).

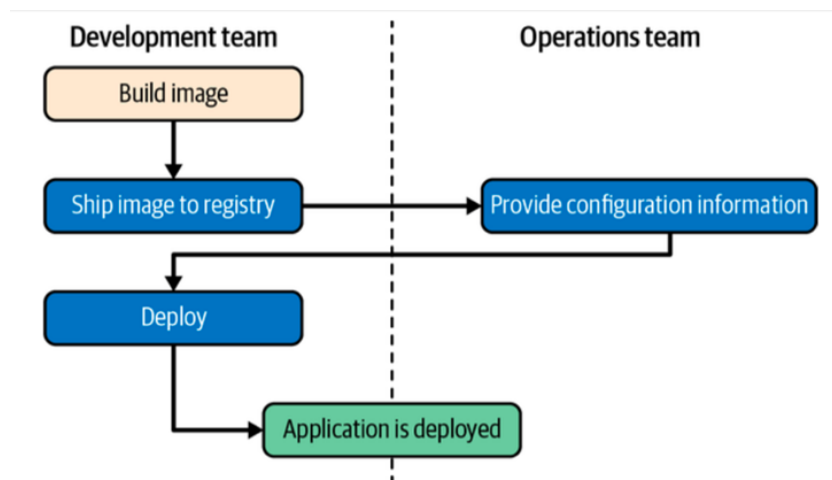


FIGURA 7.- UN DESARROLLO DE UN WORKFLOW POR MEDIO DE DOCKER. (KANE Y MATTHIAS, 2023)

Además, podemos resaltar como Docker nos da la útil y eficaz herramienta de Docker Compose. La cual, se trata de un tipo de herramienta que trata de reemplazar a las herramientas ad hoc; a su vez, podemos ver que el Docker Compose se almacena y se configura como un archivo YAML, el cual puede ser encontrado en nuestro proyecto bajo el nombre *docker-compose.yml*. (Kane y Matthias, 2023) También, esta herramienta ofrecida por Docker se visualiza a través de la manipulación de un modelo interactivo, el cual nos sirve como referencia de implementación para demostrar una aproximación en la práctica, y usa un prototipo como una manera empírica de validación aproximada de la realidad. (Reis et al., 2022) Tal y como se muestra el ejemplo de un diagrama prototipo de un Docker Compose en la figura (8)

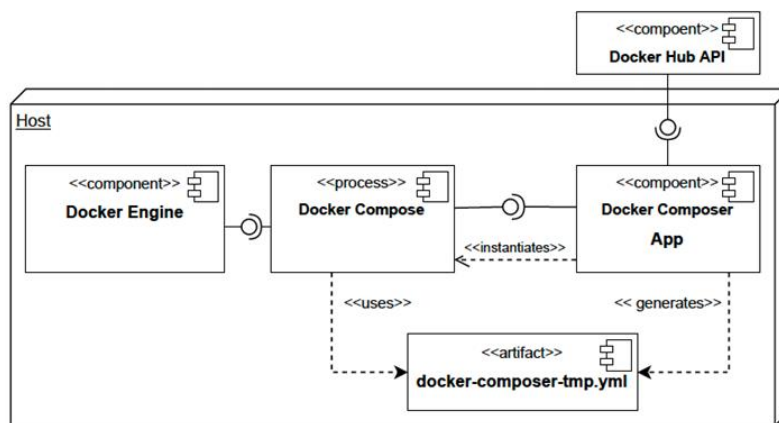


FIGURA 8.- UN EJEMPLO DE UN DIAGRAMA PROTOTIPO DE UN DOCKER COMPOSE. (REIS ET AL., 2022)

4.6. Neo4j

Esta herramienta es una de las ocupadas en el desarrollo de proyectos relacionados con grafos, ya que debido a su gestión y manejo de la base de datos relacionada con grafos. Donde se destaca su consistencia, transacción, disponibilidad, esquema flexible y las ventajas que ofrecen las queries, en conjunto con su escalabilidad y relatividad facilidad con la que puede compartir bases de datos NoSQL. (Wiese, 2019) la hace una herramienta sumamente completa y efectiva para la gestión de datos para nuestro proyecto. Siguiendo con la definición que Baton y Van Bruggen, (2018, pg 35) nos ofrecen sobre Neo4 nos dicen que “es un completo manejo de una base de datos acerca de grafos, sino más bien, una librería de grafos que la gente pueda ayudarse de esta para configurar sus estructuras de datos de una manera mucho más sencilla.” También podríamos acotar como múltiples estudios señalan la eficacia de este

programa ya que al ser un complemento de la base de datos ACID. (Baton Y Van, 2018) A su vez la forma en que actúa este sistema es a través de la duración entre el requisito de la web a la base de datos y la respuesta de la web ante la petición; lo que se da en milisegundos. En la actualidad según Baton y Van (2018) “existen dos tipos de sistemas de gestión de datos la OLTP (Online transaction Processing); y la OLAP (Online Analytical processing).” En la figura (9) se muestra con más detalle las diferencias entre ambas.

La ciencia de datos de Neo4j como su escritorio, buscador, Neo4J y además el manejo transaccional y analítico del proceso al trabajar con grafos, se muestra en la Figura (9).

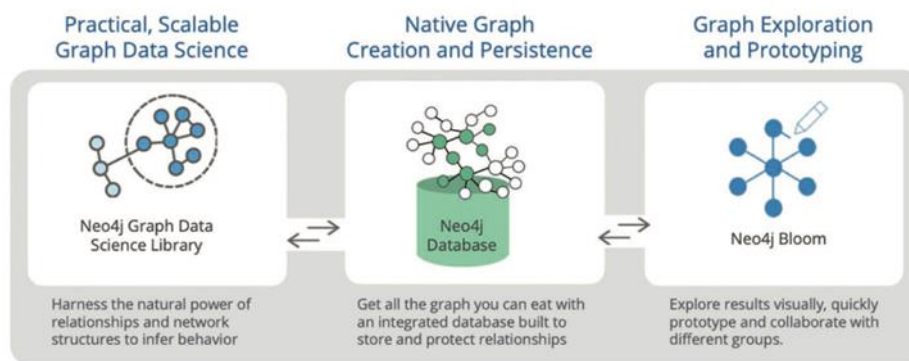


FIGURA 9.- CIENCIA DE GRAFOS PARA TECNOLOGÍA NEO4J. (BADER, 2022)

El manejo de la base de datos que ofrece Neo4j a sus usuarios, es gracias a su enorme base de datos. Bader, (2022) resalta como esta tecnología gracias a su facilidad para el trabajo con grafos, tal y como se muestra en la figura (9); muestra como este sistema puede mantener los atributos numéricos, mejor conocidos como propiedades. Por consiguiente, lo que múltiples autores señalan que hace que esta herramienta sea la más adecuada con el manejo de grafos. Apunta a que sea el lenguaje Cypher, ya que este lenguaje al tener las propiedades de la bases de datos de grafos, dicha similitud se aprecia en uso del tipo de sintaxis ASCII- Art. (Bader, 2022; Stothers & Nguyen, 2020) Las propiedades de construcción de “Keys”, como la caracterización axiomática, la caracterización de algoritmos, genere que el programa en si sea beneficioso al momento de actualizar y consultar operaciones en propiedades de grafos, esto gracias a las oportunidades físicas y lógicas sobre el manejo de los datos con Neo4j. (Link, 2019)

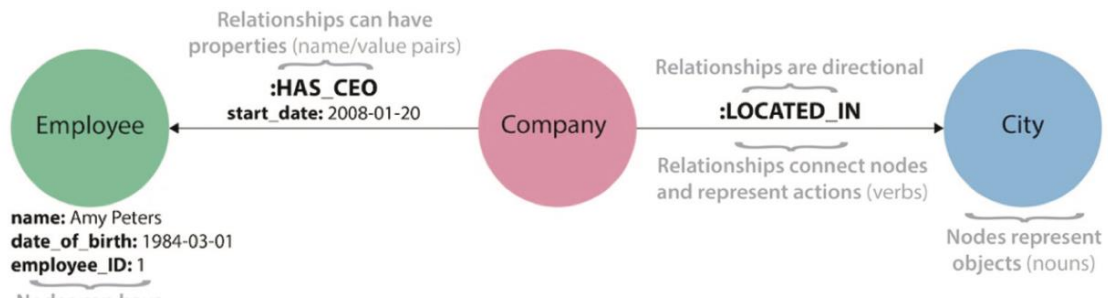


FIGURA 10.- PROPIEDADES DE UN GRAFO DENTRO DE UN CONTENEDOR. (BADER, 2022)

4.7. MongoDB

Esta aplicación NoSQL otorga múltiples soluciones para el almacenamiento de datos, este tiene la cualidad de permitir trabajar desde 0 o más bases de datos actuando a un alto nivel de contención para todos los datos. (Seguin, 2018) Además, podemos acotar que esta base de datos provee mejor rendimiento, más disponibilidad, y un escalado automático; dicha data se estructura en una carpeta compuesta y valores pares, a su vez, se destaca su similaridad con los objetos JSON. (Chauhan, 2019) También, soporta consultas basadas en cursor, estilo unión, tipos de datos complejos, coincidencia de claves, clasificación de filtros, agregaciones, uniones, disparadores, documentos anidados y referencias. (Makris et al., 2021; Treviño-Villalobos et al., 2019) Por último podríamos resaltar como este tiene buena sincronización con los sistemas de propiedades ACID. (Matallah et al., 2021) también el uso del Framework Sistema de infraestructura distribuida (DSI) los cuales se utilizan para que funcionen cientos de bancos de referencias diario. (Ingo & Daly, 2020) Podríamos enumerar las ventajas de la utilización de esta aplicación en “Un alto rendimiento, Alta disponibilidad de datos, su estabilidad horizontal fragmentando y distribuyendo sus datos a un *clúster*. (Chauhan, 2019, pg 90) .

También se podría remarcar que MongoDB gracias a su arquitectura de replicación de datos, tal y como se muestra en la figura (11); se puede apreciar como esta herramienta funciona a través de la documentación orientada al almacenamiento de las bases de datos JSON, las cuales también son llamadas BSON.

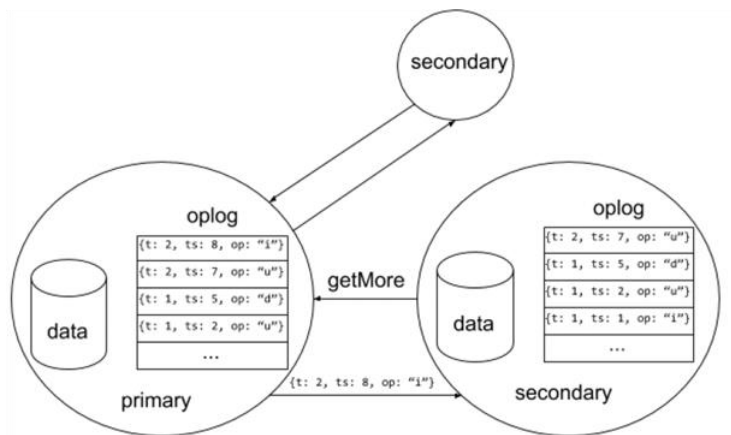


FIGURA 11.- ARQUITECTURA DE MODELO DE REPLICACIÓN OFRECIDO POR MONGODB.

(SCHULTZ ET AL, 2019)

4.8. Web Crawler

Podemos destacar que estos fueron inventados en 1989-1990, uno de los primeras publicaciones de web crawlers fue la de Inktomi en 1996.(Bergman y Popov, 2023) Los web crawler vienen a ser una parte fundamental del web scraping, donde se resalta que las web crawler cruzan la red, a su vez, su núcleo se describe como un elemento de recursión, donde muchas de los contenido URL de las páginas examinan otros datos URL y devolviéndola a sí misma; *ad infinitum*. (Mitchell, 2018) donde podemos destacar la red de *Crawlers Scrapy*, la cual según Chapagain, (2019); la cual define a estos como un recurso Framework abierto y colaborativo para su extracción de datos, rápido, sencillo y extensible, el cual provee de base de datos para futuras aplicaciones de proyectos. Podríamos destacar que la adquisición de datos se da por medio de la recolección de recursos privados, como ventas del cine, o reportes financieros; donde, estas técnicas de análisis web destacan en conjunto con la organización datos. (Khder, 2021) Se puede inferir que estas tres tecnologías, las webs crawlers, los sitios web analizadores de datos, y las plataformas de organización de datos son las más conocidas y las cuales coinciden en el uso del protocolo HTTP. La arquitectura de los web crawler, consiste en 1) la implementación de una solicitud destinada a la creación de una base de datos diseñada para la API, 2) después le sigue la creación de un web crawler funcional el cual cubra con los procesos de rastreo de la información, 3) por último, se debe hacer un análisis estadístico de los datos.(Klymash et al., 2020) En la figura (12) se puede apreciar el prototipo de una arquitectura de un web crawler.

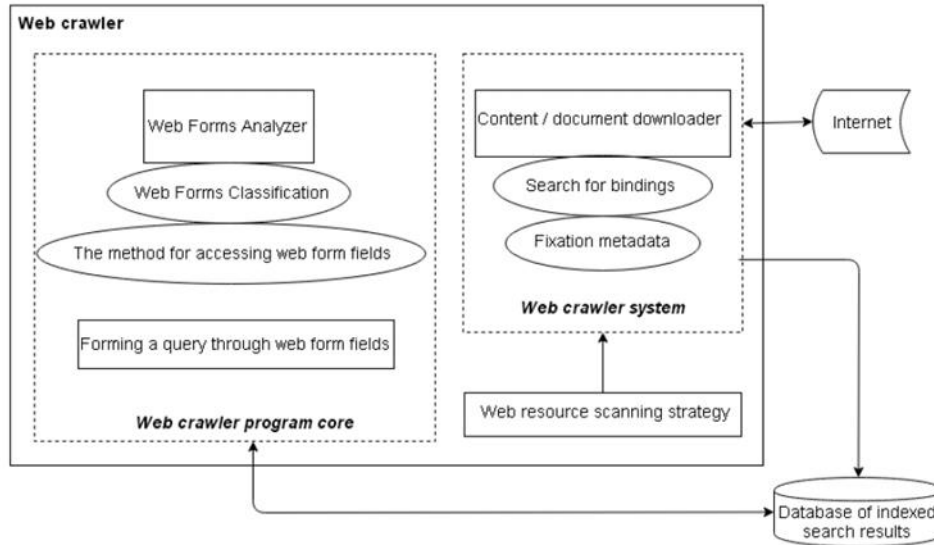


FIGURA 12.- PROTOTIPO DE UN WEB CRAWLER. (KLYMASH ET AL, 2020)

En el mercado según Bergman y Popov, (2023) nos dicen que Nutch, Crawler4j y Mercator. Son en la actualidad los softwares más utilizados en el desarrollo de páginas HTTP. (Abdulrahman et al., 2023) esta herramienta al ser de un carácter de búsqueda dentro de las bases de datos de web, nubes o de Bigdata. Esta ofrece un minado de datos efectivo, simple y acorde a los parámetros que se le imparta. Propuestas como las de Qudus Khan et al., (2021) sobre un algoritmo que sea capaz de realizar estas funciones, concluyeron en que al basar un algoritmo dividido; primero en la generación de una lista de páginas donde se da una autocorrección de las plantillas; y en segundo, esto provocara que al almacenar los datos para futuras actualizaciones sea mucho más simple. En conclusión, estas herramientas son muy útiles al momento de la búsqueda automatizada, sin embargo, se debe tener en cuenta todas las condiciones éticas de esta búsqueda de datos, para lo cual Brewer et al., (2021) nos explica como el llevar un manejo ético sobre el uso de estas herramientas, lleva a que estas tecnologías puedan ser usadas incluso dentro de la criminología, como una herramienta para la resolución de problemas sociales muy complejos, y sin respuesta aparente.

5. Marco metodológico

En esta sección indicaremos que la metodología propuesta para el desarrollo de esta aplicación es la metodología “SCRUM” por ser una metodología ágil, aporta una elevada simplificación y no renuncia a las practicas esenciales para asegurar la calidad del producto.

5.1. Metodología de desarrollo ágil Scrum

El desarrollo de software utiliza el enfoque ágil SCRUM , para ello se comienza creando Sprints en base a los requisitos establecidos con anterioridad , teniendo en cuenta cada uno de las condiciones que el sistema requiere, se necesita desarrollar un cronograma con todas fases, una vez finalizada un Sprint tiene que verificarse que se han completado cada una de las tareas y luego seguir con el siguiente Sprint , Cuando el producto final requiere cambios ,mejores ,o agregar otras funcionalidades es importante saber que se tiene que implementar nuevos requisitos,

- **Asignación de Roles**
 - **ProductOwner:** Ing. Remigio Hurtado, se delegará las funciones que tiene el proyecto
 - **Scrum Master:** Jonnathan Simbaña, es el responsable de describir todas las funcionalidades a los integrantes del proyecto a su respectivo rol.
 - **DevelopmentTeam:** Jonnathan Simbaña, encargado de codificar, probar e implementar la aplicación móvil desarrollada durante la vida del proyecto

5.2. Descripción de las Actividades

- **OE1. Revisar el estado del arte sobre sistemas de recomendación basados en grafos.**

No.	Actividades
1	Revisar la bibliografía referente a recomendaciones en grafos
2	Investigar conceptos de los sistemas de recomendación utilizando grafos.
3	Identificar las principales herramientas o librerías de programación para la implementación de grafos y sistemas de recomendación.

TABLA 1.-ACTIVIDADES A DESARROLLAR DEL OE1

- **OE2. Diseñar e implementar un repositorio de contenidos educativos.**

No.	Actividades
1	Recopilación de datos educativos para el análisis de requerimientos del sistema.
2	Desarrollar un web crawler para la recopilación de información.
3	Diseñar e implementar una infraestructura de grafos junto a sus modelos, lo que mejorara la eficiencia de los algoritmos.

TABLA 2.-ACTIVIDADES A DESARROLLAR DEL OE2

- **OE3.Diseñar e implementar una arquitectura para el sistema de acopio y distribución.**

No.	Actividades
1	Desarrollar una infraestructura basada en contenedores
2	Desarrollar e implementar los contenedores

TABLA 3.-ACTIVIDADES A DESARROLLAR DEL OE3

- **OE4.Desarrollar un módulo inteligente para la recomendación de contenidos educativos**

No.	Actividades
1	Conocer los tipos de recomendaciones junto con sus algoritmos aplicables.
2	Definir el tipo de algoritmo de recomendación.
3	Implementar los algoritmos aplicables al sistema recomendador basado en grafos.

TABLA 4.-ACTIVIDADES A DESARROLLAR DEL OE4

- **OE5. Probar y validar la recomendación basada en encuestas.**

No.	Actividades
1	Conocer los tipos de recomendaciones junto con sus algoritmos aplicables.
2	Definir el tipo de algoritmo de recomendación.
3	Implementar los algoritmos aplicables al sistema recomendador basado en grafos.

TABLA 5.-ACTIVIDADES A DESARROLLAR DEL OE5

- **OE6. Automatizar el despliegue de la plataforma y el sistema recomendador en la nube**

No.	Actividades
1	Desplegar neo4j dentro de un contenedor en la nube
2	Desplegar MongoDB dentro de un contenedor en la nube
3	Desplegar los servicios web en la nube y probar el funcionamiento de las Apis
4	Desplegar la aplicación móvil en Google play

TABLA 6.-ACTIVIDADES A DESARROLLAR DEL OE6

5.3. Fases del scrum

Para este proyecto se aplicó la metodología SCRUM, ya que nos permite la entrega de avances de forma incremental aquí se define los roles principales.

Rol	Cargo
Product owner	Ing. Remigio Hurtado
SCRUM Master	Jonnathan Simbaña
Deveploment Team	Jonnathan Simbaña

TABLA 7.-ROLES DEL PROYECTO

Una vez definido los roles, se procede a definir cada uno de los sprint para la creación del proyecto, para iniciar la creación de los sprint se utiliza como punto referencia los objetivos específicos definidos previamente.

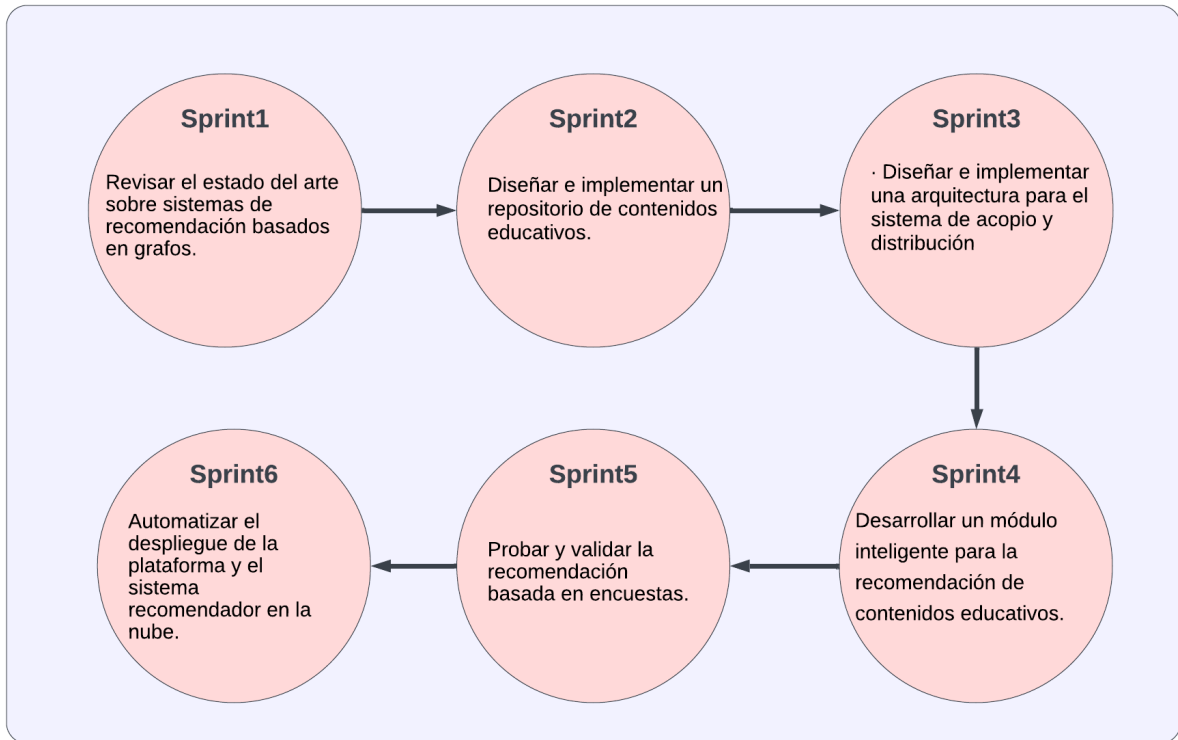


FIGURA 13.-SPRINTS DEL PROYECTO

Con esta planificación se obtuvo la distribución necesaria para llevar los procesos de desarrollo, revisión, validación y reestructuración.

- **Primera Fase**

Consiste en estudiar y comprender el estado del arte de los sistemas de recomendación basados en grafos y las herramientas adecuadas para la implementación de grafos junto a sus algoritmos y de cómo implementar el uso de los nodos con sus relaciones.

A. Estado del arte

En esta sección se estudiarán los diferentes tipos de SR las que son más utilizadas dentro de cada una de ellas se explicaran sus técnicas y los diferentes algoritmos que tienen

B. Sistemas basados en contenido

Por un lado, al referirnos a los sistemas basados en el contenido por medio de analizar contenido similar y crear etiquetas, se puede llegar a calcular su similaridad.

Por ejemplo, "Suppose we have a super set A consisting of all the possible movies in the given platform. Now we have another set, B, consisting of all the movies that a user has watched and given us some information about, such that $\forall B \in A$ and $|B| \ll |A|$, which makes sense as we're taking a considerable amount of data for training our recommendation model. We also introduce another set, C, consisting of the movies the user has yet not rated, $\forall C \in A$ and $|C| \ll |A|$ and a function $f(x)$ that can also be referred to an interest function of the user, which denotes a positive (1) or a negative (0) value and helps us to derive another function $g(x)$, which will estimate the value of $f(x)$ for every element of the set A to effectively recommend appropriate content to the user based on his or her personal choice." [Suponiendo que tenemos un superconjunto de películas A formado por todas las películas posibles en la plataforma, ahora, tenemos un conjunto B formado por las películas calificadas por los usuarios, de tal forma que se podría inferir para cada grupo B que pertenezca a un grupo A, suscitara que el valor absoluto de B sea sumamente inferior que el de A, o expresado de otra manera, $\forall B \in A$, y $|B| \ll |A|$, luego de introducir otro conjunto C, que sería las películas que el usuario aún no ha valorado dándonos como resultado que $\forall C \in A$ y $|C| \ll |A|$, y una función $f(x)$ formada a partir de estos datos se la denominaría función de interés del usuario, donde al indicar un valor positivo (1), y uno negativo (0); nos ayuda al poder derivar a otra función $g(x)$ que estimara el valor de $f(x)$ para cada elemento del conjunto A.] (Rodriguez, 2023)

Así mismo, como se muestra en la figura (14), donde se muestra como aprende el filtro basado en contenido, y, en la figura (15), se aprecia como se calcula la recomendación basada en un filtro de contenido.

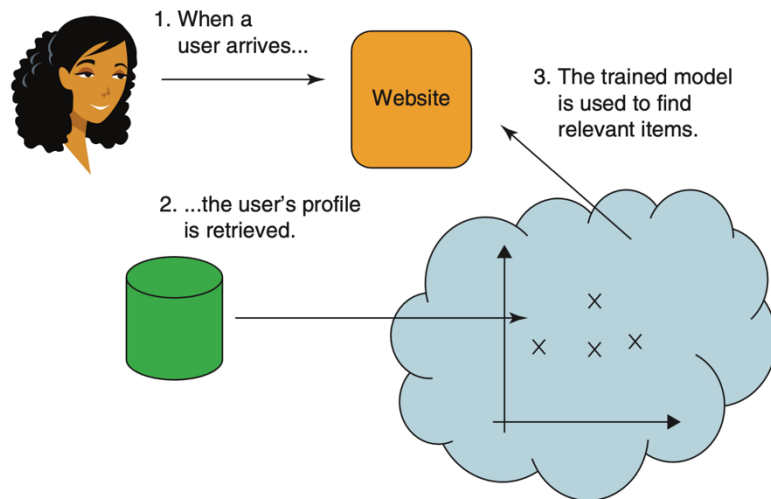


FIGURA 14.-ENTRENAMIENTO DE FILTRO ONLINE. (FALK,2019)

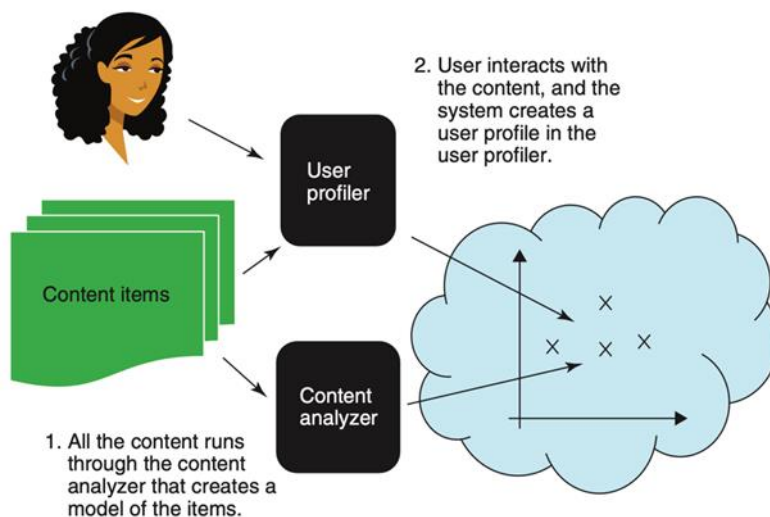


FIGURA 15.-EL RECUPERADOR DE ÍTEMS DEVUELVE UNA RECOMENDACIÓN EN BASE AL FILTRO. (FALK,2019)

C. Sistemas basados en un filtro colaborativo

Estos sistemas básicamente se basan en calificaciones que van realizando los usuarios por medio de encuestas de satisfacción, estos a su vez se dividen en filtro basados en usuarios y en contenido. (Partida, 2021) donde podemos encontrar al *Neighborhood-based filtering*, el cual funciona por medio de un filtro de usuarios y un filtro de contenido.

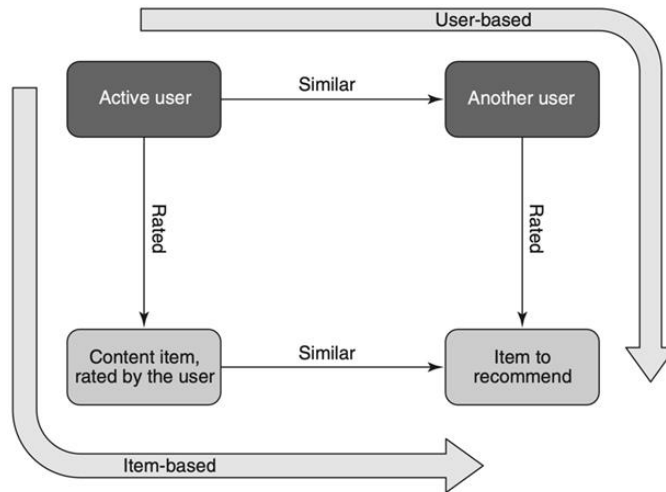


FIGURA 16.- LAS DOS FORMAS DIFERENTES EN LAS QUE TRABAJA EL SISTEMA BASADO EN VECINOS CERCANOS. (FALK, 2019).

Podemos apreciar que este sistema puede ver las similitudes existentes entre los gustos de los usuarios y el contenido que consumen, sin embargo, la similitud que muestran este sistema puede ser utilizado dentro de una formulación estadística, donde el uso de diferentes algoritmos mostrados en la tabla. 1, se puede ver como la correlación de Pearson tiene cierto grado de resonancia en la precisión de los resultados con el coseno ajustado. (Falk, 2019) A su vez, en la figura (4), se puede observar algunos de los elementos en la formulación de la similitud.

Name	Definition	Example
r_{ui}	Rating of user u for item	$r_{sara,star\ trek} = 4,5$ indicates that Sara rated <i>Star Trek</i> 4,5
\bar{r}_u	Average rating of the user u .	The mean of all the ratings given by user u . If Peter rated <i>Star Trek</i> 4, <i>Star Wars</i> 3, then $\bar{r}_{Peter} = 3,5$
$P_{a,b}$	Set of items rated by both a and b	The set of items, if you have Sara and Peter from the previous examples. $P_{sara,Peter} = \{Star\ Trek\}$ since they both rated that movie.

TABLA 8.- ELEMENTOS DE LA FUNCIÓN DE LA SIMILARIDAD. (FALK, 2019)

Algoritmo	Formulación	Relevancia
<i>Jaccard distance</i>	1) $similarity_{Jaccard}(i,j) = \frac{\#users\ that\ bought\ both\ items}{\#users\ who\ bought\ either\ i\ or\ j}$	Al revisar dos grupos de películas (i,j) diferentes, donde al comparar si es que se ha vendido o no, se lograra realizar una recomendación en base a los datos de cada grupo de usuarios.
<i>Euclidian norm (L₂-norm)</i>	1) $distance(Sara,Pietro) = \ r_{sara} - r_{pietro}\ _2 = \sqrt{\sum_{i=1}^n r_{sara,i} - r_{pietro,i} ^2}$ 2) $Root\ Mean\ Squared\ Error\ (RMSE) = \frac{1}{n} \sum_{i=1}^n r_{sara,i} - r_{pietro,i} ^2$	1) Se debe calcular la distancia entre los puntos, 2) luego debemos obtener el error cuadrático medio (RMSE).
<i>Coseno de similitud</i>	1) $sim(i,j) = \frac{r_i \cdot r_j}{\ r_i\ _2 \ r_j\ _2} = \frac{\sum_u r_{i,u} r_{j,u}}{\sqrt{\sum_u r_{i,u}^2} \sqrt{\sum_u r_{j,u}^2}}$ $sim(i,j) = \frac{\sum_u (r_{i,u} - \bar{r}_u)(r_{j,u} - \bar{r}_u)}{\sqrt{\sum_u (r_{i,u} - \bar{r}_u)^2} \sqrt{\sum_u (r_{j,u} - \bar{r}_u)^2}}$ 2)	1) Para poder crear vectores en una matriz, es necesario obtener su coseno de similitud. 2) sin embargo para comparar diferentes calificaciones de usuarios en toda la red, se debe utilizar la formula del coseno ajustado.
<i>Pearson's correlation coefficient</i>	$sim(i,j) = \frac{\sum_{e \in U} (r_{i,u} - \bar{r}_i)(r_{j,u} - \bar{r}_j)}{\sqrt{\sum_{e \in U} (r_{i,u} - \bar{r}_i)^2} \sqrt{\sum_{e \in U} (r_{j,u} - \bar{r}_j)^2}}$	Esta correlación al mostrar cuan diferentes son algunos puntos correlaciones entre gustos de usuarios, donde se trata de identificar cual tendencia es más parecida con la otra.

TABLA 9.- TIPOS DE ALGORITMOS UTILIZADOS EN LOS SISTEMAS DE FILTRO

COLABORATIVO (FALK,2019)

- **Segunda Fase**

Consiste en el diseño e implementación de un repositorio de contenidos educativos con los grafos es decir vamos a categorizar y diseñar cada uno de los nodos con sus respectivos datos, ya sean archivos multimedia, artículos, frases, actividades y recursos. Para cumplir con los objetivos de esta fase lo primero que se tiene que tomar en cuenta es la recopilación de datos de contenidos educativos referente a niños.

A. Recopilación de datos educativos para el análisis de requerimientos

La información recolectada se extrajo de Wikinclusion.org que es una base de conocimiento que ofrece software, vídeos y material imprimible para facilitar la comunicación y el conocimiento a todas las personas, tengan o no discapacidad, dificultades de aprendizaje o diversidad funcional. Gestionada por Creática FREE Iberoamericana para la Cooperación. Uruguay y la Fundación Red Especial España con la colaboración de la Cátedra UNESCO. Ecuador y CIAPAT Argentina.(Montoya R).

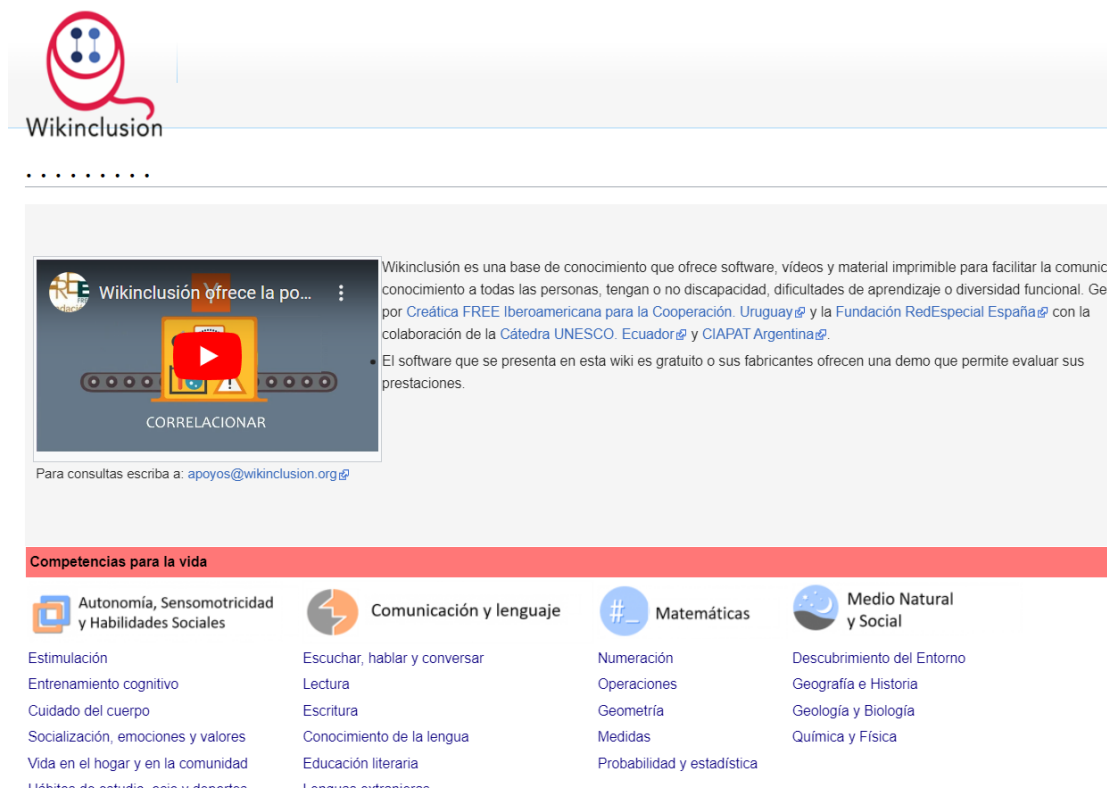


FIGURA 17.-PÁGINA WIKIINCLUSION.ORG (MONTROYA.R)

B. Desarrollar un web crawler para la recopilación de información

Se procedió a crear un web crawler para la captura de información en esta parte lo que hacemos es utilizar Python y la librería beautifulsoup4 ya que nos permite extraer datos de archivos HTML y XML. En la librería beautifulsoup4 es importante tener en cuenta que para extraer la información hay que tener dos puntos claves, que son si extraer toda la información de la página completa o extraer la información de acuerdo con las necesidades que necesitemos.

Para el sistema vamos a extraer la información que nosotros necesitamos, en este caso es necesario saber en qué tipo de elemento HTML esta la información junto a sus propiedades ya que con eso se obtendrá una búsqueda más exacta de la información.



FIGURA 18.-. ESTRUCTURA DE UN ELEMENTO HTML JUNTO A SUS PROPIEDADES (RAMOS, G, 2021).

```

<!DOCTYPE html>
<html class="client-js" lang="es" dir="ltr" wtx-context="3148CB0C-96A4-4ED4-A3E2-BC0D4E28B370">
  <head> </head>
  <body class="mediawiki ltr sitedir-ltr mw-hide-empty-elt ns-0 ns-subject page-Página_principal r
ootPage:Página_principal skin-vector action-view">
    <div id="mw-page-base" class="noprint"></div>
    <div id="mw-head-base" class="noprint"></div>
    <div id="content" class="mw-body" role="main">
      <a id="top"></a>
      <div class="mw-indicators mw-body-content"></div>
      <h1 id="firstHeading" class="firstHeading" lang="es">. . . . .</h1>
      <div id="bodyContent" class="mw-body-content">
        <div id="siteSub" class="noprint">De wikiclusión</div>
        <div id="contentSub"></div>
        <div id="jump-to-nav"></div>
        <a class="mw-jump-link" href="#m-head">Ir a la navegación</a>
        <a class="mw-jump-link" href="#s-search">Ir a la búsqueda</a>
        <div id="mw-content-text" lang="es" dir="ltr" class="mw-content-ltr">
          <div class="mw-parser-output">
            <table style="width:100%; background:#F5F5F5; cellpadding="6"></table>
            <table style="width:100%; border:1px solid #FFFFFF; background:#FFF777; color:#000000;"
              align="left;" cellpadding="4" cellspacing="0"></table>
          <table>
            <tbody>
              <tr></tr>
              <tr>
                <td>== $@
                  <a href="/index.php/Estimaci%3B3j" title="Estimación">Estimación</a>
                </td>
                <td></td>
                <td></td>
                <td></td>
              </tr>
            </tbody>
          </table>
        </div>
      </div>
    </div>
  </body>
</html>

```

FIGURA 19.-EJEMPLO DE MODO INSPECCIÓN PARA IDENTIFICAR LOS ELEMENTOS HTML

Una vez identificada cada uno de los elementos de los elementos he información que se quiere extraer el resultado sería el siguiente:

Nombre	Elemento HTML	Clase del elemento	Total, de información recolectada
Áreas	<table><tbody>	“ ”	7
Bloques	<tr><td>		27
Competencia	<a>		151
Habilidades			1640
Videos	<iframe>	“”	4353
Software	<p>	“”	Varía de acuerdo con la petición por habilidad

TABLA 10.-TABLA DE LA CANTIDAD DE INFORMACIÓN RECOLECTADA

C. Desarrollar e implementar una infraestructura de grafos junto a sus modelos

En esta actividad se diseñó un modelo para la creación de los nodos y aristas, para la creación de este modelo se utilizó para el apartado de grafos en NEO4J y para la base de datos MONGODB

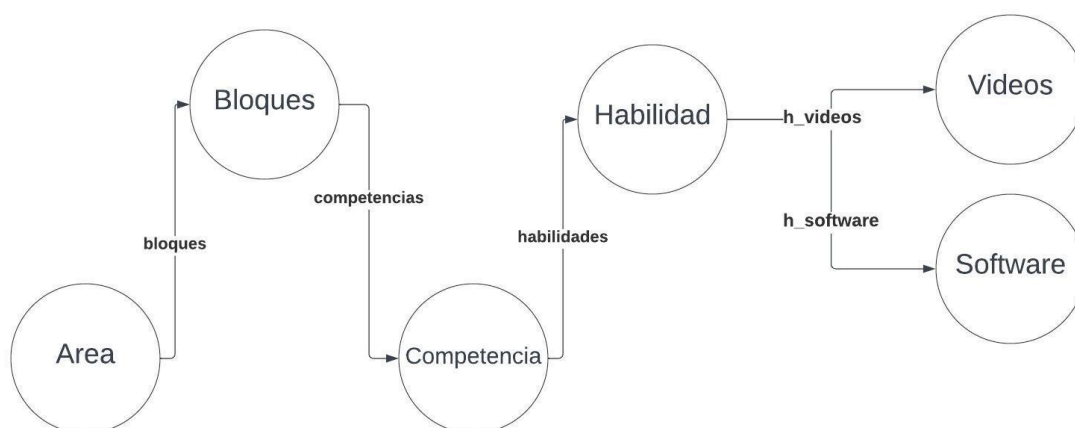


FIGURA 20.-. CREACIÓN DE LA INFRAESTRUCTURA DE GRAFOS CON LOS MODELOS

- **Tercera Fase**

En esta fase está conformada por el diseño de la capa de acceso a los datos juntamente con el sistema de recomendación a través de neo4j y MongoDB y el Framework FastApi esta parte es clave ya que con esto nos ayudara el funcionamiento del sistema.

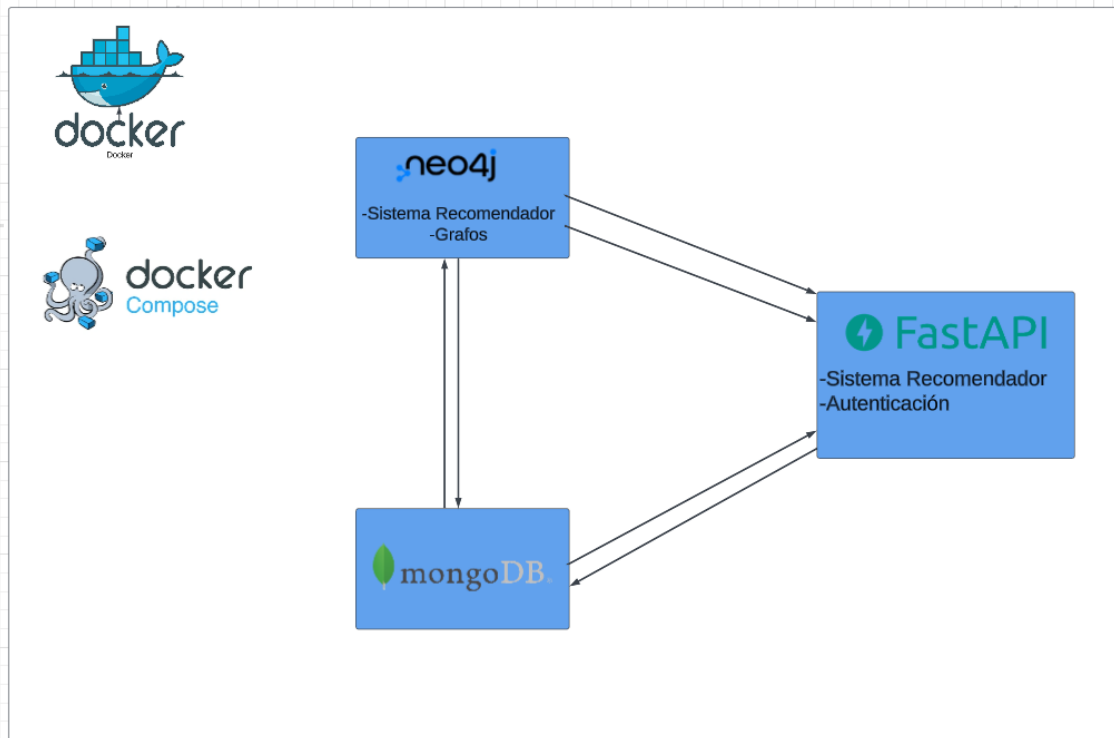


FIGURA 21.-LÓGICA DE NEGOCIO

Para empezar en la implementación del sistema lo primero que se tiene que definir son las tecnologías con las que se va a usar para la construcción del sistema.

Para la base de datos se utilizará MongoDB es su última versión, se hace uso de esta herramienta para el almacenamiento de los datos. Se utilizará el paquete pymongo ya que contiene los controladores que nos permite comunicar Python con MongoDB.

A continuación, se mostrará la implementación de los contenedores.

A. Contenedor Neo4j

```
version: "3.8"
services:
  neo4j:
    image: neo4j:4.4.5-enterprise
    ports:
      - "7474:7474"
      - "7687:7687"
    restart: unless-stopped
    container_name: tesis-docker-final2
    environment:
      - NEO4J_AUTH=neo4j/Admin12345
      - NEO4J_apoc_export_file_enabled=true
      - NEO4J_apoc_import_file_enabled=true
      - NEO4J_ACCEPT_LICENSE_AGREEMENT=yes
      - NEO4J_apoc_import_file_use__neo4j__config=true
      - NEO4JLABS_PLUGINS=["graph-data-science"]
      - server.http.listen_address
      - NEO4J_dbms.security.procedures.unrestricted=gds.*
      - dbms.connector.bolt.listen_address=0.0.0.0:7687
      - dbms.connector.http.enabled=true
      - dbms.connector.https.enabled=true
    volumes:
      - ./db/data:/data
      - ./db/conf:/conf
      - ./db/logs:/logs
      - ./db/plugins:/plugins
```

FIGURA 22.-CREACIÓN DEL CONTENEDOR UTILIZANDO DOCKER-COMPOSE

Dentro del archivo compose es importante poner el plugin para que se instale automáticamente sin eso no podríamos utilizar los algoritmos de recomendación y es importante habilitar para que cualquier dirección IP pueda conectarse ya sea local o en la nube.

B. Contenedor MongoDB

Luego de que toda la configuración este correctamente, se configura con el código para el despliegue dentro de un Docker, se creara un Docker-compose, si es necesario se definen los requerimientos que necesite el MongoDB para su uso y se procede a construir el contenedor.

```

version: '3.5'

services:
  mongo:
    image: mongo
    restart: always
    ports:
      - "27017:27017"
    container_name: mongo-0
    hostname: host-mongo
    environment:
      - MONGO_INITDB_ROOT_USERNAME=root
      - MONGO_INITDB_ROOT_PASSWORD=example

  volumes:
    - "./data:/data/db"
    - "./data/log:/var/log/mongod"

  mongo-express:
    image: mongo-express
    container_name: mongo-express
    restart: always
    ports:
      - 8081:8081
    environment:
      - ME_CONFIG_MONGODB_ADMINUSERNAME=root
      - ME_CONFIG_MONGODB_ADMINPASSWORD=example
      - ME_CONFIG_MONGODB_SERVER=mongo
      #ME_CONFIG_MONGODB_URL: mongod://root:example@mongo

```

FIGURA 23.-CREACIÓN DEL CONTENEDOR UTILIZANDO DOCKER-COMPOSE

Dentro del Docker-compose tenemos las configuraciones importantes que son en que puertos se va a correr el contenedor, el usuario y la contraseña del usuario.

C. Contenedor Servicios web

Después de que este todo funcionando correctamente lo siguiente es crear un Docker-compose para subir el proyecto en este caso es importante generar un archivo de requerimientos para que cuando se suba al contenedor Docker se instale los requerimientos.

```

1  version: '3.5'
2
3  services:
4    fastapi:
5      build:
6        context: fastapi
7        dockerfile: Dockerfile
8      image: fastapi:latest
9      container_name: fastapi
10     command: uvicorn app.main:app --host 0.0.0.0 --port 8000
11     restart: always
12     ports:
13     - "8000:8000"

```

FIGURA 24.-CREACIÓN DEL CONTENEDOR UTILIZANDO DOCKER-COMPOSE

- **Cuarta Fase**

En esta fase desarrollaremos un módulo inteligente para la recomendación del sistema de contenido educativo, detallaremos cada una de las actividades que se desarrolló.

Existen varios algoritmos que nos permiten el uso de sistemas de recomendación basados en filtros colaborativos y basados en contenido para este proyecto se utilizó los algoritmos de similitud.

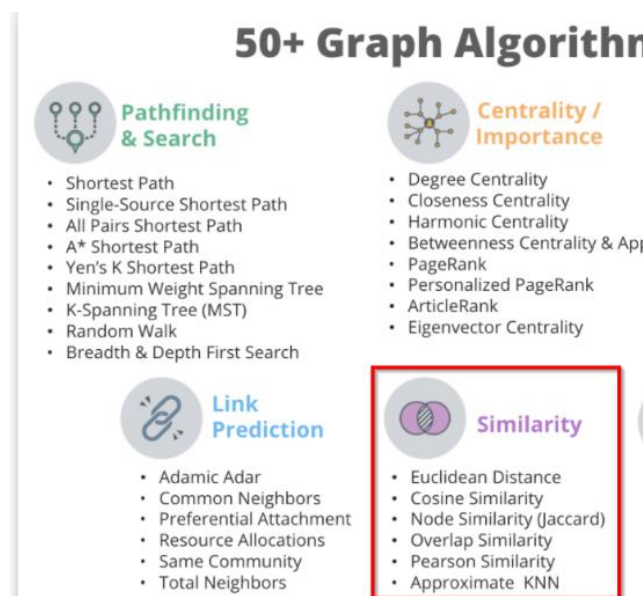


FIGURA 25.-ALGORITMOS DE SIMILARIDAD

Dentro de esta categoría se utilizó el algoritmo de “Cosine Similarity” ya que se encarga de encontrar el ángulo de dos vectores, es decir cumplen una con la condición de ser máximamente similares, si los dos vectores tienen la misma orientación tienen una similitud de 1 y dos vectores opuestos tienen una similitud de 0.

Para implementar los diferentes tipos de algoritmos en neo4j es importante instalar el paquete GDS, ya que contiene cada una de las herramientas que nos permitirán el uso de esta, para la instalación dentro de un Docker file es necesario importar agregar esta línea de comando.

`“NEO$JLABS_PLUGINS=[‘graph-data-science’]”`

Para instalar de manera local es necesario descargar el programa “Neo4j Desktop”, una vez creada la base de datos, tenemos que irnos a la pestaña de plugins y se procede a instalar el GDS.

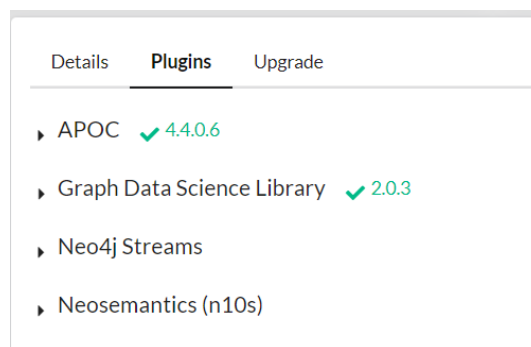


FIGURA 26.-INSTALACIÓN DE PLUGINS

- **Quinta Fase**

En esta fase se diseñó un plan de experimentación que es la validación de las recomendaciones basados en contenido. Dentro de la pantalla de inicio Figura (35), nos saldrá un apartado con todas las recomendaciones de acuerdo con los gustos del usuario.

Para las validaciones de estas recomendaciones utilizamos encuestas de satisfacción en las cuales nos permite identificar si al usuario le gustó o no la recomendación para ello tenemos las siguientes preguntas.

- ¿Te gustó la recomendación dada?
- ¿Los videos te ayudaron en el aprendizaje?
- ¿El software te ayudo a entender mejor la habilidad?
- ¿Recomendarías esta habilidad?

Las valoraciones de satisfacción de preguntas son las siguientes:

Estado	Valoración
Excelente	≥ 5
Muy bueno	≥ 4 y < 5
Bueno	≥ 3 y < 4
Regular	≥ 2 y < 3
Malo	≥ 1 y < 2

TABLA 11.-TABLA DE VALORACIONES

- **Sexta Fase**










Una vez que todo esté funcionando correctamente de manera local, el siguiente paso utilizar un servicio en la nube en este caso utilizamos Digital Ocean, y para la aplicación móvil se sube a la consola de Play Store.

Para creación de los contenedores en Digital Ocean, hay que crear Droplets, dentro del asistente de configuración hay q elegir la región en la que está el servidor, luego hay que seleccionar una imagen, que dentro de estas imágenes están los sistemas operativos y la versión que se necesite y al último seleccionar el plan, es decir la cantidad de memoria, RAM y CPU.

Create Droplets

Droplets are virtual machines that anyone can setup in seconds. You can use droplets, either standalone or as part of a larger, cloud based infrastructure.

Choose Region

 New York	 San Francisco	 Amsterdam
 Singapore	 London	 Frankfurt
 Toronto	 Bangalore	 Sydney

Datacenter

San Francisco • Datacenter 3 • SFO3



Tip: Select the datacenter closest to you or your users

Avoid any potential latency by selecting a region closest to you - a region is a geographic area where we have one or more datacenters.

[Dismiss](#)

VPC Network - default-sfo3 DEFAULT

FIGURA 27.-CREACIÓN DEL DROPLET

Es importante saber las características que necesite cada herramienta para que funcione de manera óptima, una vez finalizado todos los pasos que pide Digital Ocean para la creación del Droplet como podemos observar en la figura (27)

docker-neo4j		version1	
Image	Ubuntu 22.04 (LTS) x64	Region	NYC1
Size	2 vCPUs 4GB / 25GB Disk (\$42/mo) Resize	IPv4	157.230.15.68
		IPv6	Enable
		Private IP	10.116.0.2
		VPC	default-nyc1

FIGURA 28.-RESULTADO DE LA CREACIÓN DEL DROPLET

Es importante saber las características que necesite cada herramienta para que funcione de manera óptima.

5.4. Arquitectura

En esta parte se diseña la estructura de todo el proyecto, donde se especifica la distribución, comunicación y las tecnologías que se van a implementar para el desarrollo del sistema, aquí nos ayudara a visualizar de forma general como va a ayudar a visualizar de cómo está estructurado el proyecto.

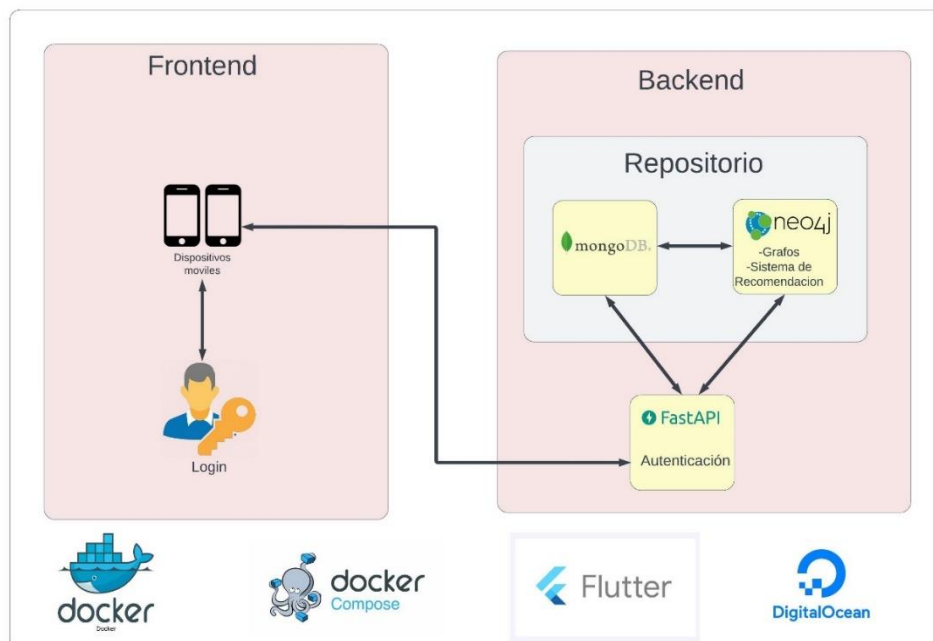


FIGURA 29.-ARQUITECTURA DEL SISTEMA

A. Backend

Dentro del backend tiene como objetivo las operaciones y funcionalidad del sistema, puede ser accedido por los usuarios, dependiendo de los permisos, es conocida como capa de negocios que será nuestro caso alojando nuestro repositorio de contenido educativo.

- **Lógica de negocio (Repositorio):** En esta capa se encuentra la base de datos y la información del contenido educativo que están dentro de MongoDB y neo4j.
- **Sistema de autenticación (FastApi):** en esta capa esta encargada de permitir el ingreso al usuario dependiendo el rol que tenga cada usuario, con la ayuda de Python y su Framework FastAPI.
- **Sistema recomendador y grafos(neo4j):** Es el motor que permite la creación de grafos y del sistema recomendador en base al contenido, ya que gracias al uso de neo4j es fácil implementar cualquier algoritmo de similitud, conjunto con la información de MongoDB que contiene la base de datos.

B. Frontend

Dentro del Frontend contiene la presentación al usuario es decir las funcionalidades dadas por el desarrollador, tanto como el ingreso o registro del sistema que se tiene con la autenticación.

Para la construcción del sistema se empleó una arquitectura DDV(Data-Domain-View), ya que con esta arquitectura tiene un conjunto de patrones de diseño que sirven para proveer la facilidad al proceso a la hora de la creación del software, así como la escalabilidad y el mantenimiento, ya que en un futuro nos permite implementar nuevos cambios a los componentes y no afecta entre si a los componentes, ya que solo va a afectar a la lógica del software.

Los componentes de la arquitectura son:

- **Data:** Esta capa es la encargada de comunicar las dependencias externas de la aplicación para la obtención de los datos, es decir el consumo de los servicios REST, SOAP, Firebase, etc.
- **Domain:** Esta capa es la encargada de la lógica de negocio de la aplicación, la cual se encuentra el código, es decir es el código reutilizable en base a los casos que presente cada funcionamiento del software.
- **View:** Esta capa es la encargada de presentar la información al usuario final, en este caso es todo el código que se encuentra desarrollado en Flutter ya que nos permite

emplear el uso de páginas y widgets y el uso de navegación como es la entrada y salida de los datos.

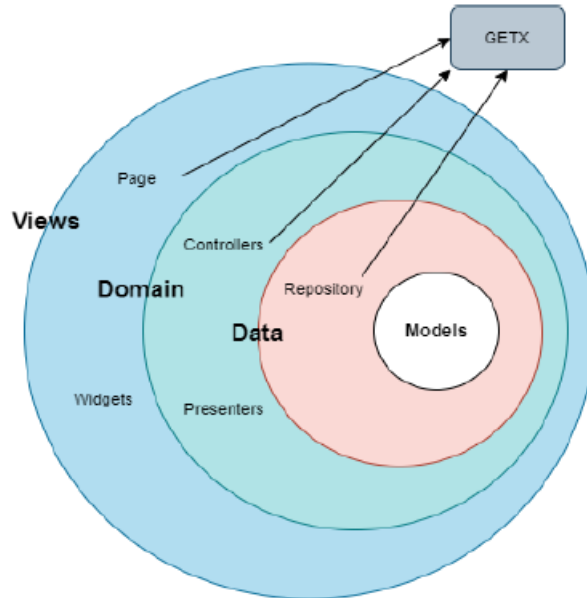


FIGURA 30.-ARQUITECTURA FRONTEND

Dentro de la arquitectura se emplea un gestor de estados q es GetX , al implementarlo con este Framework, se maneja los estados, enrutamientos y la inyección de dependencias de manera fácil y eficiente, El manejador de estados Getx se utiliza dentro de la capa domain por medio de este controlador evita el uso de StatefulWidget, ya que provee su propio ciclo de vida ,con la finalidad de separar la lógica del negocio con la vista, ya que si sucede un algún tipo de cambio dentro del controlador ,la vista se actualice automáticamente.

6. Resultados

En esta sección se presentará todos los resultados que hemos obtenido en el proceso de desarrollo de esta aplicación, y para ello presentaremos las especificaciones que necesita el sistema.

6.1. Requerimientos

6.1.1. Neo4j para uso local

CPU	Intel x86-x64 Core i3 mínimo, Core i7 recomendado. AMD x86-x64, Mac ARM.
Memoria	Mínimo 2 GB, 16 GB o más recomendados.
Almacenamiento	Mínimo SATA de 10 GB, SSD con SATA Express o NVMe recomendado.

TABLA 12.-REQUERIMIENTOS MÍNIMOS PARA EL USO DE NEO4J PARA USO PERSONAL

6.1.2. Neo4j para uso en la nube

CPU	2vCPU mínimo, 16 + recomendado.
Memoria	2GB mínimo. Los requisitos reales dependen de las cargas de trabajo. En algunos casos, se recomienda usar instancias con memoria que se adapten al tamaño del gráfico en uso.
Almacenamiento	Almacenamiento en bloque mínimo de 10 GB, se recomienda el SSD NVMe adjunto. El tamaño de almacenamiento depende del tamaño de las bases de datos.

TABLA 13.-REQUERIMIENTO PARA EL USO DE NEO4J EN LA NUBE

6.2. Diseño

6.2.1. Diseño de la Aplicación

A. Diseño de los nodos en neo4j

Los recursos obtenidos se almacenan en la base de datos neo4j y MongoDB que permiten la creación de relaciones para generar búsquedas, aplicar algoritmos, obtener recomendaciones y recursos de su base de datos. Para el sistema de grafos se crearon 7 entidades la única diferencia con MongoDB es que sus atributos son solo los códigos de las relaciones:

- **Área:** Se creó el nodo con un atributo que es su código.
- **Bloque:** Se creó el nodo con dos atributos que es su código y el código del área.
- **Competencia:** Se creó el nodo con dos atributos que es su código y el código del bloque.
- **Habilidad:** Se creó el nodo con dos atributos que es su código y el código de la competencia.
- **Usuario:** Se creó el nodo con dos atributos que es su código y el código del usuario

Cuando se definió las entidades y nodos que se van a utilizar procedimos a la creación del diseño de los nodos y aristas, utilizando el lenguaje Cypher Query de neo4j procedimos a la creación del script que se encuentra disponible en el (Anexo1.), para ello vamos a visualizar la creación completa de los nodos y las aristas.

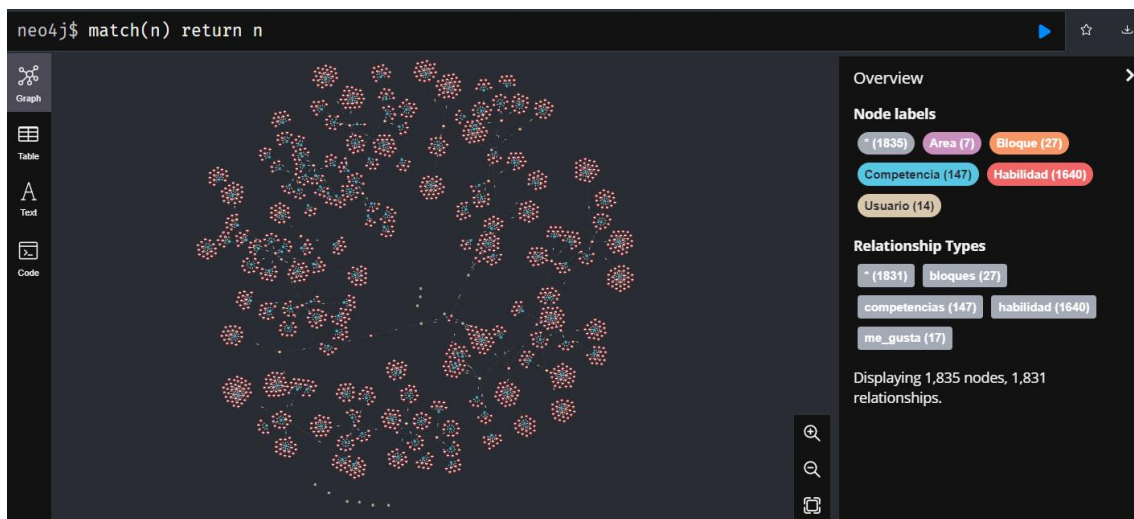


FIGURA 31.-VISUALIZACIÓN DE NODOS CREADOS

B. Diseño de las colecciones en MongoDB

Para la base de datos en MongoDB se crearon 7 entidades que en MongoDB hacen referencia a documentos(colecciones), esto quiere decir que, en vez de guardar datos en registros, guarda los datos en documentos, ya q son almacenados en BSON (similar a JSON), aquí obtendremos los datos completos de cada uno de los documentos que son:

- **Área:** Colección de datos
 - **_id:** dato que se genera automáticamente
 - **código:** Código del área

- **nombre:** Nombre del área
- **imagen** nombre de una imagen que se utilizara como fondo
- **Bloque:** Colección de datos
 - **_id:** dato que se genera automáticamente
 - **código:** Código del bloque
 - **nombre:** Nombre del bloque
 - **codigo_area:** Código del área a la que pertenece
- **Competencia:** Colección de datos
 - **_id:** dato que se genera automáticamente
 - **codigo:** Código de la competencia
 - **codigo_bloque:** Código del bloque al que pertenece
 - **nombre:** Nombre de la competencia
- **Habilidad:** Colección de datos
 - **_id:** dato que se genera automáticamente
 - **codigo:** Código de la habilidad
 - **nombre:** Nombre de la habilidad
 - **cod_competencia:** Código de la competencia a la que pertenece
 - **identificador:** código identificador de tiene la habilidad
- **Usuario:** Colección de datos
 - **_id:** dato que se genera automáticamente
 - **usuario:** Nombre del usuario
 - **Nombre:** Nombres del usuario
 - **Apellido:** Apellidos del usuario
 - **correo:** Correo electrónico del usuario
 - **rol:** Rol que tiene ese usuario (Usuario-Administrador)
 - **contraseña:** Contraseña del usuario
- **Video:** Colección de datos
 - **_id:** dato que se genera automáticamente
 - **codigo:** Código del video
 - **link:** Enlace del video
 - **descripción:** Titulo del video
 - **identificador:** Identificador de la habilidad a la que pertenece
- **Encuesta:** Colección de datos

- **_id:** dato que se genera automáticamente
- **pregunta1:** valor que dio el usuario a esa pregunta
- **pregunta2:** valor que dio el usuario a esa pregunta
- **pregunta3:** valor que dio el usuario a esa pregunta
- **pregunta4:** valor que dio el usuario a esa pregunta
- **codigo_usuario:** Codigo del usuario que hizo la encuesta
- **código_habilidad:** Codigo de la habilidad de a que se hizo la encuesta

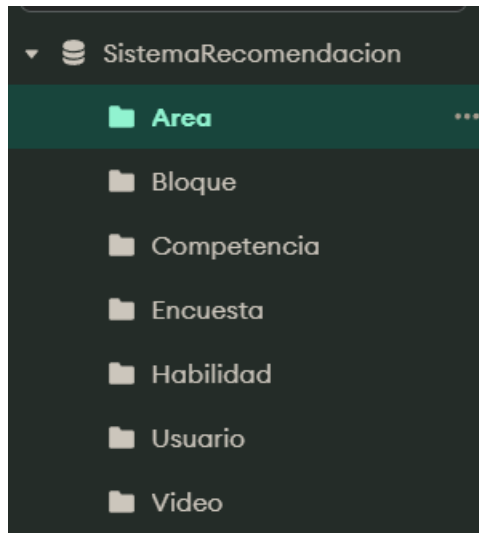


FIGURA 32.-CREACIÓN DE LAS COLECCIONES EN MONGODB

6.2.2. Diseño de la aplicación móvil

Se hizo una aplicación llamativa para que sea amigable con el usuario, A continuación, se mostrará las interfases que se desarrolló para la aplicación móvil:

- **Login:** En esta pantalla permite ingresar a la aplicación por medio de las credenciales, en caso de no tener un usuario, se le presentara un botón en el cual puede realizar dicho proceso.

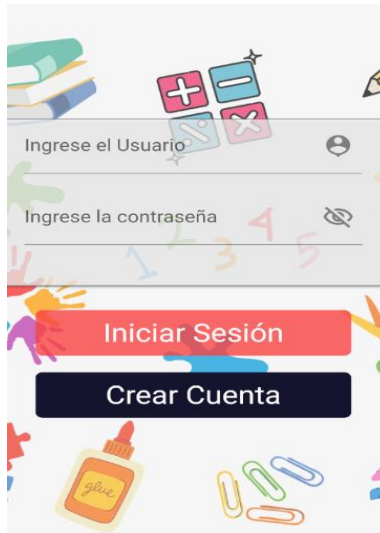


FIGURA 33. PANTALLA DEL LOGIN

- **Registro Usuario:** En esta pantalla permite al usuario realizar el proceso de creación de cuenta en la aplicación, para ellos se solicitan los datos personales del usuario que son nombre, apellido, usuario, contraseña, etc. Una vez creado el usuario podrá ingresar al sistema.



FIGURA 34. PANTALLA DE REGISTRO

- **Home:** En esta pantalla nos muestra principalmente el sistema de recomendación junto con el listado de áreas que tiene el sistema, cuando se da clic en cualquier área procede a pasar a la siguiente pantalla que es el de bloques.



FIGURA 35.PANTALLA HOME

En la pantalla de inicio tiene dos apartados más que son:

- **Favoritos:** En esta pantalla lo que nos va a permitir es visualizar un listado de las habilidades que han sido seleccionadas como favoritas para el usuario, en el cual tiene un botón en el cual va a permitir eliminar la habilidad en caso de que ya no le guste.

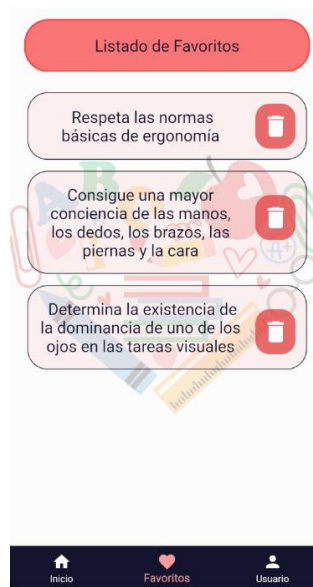


FIGURA 36.PANTALLA DE FAVORITOS

- **Perfil:** En esta pantalla nos mostrará los datos del usuario y también permitirá al usuario modificar algún dato de ellos.

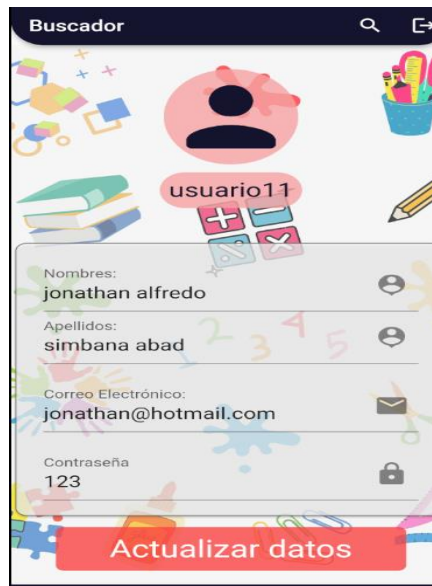


FIGURA 37. PANTALLA DEL PERFIL DEL USUARIO

- **Bloque:** En esta pantalla nos muestra un listado de los bloques al que pertenece el área seleccionada por el usuario, en caso de dar clic en uno de los bloques, se procederá a la siguiente pantalla que es de competencias.

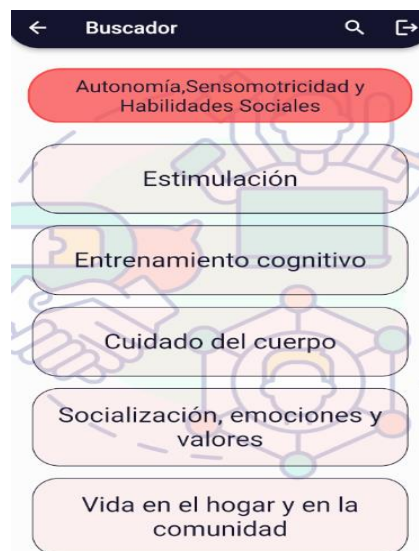


FIGURA 38. PANTALLA DE BLOQUES

- **Competencia:** En esta pantalla nos muestra un listado de competencias al que pertenece al bloque seleccionada por el usuario, en caso de dar clic en una de las competencias, se procederá a la siguiente pantalla que es de habilidades.



FIGURA 39.PANTALLA DE COMPETENCIA

- **Habilidad:** En esta pantalla nos muestra un listado de habilidades a las que pertenecen en la competencia seleccionada por el usuario, en caso de dar clic en una de las habilidades, se procederá a la siguiente pantalla que es el detalle.

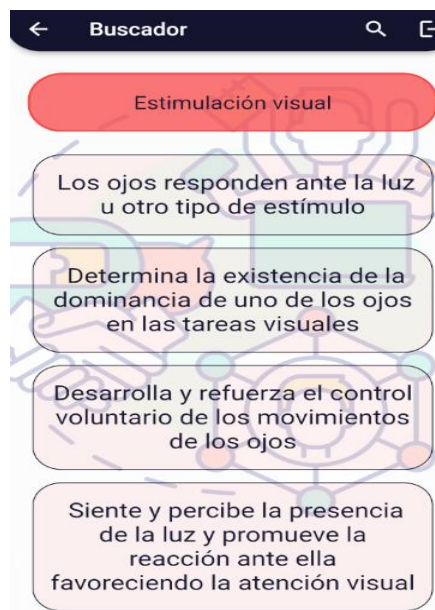


FIGURA 40.PANTALLA DE HABILIDAD

- **Detalle Habilidad:** En esta pantalla nos muestra la información de la habilidad que el usuario ha seleccionado, dispone de un botón en el cual permite al usuario dar me gusta en caso de que le haya gustado la habilidad. En esta pantalla se dispone de dos apartados que son:

- **Videos:** En esta pantalla se muestra un listado de videos relacionados con la habilidad



FIGURA 41.PANTALLA DETALLE DE HABILIDAD

- **Software:** En esta pantalla se muestra un listado de software relacionados con la habilidad.

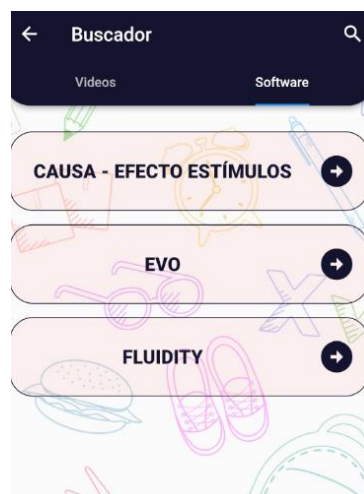


FIGURA 42,-PANTALLA DE LOS SOFTWARE DISPONIBLES

6.2.3. Diseño del modelo de recomendación

Para el sistema de recomendación se diseñó que dentro de la aplicación en el apartado de habilidades tenemos una opción en la cual nos permite darle me gusta y calificar entre 1 y 5 la habilidad de acuerdo con los gustos del usuario, después cuando se da me gusta automáticamente se crea la relación entre el usuario y la habilidad, ya que con esto nos permitirá realizar el sistema de recomendación como podremos ver en el siguiente ejemplo:

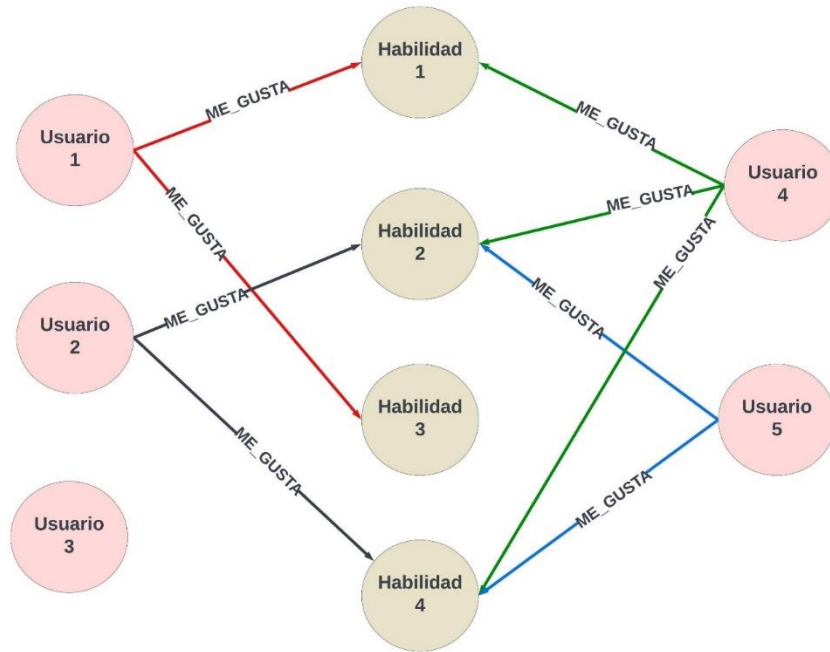


FIGURA 43.-EJEMPLO DEL MODELADO EN NEO4J

Para entender el funcionamiento del sistema de recomendación, es importante explicar paso a paso de cómo está funcionando el modelo basado en grafos, cuando el usuario ingresa a la aplicación el usuario tendrá que navegar de la manera como lo muestra la Figura (44). Es decir el usuario accede a las habilidades de acuerdo a las diferentes áreas y dentro de cada área existen un conjunto de bloques a la vez dentro de cada bloque existen varias competencias y dentro de las competencias existen varias habilidades en las cuales el usuario podrá acceder a un conjunto de videos e información o diferentes materiales de apoyo, aquí el usuario podrá dar un like y calificar de 1 a 5 que tanto le gusto esa habilidad.

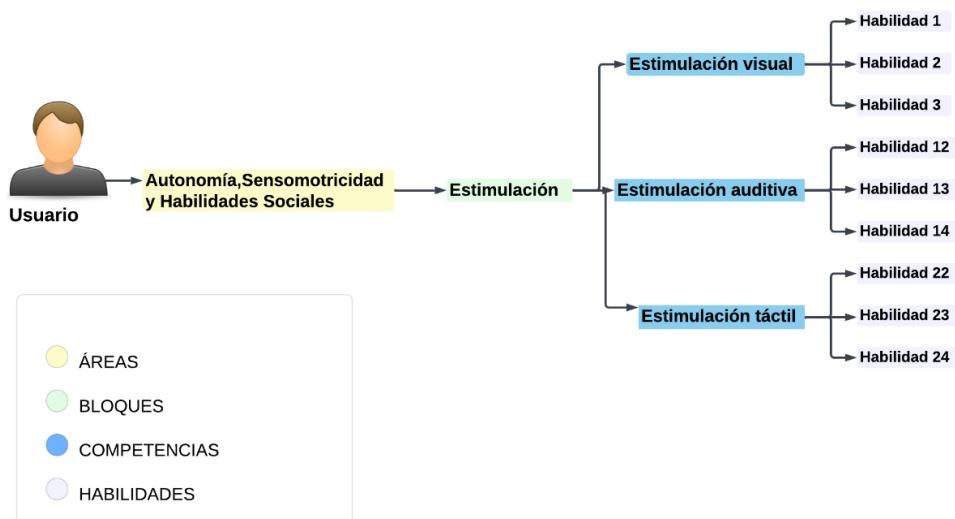


FIGURA 44.-EJEMPLO DEL MODELO BASADO EN GRAFOS

Una vez entendido como se hizo el modelado en la base de datos basados en grafos y como el usuario tiene que acceder a las habilidades es momento de explicar cómo funciona el sistema recomendador en base al contenido y donde se encuentra.

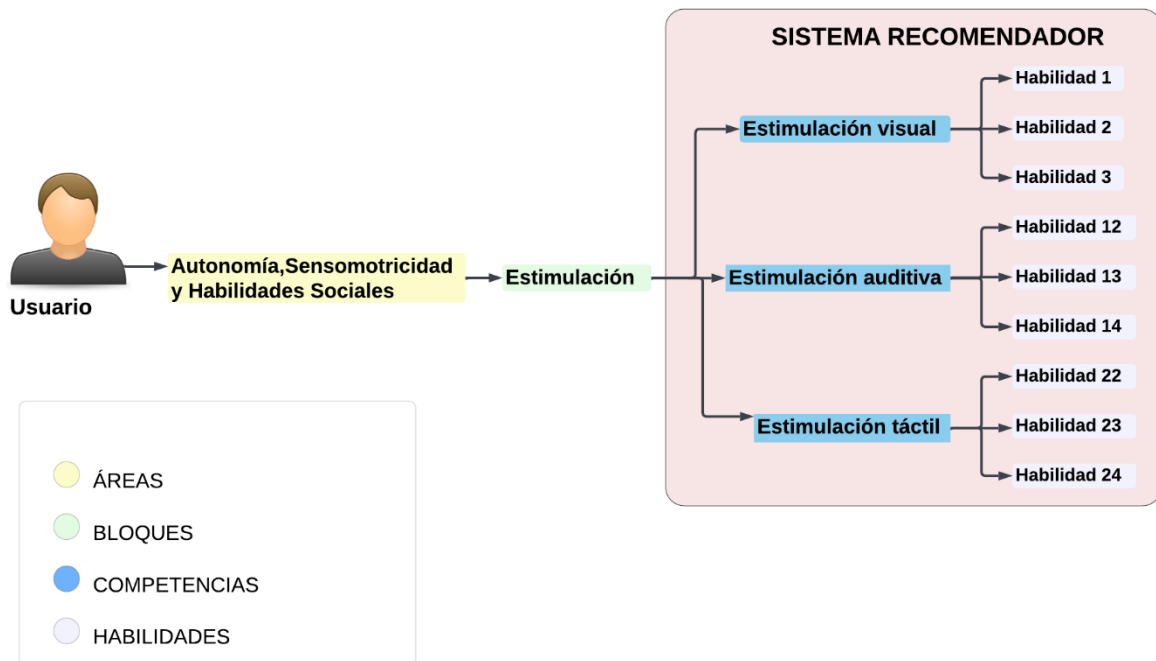


FIGURA 45.-EJEMPLO DEL SISTEMA RECOMENDADOR

Como podemos observar en la Figura (45) el sistema recomendador se encuentra dentro de las competencias ya que dentro de las competencias tendremos el conjunto de habilidades y si el usuario le ha gustado cualquier habilidad, el sistema recomendará habilidades de acuerdo con el contenido seleccionado.

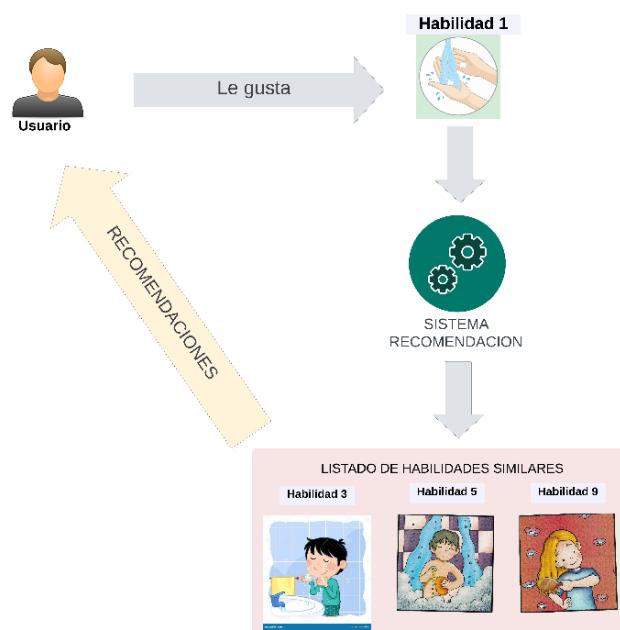


FIGURA 46.-FUNCIONAMIENTO DEL SISTEMA RECOMENDADOR BASADO EN CONTENIDO

El siguiente paso se creará dos consultas distintas, para hacer las recomendaciones en base a dos criterios. La primera es recomendar habilidades de acuerdo a gustos similares por usuario, el segundo criterio es recomendar habilidades de acuerdo con los gustos del usuario.

```

1 MATCH (p1:Usuario {name:'usuario1'})-[likes1:LIKES]->(cuisine)
2 MATCH (p2:Usuario)-[likes2:LIKES]->(cuisine2) where p1<>p2
3 WITH p1, collect(likes1.score) AS p1L, p2, collect(likes2.score) AS p2L
4 RETURN p1.name AS Usuario, p2.name AS Usuarios,
5 gds.similarity.cosine(p1L, p2L) AS similitud ORDER BY similitud DESCENDING
6

```

FIGURA 47.-ALGORITMO DE SIMILITUD DE COSENO, UTILIZANDO GDS

Usuario	Usuarios	similitud
"usuario1"	"usuario2"	0.9878048780487805
"usuario1"	"usuario5"	0.9745831841044724
"usuario1"	"usuario4"	0.8729380882533108

FIGURA 48.-RESULTADO DE LOS VECINOS MÁS CERCANOS

El resultado de la Figura (48) nos muestra los usuarios más cercanos que tiene el usuario 1. El siguiente algoritmo nos ayuda a seleccionar las habilidades que se encuentran dentro de la misma competencia y nos ayuda a eliminar las habilidades que estén repetidas.

```

MATCH (p1:Usuario)-[:me_gusta]->(b1:Habilidad)<-[:me_gusta]->(p2:Usuario)-[:me_gusta]->(b2:Habilidad)
WHERE b1.codigo<>b2.codigo and p1.codigo=""+str(usuario1)+""
and p2.codigo=""+str(usuario2)+"" and b1.codigo_competencia=b2.codigo_competencia
RETURN b2.codigo as HabilidadRecomendada

```

FIGURA 49.-ALGORITMO PARA LISTAR LAS HABILIDADES

6.2.4. Desarrollo

A. Módulo de la base de datos en MongoDB

Para el desarrollo de la base de datos en MongoDB se utilizará el paquete PyMongo ya que contiene los controladores que nos permite comunicar Python con MongoDB. Para conectarse con MongoDB, se utilizará el MongoClient<'<MongoDB URL>'> para hacer esto.

```

from pymongo import MongoClient
from neo4j import GraphDatabase
conn=MongoClient('mongodb://root:example@157.230.15.68:27017/?authMechanism=DEFAULT')

```

FIGURA 50.-CONEXIÓN A MONGODB

En nuestro caso el MongoDB lo tenemos instalado en la nube entonces configuramos de la siguiente manera primero es usuario y contraseña luego la dirección IP ya sea local o en la nube.

```
def listarBloqueAreaM(codArea):
    datos=pd.DataFrame(list(conn.SistemaRecomendacion.Bloque.find({"codigo_area": codArea},{'_id':False})))
    return datos

def listarCompetenciaBloque(codBloque):
    datos=pd.DataFrame(list(conn.SistemaRecomendacion.Competencia.find({"codigo_bloque": codBloque},{'_id':False})))
    return datos

def listarHabilidadCompetencia(codCompetencia):
    datos=pd.DataFrame(list(conn.SistemaRecomendacion.Habilidad.find({"cod_competencia": codCompetencia},{'_id':False})))
    return datos

def buscadorHabilidad(palabra):
    datos=pd.DataFrame(list(conn.SistemaRecomendacion.Habilidad.find({"nombre": '/' + palabra + '/i' })))
    return datos

def listarHabilidades():
    datos=pd.DataFrame(list(conn.SistemaRecomendacion.Habilidad.find({},{'_id':False})))
    return datos

def listarHabilidades2(codigo):
    datos=conn.SistemaRecomendacion.Habilidad.find_one({'codigo':codigo},{'_id':False})
    return datos
```

FIGURA 51.MÉTODOS USADOS EN EL SISTEMA

Como se muestra en la Figura.18 tendremos algunos métodos que nos permitirán la búsqueda de información dentro de cada una de las conexiones que existen dentro de la base de datos.

B. Modulo del sistema recomendador Neo4j

Para la parte del sistema de recomendación y la creación de nodos se utilizara neo4j , se instalara el plugin de Graph Data Science(GDS) este plugin nos permitirá el uso de los sistemas de recomendación ya sean basados en filtro colaborativo o basados en contenido ya que contiene cada uno de los algoritmos que utilizaremos para este proyecto , para utilizarlo en Python se instalará el paquete GraphDatabase , ya que contiene los controladores que nos permitirá la comunicación entre Python y neo4j.

Para poder hacer la conexión entre Python y neo4j, necesitamos del método controlador dentro de este método nos pedirá la siguiente información:

- Enlace que es la dirección ip local o de la nube
- El usuario creado en neo4j
- La contraseña

```

def connection():
    driver=GraphDatabase.driver("neo4j://157.230.15.68:7687",
                                auth=("neo4j",
                                       "Admin12345"))
    return (driver)

```

FIGURA 52.-CONEXIÓN DE NEO4J CON PYTHON

La implementación de métodos es muy importante ya que nos permite usar el lenguaje Cypher de neo4j en Python para ello lo primero q se tiene q hacer es crear una sesión. Luego en una variable query se guarda la sentencia que necesitamos para mostrar la información y luego se manda a ejecutar y guardar en un datagrama.

```

def listarFavoritos(usuarios):
    driver_neo4j=connection()
    session = driver_neo4j.session()
    query="""
    MATCH (u:Usuario {codigo: $usuarios})-[r:me_gusta]->(m:Habilidad)
    RETURN m.codigo as codigo
    """

    resultado=session.run(query,usuarios=usuarios)

    return pd.DataFrame(resultado)

```

FIGURA 53.EJEMPLO DE UN MÉTODO NEO4J

Cuando todo esté funcionando correctamente lo siguiente es crear un Docker-compose para subir el código dentro de un contenedor Docker.

C. Módulo de los servicios web

Para el desarrollo de los servicios web se utilizará el servidor FastApi, en este apartado lo que nos va a permitir es construir Apis de forma sencilla y rápida, ya que es una herramienta fácil de aprender a utilizar, nos permite hacer los métodos necesarios para las operaciones CRUD sobre las tablas, el uso de Reuters que ayuda en la organización de recurso.

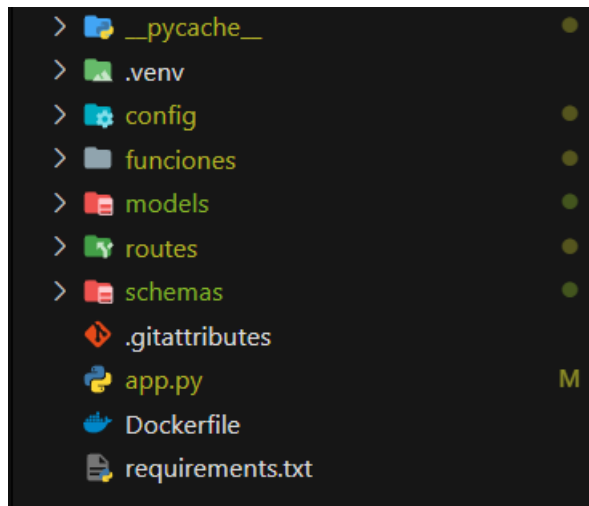


FIGURA 54.-ESTRUCTURA DEL SERVICIO FASTAPI

Dentro de nuestra Api tenemos una carpeta llamada funciones dentro de esta están dos archivos en los cuales están todos los métodos que se van a utilizar en meo4j y MongoDB, dentro de la Carpeta routers están los servicios que se crearon para su consumo.

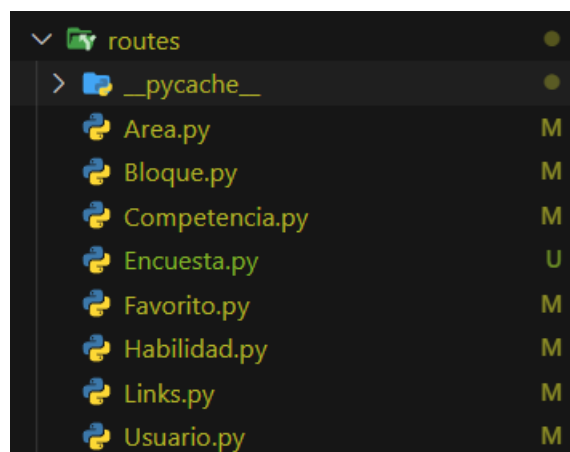


FIGURA 55.ESTRUCTURA DE LOS ROUTERS

Para poder realizar las diferentes funciones de las Apis que se conectan a la base de datos y servicios web de terceros, para entender mejor el uso de neo4j y MongoDB mostraremos el siguiente método que tiene la Figura (55).

```

@bloque.get('/bloques')
async def obtenerBloquesAreas(codigo:str):

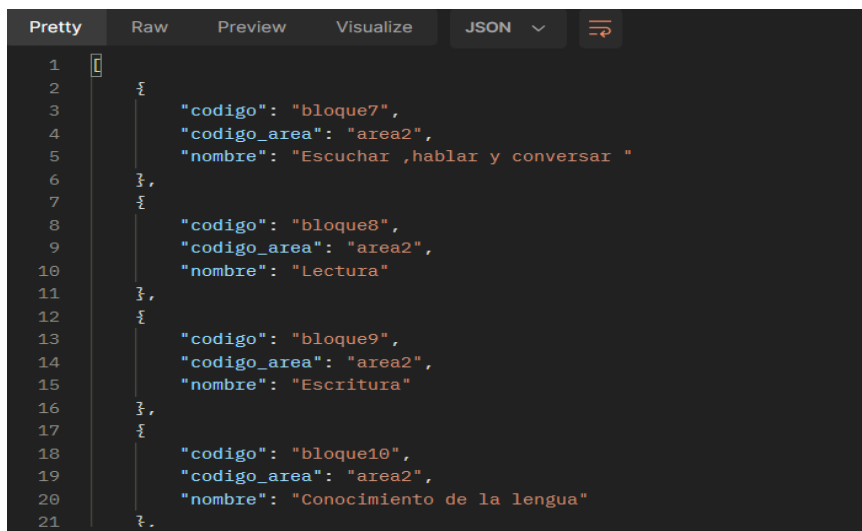
    resultado = []
    resu_query =FuncionesNeo4j.listarBloqueArea(codArea=codigo)
    if resu_query.empty == True:
        return JsonResponse(status_code=404, content={"message": "Item no encontrado"})
    else:
        mongo = pd.DataFrame(FuncionesMongo.listarBloqueAreaM(codArea=codigo))
        for i in resu_query.iloc[:,0]:
            comparar = mongo.loc[mongo['codigo'] == i]
            if (len(comparar) > 0):
                comparar = comparar.to_dict('records')
                for s in comparar:
                    resultado.append(s)
        fina=json.dumps(resultado)

        return JsonResponse(status_code=200, content=json.loads(fina))

```

FIGURA 56.CREANDO UN SERVICIO GET CON MONGODB Y NEO4J

Lo que hacemos primero es guardar el método de neo4j y la variable es nula se devuelve un “status code 404” caso contrario se procede al siguiente variable q es guardar el método de MongoDB luego lo que se hace es recorrer la variable donde contiene la información de neo4j y le va comparando con la variable que contiene información del MongoDB, es decir compara cada el código de neo4j y el código de MongoDB y si son iguales le guarda en un arreglo caso contrario no se le toma esa variable como resultado vinal nos devuelve los datos solicitados.



```

1  [
2    {
3      "codigo": "bloque7",
4      "codigo_area": "area2",
5      "nombre": "Escuchar ,hablar y conversar "
6    },
7    {
8      "codigo": "bloque8",
9      "codigo_area": "area2",
10     "nombre": "Lectura"
11   },
12   {
13     "codigo": "bloque9",
14     "codigo_area": "area2",
15     "nombre": "Escritura"
16   },
17   {
18     "codigo": "bloque10",
19     "codigo_area": "area2",
20     "nombre": "Conocimiento de la lengua"
21   }

```

FIGURA 57.EJEMPLO DEL CONSUMO DE UNA API

- **Sistema de recomendación**



FIGURA 58.-INICIO DE UN USUARIO NUEVO

Como se puede observar en la figura (54) es el inicio del sistema para un usuario nuevo, el sistema recomendador no se muestra ya que el usuario no tiene registrado habilidades favoritas.



FIGURA 59.-INICIO DEL SISTEMA RECOMENDADOR

Como se observa en la figura (55), es el sistema recomendador funcionando, es decir en base a las habilidades favoritas del usuario, lo que hace el sistema es buscar habilidades similares de otros usuarios y recomienda al usuario de acuerdo con los gustos.

- **Encuestas**

En nuestro proyecto se implementó un sistema de encuestas basados en satisfacción, es decir, que de acuerdo con el criterio del usuario sabemos que tan satisfecho está con la recomendación dada.



FIGURA 60.-PANTALLA DE ENCUESTAS

Como se observa en la figura (56) se ve el listado de preguntas en la cual el usuario puede calificar cada una de las preguntas de la encuesta según su criterio, una cosa importante es que cada usuario puede realizar una encuesta de acuerdo con cada habilidad, es decir el usuario no puede realizar dos encuestas sobre la misma habilidad.

- **Validación del sistema de recomendación**

Para la validación del sistema recomendador en base a las encuestas, se creó un apartado donde se puede observar el listado de habilidades por encuestas respondidas, es decir se tiene un total de encuestas respondidas, un promedio general de cada pregunta.



FIGURA 61.-REPORTE DE ENCUESTAS POR HABILIDAD

Para identificar que tan recomendable es la habilidad según los usuarios que han llenado la encuesta, lo único es ingresar a cualquier habilidad y se nos abre otra pantalla donde tendremos lo siguiente, información de la habilidad, total de encuestas, un detalle en general de cada pregunta, donde nos mostrara si es rentable la habilidad recomendada.

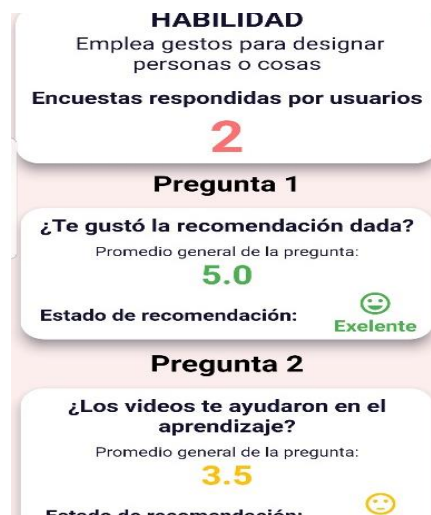


FIGURA 62.-DETALLE DE LA ENCUESTA POR PREGUNTA

7. Cronograma

Nombre de tarea	Duración	Inicio	Finalización	Encargado
Proyecto	400 horas	lun 18/04/22	vie 3/1/24	JS
Tarea 1	100 horas	lun 18/04/22	vie 29/04/22	JS
OE. 1	100 horas	lun 18/04/22	vie 06/05/22	JS
ACT.1. Revisar la bibliografía referente a recomendaciones en grafos.	30 horas	lun 18/04/22	vie 13/05/22	JS
ACT.2. Investigar conceptos de los sistemas de recomendación utilizando grafos.	30 horas	lun 25/04/22	25/04/22	JS
ACT.3. Identificar las principales herramientas o librerías de programación para la implementación de grafos y sistemas de recomendación.	30 horas	lun 02/05/22	Vie 02/05/22	JS
REVISIÓN/CORRECCIÓN	10 horas	lun 09/05/22	vie 13/05/22	RH
Tarea 2	100 horas	lun 16/05/22	vie 17/05/22	JS
OE. 2	100 horas	lun 16/05/22	vie 17/05/22	JS

ACT.1. Recopilación de datos educativos para el análisis de requerimientos del sistema.	20 horas	lun 16/05/22	vie 20/05/22	JS
ACT.2. Desarrollar un web crawler para la recopilación de información.	20 horas	lun 23/05/22	vie 27/05/22	JS
ACT.3. Diseñar e implementar una infraestructura de grafos junto a sus modelos, lo que mejorara la eficiencia de los algoritmos.	20 horas	lun 30/05/22	vie 3/06/22	JS
ACT.4. Generar sus nodos y aristas progresivamente y en tiempo real, cuando sucedan las acciones que deban persistir.	30 horas	lun 06/06/22	vie 10/06/22	JS
REVISIÓN/CORRECCIÓN	10 horas	lun 13/06/22	vie 17/06/22	RH
Tarea 3	100 horas	lun 20/06/22	vie 15/07/22	JS
OE. 3	100 horas	vie 15/07/22	vie 08/08/22	JS
ACT.1. Conocer los tipos de recomendaciones junto con sus algoritmos aplicables.	30 horas	lun 20/06/22	vie 24/06/22	JS
ACT.2. Definir el tipo de algoritmo de recomendación.	30 horas	lun 27/06/22	vie 1/07/22	JS
ACT.3. Implementar los algoritmos aplicables al sistema recomendador basado en grafos.	30 horas	lun 4/07/22	vie 8/07/22	JS

REVISIÓN/CORRECCIÓN	10 horas	lun 11/07/23	vie 15/07/23	RH
Tarea 4	100 horas	lun 11/07/23	vie 05/01/24	JS
OE. 4	100 horas	lun 11/07/23	lun 05/01/24	JS
ACT.1. Diseñar un plan de experimentación.	10 horas	lun 11/07/23	Vie 9/07/23	JS
ACT.2. Probar el sistema recomendador basados en grafos.	20 horas	Vie 9/07/23	Vie 21/06/23	JS
ACT.3. Crear encuestas para validar el sistema de recomendación.	20 horas	lun 21/06/23	12/07/23	JS
ACT.4. Ejecutar el plan de experimentación.	40 horas	Jue 21/09/23	28/09/23	JS
REVISIÓN/CORRECCIÓN	10 horas	03/0124	05/01/24	RH

8. Presupuesto

DENOMINACIÓN	CANT.	COSTO UNITARIO	COSTO TOTAL
	Unidades	Dólares	dólares
1. Bienes			
Papel Bond A-4	100	0,05	5
Impresiones	100	0,10	10
Empastados	1	10	10
2. Tecnológico			
Computadora portátil	1	1100	1100
Celular Inteligente	2	300	600
Servidor en la nube	1	300	300
3. Servicios			
Servicio de transporte	40	2.00	360
Servicios de Internet	1	25	25
Alimentación	180	2.50	450
4. Personal			
Horas de desarrollo	400	8	1600
3. Otros			
Imprevistos	1	200	200
TOTAL	429	932.65	4660

TABLA 14.-PRESUPUESTO DEL DESARROLLO DE LA APLICACIÓN

Conclusiones y trabajos futuros

En base al estado del arte los sistemas de recomendación y las técnicas empleadas para la generación de recomendaciones ya sean por filtro colaborativo o basado en contenido, se comprobó que gracias a estos sistemas ayudan a los usuarios con facilidad las búsquedas de a las necesidades del usuario o ya sea comparando los gustos de otros usuarios.

En base a lo desarrollado se procedió a diseñar encuestas, para validar la recomendación que dio el sistema y los recursos que tiene, por lo cual se obtuvo que el 100% de niños encuestados, el 90% está de acuerdo con las recomendaciones dadas, esto nos permite verificar que cada habilidad recomendada por el sistema esta correctamente funcionando.

Se adquirió satisfactoriamente los datos necesarios para la implementación del sistema; tomados del wikinclusion, donde se extrajo la información utilizando webcrawler para convertirlos en CSV y cargarlos al base de datos de NEO4J y MONGODB.

Se debe tener en cuenta que el propósito de este proyecto es contribuir de cierta manera a combatir el problema planteado sobre el contenido consumido por nuestros infantes. Ya que múltiples estudios señalan que el educar a los niños controlando y guiando de una manera correcta a los menores en la navegación por la red, es la finalidad de la aplicación.

En este trabajo se implementó un repositorio de contenido educativo y un sistema recomendador basado en contenidos usando grafos, para la base de datos se utilizó MongoDB y para el sistema de recomendación Neo4j, todo el sistema se implementó dentro de Docker (Docker-compose -DockerFile) en base a ello, se describen ideas para mejorar el sistema en rendimiento y despliegue de este.

- Para el sistema de recomendación proveer más datos para mejorar las predicciones.
- Implementar una aplicación web progresiva (PWA).
- Desarrollar el despliegue del sistema para diferentes plataformas.
- Para el repositorio de contenido educativo, implementar más habilidades.
- Dentro de las Habilidades agregar más contenido ya sea multimedia otros tipos de software para el uso de esa habilidad.
- Se podría implementar dentro de las habilidades un apartado de diga material, es decir, que contenga material descargable, dibujar o libro.

Recomendaciones

En base al desarrollo del proyecto, para el sistema recomendador se debe tener una mayor fuente de datos ya que con esos datos nos permite entrenar y tener predicciones más acertadas ya que cuando el sistema está en sus estados iniciales se posee poca información para tener una predicción acertada.

El momento de hacer las importaciones a las colecciones dentro de MONGODB es importante tener en cuenta los campos que este estructurados adecuadamente ya que con eso puede generar pérdida de datos al momento de la importación.

Con respecto a Flutter se debe tener en cuenta el manejo de sus rutas, así como el manejo de sus estados para tener una mayor calidad de rendimiento y funcionamiento. Se recomienda el uso adecuado de los widgets para así lograr una combinación de rendimiento y una interfaz intuitiva.

El sistema posee una arquitectura basada en contenedores utilizando Docker, ya que dentro de Docker nos permite la creación de contenedores que permiten la interacción de la base de datos para el repositorio del contenido educativo, es importante tener en cuenta que los puertos que se están usando no están generando conflicto o están ocupado por otro servicio, es importante generar adecuadamente el Docker-compose para la generación de los contenedores ya que cada contenedor tiene una tarea específica de acuerdo con sus configuraciones.

El contenido educativo se debe tener un indicador de evaluación, es decir, área, bloques, competencias y habilidades que son las actividades para obtener con mayor precisión los recursos educativos.

Referencias bibliográficas

- Abdulrahman, L. M., Ahmed, S. H., Rashid, Z. N., Jghef, Y. S., Ghazi, T. M., & Jader, U. H. (2023). Web Phishing Detection Using Web Crawling, Cloud Infrastructure and Deep Learning Framework. *Journal of Applied Science and Technology Trends*, 4(01), 54–71. <https://doi.org/10.38094/jastt401144>
- Adeshina, A. A. (2022). *Building Python Web APIs with Fast API a fast -paced guide to building high-performance, robust web APIs with very little boilerplate code.*
- Alfonso Azcuy, R. (2021). Módulo de recomendación de patrones de diseño para EGPat. *Revista Cubana de Ciencias Informáticas*, 15(2), 118–137.
- Bader, D. A. (2022). Massive graph analytics. In *Massive Graph Analytics*. <https://doi.org/10.1201/9781003033707>
- Bailey, T., Biessek, A., & Wills, T. (2021). *Flutter for Beginners - Second Edition*.
- Baton, J., & Van Bruggen, R. (2018). *Learning Neo4j 3.x: Effective Data Modeling, Performance Tuning and Data Visualization Techniques in Neo4j*.
- Bergman, J., & Popov, O. B. (2023). Exploring Dark Web Crawlers: A Systematic Literature Review of Dark Web Crawlers and Their Implementation. *IEEE Access*, 11(February), 35914–35933. <https://doi.org/10.1109/ACCESS.2023.3255165>
- Brewer, R., Westlake, B., Arauza, O., & Hart, T. (2021). The Ethics of Web Crawling and Web Scraping in Cybercrime Research: Navigating Issues of Consent, Privacy, and Other Potential Harms Associated with Automated Data Collection. *Researching Cybercrimes: Methodologies, Ethics, and Critical Approaches*, 435–456. https://doi.org/10.1007/978-3-030-74837-1_22
- Chapagain, A. (2019). *Hands-On, Web Scraping with Python*. Packt Publishing.
- Chauhan, A. (2019). A review on various aspects of auxetic materials. *AIP Conference Proceedings*, 2105(05), 90–92. <https://doi.org/10.1063/1.5100689>
- EFE, A., ASLAN, U., & KARA, A. M. (2020). Securing Vulnerabilities in Docker Images. *International Journal of Innovative Engineering Applications*, 4(1), 31–39. <https://doi.org/10.46460/ijiea.617181>
- Estefanell, L. (2020). Pantallas en casa. *Unicef*, 38. <https://www.unicef.org/uruguay/informes/pantallas-en-casa>
- Eyada, M. M., Saber, W., El Genidy, M. M., & Amer, F. (2020). Performance Evaluation of IoT Data Management Using MongoDB Versus MySQL Databases in Different Cloud Environments. *IEEE Access*, 8, 110656–110668.

<https://doi.org/10.1109/ACCESS.2020.3002164>

- Falk, K. (2019). *Practical Recommender Systems* (S. Island (ed.); Primera). Manning Publications Co.
- Flores-Vivar, J. M., & García-Peñalvo, F. J. (2023). Reflections on the ethics, potential, and challenges of artificial intelligence in the framework of quality education (SDG4). *Comunicar*, 30(74), 35–44. <https://doi.org/10.3916/C74-2023-03>
- IBRAHIM, H. (2019). *A Study of the Use of Docker Compose and DockerHub Images*. September.
- Ingo, H., & Daly, D. (2020). Automated system performance testing at MongoDB. *Proceedings of the Workshop on Testing Database Systems, DBTest 2020*, 2. <https://doi.org/10.1145/3395032.3395323>
- Kane, S. P., & Matthias, K. (2023). *Docker Up & Running, Shipping reliable containers in production* (K. Hightower, P. D. Advocate, L. Rice, C. Open, S. Officer, M. Todor, S. P. Engineer, & C. Orchestration (eds.)).
- Khder, M. A. (2021). Web scraping or web crawling: State of art, techniques, approaches and application. *International Journal of Advances in Soft Computing and Its Applications*, 13(3), 144–168. <https://doi.org/10.15849/ijasca.211128.11>
- Kithulwatta, W. M. C. J. T., Jayasena, K. P. N., Kumara, B. T. G. S., & Rathnayaka, R. M. K. T. (2021). Docker incorporation is different from other computer system infrastructures: A review. *Proceedings - International Research Conference on Smart Computing and Systems Engineering, SCSE 2021*, 230–236. <https://doi.org/10.1109/SCSE53661.2021.9568323>
- Klymash, M., Demydov, I., Uryvskiy, L., & Pyrih, Y. (2020). A Brief Survey on Architecture of Feedback Systems for Interactive E-Government ICT Platforms. *Proceedings - 15th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering, TCSET 2020, July*, 458–461. <https://doi.org/10.1109/TCSET49122.2020.235475>
- Link, S. (2019). Neo4j Keys. *The University of Auckland*, 14.
- Lubanovic, B. (2023). FastAPI Modern Python Web Development. In *Angewandte Chemie International Edition*, 6(11), 951–952. (Vol. 3, Issue 1). <https://medium.com/@arifwicaksanaa/pengertian-use-case-a7e576e1b6bf>
- Makris, A., Tserpes, K., Spiliopoulos, G., & Anagnostopoulos, D. (2019). Performance evaluation of MongoDB and PostgreSQL for spatio-temporal data. *CEUR Workshop*

Proceedings, 2322.

- Makris, A., Tserpes, K., Spiliopoulos, G., Zissis, D., & Anagnostopoulos, D. (2021). Correction to: MongoDB Vs PostgreSQL: a comparative study on performance aspects (GeoInformatica, (2020), 10.1007/s10707-020-00407-w). *GeoInformatica*, 25(1), 241–242. <https://doi.org/10.1007/s10707-020-00424-9>
- Matallah, H., Belalem, G., & Bouamrane, K. (2021). Comparative Study Between the MySQL Relational Database and the MongoDB NoSQL Database. *International Journal of Software Science and Computational Intelligence*, 13(3), 38–63. <https://doi.org/10.4018/ijssci.2021070104>
- Melamud, A., & Waisman, I. (2019). Pantallas: discordancias entre las recomendaciones y el uso real. *Archivos Argentinos de Pediatría*, 117(5), 349–351. <https://doi.org/10.5546/aap.2019.349>
- Mendoza Olguín, G. E., Laureano de Jesús, Y., & Pérez de Celis Herrero, M. de la C. (2019). Métricas de similaridad y evaluación para sistemas de recomendación de filtrado colaborativo. *Revista de Investigación En Tecnologías de La Información*, 7(14), 224–240. <https://doi.org/10.36825/riti.07.14.019>
- Mitchell, R. (2018). Web Scraping with Python. In *Journal of Chemical Information and Modeling* (Vol. 53). O'Reilly Media, Inc.
- Monideepa, R., Pushpendu, K., & Sujoy, D. (2023). *Recommender Systems (Erasmus IP) Recommender Systems*.
- Needham, M., & Hodler, A. (2019). Graph Algorithms. In J. Hassel, J. Bleiel, D. Baker, T. Brown, R. Head, J. McConville, & D. Futato (Eds.), *Paper Knowledge . Toward a Media History of Documents* (1st ed.). O'Reilly Media, Inc.
- Oscar, E., & Marcellin, S. (2018). Estado del arte en los sistemas de recomendación. *Research in Computing Science*, 135(1), 25–40. <https://doi.org/10.13053/rcs-135-1-2>
- UNDP. (2023). *Objetivos de Desarrollo Sostenible*. Programa de las Naciones unidas para el desarrollo. <https://www.undp.org/es/sustainable-development-goals>
- Partida, S. (2021). *INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE OCCIDENTE Departamento de Estudios Socioculturales* (Issue 15018). <https://rei.iteso.mx/handle/11117/5572%0Ahttp://hdl.handle.net/11117/5303>
- Qudus Khan, F., Tsaramirsis, G., Ullah, N., Nazmudeen, M., Jan, S., & Ahmad, A. (2021). Smart algorithmic based web crawling and scraping with template autoupdate capabilities. *Concurrency and Computation: Practice and Experience*, 33(22), 1–10.

<https://doi.org/10.1002/cpe.6042>

- Raj, A. J. (2020). Building Microservices with Docker Compose. *Researchgate.Net*, May. https://www.researchgate.net/profile/Aditya-Raj-12/publication/351626260_Building_Microservices_with_Docker_Compose/links/60a24f02299bf15ca390c8c5/Building-Microservices-with-Docker-Compose.pdf
- Reis, D., Piedade, B., Correia, F. F., Dias, J. P., & Aguiar, A. (2022). Developing Docker and Docker-Compose Specifications: A Developers' Survey. *IEEE Access*, 10. <https://doi.org/10.1109/ACCESS.2021.3137671>
- Riquelme, J. C., & Botella, P. (2020). XXI Jornadas de Ingeniería del Software y Bases de Datos, JISBD'2006. In *IEEE Latin America Transactions* (Vol. 5, Issue 4). <https://doi.org/10.1109/TLA.2007.4378506>
- Rodriguez, D. (2023). *Recommender Systems (Erasmus IP) a multi-disciplinary approach*. (R. Monideepa, K. Pishpendu, & D. Sujoy (eds.)). Taylor & Francis Group.
- Rosado, K., & Ruiz, J. (2022). *DESARROLLO DE UN PODCAST INFORMATIVO PARA PADRES PARA PREVENIR EL USO INADECUADO DE TECNOLOGÍA EN NIÑOS MENORES DE 10 AÑOS*.
- Rose, R. (2022). Flutter and dart. In <https://Flutter.Dev/>.
- Rodella, F. (2018, July 6). *Polémica por la violación del avatar de una niña de siete años en un popular videojuego*. Ediciones EL PAÍS S.L. https://elpais.com/tecnologia/2018/07/06/actualidad/1530871736_133106.html?event=regonetap&event_log=regonetap&prod=REGONETAP&o=regonetap
- Salcedo Aparicio, D., Fuentes Torres, B. J., Villamar Cedeño, E. D., & Salcedo Aparicio, P. R. (2021). Los niños de era digital: estilos de aprendizaje y los retos de la participación. *Recimundo*, 5(4), 37–44. [https://doi.org/10.26820/recimundo/5.\(4\).oct.2021.37-44](https://doi.org/10.26820/recimundo/5.(4).oct.2021.37-44)
- Schultz, W., Avitabile, T., & Cabral, A. (2019). Tunable consistency in MongoDB. *Proceedings of the VLDB Endowment*, 12(12), 2071–2081. <https://doi.org/10.14778/3352063.3352125>
- Sedgewick, R., & Kevin, W. (2018). *Algorithms* (4th ed.). Pearson education, Inc.
- Seguin, K. (2018). The little MongoDB Book. *Jurnal Penelitian Pendidikan Guru Sekolah Dasar*, 6(August), 128.
- Stothers, J. A. M., & Nguyen, A. (2020). Can Neo4j Replace PostgreSQL in Healthcare? *AMIA Joint Summits on Translational Science Proceedings*. *AMIA Joint*

Summits on Translational Science, 2020, 646–653.
<http://www.ncbi.nlm.nih.gov/pubmed/32477687>
<http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC7233060>

- Treviño-Villalobos, M., Viquez-Acuña, L., Quirós-Oviedo, R., & Esquivel-Vega, G. (2019). Una comparación de rendimiento entre bases de datos NoSQL: MongoDB y ArangoDB. *Revista Tecnología En Marcha*, 32, 5–15. <https://doi.org/10.18845/tm.v32i6.4223>
- Wiese, L. (2019). Tutorial: Data analytics with graph algorithms - A hands-on tutorial with Neo4j. *Lecture Notes in Informatics (LNI), Proceedings - Series of the Gesellschaft Fur Informatik (GI), P-290*, 259–261. <https://doi.org/10.18420/btw2019-ws-26>

Anexos

En este apartado se encuentra disponible en el repositorio de github en la cual encontramos los recursos que se utilizaron en este proyecto, dentro de este enlace se encuentra la siguiente información:

- Colección de datos: dentro de esta carpeta se encuentran archivos csv para importar las colecciones a mongoDB
- Script Nodos: En esta parte se encuentra el script completo que se ejecuta dentro de neo4j para la creación de todos los nodos
- Docker-Compose: Son archivos para la creación de los contenedores dentro de Docker

Para acceder a este repositorio de github, hacer clic al siguiente enlace:

<https://github.com/jonathans2200/Recursos-Tesis>