Servicio para facilitar la selección de artículos de investigación relevantes y no relevantes a una temática basado en una consulta, resultados de SCOPUS y visualización de grupos de documento

Juan Fernando Campo Mosquera
Laura Isabel Chaparro Navia

Director: PhD. Carlos Alberto Cobos Lozada

Universidad del Cauca
Facultad de Ingeniería Electrónica y Telecomunicaciones
Departamento de Sistemas
Grupo de I+D en Tecnologías de la Información
Área de interés en Gestión de la Información y Sistemas Inteligentes
Popayán, noviembre 2023

TABLA DE CONTENIDO

LISTA DE FIGURAS

LISTA DE TABLAS

LISTA DE ANEXOS

RESUMEN

Uno de los problemas más comunes en el desarrollo de proyectos de investigación es la búsqueda de artículos relevantes a las necesidades de los investigadores. Sin mencionar el hecho de la gran cantidad de artículos presentes en las bases de datos bibliográficas, lo que dificulta los procesos de búsqueda y selección de los documentos apropiados. Actualmente las herramientas que permiten consultar las bases de datos bibliográficas entregan los resultados en forma de listas, lo cual obliga el usuario a realizar una búsqueda secuencial y que consume mucho tiempo entre los resultados obtenidos para encontrar los artículos relevantes y descartar aquellos no relevantes a su consulta. Por lo anterior, el presente trabajo de investigación propone una aplicación web que parte de una cadena de búsqueda la cual se realiza a Scopus y con los resultados obtenidos se realiza un proceso de agrupamiento con el fin de generar una visualización alternativa de los resultados para el usuario en grupos temáticos que faciliten descartar los artículos no relevantes a la consulta y seleccionar los relevantes basado en bloques o grupos de estos. Para ello se implementan los algoritmos de agrupamiento K-means, Spectral, Fuzzy C-means, Lingo y STC y se evalúan los resultados obtenidos haciendo uso de los conjuntos de datos AAAI13, AAAI14, Arxiv y Topic Modeling, mediante el uso de métricas clásicas tales como precisión, recuerdo, medida F y exactitud, donde se obtuvo que K-means y Spectral son los algoritmos que obtienen mejores resultados. Además, se implementaron los algoritmos Yake, Graph topic rank, Inverse transform, Noun phrases y Semantic frequency para generar los títulos o etiquetas de los grupos. Se desarrolló un backend implementado en Django que ofrece los servicios de agrupamiento y etiquetado, configuración de preferencias del perfil, entrega de los resultados, entre otros. Y una aplicación web desarrollada en Angular que consume los servicios prestados por el back. Los servicios propuestos fueron evaluados desde el aplicativo web por un grupo de investigadores mediante una encuesta no estructurada que evalúa su nivel de satisfacción, donde se concluyó que la aplicación es intuitiva y tiene un alto valor para los investigadores, aunque presenta elementos por mejorar.

DEDICATORIA

A nuestros padres, quienes con su amor y sacrificio han sido los pilares que sostienen nuestros sueños.

A los jóvenes, que su búsqueda de conocimiento esté siempre guiada por la curiosidad y que su deseo de aprender nunca se agote.

AGRADECIMIENTOS

Me agradezco por nunca perder la motivación y por perseverar en esta investigación, que marca el fin de una etapa importante en mi vida. Este trabajo representa la realización de un sueño, y me siento agradecido con Dios por la oportunidad de convertirlo en realidad. Gracias a mis padres por que han sido una fuente constante de inspiración y apoyo en mi vida. Sus palabras alentadoras, su apoyo inquebrantable y su fe en mis capacidades me han sostenido en los momentos más desafiantes.

No puedo dejar de agradecer a Laura, mi compañera de trabajo de grado. Su colaboración y apoyo constante fueron fundamentales para superar los retos que encontramos en este camino y para mantener el enfoque cuando parecía perderse. Espero que este trabajo no solo represente el final de una etapa, sino también el inicio de nuevas oportunidades y logros en el camino que tengo por delante.

Juan Fernando Campo Mosquera

Quiero comenzar expresando mi agradecimiento a Dios por permitirme llegar a este punto en mi camino académico. A mi madre, le dedico un agradecimiento especial por su apoyo y respaldo en cada paso de este viaje y por brindarme la oportunidad de estudiar y convertirme en una profesional. Su amor y dedicación son la razón de mi perseverancia. A Manuel, así como a mi padre, a mi familia y seres queridos les agradezco por su apoyo incondicional, por su paciencia y por creer en mí a lo largo de este recorrido.

A mi compañero de grado, Fernando, le agradezco por su apoyo y por desafiarme a superarme cada día, así como por acompañarme en este reto tan importante.
Este logro no habría sido posible sin la valiosa contribución de cada uno de ustedes, y estoy profundamente agradecida por ello.

Laura Isabel Chaparro Navia

A nuestro director de trabajo de grado, PhD. Carlos Alberto Cobos Lozada, queremos expresarle nuestra gratitud por su orientación experta y su compromiso. Sus valiosas sugerencias y dedicación fueron esenciales para el éxito de este proyecto. Este logro es el resultado del esfuerzo conjunto de muchos, agradecemos a todas las personas que apoyaron este proceso, compañeros, docentes, evaluadores, Universidad del Cauca y Facultad de Ingeniería electrónica y Telecomunicaciones. Su disposición para compartir sus conocimientos y dedicar su tiempo aportó de manera significativa a nuestro trabajo.

Juan Fernando Campo Mosquera y Laura Isabel Chaparro Navia

● CAPÍTULO 1

1 INTRODUCCIÓN

1.1 PLANTEAMIENTO DEL PROBLEMA

Los proyectos de investigación dan a conocer sus resultados de diferentes formas y por diversos medios, entre ellos, uno de los más conocidos y divulgados son los artículos científicos, donde se resumen los principales aportes de las investigaciones en diferentes áreas del conocimiento; estos artículos se dan a conocer a través de su publicación en revistas, seminarios, conferencias, congresos, por nombrar algunos [1]. Por otro lado, los investigadores cuando realizan el planteamiento de proyectos o la ejecución de estos, deben continuamente hallar información de fuentes confiables acerca de los temas concretos de sus investigaciones y para ello deben buscar, leer, estudiar y analizar múltiples documentos expresados como artículos de investigación, capítulos de libros, entre otros, en un proceso que generalmente implica mucho tiempo [2], puesto que la cantidad de información disponible en las bases de datos es gigante, tan solo tomando como referencia a Scopus, al mes de Junio del 2022 se reportaban más de 87 millones de documentos, y diariamente se publicaban aproximadamente 11.000 nuevos escritos [3]. Por ejemplo, una consulta en el área de las ciencias de la salud puede dar como resultado miles o decenas de miles de resultados [3]. Además, en las bases de datos científicas los resultados de una búsqueda se muestran a través de una lista ordenada por relevancia (relevancia frente a los criterios de búsqueda expresados por el usuario), la revisión de extensas páginas de resultados con documentos relevantes y no relevantes hace más lenta y tediosa la tarea de búsqueda y análisis bibliográfico. Se precisa una forma de soportar mejor este proceso, por ejemplo, ayudando a expandir los criterios de búsqueda para que el usuario reciba resultados más relevantes o agrupando los documentos que manejan temas similares a modo de categorías que le ayude al usuario a descartar conscientemente esos grupos de documentos que no se relacionan con sus necesidades de búsqueda y acotar revisión a los grupos que tienen mayor posibilidad de estar relacionados con su interés [4].

En cuanto a la simplificación de la revisión y selección de artículos de investigación se encuentra una amplia gama de herramientas y estrategias. En 2011, se presentó una metodología basada en las relaciones de citación, utilizando un grafo de distancias y relaciones para medir la relevancia entre dos artículos [5]. En 2016, se presentó una técnica llamada AKR que proporciona una lista inicial de artículos relevantes basados en las palabras clave especificadas por el autor [6]. En el mismo año, [7] presentó un enfoque basado en modelos de

contenido y bibliométricos haciendo uso de minería de datos y aprendizaje de máquina.

En 2019, se presentó el asistente inteligente FAST2 que utiliza métodos basados en resúmenes y aprendizaje de máquina para apoyar la selección de artículos relevantes [8]. Además, en el mismo año se realizó un estudio sobre la recomendación de artículos científicos, encontrando que se pueden aplicar diferentes métodos como filtrado basado en contenido, filtrado colaborativo, método basado en grafos y método de recomendación híbrida [9]. También en 2019, Chen y Ban [10] presentaron un sistema de recomendación que particiona las publicaciones de un investigador en grupos de interés haciendo uso del algoritmo de agrupamiento K-means y LDA (Latent Dirichlet Allocation) para puntuar los artículos. Este mismo año, en [11] se propuso una aplicación web en PHP para el "agrupamiento" de artículos usando similitud de cosenos y ponderación TF-IDF. En 2020, se presentó un método automático para agrupar artículos de investigación basados en sus títulos y palabras clave utilizando el algoritmo SOM [12]. En 2021, se propuso un nuevo enfoque de agrupación no supervisada de artículos usando incrustación de frases hallando un incremento en la calidad de los resultados en un 47,94% cuando se usa la distancia de coseno [13]. También en 2021, Jalal y Ali [14] presentaron el uso de "Document Clustering", TF-IDF, y el uso de similitud de cosenos para agrupar los artículos en categorías significativas. Para el 2022, se presentó la agrupación de artículos usando NLP, K-means y ponderación basada en TF y TF-IDF [15]. El uso de K-means en estas investigaciones da como resultado grupos sin solapamiento, es decir, un documento pertenece a un solo grupo y en el agrupamiento de documentos científicos esto puede ser inapropiado dado que, por ejemplo, una revisión sistemática puede estar relacionada con diversos tópicos de un área de interés.

A pesar de los avances mencionados anteriormente, y los que se detallan más adelante en la sección de "estado del arte", aún persiste el problema para los investigadores, ya que, al momento de consultar los artículos de investigación en bases de datos científicas se dificulta poder descartar de manera consciente aquellos documentos que no son relevantes y ubicar aquellos que sí son relevantes. Muchos de los avances realizados hasta el momento proponen la agrupación a partir de conjuntos de documentos, pero todavía quedan varios temas que no se han tenido en cuenta. El interés de esta investigación se motivó en que no existe una herramienta con la que un investigador pueda realizar sus consultas en Scopus (uno de los índices de documentos científicos más usado en el mundo) y pueda visualizar sus resultados agrupados de acuerdo con la similitud de su contenido temático y teniendo en cuenta que un documento puede estar en uno o más grupos, de tal manera que pueda identificar rápidamente los artículos más relevantes de acuerdo con su consulta. De acuerdo con lo anterior, en esta investigación se buscó resolver la siguiente pregunta de investigación: ¿Cómo

ayudar a que un investigador descarte de manera consciente grupos de artículos no relevantes a sus necesidades de información, documentos que son resultado de una consulta en Scopus?

Para acotar el alcance de la anterior pregunta de investigación a un trabajo de grado de pregrado en la FIET que fuese realizable en nueve (9) meses, se definió un enfoque de solución y basado en este se buscó resolver la siguiente pregunta de investigación especifica: ¿Cómo ayudar a que un investigador descarte de manera consciente grupos de artículos no relevantes a sus necesidades de información, mediante una aplicación que haga uso de técnicas de agrupamiento existentes, resultados de una consulta en Scopus y la visualización de los resultados por grupos temáticos? Por lo tanto, en este trabajo de investigación se desarrolló una solución, expresada en una aplicación web que permite realizar la búsqueda y selección de artículos basándose en una consulta construida por el usuario, esta aplicación toma los resultados obtenidos de la base de datos de Scopus y los muestra en grupos de acuerdo con el contenido temático de los documentos, grupos que permiten solapamiento de documentos (documentos multi tema como las revisiones sistemáticas pueden pertenecer a varios grupos) y con títulos o etiquetas que ayudan a identificar el contenido de los grupos, con lo que se le ayuda al usuario a tener una mejor comprensión de los documentos y su relevancia.

1.2 APORTES DEL PROYECTO

Desde la perspectiva de investigación el aporte del presente trabajo de grado se enfocó en la generación de nuevo conocimiento centrado en un nuevo método de visualización de grupos temáticos que permiten el solapamiento de documentos resultados de una consulta sobre Scopus, ordenados en cada grupo por su relevancia frente a la consulta y con títulos o etiquetas que buscan ayudar a identificar el contenido de los grupos. Lo anterior, buscando que el investigador invierta menos tiempo en identificar grupos de documentos relevantes y no relevantes a sus necesidades de información.

Desde la perspectiva de innovación, desarrollo y transferencia en este trabajo se desarrolló una aplicación web que permite hacer uso del método de visualización propuesto, que estará disponible para su uso a todos los investigadores de la Universidad del Cauca (como usuarios autorizados de Scopus) y con la cual se espera ayudar a disminuir el tiempo y esfuerzo requerido en cada investigación para definir el estado del arte relevante a un tema específico.

Es de resaltar que la información entregada por Scopus presenta variabilidad en términos de la estructura de los datos. Esto puede incluir cambios en los nombres de los campos, la organización jerárquica de los datos o incluso en la forma en que se representan los tipos de datos. Esta variabilidad de los datos implicó realizar validaciones para abarcar las diferentes estructuras que utiliza Scopus

para entregar la información, además, se realizó un filtrado de artículos que al menos contenga la información del resumen para efectos de realizar un correcto uso y procesamiento de los datos. Asimismo, fue necesario realizar modificaciones a los algoritmos de agrupamiento: Spectral, K-means y el índice de *SilhouetteScore* de la librería Sklearn (en Python) con el propósito de cambiar la medida de distancia utilizada por estos (Euclidiana); se realizaron modificaciones en su código fuente para hacer uso de la distancia de cosenos debido a que no se permitía esta configuración desde su implementación original, además, se agregó el parámetro de entrada *isunidimensional* con el fin de realizar un uso uniforme del método sin importar si los datos son unidimensionales o multidimensionales.

Se incorporó el uso de los algoritmos STC y Lingo dentro del servicio prestado por la aplicación; para ello se tomó la implementación realizada en carrot2 para lo cual fue necesario desplegar la REST API proporcionada por ellos en su documentación[1] mediante una imagen de Docker alojada en Google Cloud.

Sumado a esto y buscando proporcionar una aplicación NRT (Non-real time) fue necesario hacer uso de Celery y Redis. Celery es una biblioteca de Python que permite la creación y administración de tareas asíncronas y en segundo plano, diseñada para gestionar tareas distribuidas y programadas. Redis es una base de datos en memoria de código abierto que al usarlo con Celery actúa como un message broker, es decir, como un transportador de mensajes intermediario que permite enviar y recibir mensajes [16]. Estos dos elementos fueron usados para procesar de forma asíncrona la obtención de datos desde Scopus, el preprocesamiento de los datos, el agrupamiento y su etiquetado con el fin de permitir el uso de la aplicación de una forma continua sin tener que esperar la respuesta de consultas anteriores. Para ello fue necesario desplegar dos instancias de Celery en Google Cloud con el fin de tener dos trabajadores disponibles para procesar los datos de las consultas realizadas.

Finalmente, para la evaluación de las etiquetas con solapamiento se propuso una nueva forma para llenar la matriz de confusión, debido a que tradicionalmente la matriz de confusión no tiene en cuenta el solapamiento dentro del agrupamiento. Además, se realizó la implementación del mapeo de las clases predichas y las clases reales haciendo uso del método llamado enlace de centroide o método del centroide.

[1] https://carrot2.github.io/release/4.5.1/doc/rest-api-basics/

1.3 OBJETIVOS

A continuación, se presentan los objetivos tal y como fueron aprobados por el Consejo de Facultad de la Facultad de Ingeniería Electrónica y Telecomunicaciones al inicio del proyecto.

1.3.1 Objetivo General

Proponer un servicio software para facilitar la selección de artículos de investigación relevantes y no relevantes a una temática, basado en una consulta definida por un investigador, los resultados entregados por Scopus y la visualización de grupos de los documentos resultantes.

1.3.2 Objetivos Específicos

- Definir un algoritmo para el agrupamiento y visualización de resultados de una consulta sobre la base de datos bibliográfica Scopus con el objetivo de facilitar la identificación de documentos relevantes y no relevantes a las necesidades de un investigador haciendo uso del patrón de investigación iterativa propuesto por Pratt [17] como enfoque metodológico, además de cuatro o más conjuntos de datos del estado del arte o sintéticos y métricas clásicas del área de recuperación de información para la evaluación del algoritmo.

- Desarrollar una aplicación web que le ofrezca a los usuarios de la Universidad del Cauca la posibilidad de usar el algoritmo de agrupamiento y visualización propuesto con un tiempo de respuesta NRT (Non-real time) usando una arquitectura de microservicios con Django (Python) en el backend y Angular en el frontend y SCRUM como proceso de desarrollo.

- Determinar el nivel de satisfacción de cinco o más investigadores de la Universidad del Cauca al usar la aplicación web para determinar las posibles mejoras en el agrupamiento de los resultados, el ordenamiento de estos en cada grupo y las etiquetas generadas para identificar los grupos, lo anterior después de que los investigadores usan la aplicación por un tiempo mínimo de 30 minutos y luego contestan una encuesta semiestructurada.

1.4 RESULTADOS OBTENIDOS

A continuación, se resumen los principales resultados del presente trabajo de grado:

1. **Monografía de trabajo de grado**: Se refiere al presente documento que contiene primero, la motivación del trabajo (problema, aportes, objetivos y resultados), segundo, el estado de arte sobre el problema, tercero, el método propuesto para agrupamiento, visualización y selección de artículos, cuarto,

una breve explicación sobre la aplicación web desarrollada, quinto, la evaluación de la aplicación implementada, sexto, las conclusiones y el trabajo futuro y por último (séptimo), las referencias y los anexos.

2. **Código fuente y recursos:** Aplicación web desarrollada en Angular (**Anexo 1)**. Aplicación backend desarrollada en Django (**Anexo 2)**. Conjuntos de datos y Aplicación en Python utilizada para la evaluación y comparación de diferentes métodos usando los conjuntos de datos (**Anexo 3)**.
3. **Artículo**: Un artículo con los resultados del trabajo de grado de investigación sometido a evaluación en la revista Científica de la Universidad Distrital Francisco José Caldas (**Anexo 4)**.

1.5 ESTRUCTURA DE LA MONOGRAFÍA

A continuación, se describe de manera general el contenido y organización de la presente monografía:

CAPÍTULO 1: INTRODUCCIÓN: Hace referencia al presente capítulo que introduce el tema de investigación, presenta el problema que originó el trabajo, los aportes al problema luego de terminada la investigación, también los objetivos (general y específicos) definidos en el anteproyecto, un breve resumen de los resultados obtenidos y finalmente la organización de la monografía.

CAPÍTULO 2: ESTADO DEL ARTE: En este capítulo se presentan trabajos previos relacionados con herramientas o estrategias para facilitar la selección y revisión de artículos de investigación organizadas en orden cronológico.

CAPÍTULO 3: ALGORITMOS DE AGRUPAMIENTO, ETIQUETADO Y METRICAS DE EVALUACIÓN: En este capítulo se presentan los algoritmos de agrupamiento y etiquetado implementados, las métricas utilizadas para la evaluación de los algoritmos, la implementación del proceso de evaluación de los algoritmos y los conjuntos de datos utilizados para ello.

CAPÍTULO 4: IMPLEMENTACIÓN DE LA REST API Y LA APLICACIÓN WEB: En este capítulo se presenta el diseño detallado del servicio desarrollado junto a los aspectos más importantes de la implementación del proyecto en su backend y frontend.

CAPÍTULO 5: RESULTADOS EXPERIMENTALES Y ANÁLISIS: En este capítulo se presentan los resultados de la evaluación realizada con los 4 conjuntos de datos reales y la evaluación desarrollada por los investigadores usando el aplicativo web.

CAPÍTULO 6: CONCLUSIONES, RECOMENDACIONES Y TRABAJO FUTURO: En este capítulo se presentan las conclusiones obtenidas al finalizar el trabajo de investigación y unas ideas para mejorar el proyecto en un trabajo futuro.

CAPÍTULO 7: BIBLIOGRAFÍA: Este último capítulo contiene las referencias bibliográficas de los artículos, páginas y libros consultados para la realización del presente proyecto.

Esta página ha sido dejada intencionalmente en blanco.

● CAPÍTULO 2

2 ESTADO DEL ARTE

La literatura en materia de herramientas o estrategias para facilitar la selección y revisión de artículos de investigación presenta diversas opciones, a continuación, se presentan en orden cronológico y muy resumidas las que se consideraron más relevantes para la presente investigación. En 2011, en [5] se planteó una metodología basada en las relaciones de citación basado en un grafo de distancias y relaciones con peso, diferenciadas por dependencia, para medir la relevancia entre dos artículos dentro del grafo de citación. En 2016, en [6] se presenta una nueva técnica llamada AKR (Author-specified Keywords based Retrieval), para ello se toma como punto de partida un tema de búsqueda que se ve reflejado en palabras clave con las cuales se realiza la búsqueda y posteriormente se realiza un filtro basado en contenido utilizando el cálculo de similitud Okapi BM25 para cotejar los textos teniendo en cuenta el título, resumen y palabras clave del autor, de lo anterior se obtiene una lista de lectura que sirve de punto de partida para la revisión de la bibliografía sobre un tema de investigación determinado, teniendo en cuenta que estos artículos puedan ser populares, de estudio, recientes y diversos. De estos artículos se seleccionan 200 para posteriormente realizar el ranking de la lista final de artículos haciendo uso principalmente del valor TPC (Topical and peripheral coverage) y así dar como resultado la recomendación de 20 artículos. En el mismo año, Rubio T. y Gulo C. [7] presentaron un enfoque fundamentado en modelos basados en el contenido del artículo, así como en las características bibliométricas y usaron la metodología de minería de datos (“Knowledge Data Discovery”, KDD), posteriormente obtuvieron la información de un repositorio que luego de ser preprocesado encontraron los atributos de los metadatos más importantes para hallar publicaciones relevantes. Finalmente, hacen uso del algoritmo de aprendizaje de máquina ID3 para definir un modelo de clasificación que aprende de datos anotados por especialistas y con esto, realiza la recomendación para una investigación concreta.

Por otro lado, en el año 2019 se presenta el asistente inteligente FAST2 [8] el cual se basa en métodos basados en resúmenes con el fin de entrenar modelos de clasificación de texto y aplicar esos modelos para apoyar la selección de artículos relevantes, lo anterior acompañado de diferentes tipos de algoritmos de aprendizaje de máquina.

Este mismo año, en [9] se realizó un estudio acerca de la recomendación de artículos científicos haciendo uso de sistemas de recomendación fundamentados en métodos de filtrado basado en contenido (Content-Based Filtering, CBF),

métodos de filtrado colaborativo (Collaborative Filtering, CF), métodos basados en grafos (Graph-Based method, GB) o métodos de recomendación híbrida (Hybrid recommend method). Asimismo, expone que el funcionamiento de CBF se basa en obtener las palabras clave de un artículo candidato y calcula la similitud con las palabras clave extraídas de un perfil de usuario, el cual es un modelo que considera las preferencias históricas de un usuario y su librería personal, para posteriormente sugerir el artículo que se encuentre mejor ranqueado en cuanto a la similitud. Por otro lado, CF se enfoca en las calificaciones o acciones sobre los elementos de otros usuarios cuyos perfiles son similares a los del usuario inicial, es decir, usuarios parecidos o vecinos. En cuanto a los métodos basados en grafos, según estudios previos, estos son construidos a partir de nodos que representan a los autores y otros nodos que representan a los artículos, donde la relación entre estos dos nodos representa los vértices del grafo; una vez construido el grafo, se utiliza un algoritmo de grafos que permita computar la similitud entre los autores y los artículos. Asimismo, indican que para el método de recomendación híbrida es común usar el filtrado basado en contenido y el filtrado colaborativo, debido a que, de esta forma los sistemas de recomendación usualmente son más precisos. Finalmente, concluyen que, de acuerdo con el análisis realizado, los métodos basados en contenido e híbridos son las técnicas más usadas en sistemas de recomendación de artículos.

En 2020, en [18] se propone un enfoque alternativo para la creación de un marco de trabajo que personaliza la recomendación de artículos basándose en las asociaciones ocultas entre un artículo de investigación y los metadatos contextuales públicos, haciendo uso de similitud colaborativa y similitud basada en contenido, teniendo en cuenta que, no es necesaria la información de un perfil de usuario a priori cuando se parte de una consulta o mensaje de búsqueda.

Es de resaltar que en 2021 van Dinter R., Tekinerdogan B. y Catal C. [19] realizaron una revisión sistemática de la literatura acerca de la automatización de la revisión sistemática de la literatura, esta revisión abarca desde enero del año 2000 hasta junio del 2020 e incluye 41 artículos publicados entre el 2006 y el 2020. Los autores definieron los resultados de su estudio dentro de una estructura de 6 preguntas, a las cuales dan respuesta con lo hallado de la siguiente manera: 1) ¿Cuál es el objetivo de realizar la automatización?, la mayoría de los artículos utilizan la automatización con el propósito de disminuir el costo de realizar la revisión sistemática. 2) ¿En qué ámbitos de aplicación se evalúan las herramientas de revisión bibliográfica sistemática automatizada?, los dominios de aplicación son predominantemente en ingeniería de software y medicina, seguidos de salud pública, medio ambiente, psiquiatría, farmacia, entre otras. 3) ¿Cuáles son las bases de datos utilizadas?, Las bases de datos más usadas fueron IEEE Xplore, PubMed, Medline, ACM Digital Library, Springer Link, entre otras. Sólo en [20] se describe el uso de un algoritmo en Python para la extracción de los metadatos (título, resumen, palabras clave, año de publicación, autores y doi) de

los artículos publicados en IEEE, ACM, Springer, Scopus, y Semantic Scholar, teniendo en cuenta que para ACM y Semantic Scholar los autores usan “Web Scraping” basado en la librería Selenium de Python. 4) ¿Cuáles son los pasos que automatizan de la revisión sistemática?, la selección de los estudios primarios es el paso que más se intenta automatizar debido a que es una de las acciones que consumen más tiempo. 5) ¿Cuáles son las técnicas aplicadas y cuales partes de un documento son utilizadas en el algoritmo? Predominantemente se hace uso de técnicas de preprocesamiento y representación de PLN (Procesamiento de lenguaje natural). Se destaca el llevar una palabra a su raíz sintáctica (Stemming), la eliminación de palabras vacías (stop words) y la aplicación del modelo de bolsa de palabras (bag of words) usando ponderación TF-IDF (Term Frequency-Inverse Document Frequency). También hallaron que la tarea del aprendizaje de máquina más usada fue el aprendizaje supervisado (clasificación) tanto para la selección de estudios primarios como para la extracción de información.

Asimismo, encontraron que en la identificación de la investigación la tarea más realizada es el “ranking” y en el momento de realizar la evaluación de la calidad de los estudios se destaca el uso de agrupamiento o análisis de grupos. La métrica que no es de aprendizaje de máquina más utilizada es WSS (Work Saved over Sampling). Además, encontraron que se hace uso de K-NN, “Latent Dirichlet Allocation”, árbol de decisión, K-means, “Bit-priori Association Classification Algorithm”, Regresión logística, redes neuronales, “Denoising Autoenconder”, Modelo lineal generalizado, “Rocchio”, Matriz de factorización no-negativa, “Supervised Distant Supervision”, “Voting Perceptron”, “Support Vector Machines” y Redes Bayesianas. En cuanto a las partes del documento utilizadas se encuentran los metadatos, términos MeSH (Medical Subject Headings), en algunos casos el documento completo, el tipo de publicación, entre otros. Finalmente, la pregunta 6) ¿Cuáles son los retos pendientes y las vías de solución?, algunos de los retos planteados son: el desequilibrio de clases en el entrenamiento del modelo de selección de los estudios primarios, la necesidad de un sistema que obtenga los artículos completos de las bases de datos, la necesidad de seleccionar las mejores características para los modelos planteados, la carencia de algún tipo de conversión de PDF para ser utilizado como entrada dentro del modelo, comentan que no es del todo una alternativa apropiada el uso de técnicas de aprendizaje activo ya que, el usuario pierde el control sobre los resultados. Por último, agregan que las maquinas no pueden comprender un modelo canónico por lo que la información debería ser traducida de tal forma que sea comprendida por la máquina.

De forma similar, Kreutz y Schenkel [21] en el 2022, realizaron una revisión bibliográfica de las ultimas publicaciones acerca de los sistemas de recomendación de artículos científicos incluyendo artículos que se encuentren entre enero del 2019 y octubre del 2021, obteniendo 65 artículos de los cuales pudieron observar que se encuentran predominantemente cuatro categorías de

enfoques, siendo estos: Filtrado basado en contenido, Filtrado colaborativo, Filtrado basado en grafos y Filtrado con sistemas híbridos. Además, categorizaron los artículos seleccionados en diferentes dimensiones encontrando que en cuanto al enfoque general de información hallaron dos categorías: por personalización (basada en perfiles de usuario) y con una entrada (basado en un artículo inicial, palabras clave, consulta, etc.). Por otro lado, también observaron que la mayoría de los artículos toman información relacionada directamente del artículo como el título, resumen, palabras clave, fragmentos de texto, citaciones, así como un historial de interacciones que tenga el usuario con la información.

Asimismo, observaron la prevalencia de métodos con los cuales pueden crear o reestructurar información, tal como la construcción de perfiles de usuario, el uso de un indicador como la popularidad o frases clave, el uso de alguna técnica de incrustación de texto o grafos, modelado de temas, grafos o grafos de conocimiento, el uso de “Meta-path”, “Random Walk”, aprendizaje de máquina avanzado, “Web Crawling” y usar la similitud de cosenos. También, observaron que 32 artículos hacen uso de perfiles de usuario los cuales en la mayoría de los casos son construidos a partir de la interacción del usuario como lo puede ser un histórico de interacciones no específicas como la media de la representación de los documentos con los cuales interactúa, el tiempo que dedica al comportamiento de interacción, documentos marcados como favoritos o destacados, documentos leídos, documentos calificados, documentos a los cuales les ha dado clic o seleccionado, tweets, interacciones sociales, entre otros. Es de resaltar que, el perfil de usuario también puede ser construido a partir de los artículos que haya escrito el usuario.

Por otro lado, encontraron que 13 artículos usan algún tipo de medida de popularidad, por ejemplo, popularidad basada en el autor, popularidad basada en el espacio y popularidad basada en el artículo: la primera se basa en la autoridad, así como en la cantidad de citaciones que recibe el autor, entre otros elementos; la segunda consiste en una noción de reputación no especificada, como también un factor de impacto, y la tercera se basa en la cantidad de artículos citados, así como también puede ser la cantidad de descargas o menciones en redes sociales. Del mismo modo hallaron que sólo 4 artículos usan alguna forma de frases clave construidas a partir de títulos y resúmenes haciendo uso métodos automáticos como TF-IDF o “Distiller Framework”.

Muchos artículos hacen uso de alguna forma de incrustación basada en métodos existentes de representación de documentos como Word2Vec, Doc2Vec, BERT, u otros. Ocho (8) artículos hacen uso de un componente de modelamiento de temas, siendo LDA el más usado para la representación de contenido de un documento. Además, 33 artículos hacen uso de grafos, aunque solo 6 utilizan los grafos de conocimiento. En cuanto a los que usan grafos encontraron que estos utilizan medidas de centralidad, alguna forma de “PageRank”, construcción de árboles

Steiner, agrupamiento o cálculo del grado de cercanía. Asimismo, solo cuatro (4) enfoques hacen uso de “Meta-paths” así como 12 artículos utilizan “Random Walk” como metodología. Veintinueve (29) artículos hacen uso de alguna forma de aprendizaje de máquina avanzado como LSTM, perceptrones multi capa, factorización de matrices, redes neuronales, entre otras. Adicionalmente, hallaron que 9 artículos incorporan en su enfoque el uso de “Crawling”. Finalmente, observaron que 31 artículos hacen uso de la similitud de cosenos aplicado entre artículos, entre usuarios o entre usuarios y artículos, entre otras formas de aplicación.

A raíz del surgimiento de la inteligencia artificial generativa han surgido numerosas aplicaciones haciendo uso de está en el campo de la investigación. La inteligencia artificial generativa según [22] “se puede definir como una tecnología que aprovecha los modelos de aprendizaje profundo para generar contenido similar al humano (por ejemplo, imágenes, palabras) en respuesta a indicaciones complejas y variadas”, por lo tanto, al ser una herramienta tan versátil la vuelve idónea en campos multidisciplinarios. Por ejemplo, en 2023 [23] menciona que ChatGPT puede usarse en la búsqueda de artículos científicos y académicos hallando que es una herramienta útil y con gran potencial si se utiliza de forma adecuada y ética; aunque agregan que debería integrarse a plataformas académicas para reducir el porcentaje de plagio y organizar el proceso de publicación. En el mismo año, en [24] se agrega que ChatGPT no es recomendable para realizar una revisión sistemática de la literatura debido a que entrega demasiados documentos falsos y por lo tanto no es un sustento valido para un proceso de investigación. Por lo tanto, a pesar de que es una herramienta de gran potencial su uso es poco confiable.

Asimismo, existen numerosas herramientas que hacen uso de inteligencia artificial enfocadas en la búsqueda de artículos científicos, por ejemplo: consensus.app[2], Elicit.org[3], Scite.ai[4], Research Rabbit[5], entre otros. Un elemento en común de estas herramientas es que los resultados que entregan tienen una estructura de lista, lo que implica que el usuario tarde más tiempo en realizar la búsqueda de los artículos relevantes a su consulta, ya que no puede desechar conscientemente grupos de documentos no relevantes.

Es de resaltar que en [25] (2007) se habla de “La hipótesis del agrupamiento” la cual consiste en que “los documentos fuertemente asociados tienden a ser relevantes para la misma consulta” [26] por lo tanto al agrupar los documentos por su contenido, éstos serán relevantes a su grupo y así el usuario no tiene que

[2] https://consensus.app/
[3] https://elicit.org/
[4] https://scite.ai/
[5] https://www.researchrabbit.ai/

revisar todos los documentos de un grupo para determinar aquellos documentos relevantes o irrelevantes a su búsqueda.

Teniendo en cuenta lo anteriormente mencionado, el enfoque de solución planteado dentro del proyecto consistente en agrupar los artículos mediante una técnica de agrupamiento para formar categorías o temáticas, y dentro de cada categoría ordenar o ranquear los artículos por relevancia a la consulta del usuario (que expresa sus necesidades de información), se procedió a analizar la literatura que aplica un enfoque similar. Es de resaltar que, la clasificación de publicaciones en investigación puede ser separada principalmente en dos categorías: enfoques basados en contenido y técnicas basadas en metadatos [27]. En 2014 en [28] se propone un modelo de recomendación de artículos en el cual se clasifican los artículos por comunidades a través de una red de citación haciendo uso del algoritmo “Greedy Clique Expansion” y utilizando del algoritmo “PaperRank” dentro de cada comunidad para hallar los artículos más relevantes. Este mismo año, en [2] proponen un nuevo método para la organización y recuperación de artículos de investigación, en el cual plantean la agrupación de artículos basada en el centroide y en las relaciones entre estos, este enfoque tiene en cuenta el título del artículo y la información relacionada con este, como las palabras clave, las frases más frecuentes relacionadas y las referencias más similares, y sus referencias más similares para agrupar. Es de resaltar que, el algoritmo planteado para agrupar trabajos de investigación similares se basa en el algoritmo K-means el cual parte de que un centroide para representar un “clúster” o grupo. Asimismo, tienen en cuenta las relaciones de un artículo como lo pueden ser las palabras en común en el título del artículo, palabras en común con las palabras clave del artículo, las palabras clave en las frases más frecuentes y las palabras más comunes en las referencias más similares, además utilizan la medida de la similitud entre artículos basándose en las relaciones.

Chen y Ban [10] en 2019 proponen un sistema de recomendación que particiona las publicaciones de un investigador en grupos de interés haciendo uso del algoritmo K-means, además construyen un modelo de necesidad académica basado en una clase de equivalencia de patrones, para la cual cada artículo que se encuentre en el mismo punto de interés es usado para construir un modelo LDA (Latent Dirichlet Allocation) y donde los patrones mejorados son generados a partir de la eliminación de palabras repetidas de los temas del LDA y la extracción de patrones frecuentes. Finalmente, calculan la puntuación de cada documento candidato por grado de coincidencia de patrones y grado de preferencia para realizar la recomendación de n artículos por cada conjunto de intereses ordenados de acuerdo con el puntaje obtenido.

En este mismo año (2019), en [29] se propone “PubTeller” un nuevo enfoque híbrido de recomendación de publicaciones utilizando información compuesta, para ello a partir de un repositorio de publicaciones y una entrada que en este

caso serían las palabras clave en las que está interesado el usuario, se realiza "Paper Ranking", el cual calcula un puntaje de clasificación de cada artículo en el depósito de publicaciones en función de su relevancia con las palabras clave introducidas, para hacer esto hacen uso de dos tipos de recomendación: basada en contenido y basada en citaciones. Por lo tanto, tienen en cuenta el contenido del artículo para construir los vectores de términos, la información sobre el lugar de publicación, el año de publicación del artículo y una red de citas. Además, proponen un algoritmo de agrupamiento para agrupar los artículos que tienen temas similares basándose en las relaciones de citación para finalmente presentar los top-k artículos más relevantes.

También en 2019, en [11] se presentó un sistema en forma de aplicación web que se encuentra dividida en dos módulos, el primero para extracción de palabras clave y el segundo para agrupamiento. El primer módulo consiste en una actividad definida para identificar automáticamente un conjunto de términos que describen el tema o sujeto del documento, además dentro de este módulo hacen uso del método de ponderación TF-IDF, asimismo, para calcular la similitud de palabras hacen uso de la similitud de cosenos teniendo en cuenta que este último es utilizado para realizar la agrupación de los artículos según su contenido. En 2020 Ahmed R., Salama C., Mahdi H. [12] proponen un enfoque para la agrupación automática de documentos de texto utilizando un Mapa Autoorganizado o "Self-Organizing Map" (SOM). Este algoritmo es aplicado con el propósito de mejorar y simplificar el proceso de obtención de información como lo son los artículos. Además, hacen uso del modelo Word2Vec para detectar la similitud entre las palabras y también usan un algoritmo genético (Genetic Algorithm, GA) para optimizar los parámetros de las entradas aleatorias de SOM. Al hacer uso de SOM los resultados se despliegan de una forma visualmente atractiva, además, de que no depende de un número de grupos o "clústeres" predeterminados.

Finalmente, en el año 2022, en [15] se propone un enfoque que consiste en la agrupación automática de artículos científicos basada en el análisis de su texto, haciendo uso de procesamiento de lenguaje natural y el algoritmo K-means para realizar la agrupación de documentos o "Document Clustering". Para lo anterior toman los artículos científicos y los convierten de PDF a archivos de texto, posteriormente el contenido de los artículos es dividido en 3 colecciones, donde la primera colección contiene el resumen de los documentos, la segunda contiene la introducción y el tercero contiene el texto completo sin el resumen; es de resaltar que las palabras clave son utilizadas para evaluar el trabajo realizado en la agrupación. Una vez hecho esto aplican métodos de NLP como "Stemming" para cambiar las palabras que comparten la misma raíz a una forma común y "Lematization" para llevar una palabra a su raíz, además hacen uso de las ponderaciones TF y TF-IDF para la creación de matrices que serán la base para crear los grupos haciendo uso del algoritmo K-means. Por último, para evaluar el

trabajo realizado en la agrupación proponen crear una matriz de conexión entre artículos.

● CAPÍTULO 3

3 ALGORITMOS DE AGRUPAMIENTO, ETIQUETADO Y MÉTRICAS DE EVALUACIÓN

3.1 METODOLOGÍA DE DESARROLLO

En este trabajo se usó el patrón de investigación iterativa (PII) propuesto por Pratt [17] y diseñado fundamentalmente para proyectos de investigación relacionados con una solución computacional. Este patrón consta de cuatro etapas principales que son: observación (observaciones de campo en el marco del problema), identificación del problema, desarrollo de la solución y prueba de la solución. Estas etapas se desarrollaron en 4 iteraciones incrementales o en ciclos de (3) semanas cada uno.

Iteración 1: Diseño Inicial y Agrupamiento

1. **Observación**: Se analizaron los algoritmos de agrupamiento del estado del arte para comprender sus características y resultados.
2. **Identificación del problema**: Se determinó cómo aplicar los algoritmos para agrupar los resultados de la consulta en Scopus.
3. **Desarrollo de la solución**: Se implementaron cinco (5) algoritmos seleccionados basados en la investigación previa.
4. **Prueba de la solución**: Los algoritmos se evaluaron mediante pruebas con 4 conjuntos de datos reales y métricas de recuperación de información.

Iteración 2: Mejoras y Optimización

1. **Observación**: Se evaluaron los resultados de la iteración anterior para identificar áreas de mejora.
2. **Identificación del problema**: Se establecieron aspectos específicos de los algoritmos que requerían optimizaciones o mejoras y se propusieron estrategias para ello.
3. **Desarrollo de la solución**: Se implementaron cambios en los algoritmos que lo requerían para mejorar los resultados.
4. **Prueba de la solución**: Se evaluaron las mejoras introducidas utilizando el mismo proceso que en la iteración anterior.

Iteración 3: Etiquetado de Grupos

1. **Observación**: Se investigaron algoritmos de etiquetado de grupos en el estado del arte, enfocados en el contenido de los artículos.
2. **Identificación del problema**: Se determinó cómo aplicar los algoritmos para etiquetar grupos de tal forma que las etiquetas halladas representaran de la forma más clara posible la temática de cada grupo.
3. **Desarrollo de la solución**: Se implementaron cinco (5) algoritmos de etiquetado en el proceso.
4. **Prueba de la solución**: Se realizó una revisión de las etiquetas entregadas por los algoritmos de etiquetado, observando la coherencia de estas con el contenido de los artículos.

Iteración 4: Versión final e Integración

1. **Observación**: Se evaluaron los componentes de la solución y seleccionaron aquellos que entregaron los mejores resultados para definir un sistema que cumpliera con lo requerido.
2. **Identificación del problema**: Se definió la forma de integrar nuevos componentes sin afectar el rendimiento existente.
3. **Desarrollo de la solución**: Se realizó la integración de los algoritmos desarrollados en una versión final del sistema.
4. **Prueba de la solución**: Se sometió la versión final a pruebas integrales para garantizar su funcionalidad y rendimiento.

Además, se usó SCRUM como guía para el desarrollo del “backend”, en el cual se exponen los algoritmos realizados con anterioridad a través del desarrollo de un servicio RESTful desarrollado en Python, haciendo uso del framework Django. Asimismo, se implementó un “frontend” haciendo uso de Angular con el fin de consumir los servicios prestados por el “backend”.

Finalmente, se realizó la evaluación del servicio desarrollado con la participación de once (11) evaluadores o investigadores mediante una encuesta dispuesta en el mismo aplicativo web. El análisis de los resultados se presenta más adelante en el **Capítulo 5**.

3.2 IMPLEMENTACIÓN DE LOS ALGORITMOS DE AGRUPAMIENTO

El agrupamiento de documentos es un proceso no supervisado que tiene como objetivo agrupar un conjunto de textos sin etiquetar, de forma que los textos de un mismo grupo sean similares entre sí [30]. Para lograr esta tarea se realizaron una serie de pasos que se describen a continuación.

3.2.1 PREPROCESAMIENTO Y PREPARACIÓN DE LOS DATOS

Los pasos realizados en el preprocesamiento y preparación de los datos para los algoritmos K-means, Spectral y Fuzzy C-means se describen en la Tabla 1.

Tabla 1. Resumen del preprocesamiento para K-means, Spectral y Fuzzy C-means

Paso 1	Transformación del texto de mayúsculas a minúsculas (Case Folding).
Paso 2	Dividir el documento en palabras o términos (Tokenización).
Paso 3	Eliminación de palabras vacías o no deseadas (Filtrado).
Paso 4	Llevar a cada termino a su palabra raíz, quitando y reemplazando sufijos de la raíz de la palabra (Stemming).
Paso 5	Construcción de la matriz TF-IDF con los textos preprocesados construyendo unigramas, bigramas y trigramas.
Paso 6	Implementación de LSA (Latent Semantic Analysis) haciendo uso de descomposición truncada del valor singular (SVD).

Los pasos realizados en el preprocesamiento y preparación de los datos para los algoritmos STC y Lingo se describen en la Tabla 2. Debido a que estos algoritmos son de terceros, reciben como entrada el texto completo, por lo tanto, no se podrían aplicar los pasos 5 y 6 de la Tabla 1, además, si se aplican los pasos 1 y 4 de la anterior tabla se podría estar modificando el funcionamiento y los resultados entregados por estos dos algoritmos.

Tabla 2. Resumen del preprocesamiento para STC y Lingo

Paso 1	Dividir el documento en palabras o términos (Tokenización).
Paso 2	Eliminación de palabras vacías o no deseadas (Filtrado).

En el proceso de realizar el agrupamiento de documentos es esencial preparar los datos de forma adecuada para facilitar el procesamiento y para obtener resultados coherentes e idóneos, por ende, en la literatura se encuentran de forma general cuatro (4) pasos para preprocesar la información: Case folding, Tokenización, Stemming y Filtrado (remoción de palabras vacías como los artículos y las preposiciones, que no superen una longitud mínima, que no se encuentren un número mínimo de veces en la colección de documentos, entre otros.) [31]. Primero se realizó la transformación del texto de mayúsculas a minúsculas (**Case folding**) con el fin de estandarizar la entrada de datos. Segundo, se tokenizaron los textos, es decir, los documentos se dividieron en palabras o términos (**Tokenización**). Tercero, se quitaron las palabras no deseadas (**Filtrado**), en este caso, las palabras vacías, las cuales se obtuvieron de la librería NLTK. Además, cuando se reciben los datos de Scopus se eliminan palabras que generan ruido dentro del texto (el texto es el resultado de la unión del resumen, las palabras clave y el título de cada artículo, capítulo de libro, memoria de evento, entre otros.) como por ejemplo la información de la editorial o el año de publicación, entre otros.

Finalmente, se llevó cada término del documento a su palabra raíz (**Stemming**), esto con el fin de facilitar el reconocimiento de palabras iguales o similares independientemente de su conjugación.

Posteriormente, se realizó la construcción de una matriz TF-IDF (Term Frequency – Inverse Document Frequency) con la unión de los textos de los documentos recuperados de Scopus que ya fueron preprocesados. Es de resaltar que está matriz indica la relevancia de un término dentro de la colección de documentos. Dentro de la construcción de esta matriz no se hace uso de la normalización (dejar los datos en un rango determinado, por ejemplo, entre 0 y 1), y se usa como representación de cada documento los unigramas, bigramas y trigramas con el fin de brindar la posibilidad de que las frases más significativas puedan representar a un grupo. Finalmente, se define que cuando una consulta reporte 200 documentos o menos se ignoren los términos que tengan una frecuencia de documentos inferior a 2, en caso contrario se ignoran los términos que tengan una frecuencia de documentos inferior a 10. Asimismo, se ignorarán los términos que tienen una frecuencia de documentos superior al 80% del número total de documentos recuperados.

Según [32] la implementación de LSA (Latent Semantic Analysis) en la matriz TF-IDF aumenta el rendimiento de los grupos al separar de mejor manera los elementos que no pertenecen a la misma clase. Este proceso se realiza mediante un recorte de las matrices entregadas por la descomposición en valores singulares (SVD) de la matriz TF-IDF. SVD mapea cada palabra en un subespacio con un número de dimensiones previamente establecido [32]. LSA se usó como entrada únicamente para algoritmos de agrupamiento K-means, Spectral y Fuzzy C-means. Los algoritmos STC y Lingo no usaron LSA, debido a que en el servicio prestado por Carrot2 requiere que los documentos se entreguen en su forma original, es decir, el texto plano completo.

Una vez realizado lo anterior se procedió a realizar el proceso de agrupamiento de los documentos.

3.2.2 ALGORITMOS DE AGRUPAMIENTO

Los pasos realizados en la implementación de los algoritmos de agrupamiento: Spectral, K-means y Fuzzy C-means se describen en la Tabla 3. Es de resaltar que, para estos pasos se toma como entrada la matriz TF-IDF con el preprocesamiento transformada con LSA.

Tabla 3. Resumen del procedimiento realizado en los algoritmos de agrupamiento: K-means, Spectral y Fuzzy C-means

Paso 1	Determinar la cantidad máxima de grupos usando la **Ecuación 1**
Paso 2	Determinar la cantidad de grupos mediante el uso del coeficiente de Silhouette o el criterio de información bayesiano (BIC) o el índice Davies-Bouldin teniendo en cuenta la cantidad máxima de grupos

	calculada en el Paso 1.
Paso 3	Realización del agrupamiento mediante K-means o Spectral o Fuzzy C-means
Paso 4	Realización del solapamiento (solo aplica para K-means y Spectral) mediante la detección de bordes atípicos dentro de cada grupo
Paso 5	Filtrado del solapamiento (solo aplica para Fuzzy C-means). Para cada documento se seleccionan aquellos grupos con los que presenta mayor pertenencia, estas pertenencias se suman hasta que el valor sea igual o mayor a 95%. Es decir, se desechan los grupos con los cuales el documento presenta menos pertenencia.

Los pasos realizados en la implementación de los algoritmos STC y Lingo se describen en la Tabla 4. Es de resaltar que, para estos pasos se toma como entrada la lista de documentos.

Tabla 4. Resumen del procedimiento realizado en los algoritmos de agrupamiento STC y Lingo

Paso 1	Determinar la cantidad máxima de grupos usando la **Ecuación 1** (solo aplica para STC)
Paso 2	Realización del agrupamiento mediante STC o Lingo.

Según [33] un algoritmo de agrupamiento es “un procedimiento de aprendizaje que busca identificar las características específicas de los grupos subyacentes a un conjunto de datos”. Dentro de la literatura se pueden encontrar diferentes algoritmos de agrupamiento de los cuales dentro del presente proyecto se hizo uso de K-means, Spectral, Fuzzy C-means, STC y Lingo.

Dentro del agrupamiento un reto importante ha sido determinar la cantidad de grupos, debido a que es un factor que afecta la calidad y coherencia del agrupamiento. Por ende, existen diversos métodos para intentar hallar este valor como el método del codo, el método de la estadística de huecos, el coeficiente de Silhouette, el índice Calinski-Harabasz, el índice Davies-Bouldin, el criterio de información bayesiano, entre otros [34]. En el presente proyecto se implementaron el coeficiente de Silhouette, el criterio de información bayesiano (BIC) y el índice Davies-Bouldin, teniendo en cuenta que BIC entregaba una cantidad de grupos más equilibrada y por lo tanto se dejó como algoritmo predeterminado. Es de resaltar que para todos los métodos se definió una cantidad máxima de grupos calculada con la regla empírica presentada en la **Ecuación** 1.

$\#\ máximo\ de\ grupos = \ ceil\left(\sqrt{\frac{N}{2}}\right), N\ es\ la\ cantidad\ documentos$	1

Lo anterior con el propósito de presentarle al usuario una cantidad de grupos apropiada, con el fin de proporcionar un valor agregado en relación con la calidad de resultados entregados.

El coeficiente de Silhouette consiste en ejecutar n veces el algoritmo K-means con k número de grupos y por cada ejecución medir el puntaje de coeficiente de Silhouette, al final se toma la cantidad de grupos con la cual se obtuvo el mayor puntaje.

El índice Davies-Bouldin consiste en ejecutar n veces el algoritmo K-means con k número de grupos y por cada ejecución medir el puntaje del índice Davies-Bouldin, al final se toma la cantidad de grupos con la cual se obtuvo el menor puntaje.

Finalmente, BIC consiste en ejecutar la distribución de probabilidad de un modelo de mezcla gaussiana probando con 4 tipos de covarianza diferentes: spherical, tied, diag y full. Además, por cada covarianza se prueban k número de grupos que van de en un rango de 2 hasta # *máximo de grupos* descrita en la **Ecuación 1** y con cada ejecución se obtiene un puntaje BIC. Al final se toma la cantidad de grupos que tenga el menor puntaje independientemente del tipo de covarianza. Es de resaltar que, si la cantidad de documentos que provienen de la consulta es mayor o igual a 100 el rango de prueba de k varía entre el # *máximo de* grupos*/3* y # *máximo de* grupos*.* Lo anterior se realiza debido a que se desea acotar el espacio de búsqueda para disminuir el tiempo de procesamiento de tal forma que sea posible representar las diferentes granularidades de los documentos.

Una vez se definen la cantidad de grupos se procede a hacer uso de los algoritmos de agrupamiento: K-means, Spectral y Fuzzy C-means.

El algoritmo de K-means divide los datos en k grupos, cada uno con un valor medio (centroide), cada individuo se coloca dentro del grupo con el cual se tiene más cercanía con respecto al centroide de ese grupo [35]. Para la implementación de este algoritmo hicimos uso de la librería de sklearn, de la cual modificamos el código fuente de este algoritmo para que use la distancia de cosenos en lugar de distancia euclidiana. Esto se debe a que en el agrupamiento de documentos la distancia de cosenos posee un mejor rendimiento en cuanto a generar grupos más coherentes y con valores de pureza más altos con respecto a los valores entregados por la distancia euclidiana [36], además en [37] se observa que, al usar distancia de cosenos, distancia Euclidiana, Euclidiana2, y Manhattan, los mejores resultados de acuerdo con las métricas, se obtienen con distancia de cosenos. Es de resaltar que, en la estrategia de reducción de dimensionalidad (LSA) se definió un número de dimensiones igual a 5 para este algoritmo.

Finalmente, se realizó la ejecución del algoritmo K-means de la librería sklearn configurando la cantidad de iteraciones máximas igual a 1000 y la inicialización de los centroides usando el método *K-means ++.* Una vez se obtienen las etiquetas generadas por el algoritmo, se realiza un proceso para seleccionar documentos

que pueden ser solapados, para ello se crea una matriz de distancia que contiene la distancia de coseno de cada documento con el centroide de cada grupo; posteriormente para cada grupo se calculan unos topes mediante el cálculo del percentil 25 y 75 de las distancias de los documentos que pertenecen a ese grupo, con estos valores se calcula el rango intercuartílico $IQR = Q3 - Q1; donde, Q3\ y\ Q1\ son\ los\ percentiles\ 75\ y\ 25\ respectivamente$. A partir del IQR se realiza la detección de bordes atípicos por cada grupo que llamaremos tope inferior y tope superior, para el tope inferior se aplicó la siguiente formula $tope\ inferior = Q1 - (1.5 * IQR)$, para el tope superior se aplicó $tope\ superior = Q3 + (1.5 * IQR)$ [38]. Dichos topes se usaron para implementar el solapamiento verificando los documentos que se encuentran por encima del tope superior de cada grupo, con cada uno de estos documentos que cumplen dicha condición se evalúa si la distancia que tienen con los centroides de los demás grupos es menor que el tope superior de cada grupo, en tal caso se solapa ese documento en ese grupo. Estas condiciones se realizan porque si un documento está por encima del valor aceptable de distancia hacia el centroide de su grupo, es probable que pueda pertenecer a otro grupo sin dejar de pertenecer al grupo donde se encuentra. Con los procesos mencionados anteriormente se termina el procesamiento realizado por K-means y se retornan las nuevas etiquetas con solapamiento.

Por otro lado, el algoritmo de Spectral es una técnica que explota las propiedades del grafo Laplaciano, cuyas aristas denotan las similitudes entre los puntos de datos. La técnica de agrupamiento Spectral divide un conjunto de datos dado en grupos diferentes más pequeños en función de algunas propiedades específicas [39], [40].

Para la implementación de este algoritmo se hizo uso de LSA de la misma forma que en K-means, teniendo en cuenta que el número de dimensiones es de 150 si la cantidad de documentos es mayor a este número, sino toma el valor de la cantidad de documentos. Además, se usó la librería de sklearn, a la cual también se le modificó su código fuente para que usara distancia de cosenos. El algoritmo se configuró con una estrategia de descomposición de valores propios de arpack, además se definió la construcción de la matriz de afinidad con cosenos, y la estrategia para asignar etiquetas en el espacio de incrustación de cluster_qr. Debido a que este algoritmo no entrega etiquetas con solapamiento, se aplica el mismo procedimiento descrito en el algoritmo de K-means para entregar etiquetas con solapamiento.

Por otra parte, Fuzzy C-means es una modificación del algoritmo K-means que utiliza la lógica difusa con la cual el algoritmo calcula el valor de pertenencia de cada punto de datos para cada centroide, asignando un valor entre 0 y 1 [41].

Para la implementación de este algoritmo se hizo uso de la función de StandarScaler de la librería de sklearn con el fin de estandarizar las características

de los datos eliminando la media y escalando a la varianza unitaria. Además, se hizo uso de LSA de la misma forma que en K-means pero usando los datos estandarizados y teniendo en cuenta que el número de dimensiones para este algoritmo es 2.

Por otro lado, también se hizo uso de un dataframe de pandas, junto con un apilamiento de matrices de las dos dimensiones o componentes. Además, el algoritmo se obtuvo de la librería skfuzzy[6] y se configuró con una cantidad de iteraciones máximas de 1000, una exponenciación de matriz aplicada a la función de pertenencia de 2, un criterio de parada (error) de 0.005 y para realizar el agrupamiento se entregan los datos en forma de pila o el apilamiento que se mencionó anteriormente. Es de resaltar que este algoritmo entrega como resultado una matriz de pertenencia, la cual muestra la medida en la que cada documento pertenece a diferentes grupos. Para realizar el filtrado del solapamiento se ordenan las pertenecías que tiene cada documento hacia todos los grupos. El objetivo principal es seleccionar los grupos con los que un documento tiene mayor pertenencia, donde la sumatoria de las pertenecías del documento con los grupos seleccionados sea de al menos el 95% de pertenencia total del documento. Debido a que este algoritmo ya entrega etiquetas con solapamiento, no fue necesario aplicar un procesamiento adicional para ello.

Por último, los algoritmos STC y Lingo son algoritmos de agrupamiento que proporciona el código abierto de Carrot2 Document Clustering Workbench. STC se basa en la construcción de un árbol de sufijos generalizado (GST) el cual recorre con el propósito de reconocer términos y expresiones que se repiten con regularidad en los textos de entrada, y así combinar conjuntos de documentos que comparten una gran similitud entre sí [42]. Por otro lado, el algoritmo Lingo utiliza métodos de disminución de la dimensión de la matriz de términos en el documento con el fin de identificar la composición temática en los datos de entrada [42].

Para realizar el uso de estos dos algoritmos fue necesario desplegar la REST API que proporciona carrot2 en su repositorio de GitHub[7]. Para implementar el uso de estos algoritmos se realiza una petición al servicio enviando un JSON con los documentos y la configuración del algoritmo que se desea usar. Es de resaltar que la respuesta del API está estructurada como una lista de grupos, indicando qué documentos lo conforman y el título que lo representa. Debido a que estos algoritmos ya entregan etiquetas con solapamiento, no fue necesario aplicar un procesamiento adicional para ello.

[6] https://scikit-fuzzy.readthedocs.io/en/latest/

[7] https://github.com/carrot2/carrot2

Cabe mencionar que, para todos los algoritmos, los artículos se ordenan dentro del grupo calculando la similitud de cosenos entre al artículo y el centroide del grupo y se determina el orden de mayor a menor similitud. Adicionalmente se ordenan los grupos asignando un orden de acuerdo con la suma de relevancias de los artículos que pertenecen a ese grupo. Las relevancias de los artículos se asignan de acuerdo con el orden en que los artículos son retornados por Scopus. Es decir, el primer artículo retornado por Scopus tendrá una relevancia igual a la cantidad de artículos obtenidos con la consulta y el valor de relevancia se disminuye en uno para cada artículo en el orden de respuesta.

3.3 ALGORITMOS DE ETIQUETADO

Los algoritmos para generar los títulos realizados son Inverse transform, Semantic frequency, Noun phrases, Graph topic rank y Yake.

Los pasos que se realizaron para el proceso de etiquetado en los algoritmos semantic frequency, Noun phrases, Graph topic rank y Yake se describen en la Tabla 5. La entrada para estos algoritmos es la lista de documentos.

Tabla 5. Resumen del procedimiento realizado en los algoritmos de etiquetado Semantic frequency, Noun phrases, Graph topic rank y Yake

Paso 1	Filtrar los documentos por grupo
Paso 2	Unificar los documentos de un grupo en un solo documento
Paso 3	Eliminar las palabras de la consulta que se encuentren en el documento unificado (limpieza)
Paso 4	Ejecución del algoritmo Semantic frequency o Noun phrases o Graph topic rank o Yake
Paso 5	Ordenar las palabras o frases generadas para el título de un grupo de acuerdo con la frecuencia de las palabras en los documentos pertenecientes al grupo
Paso 6	Realizar un post procesamiento de los títulos generados por el algoritmo con la eliminación de palabras repetidas dentro de los títulos de un grupo y entre grupos

Los pasos que se realizaron para el proceso de etiquetado en el algoritmo Inverse transform se describen en la Tabla 6. La entrada para este algoritmo también es la lista de documentos.

Tabla 6. Resumen del procedimiento realizado en el algoritmo de etiquetado Inverse transform

Paso 1	Eliminar las palabras de la consulta que se encuentren en cada documento (limpieza)
Paso 2	Construir una matriz TF-IDF con la lista de documentos construyendo trigramas
Paso 3	Ejecución del algoritmo Inverse transform
Paso 4	Ordenar las palabras o frases generadas para el título de un grupo de

	acuerdo con la frecuencia de las palabras en los documentos pertenecientes al grupo
Paso 5	Realizar un post procesamiento de los títulos generados por el algoritmo con la eliminación de palabras repetidas dentro de los títulos de un grupo y entre grupos

La tarea de asignar títulos a los grupos tiene cierto grado de dificultad puesto que dichos títulos deben representar el tema principal o transversal de todo un grupo y aunque existen diferentes formas, la mayoría de los métodos de extracción automática de títulos se basa en extraer los términos más significativos de los documentos de un grupo [43]. Es de resaltar que cada algoritmo tiene una forma diferente de calcular la relevancia de los términos.

Por ejemplo, existen métodos de extracción de palabras clave que se basan en la frecuencia de aparición de un término dentro del documento. Estos métodos consisten en extraer todas las palabras clave del documento, luego se cuentan las veces que se repiten dentro del documento y se guarda un recuento de la frecuencia de cada palabra para finalmente ordenar las palabras clave por su frecuencia de forma descendente [44]. De forma similar funciona Term Frequency-Inverse Document Frequency (TF-IDF), la diferencia es que TF-IDF considera tanto la frecuencia de una palabra clave en un documento específico como su frecuencia en el conjunto completo de documentos, y que además calcula la Frecuencia Inversa de Documentos (IDF) dividiendo la cantidad total de documentos entre el número de documentos que incluyen un término específico [44].

Por lo tanto, para el primer algoritmo implementado autodenominado *inverse_transform* se parte de una matriz TF-IDF realizada a partir de los artículos a la cual se le realiza una reducción de dimensionalidad (LSA) usando la clase TruncatedSVD de la librería sklearn, con el fin de tener homogeneidad en los datos utilizados tanto en el agrupamiento como en la generación de títulos o palabras clave de cada grupo. Posteriormente, se realiza el cálculo de los centroides de cada grupo, después por cada uno se revierte la transformación realizada con el LSA usando el método *inverse_transform* de la clase TruncatedSVD. Una vez hecho esto se extraen los índices de los términos con puntaciones más altas tomadas a partir de lo retornado por el método *inverse_transform.* Finalmente, se extraen los términos que conforman la matriz TF-IDF usando el método *get_feature_names_out* y se retornan las 10 primeras palabras con mejor puntación por cada centroide (grupo) teniendo en cuenta los índices obtenidos anteriormente.

Por otro lado, se implementó un segundo algoritmo autodenominado *semantic_frecuency* en el cual se intentó mezclar la frecuencia de los términos junto con la importancia semántica que aportan los mismos. Para determinar la

frecuencia se contó la cantidad de repeticiones de las palabras dentro de la bolsa de palabras. Para hallar la importancia semántica de cada una se hizo uso de la librería Spacy[8], con la cual obtuvimos la norma del vector normal de cada palabra, ésta representa la importancia de una palabra en relación con otras palabras en términos de su significado y contexto, cuanto mayor sea el valor más peso puede tener la palabra. Dicha norma se normalizó teniendo en cuenta el intervalo de los valores de frecuencia encontrados anteriormente, con el objetivo de poder sumar éstas dos puntuaciones, para finalmente retornar las cinco (5) palabras con mayor puntuación por cada grupo.

En el tercer algoritmo implementado autodenominado *noun_phrases* se utilizan las frases sustantivas como título de cada grupo. Según [45] una frase sustantiva "está formada por un sustantivo o pronombre, que se denomina cabeza, y las palabras dependientes que aparecen antes o después de la cabeza. Las palabras dependientes proporcionan información específica sobre la cabeza". Es decir, es una secuencia de palabras consecutivas en un texto que forman un sustantivo junto con los modificadores y adjetivos que lo acompañan y representan entidades o conceptos en el texto. Para hacer esto se unen los resúmenes de los artículos de un grupo en un solo texto y se procesan mediante el uso de nlp de Spacy, posteriormente se hace el análisis de sintaxis obteniendo las frases sustantivas a partir de noun_chunks de la misma librería. Cabe mencionar que este algoritmo no se encuentra dentro de los algoritmos de etiquetado de la aplicación, debido a que realizando pruebas se identificó que los títulos entregados son poco comprensibles y muy extensos.

El cuarto algoritmo implementado es Graph Topic Rank. Esta es una técnica basada en grafos, lo que quiere decir que la estructura se compone de nodos que representan palabras o grupos de palabras conectadas entre sí. Por lo tanto, la relevancia de una palabra en el texto está relacionada con las conexiones con otras palabras. Este algoritmo en particular crea nodos que representan temas, estos nodos son creados utilizando un algoritmo de agrupamiento aplicado a los grupos iniciales. Cada nodo (tema) contiene un peso y el peso entre dos nodos se basa en la fuerza de relación semántica entre ellos. Finalmente, las frases clave se extraen a partir de los temas o nodos mejor clasificados [46].

Para la implementación de este algoritmo primero se construye un grafo de coocurrencia de las palabras del texto usando la librería NetworkX. Una vez creado el grafo se calculan los puntajes de importancia utilizando el algoritmo PageRank. Posteriormente, se ordenan las palabras clave por puntaje de importancia y finalmente se seleccionan las 5 palabras clave principales.

[8] https://spacy.io/

El último algoritmo implementado es Yake, es un enfoque liviano no supervisado para extraer automáticamente palabras clave. Se apoya en características estadísticas derivadas de textos individuales para identificar las palabras clave más relevantes en un documento [47]. Para implementar este algoritmo se usó la librería YAKE[9], la cual tiene una clase KeywordExtractor que se instancia y dispone de una función llamada extract_keywords, a esta se le pasa como argumento un texto que reúne los resúmenes de los documentos de un grupo y retorna las ocho (8) palabras que el algoritmo determinó como más importantes.

Es de resaltar que se escogieron los algoritmos Graph Topic Rank y Yake debido a que [46] realizó un estudio comparativo entre técnicas extractivas basadas en métodos de extracción de palabras clave, tales como C-TF-IDF, Yake, KeyBERT, Graph Position Rank, Graph Topic Rank y Graph Text Rank. Donde hallaron que Graph Topic Rank presenta la puntuación de inteligibilidad más alta de los métodos basados en grafos. Y Yake cuenta con un comportamiento similar debido a que también presenta una alta inteligibilidad, aunque cabe mencionar que compromete un poco la precisión. Además, encontraron que los métodos basados en incrustaciones presentan un comportamiento deficiente en cuanto a precisión e inteligibilidad por lo que descartamos su uso.

Cabe mencionar que antes de usar cualquiera de los algoritmos de etiquetado se realizó una limpieza de los datos al quitar las palabras que se encuentran en la consulta que realiza el usuario, y así evitar ruido que pueda resultar en títulos que contengan las mismas palabras de la consulta y genere un aporte deficiente en la diferenciación de los grupos. Además, una vez cada algoritmo entrega un resultado, se realiza un ordenamiento de las palabras o frases generadas para el título de cada grupo realizando un conteo de la frecuencia de cada palabra dentro de todos los documentos del grupo al que pertenecen y se toman las 5 palabras o frases con mayor frecuencia. Finalmente, se realiza un post procesamiento a través la lematización de las palabras para posteriormente eliminar las palabras que se repiten en todos los grupos y así mejorar la forma en la cual se diferencian los grupos. Asimismo, se eliminaron las palabras repetidas dentro de cada título generado para cada grupo y se eliminaron los títulos que se repetían dentro del mismo grupo, con el fin de generar títulos en la medida de lo posible únicos que representen de una forma entendible y coherente los artículos del grupo.

3.4 MÉTRICAS DE EVALUACIÓN CON CONJUNTOS DE DATOS

Para la realizar la evaluación de los algoritmos de agrupamiento implementados se utilizaron algunas de las métricas clásicas del área de recuperación de información, tales como Precisión (Precision) la cual es la proporción de

[9] http://yake.inesctec.pt/install.html

documentos recuperados que son relevantes, el Recuerdo (Recall) el cual es la proporción de documentos relevantes que fueron recuperados con respecto al total de documentos relevantes existentes, y la Medida F (F-measure) la cual es la media armónica entre la precisión y el recuerdo. Además, se tuvieron en cuenta otras medidas como la cantidad de instancias correctamente agrupadas, el porcentaje de instancias correctamente agrupadas, Exactitud promedio ponderada (Accuracy), Rata ponderada de verdaderos positivos (WTPR), Rata ponderada de falsos positivos (WFPR), Negative predictive value (NPV) y Rata ponderada de falsos descubrimientos (FDR).

Para la evaluación de estas métricas se promedia el resultado de cada una en 31 ejecuciones independientes, con el fin de obtener un valor más real (Teorema del límite central). Los resultados se ven en el capítulo 5.

Para la implementación de la evaluación de las métricas, el primer reto fue determinar la forma para construir una matriz de confusión teniendo en cuenta que los documentos podían estar solapados, sin perder el concepto de la matriz de confusión tradicional. Para ello se creó una matriz de confusión n*m donde n=cantidad de clases predichas +1 y m=cantidad de clases reales +1, estas posiciones adicionales en ambas dimensiones se usan para representar cuando un documento tiene asignadas más clases reales que clases predichas o viceversa.

Para llenar la matriz de confusión primero se debe asignar a cada clase predicha una clase real, para ello se calculó la similitud de todos los centroides de los grupos predichos y todos los centroides de los grupos reales (Centroid Linkage), y se asigna a cada clase predicha la clase real con la cual se tuvo mayor similitud. Una vez realizado lo anterior se procede a llenar la matriz de confusión, iniciando por contar todos los aciertos, para ello como existe solapamiento por cada documento se obtiene la intersección entre las clases predichas y las clases reales, y por cada posición en la coordenada [clase predicha] [clase real] que pertenezcan a la intersección se incrementa el valor de uno (1) en la matriz de confusión. Posteriormente, se cuentan todos los errores incrementando en uno (1) cada posición en la coordenada [clase predicha] [clase real] que no pertenecen a la intersección. Finalmente se cuentan lo que hemos denominado clases reales huérfanas y clases predichas huérfanas, estos casos se dan cuando al hacer la intersección quedaron clases reales fuera de la intersección, pero no quedaron clases predichas fuera de la intersección por lo tanto en este caso es donde se cuenta una clase real huérfana incrementando la matriz de confusión en la coordenada [última fila(cantidad de clases predichas)] [clase real] y el caso contrario cuando al hacer la intersección quedaron clases predichas fuera de la intersección, pero no quedaron clases reales fuera de la intersección por lo tanto, en este caso es donde se cuenta una clase predicha huérfana incrementando la

matriz de confusión en la coordenada [clase predicha] [última columna(cantidad de clases predichas)].

Para ilustrar este concepto en la Tabla 7 se presenta un ejemplo, donde se cuenta con los siguientes datos: Clases Reales= [[1, 2], [3], [4, 5]] y Clases predichas = [[1, 2], [2, 3], [4]].

Tabla 7. matriz de confusión extendida con huérfanos

Clases Predichas \ Clases Reales	**1**	**2**	**3**	**4**	**5**	**H**
1	1	0	0	0	0	0
2	0	1	0	0	0	1
3	0	0	1	0	0	0
4	0	0	0	1	0	0
H	0	0	0	0	1	

Una vez se tiene la matriz de confusión construida se realiza el cálculo de las métricas teniendo en cuenta las **Ecuaciones 2, 3, 4, 5 y 6.**

$$Precision\ x\ clase\ predicha = \frac{relevantes * 100}{recuperados}$$ $$donde:$$ $$relevantes = correctamente\ agrupados\ x\ clase\ predicha,$$ $$recuperados = total\ de\ documentos\ x\ clase\ predicha$$	2
$$Recuerdo\ x\ clase\ predicha = \frac{relevantes * 100}{total\ relevantes}$$ $$donde$$ $$relevantes = correctamente\ agrupados\ x\ clase\ predicha,$$ $$total\ relevantes = total\ documentos\ correctamente\ agrupados\ x\ clase\ real$$	3
$$F\ Score = (2 * precision * recuerdo)/(precision + recuerdo)$$	4
$$Instancias\ correctamente\ agrupadas = \sum relevantes\ x\ cada\ clase\ predicha$$	5
$$\%\ Instancias\ correctamente\ agrupadas = \frac{Instancias\ correctamente\ agrupadas * 100}{total\ de\ instancias}$$ $$donde\ instacias\ son\ los\ documentos$$	6

Por otro lado, para calcular la Rata Ponderada de Verdaderos Positivos (WTPR, por sus siglas en inglés) se calcula la cantidad total de verdaderos positivos y se divide entre el total de documentos. Para calcular el total de verdaderos positivos es necesario hallar la cantidad de verdaderos positivos para la clase, es decir, hallar la cantidad de instancias correctamente agrupadas para esa clase, y dividirlo entre la cantidad de documentos real de esa clase. El valor de esta división se multiplica por la cantidad de documentos de esa clase real y se realiza

la sumatoria de este valor por cada clase real. Finalmente, el valor de WTPR corresponde a la división de la sumatoria mencionada anteriormente (total de verdadero positivos) por la cantidad de documentos total, lo que proporciona una medida de la tasa de verdaderos positivos ajustada por la distribución de clases en el conjunto de datos.

El cálculo de la Rata Ponderada de Falsos Positivos (WFPR) se realiza dividiendo el total de verdaderos negativos sobre el total de documentos. Para obtener el total de verdaderos negativos se multiplica la tasa de falsos positivos por el total de documentos en la clase real, el cálculo de la tasa de falsos positivos es la división de los falsos positivos entre la resta del total de documentos y los documentos en la clase real, y los falsos positivos son las instancias que se asignaron a una clase que en realidad no era.

Para el cálculo de la exactitud promedio ponderada (Accuracy) primero se determina la tasa de exactitud, para esto, por cada clase real se realiza la suma de los verdaderos positivos y los verdaderos negativos y se divide este valor entre el total de documentos. Posteriormente, se realiza la sumatoria de la tasa de exactitud por cada clase real multiplicando por el total de documentos para esa clase y está sumatoria se divide entre la cantidad total de documentos.

El cálculo de Negative Predictive Value (NPV), se realizó con la sumatoria del producto de la tasa NPV por el total de documentos por cada clase real todo esto dividido por el total de documentos, donde la tasa NPV es los verdaderos negativos dividido entre la suma de verdaderos negativos más los falsos negativos.

Para el cálculo de la Rata ponderada de Falsos Descubrimientos (FDR), se realizó con la sumatoria del producto de la tasa FDR por el total de documentos por cada clase real todo esto dividido por el total de documentos, donde la tasa FDR corresponde a los falsos positivos dividido entre la suma de falsos positivos y verdaderos negativos.

3.5 CONJUNTOS DE DATOS PARA LA EVALUACIÓN Y COMPARACIÓN

En cuanto a la selección de los conjuntos de datos se tuvo en cuenta que estos contuvieran artículos científicos y en especial que al menos contaran con el atributo del resumen y la categoría (clase real) a la cual pertenece cada artículo. Por lo tanto, para realizar la evaluación se seleccionaron 4 conjuntos de datos, dos de ellos se obtuvieron a partir de un artículo del estado del arte [12] y los otros dos se hallaron a través de una búsqueda en la web.

Los conjuntos de datos utilizados son AAAI 2014 Accepted Papers[10] y AAAI 2013 Accepted Papers[11] ambos provenientes del repositorio de Machine Learning de la Universidad de California en Irvine. Los otros dos conjuntos de datos están disponibles en Kaggle, el primero se llama Topic Modeling for Research Articles 2.0[12], y el último se llama arXiv[13].

[10] https://archive.ics.uci.edu/ml/datasets/AAAI+2014+Accepted+Papers

[11] https://archive.ics.uci.edu/ml/datasets/AAAI+2013+Accepted+Papers

[12] https://www.kaggle.com/datasets/anmolkumar/topic-modeling-for-research-articles-20

[13] https://www.kaggle.com/datasets/Cornell-University/arxiv

CAPÍTULO 4

4 IMPLEMENTACIÓN DE LA REST API Y LA APLICACIÓN WEB

4.1 DIAGRAMAS

El primer paso para el desarrollo de la REST API fue la creación de los diferentes diagramas comenzando por el de la base de datos y los diagramas del modelo C4 de documentación para la arquitectura de software, tales como los diagramas de contexto, contenedores y componentes. Esto se realizó con el fin de tener una visión general de alto nivel de la estructura y las relaciones entre los componentes del sistema. Los diagramas de base de datos y modelo C4 se encuentran disponibles en el **Anexo 5**.

Por otro lado, para la estructuración del frontend se realizaron los mockups para plasmar la idea inicial del diseño de la página web.

4.1.1 DIAGRAMA DE BASE DE DATOS

El diagrama de base datos abstrae la información necesaria para realizar el proceso de agrupamiento, almacenar los resultados y mostrarle al usuario la información más relevante de los artículos. En la Figura 1, se muestra el modelo conceptual realizado donde se puede observar que todo comienza a partir de un usuario. Un usuario puede **crear** cero a muchas consultas, sin embargo, una consulta debe ser realizada por un solo usuario. Una consulta **contiene** desde cero hasta muchos artículos, pero un artículo debe pertenecer a una o muchas consultas. Debido a esta relación de muchos a muchos entre estas dos entidades en el diagrama físico se crea una tabla intermedia que no solo guarda la relación entre las dos entidades, sino que también almacena la relevancia de un artículo hacía la consulta. De igual forma, se persiste la información de los artículos donde un artículo puede ser escrito por uno o muchos autores y un autor **escribe** uno o muchos artículos. Entre estas dos entidades también se contará con una tabla intermedia.

Puesto que se necesita tener disponibilidad de los resultados, una consulta **tiene** cero o muchos grupos y un grupo solo pertenece a una consulta, además, cada grupo tiene uno o muchos artículos y cada artículo **pertenece** a uno o varios grupos. Por lo tanto, estas entidades se relacionarán mediante una tabla intermedia que además guarda el orden de los artículos en el grupo y la evaluación de la pertenencia de ese artículo en el grupo.

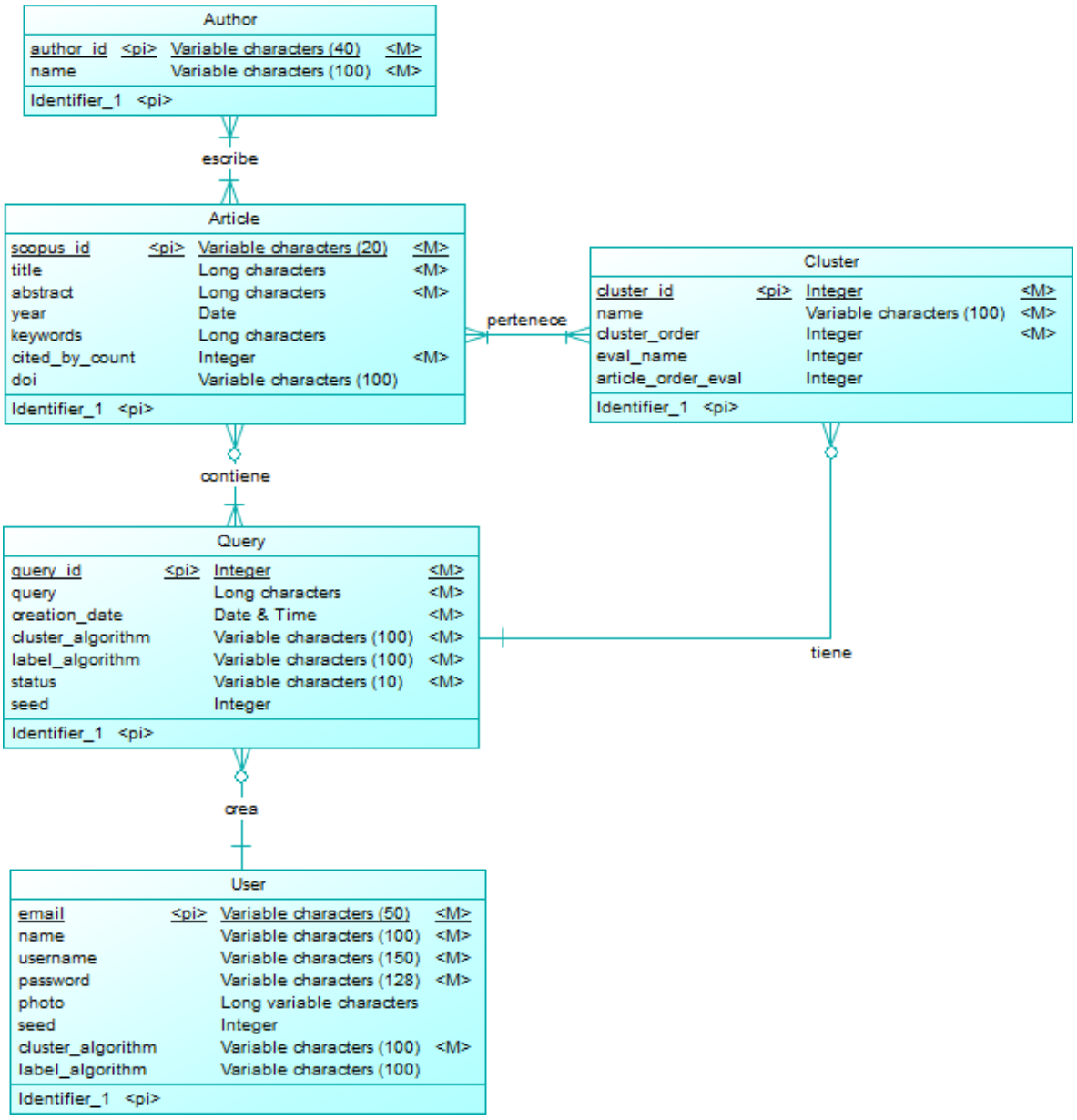


Figura 1. Modelo Conceptual de Base De Datos

4.1.2 MODELO C4

El modelo C4 es un método de diagramación de arquitectura que se constituye de un conjunto de abstracciones y diagramas jerárquicos [48]. Para este proyecto se desarrollaron los siguientes diagramas: contexto, contenedores y componentes.

4.1.2.1 DIAGRAMA DE CONTEXTO

Este diagrama muestra una abstracción del sistema y permite visualizar como interactúa con otros sistemas [49].

El diagrama de contexto (ver Figura 2) indica que la aplicación realizará consultas a 3 sistemas externos: La API de Scopus para realizar las consultas de documentos de investigación pertenecientes a Elsevier con las cadenas de búsqueda ingresadas por el usuario, el sistema de agrupamiento de Carrot2 para hacer uso de los algoritmos de STC y Lingo y el sistema de autenticación de Google Authentication para poder realizar un inicio de sesión restringido solo para usuarios pertenecientes a la Universidad del Cauca. Además, se precisa que el usuario que utilizará el sistema realizado será un investigador o interesado en material académico.

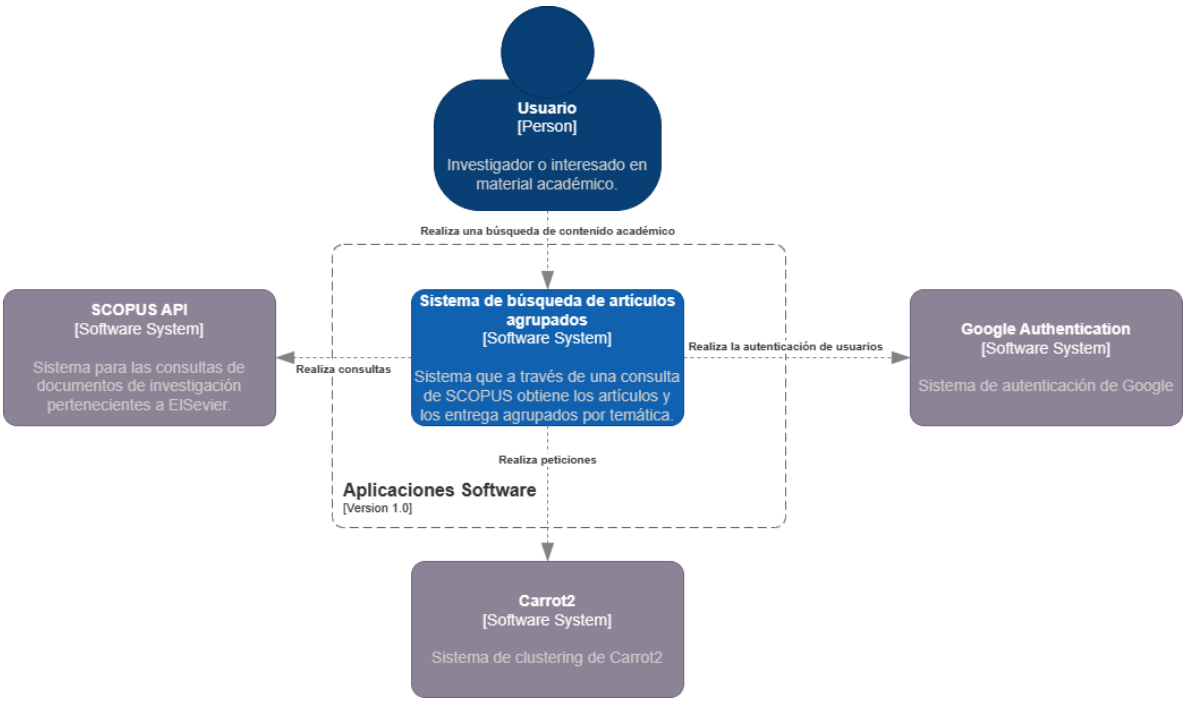


Figura 2. Diagrama de Contexto

4.1.2.2 DIAGRAMA DE CONTENEDORES

En este diagrama se amplía el sistema software para detallar los contenedores asociados a este, es decir, aplicaciones, almacenamiento de datos, microservicios, entre otros [49].

En la Figura 3 se puede observar como el sistema realizado se compone de 3 elementos. El elemento central es la API de aplicación la cual expondrá los servicios para realizar la búsqueda de documentos, el proceso de agrupamiento, la personalización de la configuración del proceso de agrupamiento, la obtención de resultados y evaluación de estos. Además, se encarga de realizar las llamadas a la API de Scopus para obtener los artículos asociados a la consulta y al sistema de Carrot2 para realizar el agrupamiento con STC y Lingo. La API de la aplicación persiste la información en una base de datos representada como otro elemento del diagrama la cual usa como tecnología a MySql. Finalmente, el elemento que

expone los servicios del API mediante una interfaz gráfica hacia el usuario es la aplicación web desarrollada con el framework de Angular. Esta aplicación web es la encargada de hacer las llamadas al sistema de autenticación de Google para verificar al usuario que desea ingresar.

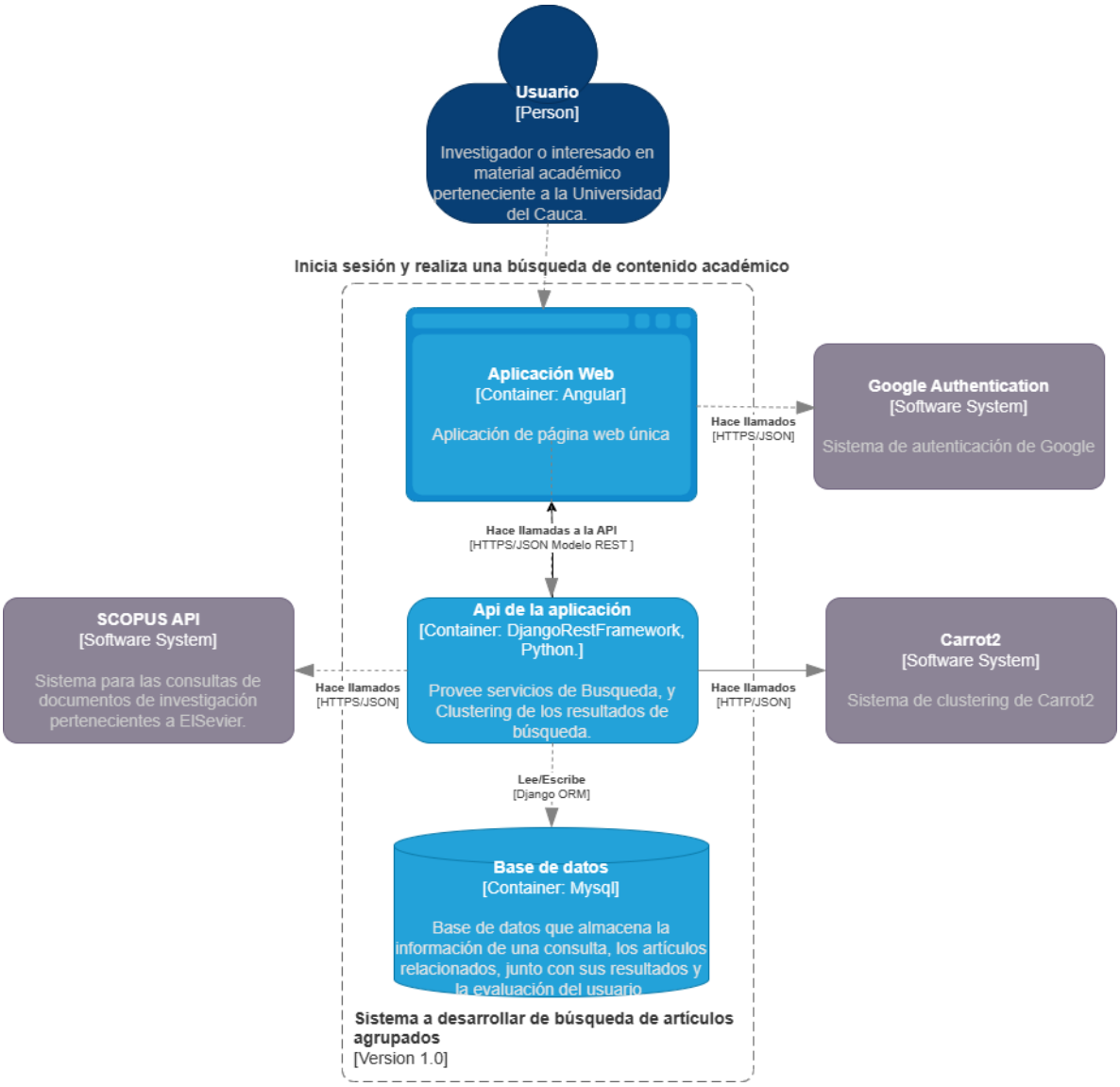


Figura 3. Diagrama de contenedores

4.1.2.3 Diagrama de Componentes

En este diagrama se amplía el contenedor de la API de la aplicación con el fin de detallar la estructura interna de los principales componentes estructurales y sus interacciones [48]. En la Figura 4 el diagrama de componentes indica que la API de la aplicación se compone de un controlador de consultas que se encarga de exponer los servicios para realizar una consulta, realizar el agrupamiento después de realizar la consulta y obtener los resultados de agrupamiento.

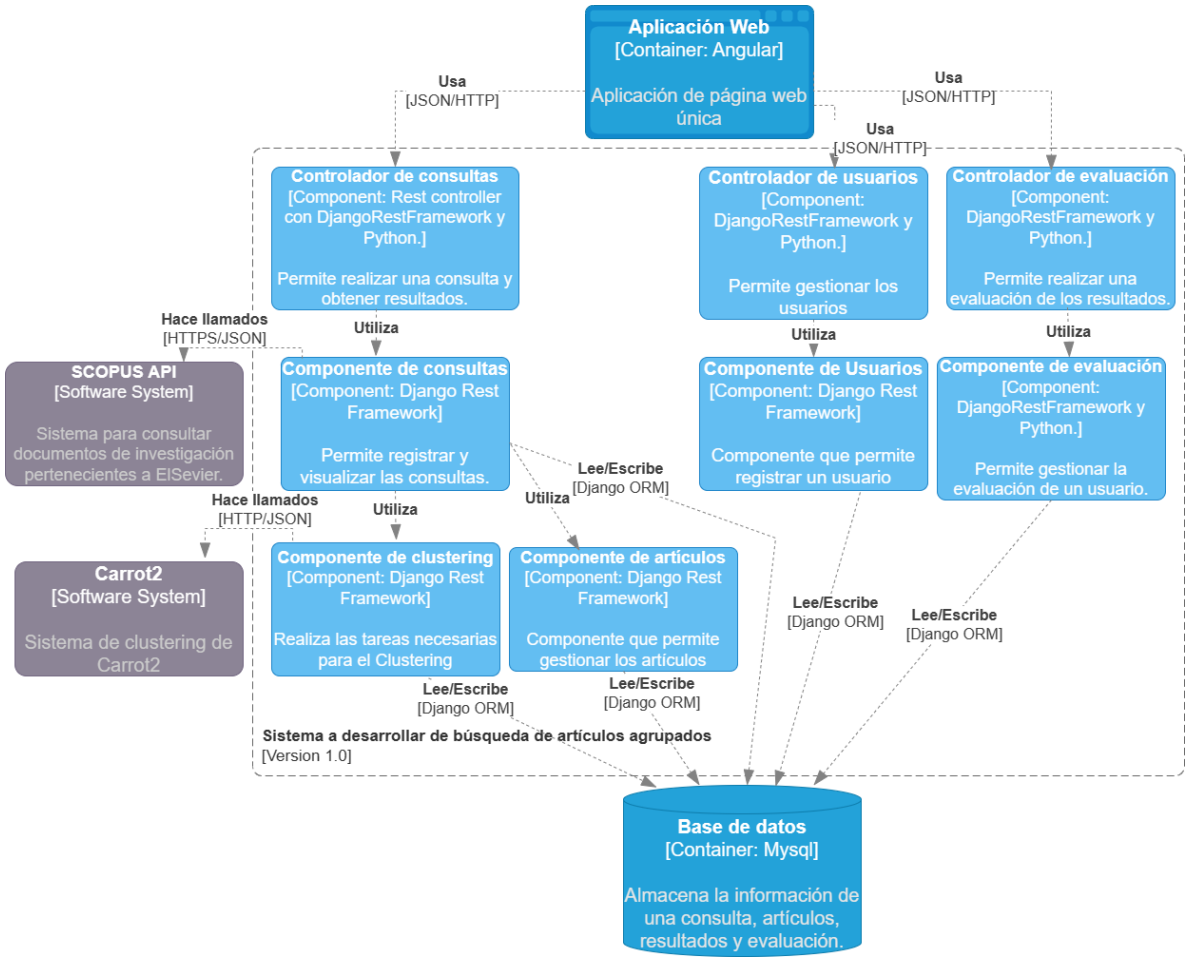


Figura 4. Diagrama de componentes

Este controlador utiliza un componente de consultas que lee y escribe en la base de datos para registrar y visualizar las consultas realizadas, además, realiza las llamadas a la API de Scopus con la consulta del usuario para obtener los documentos resultantes. A su vez este componente de consultas hace uso del componente de artículos el cual tiene la función de gestionar los artículos obtenidos desde Scopus, es decir, guardar y obtener artículos junto con los autores asociados, y hace uso de un componente de agrupamiento el cual se encarga de realizar los procesos asociados al agrupamiento (preprocesamiento, agrupamiento, etiquetado) y también hace las llamadas al sistema de Carrot2 cuando se realiza el agrupamiento con STC o Lingo. Por otro lado, también se cuenta con un controlador de usuarios el cual expone los servicios relacionados con la gestión de usuarios; este utiliza un componente de usuarios encargado de registrar, editar u obtener un usuario de la base de datos. Finalmente, se cuenta con un controlador de evaluación el cual se encarga de exponer los servicios para realizar la evaluación de los resultados obtenidos en el agrupamiento. Este controlador utiliza un componente de evaluación que tiene la función de persistir los resultados de evaluación en la base de datos.

4.2 DISEÑO DE MOCKUPS

Para el diseño de los mockups se usó la aplicación Marvel[14] con el fin de lograr una representación simple y entendible del diseño de la aplicación web. Es de resaltar que los mockups son una idea inicial de la página y en la implementación se pueden tomar algunos de los aspectos plasmados dentro de ellos. Los mockups se encuentran disponibles en el **Anexo 6**.

4.3 TECNOLOGÍAS

Se usó el framework Django para el desarrollo de la aplicación REST API, ya que, los códigos fuentes de los algoritmos de agrupamiento se encuentran disponibles y se pueden modificar, además, el lenguaje Python tiene una curva de aprendizaje rápida. Por otro lado, la base de datos se implementó en MySql debido a que es un sistema de gestión de bases de datos (DBMS) de código abierto ampliamente utilizado en la industria que ofrece un muy buen rendimiento y es capaz de manejar grandes cantidades de datos.

En cuanto a las tecnologías de terceros se utilizaron Firebase Authentication para verificar que los usuarios pertenezcan a la universidad debido a restricciones de Scopus para el uso de su API. Por otro lado, se usó Carrot2 como servicio de agrupamiento para usar los algoritmos de STC y Lingo. También, se utilizó Celery junto con Redis para ejecutar tareas asíncronas dentro de la aplicación.

Finalmente, para el despliegue del frontend se usó de Firebase Hosting y para el despliegue del backend se usó Docker para organizar la aplicación en contenedores que son alojados por los servidores de la División de TICs de la Universidad del Cauca, sin embargo, se presentaron dos retos: el primero consistió en la imposibilidad de comunicación entre la página web y los servicios del backend debido a que la página web tiene el protocolo https y el servicio maneja el protocolo http, lo que generó que al tratar de realizar una petición desde la página web esta fuese bloqueada por el navegador debido a políticas de seguridad. El segundo reto fue que los servicios del backend solo eran accesibles dentro de la red de la Universidad del Cauca, lo que impedía que se pudiera acceder públicamente al momento de realizar las pruebas y la evaluación. Por lo tanto, ante estos inconvenientes se decidió realizar el despliegue con el backend haciendo uso de Google Cloud Run, Google Cloud Sql y Redis lo que permitió tener un acceso público de los servicios por 91 días con un presupuesto de créditos brindados por Google de 400 dólares.

[14] https://marvelapp.com/

4.4 IMPLEMENTACIÓN DE LA API REST

Para la implementación de ésta REST API, se usó Django y Django Rest Framework. El proyecto se dividió en 5 módulos establecidos de la siguiente manera: un módulo para la gestión de los artículos, un módulo para el agrupamiento, un módulo para la gestión de la evaluación, un módulo de gestión de consultas y finalmente un módulo para la gestión de usuarios. Es de resaltar que se usó como motor de base de datos MySql y para la comunicación y gestión de esta base de datos se usó el ORM que proporciona Django.

Para hacer uso de los servicios proporcionados por los módulos fue necesario agregar dentro del encabezado de la petición una llave con el nombre *Authorization* la cual debe contener como valor *Token <<token del usuario>>.* El token del usuario se obtiene haciendo uso del servicio de Login o inicio de sesión.

4.4.1 MÓDULO DE GESTIÓN DE ARTÍCULOS

Este módulo proporciona las funcionalidades para la persistencia de artículos junto con sus autores y poder recuperar la información de un artículo mediante su scopus_id. Estas funcionalidades son de uso interno de la aplicación, ya que son utilizadas por el módulo de consultas y por lo tanto no se exponen dentro de los servicios REST.

4.4.2 MÓDULO DE AGRUPAMIENTO

Este módulo proporciona las funcionalidades para realizar el agrupamiento de los resultados de una consulta, es decir, aquí se encuentran las implementaciones descritas en el capítulo 3. los 5 algoritmos de agrupamiento, los 5 algoritmos de etiquetado y la persistencia de los resultados de agrupamiento.

Es de resaltar que la estructuración de este módulo facilita la adición futura de más algoritmos tanto de agrupamiento como de etiquetado.

4.4.3 MÓDULO DE GESTIÓN DE LA EVALUACIÓN

Para la evaluación existen 3 tipos de respuestas, -1 para una respuesta negativa, 0 para una respuesta insegura, es decir que no sabe y 1 para una respuesta afirmativa o positiva. Este módulo proporciona tres servicios.

- Servicio para realizar la evaluación del nombre del grupo
  Verbo: PUT
  Url: http://127.0.0.1:8000/api/evaluation/cluster?cluster_id=<<id del cluster>>
  Body:

```
{
    "eval":1
}
```

Respuesta: `{"rta":True}` Si el proceso fue exitoso. `{"rta":False}` de lo contrario.

- Servicio para evaluar el orden de los artículos en el grupo
  Verbo: PUT
  Url: http://127.0.0.1:8000/api/evaluation/cluster/articleorder?cluster_id=<<id del cluster>>
  Body:

```
{
    "eval":1
}
```

  Respuesta: `{"rta":True}` Si el proceso fue exitoso. `{"rta":False}` de lo contrario.

- Servicio para evaluar la pertenencia de un artículo a un grupo
  Verbo: PUT
  Url: http://127.0.0.1:8000/api/evaluation/article?cluster_id=<< id del cluster >>&scopus_id=<<scopus id>>
  Body:

```
{
    "eval":1
}
```

  Respuesta: `{"rta":True}` Si el proceso fue exitoso. `{"rta":False}` de lo contrario.

4.4.4 MÓDULO DE GESTIÓN DE CONSULTAS

Este módulo recibe la consulta realizada por el usuario y cada consulta es realizada a Scopus mediante un conjunto de servicios prestados por Elsevier Developer Portal[15]. Para hacer uso de estos servicios con algunos privilegios, se solicitó a Elsevier una API Key y un token institucional, los cuales se usan como autenticación en los servicios de Scopus Search API y Abstract Retrieval API. Es de resaltar que existe un límite de 50.000 peticiones a este último servicio por cada 7 días, es por esto por lo que dentro de la aplicación se limita a que se procesen máximo 1000 artículos por consulta para evitar consumir la cuota mensual establecida. Con la primera API se realiza la consulta y se obtienen los resultados en orden de relevancia según criterios de Scopus y con la segunda API se consulta la información de cada artículo. Se usó Celery para desligar el proceso de agrupamiento del hilo principal de ejecución. Para ello una vez se usa el servicio de realizar la consulta este llama al trabajador de Celery y dentro de su

[15] https://dev.elsevier.com/

tarea se realiza el proceso de agrupamiento y persistencia de los resultados haciendo uso de las funcionalidades proporcionadas por el módulo de agrupamiento descrito anteriormente. Además, dentro de este módulo se realizó la implementación para enviar un correo de notificación al usuario una vez se persisten los resultados del agrupamiento, con el fin de informarle que la consulta terminó de procesarse y ya se tienen los resultados. En la Figura 5 se muestra un ejemplo del correo que se le presenta al usuario.

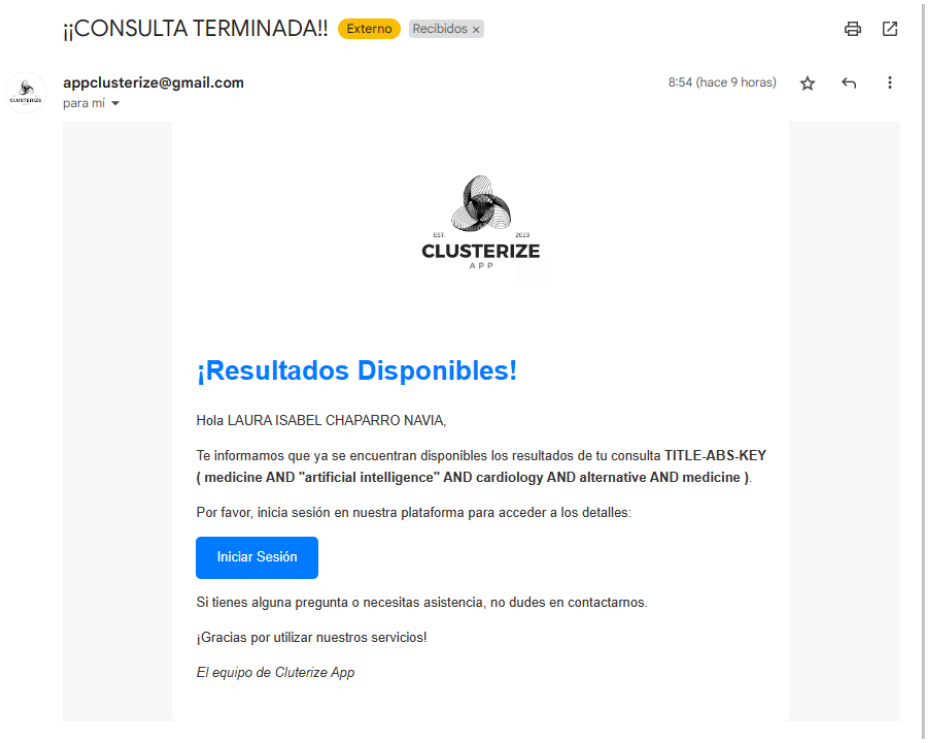


Figura 5. Correo de notificación

El módulo implementado proporciona seis servicios:

- Servicio para realizar una consulta
  Verbo: POST
  Url: http://127.0.0.1:8000/api/query

  Body:

```
{
   "query":"TITLE-ABS-
   KEY ( medicine AND \"artificial intelligence\" AND cardiology AND a
   lternative AND medicine ) ",
      "label_algorithm":"noun_phrases",
      "email":"juancamm@unicauca.edu.co",
      "cluster_algorithm": "fuzzy",
      "seed": 1511
}
```

Respuesta:


```
{
        "query_Id": 15,
    "query":"TITLE-ABS-
    KEY ( medicine AND \"artificial intelligence\" AND cardiology AND a
    lternative AND medicine ) ",
        "label_algorithm":"noun_phrases",
        "email":"juancamm@unicauca.edu.co",
        "cluster_algorithm": "fuzzy",
        "creationDate": "2023-08-28T13:54:15.759705Z",
        "status": "IN_PROCESS",
        "seed": 1511
}
```


- Servicio para consultar la cantidad de artículos que tiene una consulta
  Verbo: GET
  Url: http://127.0.0.1:8000/api/query/search
  Body:


```
{
    "query":"TITLE-ABS-
    KEY ( medicine AND \"artificial intelligence\" AND cardiology AND a
    lternative AND medicine ) "
}
```


  Respuesta:


```
{
    "totalResults": 1
}
```


- Servicio para consultar las consultas realizadas por un usuario
  Verbo: GET
  Url:
  http://127.0.0.1:8000/api/query/history?email=<<correo>>&status=<<COMPLETE o IN_PROCESS>>
  Body: no es necesario
  Respuesta:


```
[
        {
            "query_Id": 1,
        "query":"TITLE-ABS-
        KEY ( medicine AND \"artificial intelligence\" AND cardiolog
        y AND alternative AND medicine ) ",
            "label_algorithm":"noun_phrases",
            "email":"juancamm@unicauca.edu.co",
            "cluster_algorithm": "fuzzy",
```

```
                "creationDate": "2023-08-28T13:54:15.759705Z",
                "status": "COMPLETE",
                "seed": 1511
        }
    ]
```


- Servicio para obtener los resultados de una consulta
  Verbo: GET
  Url: http://127.0.0.1:8000/api/query/results/<< id de la consulta>>
  Body: no es necesario
  Respuesta:


```
[
  {
    "cluster_id": 5,
    "articles": [
      {
        "scopus_id": "85128652475",
        "authors": [
          {
            "author_id": "55791521100",
            "name": "Magdy A."
          },
          {
            "author_id": "57606968500",
            "name": "Mahmoud O."
          }
        ],
        "title": "Robo-
Nurse Healthcare Complete System Using Artificial Intelligence",
        "abstract": ", .Significant with COVID-
19 pandemic breakout , and the high risk of acquiring this infection that
 is facing the Healthcare Workers ( HCWs ) , a safe alternative was neede
d . As a result , robotics , artificial intelligence ( AI ) and internet
of things ( IoT ) usage rose significantly to assist HCWs in their missio
ns . ",
        "year": "2022-01-01",
        "keywords": "Artificial intelligence,Face recognition,Healthcare,
Humanoid robot,Infection control,Internet of Things,Nursing robot",
        "citedByCount": 0,
        "doi": "10.1007/978-3-031-03918-8_17",
        "publisher": "Springer Science and Business Media Deutschland Gmb
H",
        "eval": null
      }
    ],
    "name": "['hcws', 'humanoid', 'robot', 'infection', 'vital']",
```

```
    "clusterOrder": 1,
    "eval_name": null,
    "article_order_eval": null,
    "query_Id": 3
  }
]
```


- Servicio para eliminar una consulta
  Verbo: DELETE
  Url: http://127.0.0.1:8000/api/query/delete/<<id de la consulta>>
  Body: no es necesario
  Respuesta: Status: 200 OK Si la eliminación fue exitosa. Código de error (500 o 404 dependiendo de la situación) en caso contrario.

- Servicio para eliminar una lista de consultas
  Verbo: DELETE
  Url: http://127.0.0.1:8000/api/query/delete
  Body: no es necesario

  Respuesta: `[1,2]`, es decir, la lista de identificadores de las consultas eliminadas. Código de error (500 o 404 dependiendo de la situación) en caso contrario.

4.4.5 Módulo de Gestión de Usuarios

Este módulo proporciona cuatro servicios:

- Servicio para guardar un usuario
  Verbo: POST
  Url: http://127.0.0.1:8000/api/user
  Body:


```
{
    "email":"juan@unicauca.edu.co",
    "name":"Juan",
    "username":"juan@unicauca.edu.co",
    "password":"juan@unicauca.edu.co",
    "photo":"https://urlphoto",
    "seed":1511,
    "cluster_algorithm":"spectral",
    "label_algorithm":"yake"
}
```


  Respuesta:


```
{
    "email":"juan@unicauca.edu.co",
```

```
    "name":"Juan",
    "username":"juan@unicauca.edu.co",
    "password":"juan@unicauca.edu.co",
    "photo":"https://urlphoto",
    "seed":1511,
    "cluster_algorithm":"spectral",
    "label_algorithm":"yake"
}
```


- Servicio para obtener un usuario
  Verbo: GET
  Url: http://localhost:8001/api/user/<<email>>
  Body: no es necesario
  Respuesta:


```
{
    "email":"juan@unicauca.edu.co",
    "name":"Juan",
    "username":"juan@unicauca.edu.co",
    "password":"juan@unicauca.edu.co",
    "photo":"https://urlphoto",
    "seed":1511,
    "cluster_algorithm":"spectral",
    "label_algorithm":"yake"
}
```


- Servicio para editar un usuario
  Verbo: PUT
  Url: http://localhost:8001/api/user/update/<<email>>
  Body:


```
{
    "email":"juan@unicauca.edu.co",
    "name":"Juan",
    "username":"juan@unicauca.edu.co",
    "password":"juan@unicauca.edu.co",
    "photo":"https://urlphoto",
    "seed":1511,
    "cluster_algorithm":"STC",
    "label_algorithm":""
}
```


  Respuesta:


```
{
    "email":"juan@unicauca.edu.co",
```

```
    "name":"Juan",
    "username":"juan@unicauca.edu.co",
    "password":"juan@unicauca.edu.co",
    "photo":"https://urlphoto",
    "seed":1511,
    "cluster_algorithm":"STC",
    "label_algorithm":""
}
```


- Servicio de autenticación
  Verbo: POST
  Url: http://127.0.0.1:8000/api/token
  Body:


```
{
    "username":"juan@unicauca.edu.co",
    "password":"juan@unicauca.edu.co"
}
```


  Respuesta:


```
{
    "token": "b5c3bb6cf97343c4e630cefaa93f478849ab5239836ea90dd606941fe2e07be0”,
    "expiry": "2023-08-29 02:38:40.110465+00:00"
}
```


4.5 IMPLEMENTACIÓN DEL FRONTEND

Para la implementación de la aplicación web se hizo uso de Angular, debido a que es un framework basado en componentes, lo que permite realizar aplicaciones web escalables. Además, se seleccionó debido a que ya se cuenta con experiencia previa para el desarrollo.

El proyecto está estructurado en seis componentes: El componente de Navbar que se encarga de renderizar un menú de navegación para ingresar a las vistas de inicio, resultados, perfil y cerrar sesión. El componente Loader que se encarga de renderizar una animación que indica que se está cargando o procesando información. En la Figura 7 se aprecia el nombre y logo de la aplicación el cual fue diseñado haciendo uso de la aplicación Canva. A continuación, se explican los demás componentes de la aplicación: Login, Home, Results y Profile.

4.5.1 COMPONENTE LOGIN

El componente de Login se encarga de renderizar el inicio de sesión con Google, y por lo tanto se encarga de crear los usuarios sino están creados y generar el token de autenticación, además de recuperar la información del usuario (ver Figura 6).

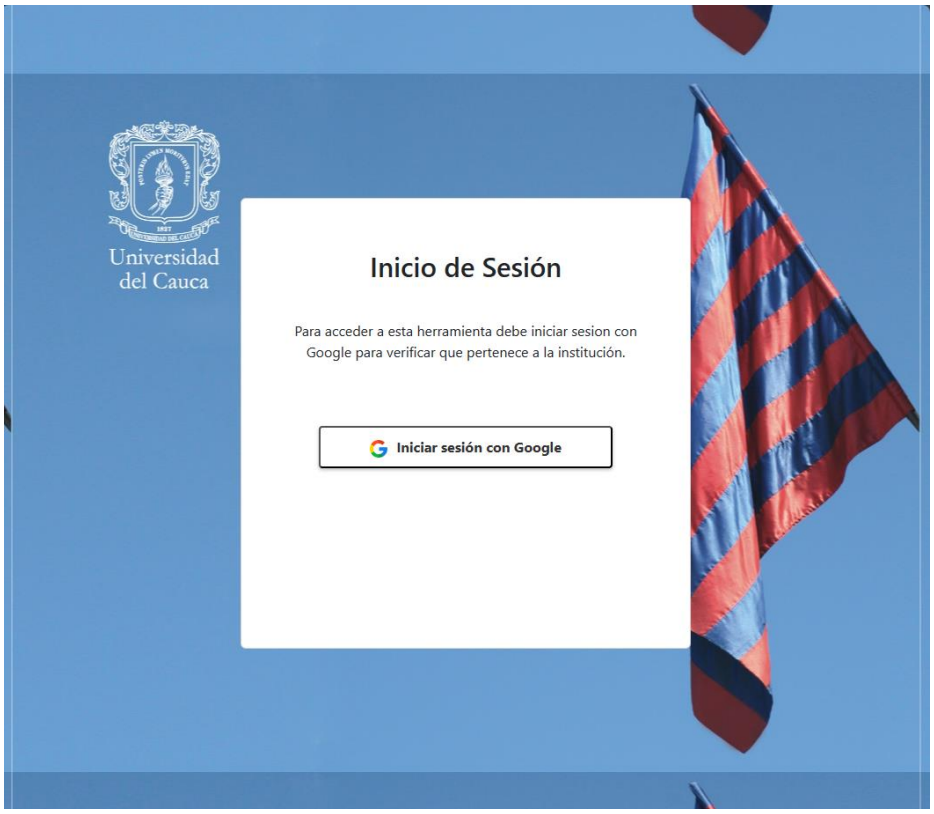


Figura 6. Vista de Inicio de sesión (Componente Login)

4.5.2 COMPONENTE HOME

Éste es uno de los componentes principales (ver Figura 7) puesto que permite realizar las consultas en inglés, permitiendo parametrizar el criterio de consulta de la misma forma que en Scopus como se puede aprecia en la Figura 8 y el rango de fechas.

Al momento en el que el usuario presiona el botón de buscar, la página indica la cantidad de artículos que retorna la consulta para validar si la desea realizar o si la quiere modificar para acotarla o expandirla.

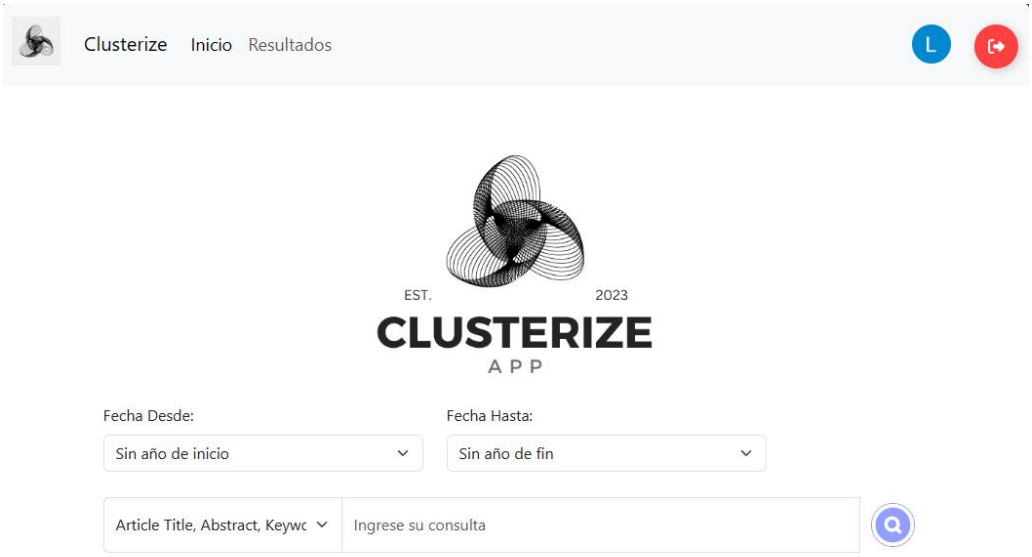


Figura 7. Vista de la Página de Principal (Componente Home)

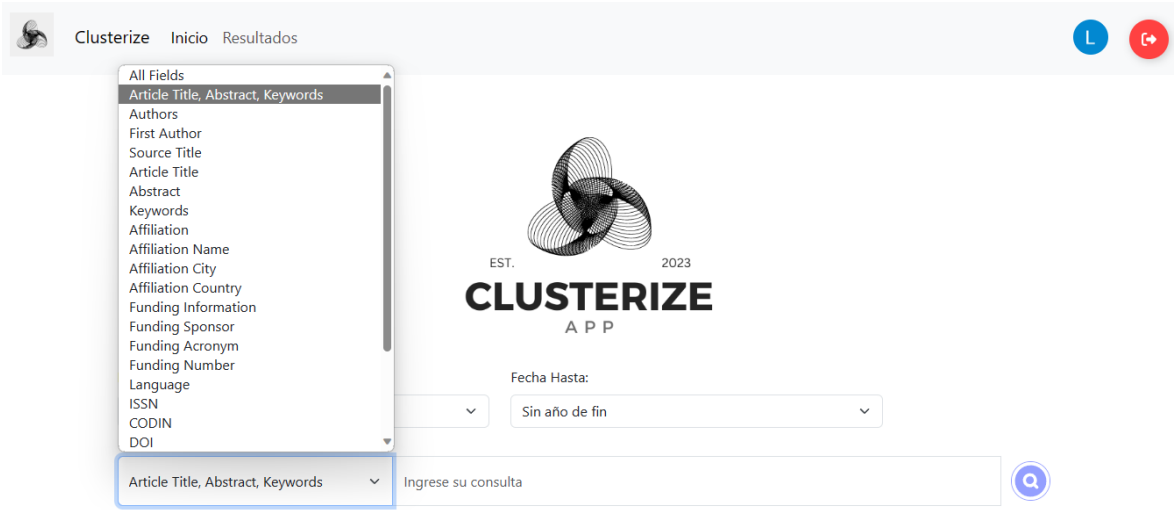


Figura 8. Vista de la Página Principal con Todos los Criterios de Búsqueda

4.5.3 COMPONENTE RESULTS

Este componte lista todas las consultas realizadas por el usuario, permitiendo ver su estado, es decir si ya se completaron o aún se encuentran en proceso y el algoritmo de agrupamiento y etiquetado usados para procesar los resultados de esa consulta. Al momento de seleccionar una consulta se listan los resultados de esta, mostrando los diferentes grupos con los títulos y el número de artículos por cada grupo. Cuando se carga el componente se listan los resultados de la

consulta más recientemente completada. El usuario puede eliminar una o varias consultas si lo desea (ver Figura 9).

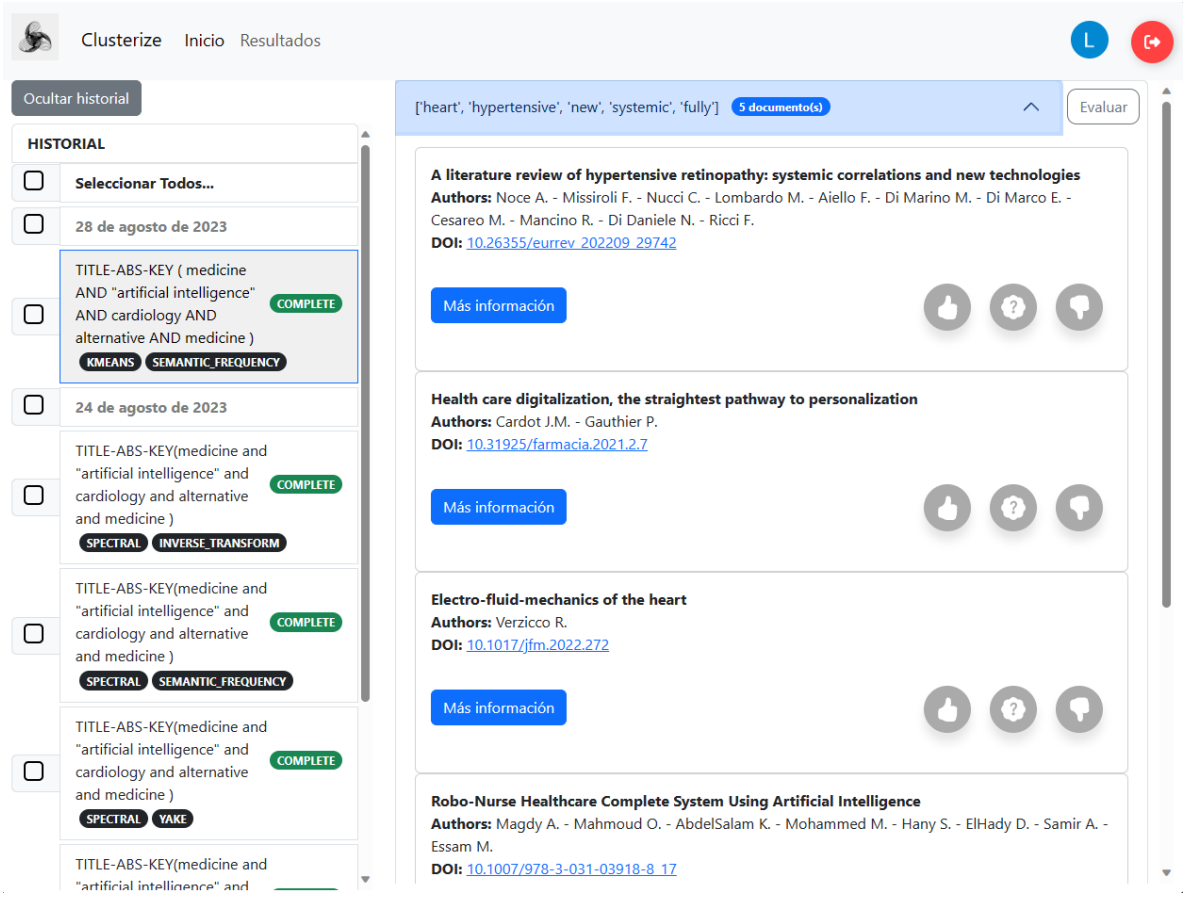


Figura 9. Vista de Resultados (Componente Results)

Es importante destacar que se encuentra un botón de evaluación en cada grupo y en cada artículo para realizar su evaluación correspondiente. Los elementos que permiten realizar la evaluación, es decir la Figura 10 y Figura 11, se pueden ocular mediante el cambio del valor del atributo *evaluation* que se encuentra dentro de una base de datos en tiempo real el proyecto de Firebase. Si el atributo se encuentra en True quiere decir que se enseñan los elementos de evaluación en la vista del componente de resultados, en caso contrario estos elementos se ocultarán. Esta funcionalidad se realizó con el fin de dejar una versión de la página libre del entorno de evaluación, es decir, dejar el diseño que se vería en un entorno de producción.

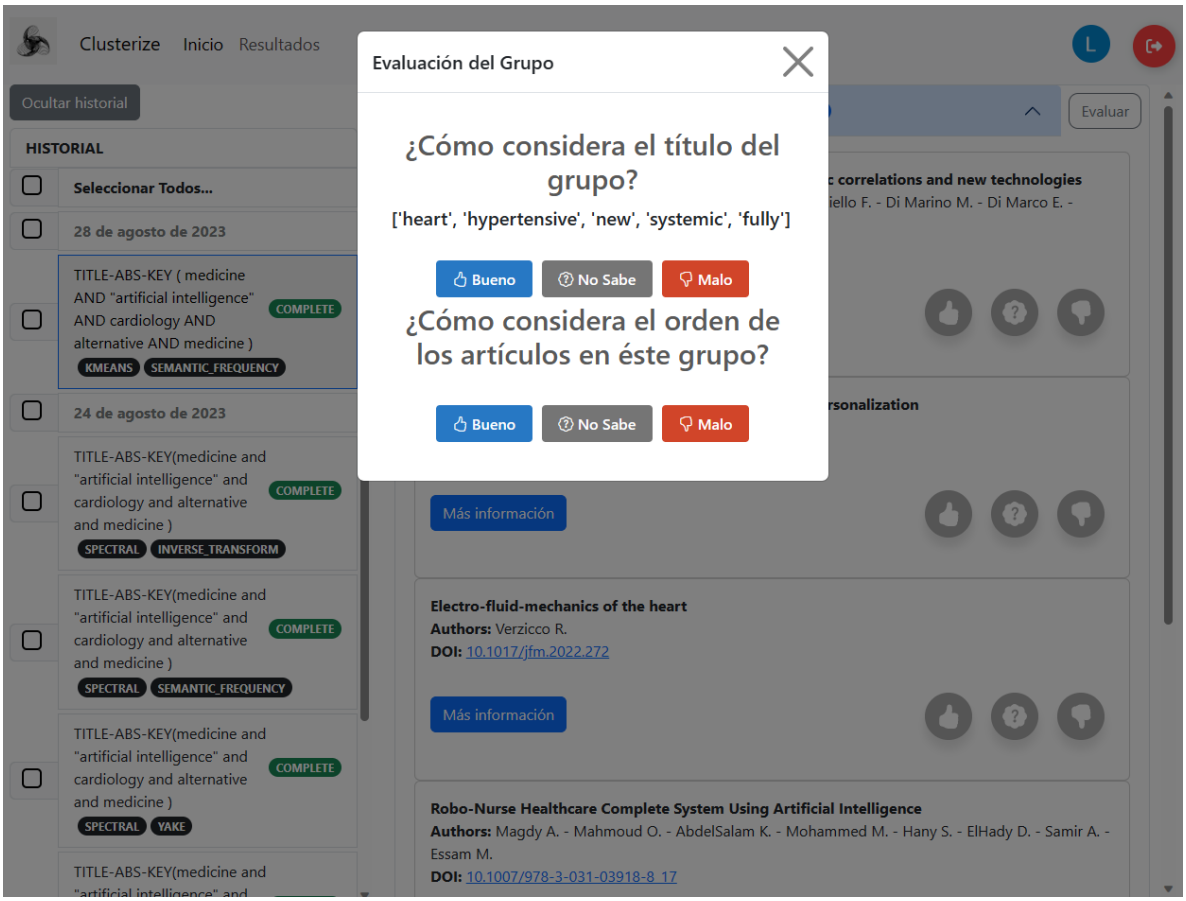


Figura 10. Vista de Evaluación de un Grupo

4.5.4 COMPONENTE PROFILE

Este componente permite enseñar la información del usuario y sus configuraciones personalizadas en cuanto al agrupamiento, el algoritmo para el etiquetado y la semilla a utilizar para realizar el proceso de agrupamiento (ver Figura 12). El algoritmo de agrupamiento por defecto es Spectral y el algoritmo de etiquetado por defecto es Yake debido a que en el resultado de las evaluaciones se encontró que Spectral entrega mejores resultados de agrupamiento y Yake entrega títulos más fáciles de comprender.

Es de resaltar que el usuario puede cambiar libremente los parámetros mencionados anteriormente, teniendo en cuenta que puede escoger entre 5 algoritmos de agrupamiento diferentes y 4 algoritmos de etiquetado. La única restricción es que si escoge STC o Lingo como algoritmos de agrupamiento no puede seleccionar un algoritmo de etiquetado debido a que estos ya realizan este proceso.

Por otro lado, la semilla que se encuentra por defecto es el primer número primo después de 1500, es decir, 1511. Este valor se puede modificar con el fin de

obtener resultados diferentes para una misma consulta usando el mismo algoritmo de agrupamiento.

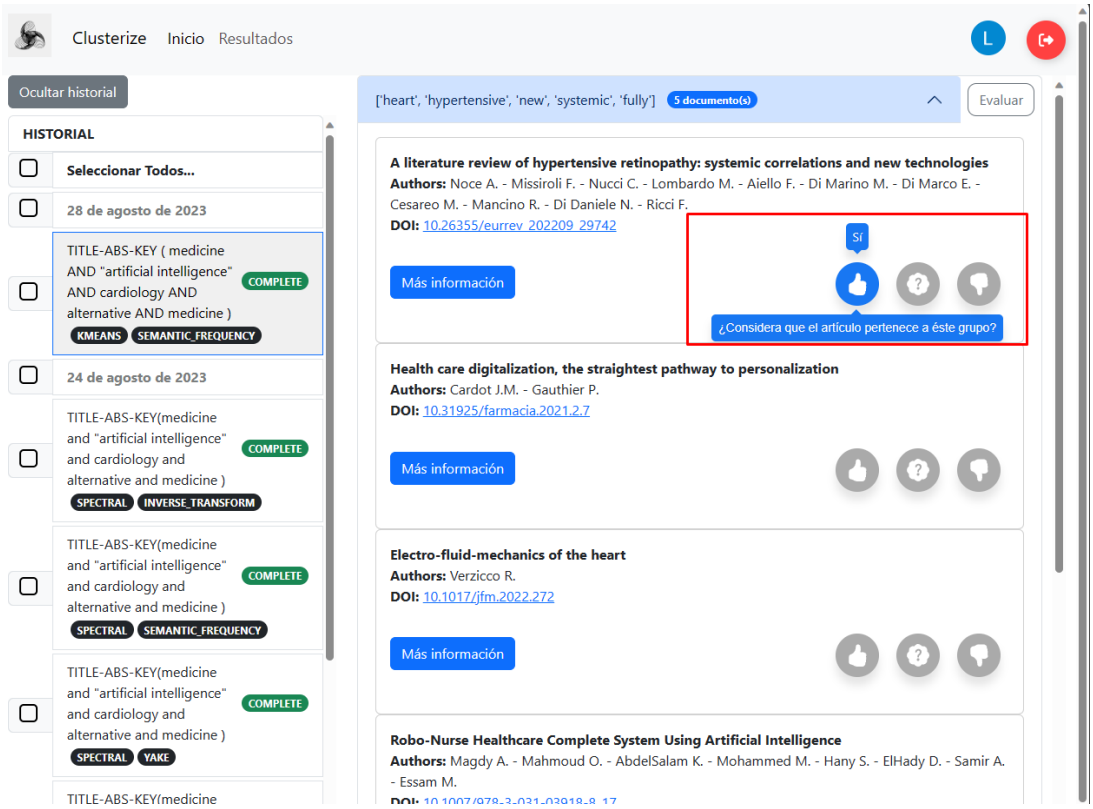


Figura 11. Vista de Evaluación de un Artículo

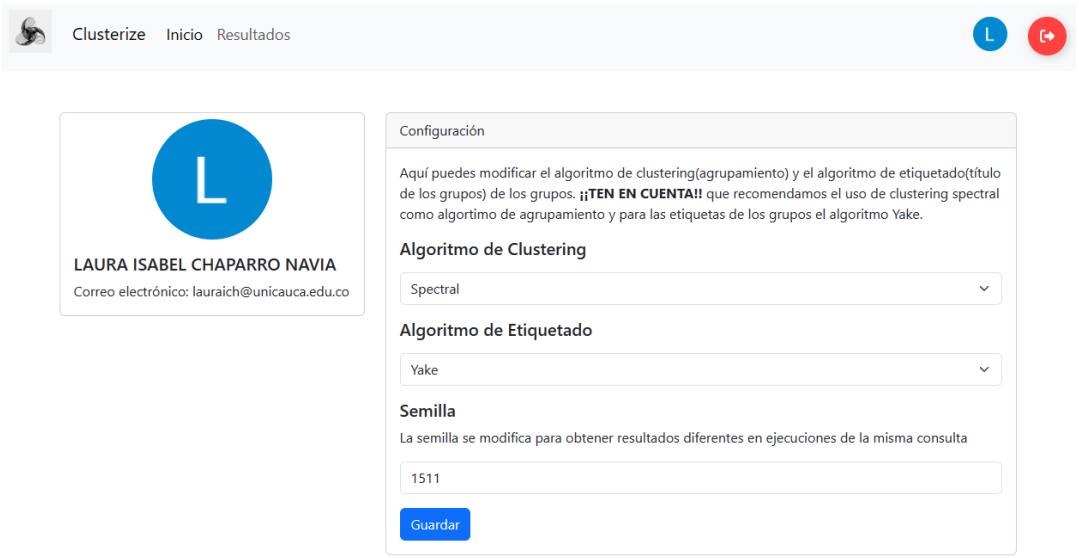


Figura 12. Vista del Perfil del Usuario (Componente Profile)

Esta página ha sido dejada intencionalmente en blanco.

CAPÍTULO 5

5 RESULTADOS EXPERIMENTALES Y ANÁLISIS

Se implementaron dos formas de realizar la evaluación del proyecto desarrollado. La primera consiste en evaluar el comportamiento de los algoritmos de agrupamiento con métricas clásicas como la precisión, recuerdo, F-measure y exactitud. La segunda forma de evaluación es determinando el nivel de satisfacción de un conjunto de investigadores de la Universidad del Cauca mediante la realización de tres (3) preguntas que se encuentran en el aplicativo web. La primera ¿Cómo considera el título del grupo?, la segunda ¿Cómo considera el orden de los artículos dentro de este grupo? Y la tercera ¿Considera que el articulo pertenece a este grupo?

5.1 EVALUACIÓN CON MÉTRICAS CLÁSICAS

Para realizar la esta evaluación se tomaron los cuatro (4) conjuntos de datos mencionados en el Capítulo 3. El conjunto de datos AAAI13 cuenta con 150 artículos. El conjunto de datos AAAI14 cuenta con 396 artículos. El conjunto de datos Arxiv cuenta con 10000 artículos y el conjunto de datos Topic Modeling cuenta con 14004 artículos. Es de resaltar, que a la hora de procesar los conjuntos de datos se removieron los artículos que tenían información incompleta, debido a que esos registros no sirven para el proceso de agrupamiento. Se calculó el promedio ponderado de los valores de las métricas, es decir, los cinco (5) algoritmos se ejecutaron 31 veces y se promedió el resultado de cada métrica. A continuación, la Tabla 8 muestra los resultados promedios de la precisión ponderada, luego la Tabla 9 los resultados de recuerdo, la Tabla 10 los resultados de F-measure y la Tabla 11 los de exactitud. En todas las tablas el peor resultado por conjunto de datos (fila) está en rojo y el mejor está en negrita.

Tabla 8. Resultados de la Métrica de Precisión

Conjuntos de datos	*K-means*	*Spectral*	*Fuzzy C-means*	*STC*	*Lingo*
AAAI13	**75.0544**	74.5402	66.7278	7.7420	35.8805
AAAI14	26.0706	**29.6435**	22.6455	7.4976	21.3987
Arxiv	31.2964	**50.2838**	28.6217	14.7653	24.1971
Topic Modeling	12.3491	8.0885	**29.6558**	19.3421	26.9899

Tabla 9. Resultados de la Métrica de Recuerdo

Conjuntos de datos	*K-means*	*Spectral*	*Fuzzy C-means*	*STC*	*Lingo*
AAAI13	26.0263	**26.3295**	22.0546	7.4458	20.0746
AAAI14	15.0551	**15.2047**	13.185	6.6017	11.5906
Arxiv	25.9867	**42.0088**	29.5426	13.4875	16.2059
Topic Modeling	15.7211	14.7481	**23.0666**	14.2499	9.6266

Tabla 10. Resultados de la Métrica de F-measure

Conjuntos de datos	*K-means*	*Spectral*	*Fuzzy C-means*	*STC*	*Lingo*
AAAI13	**33.8587**	33.61760	27.5064	5.4318	19.6599
AAAI14	16.888	**17.3585**	14.4035	5.9274	12.1059
Arxiv	24.6405	**42.3861**	27.3239	12.7057	15.6886
Topic Modeling	11.9039	8.9783	**22.7102**	13.9096	10.5727

Tabla 11. Resultados de la Métrica de Exactitud

Conjuntos de datos	*K-means*	*Spectral*	*Fuzzy C-means*	*STC*	*Lingo*
AAAI13	0.9862	**0.9865**	0.9823	0.6003	0.8475
AAAI14	**0.9303**	0.9159	0.9175	0.6379	0.8267
Arxiv	0.8500	**0.9111**	0.8025	0.7215	0.7531
Topic Modeling	0.9069	0.9082	**0.9165**	0.8803	0.8659

En la **Tabla 8** se puede observar que en dos (2) de los cuatro (4) conjuntos de datos (AAAI14 y Arxiv) el algoritmo Spectral cuenta con los valores más altos de precisión. Para el conjunto de datos AAAI13 el mejor valor de precisión se obtuvo con K-means y para el conjunto de datos Topic Modeling se obtuvo con Fuzzy C-means.

En la **Tabla 9** se puede apreciar que Spectral obtiene los valores más altos de recuerdo en tres (3) de los conjuntos de datos (AAAI13, AAAI14 y Arxiv). Por otro lado, Fuzzy C-means obtuvo el valor más alto para el conjunto de datos de Topic Modeling.

En la **Tabla 10** se puede observar que para la métrica F-measure el algoritmo Spectral obtuvo mejores valores en dos (2) de los cuatro (4) conjuntos de datos (AAAI14 y Arxiv). Para el conjunto de datos AAAI13 el mejor valor de F-measure se obtuvo con el algoritmo K-means y para el conjunto de datos Topic Modeling se obtuvo con el algoritmo Fuzzy C- means.

En la **Tabla 11** se observa que en dos (2) de los conjuntos de datos (AAAI13 y Arxiv), Spectral obtiene los valores más altos de exactitud. En el conjunto de datos AAAI14, K-means obtiene el valor más alto de esta métrica y en el conjunto de datos Topic Modeling es Fuzzy C-means el que obtiene mejores resultados.

Con lo observado en las tablas anteriores se puede considerar que en general, Spectral y K-means parecen ser consistentemente buenos en términos de precisión, recuerdo, F-measure y exactitud en la mayoría de los conjuntos de datos, esto quiere decir que presentan fortaleza en diferentes tamaños de conjuntos de datos, lo que sugiere su capacidad para identificar patrones y estructuras en diferentes escalas. Fuzzy C-means y Lingo también ofrecen un rendimiento competitivo, mientras que STC muestra tener un rendimiento bajo en comparación con los otros algoritmos basado en las métricas calculadas.

Además, se puede observar que en términos generales Fuzzy C-means se comporta de mejor manera cuando es mayor la cantidad de datos como lo es el caso de conjunto de datos Topic Modeling.

Debido a que STC y Lingo son algoritmos del estado del arte que se utilizan desde la API de Carrot2, a los cuales no se les ha hecho ninguna modificación y los datos de entrada no reciben ningún procesamiento, estos se toman como base para compararlos contra el resto de los algoritmos implementados (Spectral, K-means, Fuzzy C-means) a los cuales sí se les modificó su código fuente, se les integró con índices de calidad de las soluciones encontradas y los conjuntos de datos que recibieron fueron previamente procesados. Al realizar esta comparación se puede observar que Spectral, K-means y Fuzzy C-means presentan un mejor comportamiento debido a que tienen los valores más altos en todas las métricas evaluadas.

Además, al realizar un análisis puntual para la precisión se puede observar que Spectral, K-means y Fuzzy C-means tienen valores más altos que STC y Lingo, esto quiere decir que para estos tres algoritmos la mayoría de los artículos asignados en un grupo se encuentran bien agrupados.

Al analizar el recuerdo se puede observar que a pesar de que Spectral, K-means y Fuzzy C-means tienen valores más altos que STC y Lingo. En general, eso sí, todos presentan valores bajos, lo que significa que en los grupos quedaron faltando artículos que deberían haber pertenecido al grupo.

Con la medida F se puede confirmar que hay un desequilibrio en cuanto a las medidas de precisión y recuerdo. Finalmente, observando la métrica de la exactitud se puede apreciar que los algoritmos de forma general están clasificando correctamente la mayoría de las instancias, aunque estos valores altos pueden estar representando solo la clase mayoritaria (desbalance de clases).

Por otro lado, también se realizó la prueba estadística de Friedman en los promedios ponderados de cada métrica, haciendo uso de la herramienta Keel[16] (Knowledge Extraction based on Evolutionary Learning). Esta prueba es un método no paramétrico basado en rangos que clasifica los algoritmos en un ranking de acuerdo con su rendimiento (rango 1, rango 2, etc.) [50]. Se utiliza para determinar si existe o no una diferencia estadísticamente significativa entre los datos proporcionados.

A continuación, se muestran los resultados obtenidos de la prueba de Friedman y el Post hoc de Holm para cada una de las métricas.

En la Tabla 12 se puede observar el ranking en la prueba de Friedman en la métrica de recuerdo la cual indica que el primer puesto se lo lleva Spectral, el segundo puesto se encuentra en empate entre K-means y Fuzzy C-means, el siguiente puesto fue para Lingo y el ultimo para STC. Este ranking muestra que al menos un algoritmo obtiene resultados diferentes al de los demás debido a que el valor P hallado fue de 0.0122 que se encuentra por debajo de 0.05. Además, según la prueba post hoc de Holm (ver Tabla 13) se puede afirmar que para la métrica del recuerdo el algoritmo de Spectral es mejor que STC y Lingo con un 95% de confianza.

En la prueba de Friedman para la métrica de exactitud (Ver Tabla 14) se puede observar que dentro del ranking el primer lugar lo obtiene Spectral, seguido de K-means en el segundo lugar, el tercero, cuarto y quinto lugar lo obtienen Fuzzy C-means, Lingo y STC respectivamente. Dado que el valor computado de significancia o valor P es de 0.01461 menor que 0.05 este ranking muestra que al menos un algoritmo obtiene resultados diferentes al de los demás. Además, según la prueba de Holm (Ver Tabla 15) el algoritmo de Spectral es mejor que STC con un 95% de confianza.

[16] https://sci2s.ugr.es/keel/index.php

Tabla 12. Resultados de la prueba de Friedman de la métrica de recuerdo. Valor de P computado por la prueba de Friedman: 0.012295523833559474

Algorithm	Ranking
Kmeans	2.25
Spectral	1.5
FuzzyCmeans	2.25
STC	4.75
Lingo	4.25

Tabla 13. Valores P para una confianza del 95% de la métrica de recuerdo.

i	algorithms	$z = (R_0 - R_i)/SE$	p	Holm
10	Spectral vs. STC	2.906888	0.00365	0.005
9	Spectral vs. Lingo	2.459675	0.013906	0.005556
8	Kmeans vs. STC	2.236068	0.025347	0.00625
7	FuzzyCmeans vs. STC	2.236068	0.025347	0.007143
6	Kmeans vs. Lingo	1.788854	0.073638	0.008333
5	FuzzyCmeans vs. Lingo	1.788854	0.073638	0.01
4	Kmeans vs. Spectral	0.67082	0.502335	0.0125
3	Spectral vs. FuzzyCmeans	0.67082	0.502335	0.016667
2	STC vs. Lingo	0.447214	0.654721	0.025
1	Kmeans vs. FuzzyCmeans	0	1	0.05

Tabla 14. Resultados de la prueba de Friedman de la métrica de exactitud. Valor de P computado por la prueba de Friedman: 0.014611900586972704

Algorithm	Ranking
Kmeans	2
Spectral	1.75
FuzzyCmeans	2.25
STC	4.75
Lingo	4.25

El ranking observado en la prueba de Friedman de la métrica de precisión indica que hay un empate para el primer puesto entre Spectral y K-means. La siguiente posición la obtiene Fuzzy C-means, el siguiente puesto es para Lingo y por último se encuentra STC. Este ranking es informativo más no concluyente debido a que el valor P hallado fue de 0.1847 que es un valor superior al 0.05 exigido por la prueba. Con los resultados del test de Friedman no se procedió a realizar el pos hoc de Holm. De lo anterior, queda claro que se requieren más conjuntos de datos para poder llegar a conclusiones generalizables en las relaciones de dominancia

(un algoritmo obtiene mejores resultados que otro) o a que en realidad no existen diferencias significativas en los resultados entregados por los algoritmos.

Tabla 15. Valores P para una confianza del 95% de la métrica de exactitud.

i	algorithms	$z = (R_0 - R_i)/SE$	p	Holm
10	Spectral vs. STC	2.683282	0.00729	0.005
9	Kmeans vs. STC	2.459675	0.013906	0.005556
8	Spectral vs. Lingo	2.236068	0.025347	0.00625
7	FuzzyCmeans vs. STC	2.236068	0.025347	0.007143
6	Kmeans vs. Lingo	2.012461	0.044171	0.008333
5	FuzzyCmeans vs. Lingo	1.788854	0.073638	0.01
4	Spectral vs. FuzzyCmeans	0.447214	0.654721	0.0125
3	STC vs. Lingo	0.447214	0.654721	0.016667
2	Kmeans vs. Spectral	0.223607	0.823063	0.025
1	Kmeans vs. FuzzyCmeans	0.223607	0.823063	0.05

Por otro lado, en la prueba de Friedman para la métrica de F-measure se puede observar que no hay un ganador claro, es decir, se presenta un empate entre los algoritmos K-means, Spectral y Fuzzy C-means. En el siguiente lugar se encuentra Lingo y de ultimo STC. Este ranking es informativo debido a que el valor P hallado fue de 0.1468.Debido a los resultados no se procede a realizar el pos hoc de Holm.

Debido a lo mencionado anteriormente, se tomó la decisión de establecer a Spectral como el algoritmo predeterminado en la configuración del usuario que se puede observar en la página web, puesto que se encuentra en las primeras posiciones para las métricas de exactitud y recuerdo, y esta bien ubicado en las de precisión y medida F.

5.2 EVALUACIÓN CON INVESTIGADORES

Para evaluar el comportamiento de la aplicación con los usuarios se realizaron dos evaluaciones. Las evaluaciones se realizaron con once (11) usuarios en total, con los resultados y sugerencias obtenidas de los cinco (5) primeros evaluadores se aplicaron mejoras dentro de la aplicación que estuvieran dentro del alcance del proyecto y con ellas se realizó una evaluación final con otros seis (6) usuarios.

5.2.1 EVALUACIÓN INICIAL

Para realizar esta evaluación se seleccionaron 5 evaluadores, 4 docentes de la FIET y 1 estudiante de la Universidad del Cauca, a los cuales se les instruyó en el uso de la aplicación. Posteriormente, cada uno usó la aplicación por un tiempo mínimo de 30 minutos y evaluó los resultados obtenidos dentro de las consultas realizadas teniendo en cuenta que el algoritmo de agrupamiento predeterminado para todas las evaluaciones fue Spectral, debido a que fue uno de los mejores algoritmos en la evaluación con métricas y se utilizó Yake como algoritmo de

etiquetado. Para evaluar los resultados se realizaron la siguientes tres (3) preguntas donde cada pregunta tiene tres opciones de respuesta:

1. ¿Cómo considera el título del grupo?

- Bueno
- No sabe
- Malo

2. ¿Cómo considera el orden de los artículos dentro de este grupo?

- Bueno
- No sabe
- Malo

3. ¿Considera que el articulo pertenece a este grupo?

- Si
- Inseguro
- No

A continuación, se presenta un resumen del perfil de los evaluadores:

Evaluador 1 (Ver Figura 13)**:**

Nombre: Luz Marina Sierra Martínez

Ocupación: Docente de la FIET

Consulta realizada: TITLE-ABS-KEY("information technology" and "project management" and pmbok)

Cantidad de Artículos encontrados: 48

Cantidad de Grupos generados: 4

Títulos asignados a los grupos:

- Decision, Knowledge, Model, Plm, Selection
- Failure, Paradigm, Projects, Success, Traditional
- Application, Information, Maturity, Performance, Projects
- Development, Information, Process, Processes, Projects

Evaluador 2 (Ver Figura 14)**:**

Nombre: Jorge Jair Moreno Chaustre

Ocupación: Docente de la FIET

Consulta realizada: TITLE-ABS-KEY(artificial and intelligence and for and finance and chatgpt)&date=2020-2023

Cantidad de Artículos encontrados: 5

Cantidad de Grupos generados: 2

Títulos asignados a los grupos:

- Auditors, Cognitive, Financial, Human, Research
- Business, Education, Gai, Generative, Industry

Evaluador 3 (Ver Figura 15)**:**

Nombre: Daniel Eduardo Paz Perafan

Ocupación: Docente de la FIET

Consulta realizada: ALL((("validation") and ("software and requirements")))&date=2000-2023

Cantidad de Artículos encontrados: 7

Cantidad de Grupos generados: 2

Títulos asignados a los grupos:

- systems, test, verification
- approach, data, study

Consulta realizada: TITLE("validation" and "software requirements")

Cantidad de Artículos encontrados: 11

Cantidad de Grupos generados: 2

Títulos asignados a los grupos:

- Consistency, Quality, Review, Srv, Techniques
- Development, Gsd, Paper, Specification, Study

Evaluador 4 (Ver Figura 16)**:**

Nombre: Hugo Armando Ordoñez Erazo

Ocupación: Docente de la FIET

Consulta realizada: TITLE-ABS-KEY(business and process and models and clustering)&date=2021-2022

Cantidad de Artículos encontrados: 184

Cantidad de Grupos generados: 9

Títulos asignados a los grupos:

- Analysis, Based, Data, Innovation, Model

- Conference, Data, Detection, Learning, Network
- Agricultural, Data, Model, Network, System
- Analysis, Data, Digital, Model, Smes
- Customer, Customers, Data, Model, Segmentation
- Analysis, Data, Learning, Machine, System
- Data, Event, Mining, Model, Time
- Analysis, Based, Cluster, Development, Model
- Based, Data, Mining, Model, System

Evaluador 5 (Ver Figura 17)**:**

Nombre: Danny Alberto Diaz Mage

Ocupación: Estudiante de ingeniería de Sistemas

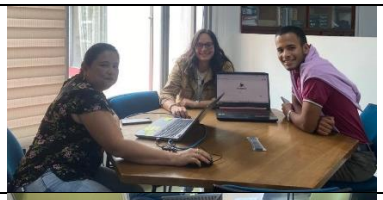
Consulta realizada: TITLE-ABS-KEY(machine and learning and psychoactive and substances)

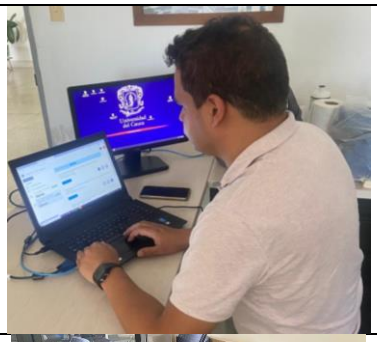
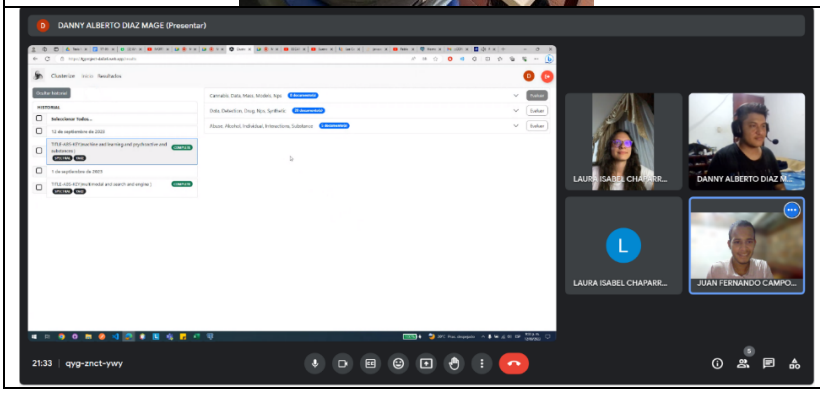
Cantidad de Artículos encontrados: 34

Cantidad de Grupos generados: 3

Títulos asignados a los grupos:

- Abuse, Alcohol, Individual, Interactions, Substance
- Cannabis, Data, Mass, Models, Nps
- Data, Detection, Drug, Nps, Synthetic

	Figura 13. Evaluación con Luz Marina Sierra Martínez
	Figura 14. Evaluación con Jorge Jair Moreno Chaustre

	Figura 15. Evaluación con Daniel Eduardo Paz Perafan
	Figura 16. Evaluación con Hugo Armando Ordoñez Erazo
	Figura 17. Evaluación con Danny Alberto Diaz Mage

Los resultados de las evaluaciones realizadas por los evaluadores son las siguientes:

En esta evaluación se obtuvieron 22 grupos, se recuperaron 289 artículos y se realizaron 6 consultas. Los resultados para cada una de las preguntas son los siguientes:

1. ¿Cómo considera el título del grupo?

- Bueno – 19 evaluaciones
- No sabe – 2 evaluaciones
- Malo – 1 evaluación

2. ¿Cómo considera el orden de los artículos dentro de este grupo?

- Bueno – 15 evaluaciones
- No sabe – 4 evaluaciones
- Malo – 3 evaluaciones

3. ¿Considera que el articulo pertenece a este grupo?

- Si – 220 evaluaciones
- Inseguro – 49 evaluaciones
- No – 19 evaluaciones

De acuerdo con la Figura 18 que presenta en porcentajes los resultados anteriores, se puede concluir que en cuanto a la pregunta ¿Cómo considera el título del grupo?, en promedio el 86.4% de los grupos tiene un buen título, en el 9.1% de los grupos no se pudo concluir si era bueno o malo y el 4.5% posee un mal título. Por otro lado, para la pregunta ¿Cómo considera el orden de los artículos dentro de este grupo? Se halló que el 68.2% de los grupos poseía un buen orden en cuanto a la relevancia de los artículos hacia el título del grupo. El 23.6% de los grupos poseía un orden malo o inadecuado y finalmente en el 18.2% de los grupos no se pudo concluir si el orden era adecuado o inadecuado.

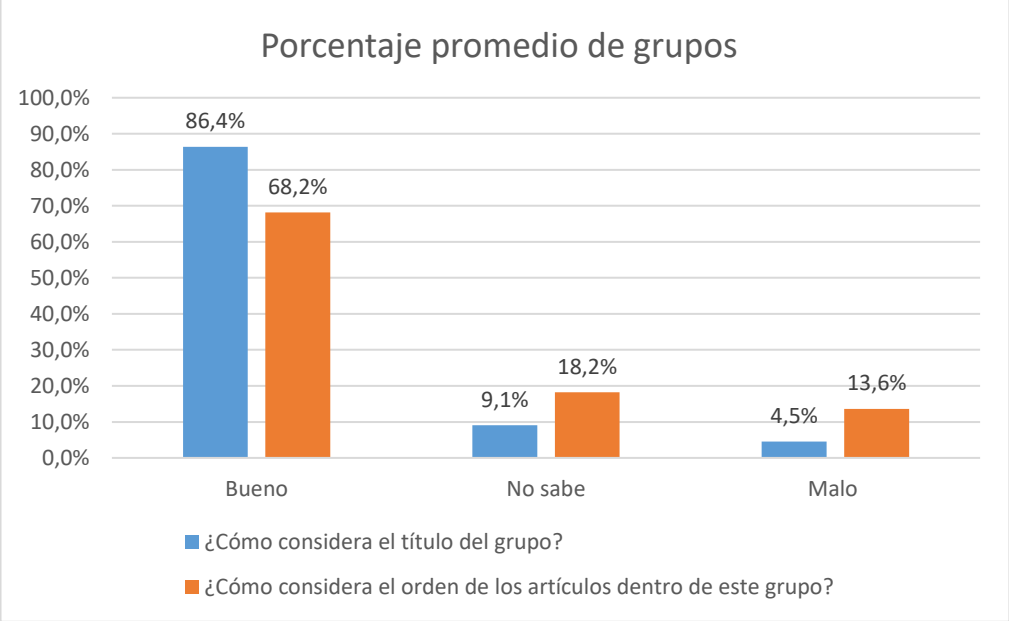


Figura 18. Porcentaje promedio de grupos para la evaluación inicial

Por otra parte, con respecto a la pregunta relacionada con el agrupamiento (ver Figura 19) ¿Considera que el articulo pertenece a este grupo?, los evaluadores consideran que el 71.12% de los artículos están bien agrupados, para un 16,96% de los artículos había inseguridad si debían o no estar en el grupo asignado y un 6,57% de los artículos se consideraron mal agrupados.

Analizando los datos obtenidos en la evaluación se puede apreciar que la mayoría de los evaluadores perciben los títulos de los grupos como "Buenos", pero una pequeña proporción no está segura o considera que son "Malos", debido a que en muchos casos los títulos generados son muy generales y no permiten diferenciar las temáticas entre los grupos o simplemente no representan la temática de los

artículos pertenecientes al grupo. Además, consideran que la mayoría de los artículos están bien agrupados en los grupos correspondientes, aunque se presentan casos en los cuales el algoritmo de agrupamiento asigna el artículo erróneamente dentro de un grupo debido a que se pueden estar generando una cantidad de grupos menor o mayor a la cantidad de temáticas que se pueden encontrar en el conjunto de artículos provenientes de Scopus.

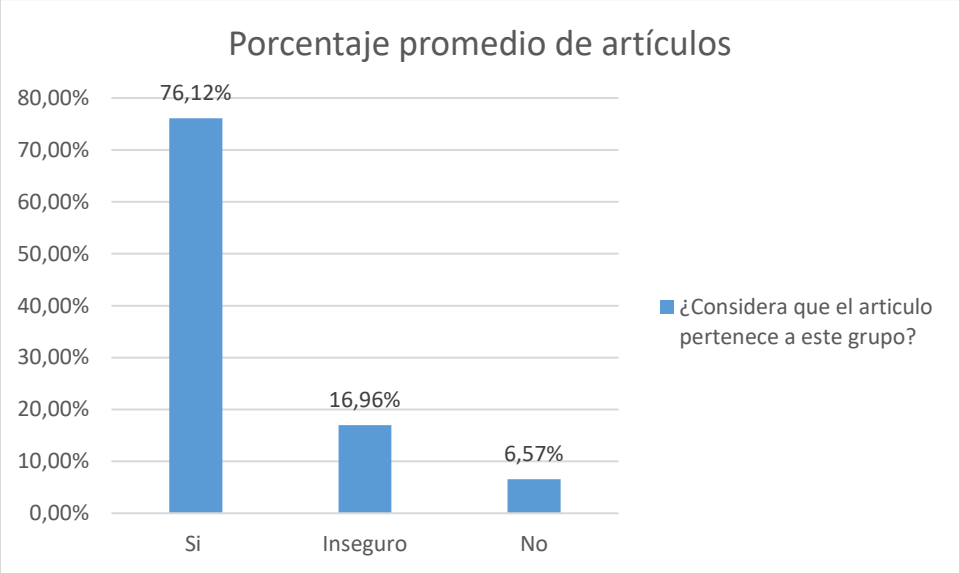


Figura 19. Porcentaje promedio de artículos para la evaluación inicial

Finalmente, en cuanto al orden de los artículos se puede a preciar que de forma general se encuentran bien porque los primeros artículos se relacionan con el título del grupo, sin embargo, el ordenamiento falla en los casos en los que se generan grupos que contienen temáticas variadas o generales y en donde no se puede definir una temática en común.

Es de resaltar que los evaluadores realizaron una serie de sugerencias y observaciones sobre el funcionamiento de la aplicación las cuales se analizaron para realizar una versión mejorada de la aplicación para realizar posteriormente una evaluación final.

Las sugerencias y observaciones fueron las siguientes:

- Agregar la funcionalidad para exportar los resultados obtenidos con la consulta
- Implementar una forma de asistencia al momento de realizar la consulta
- Realizar un filtrado o mejorar la calidad de los artículos obtenidos con la búsqueda de Scopus antes de realizar el agrupamiento
- Mejorar el mecanismo de evaluación utilizando formas de respuesta en escala que permitieran respuestas intermedias

- Mejorar la usabilidad para considerar diferentes tipos de usuarios debido a que por el momento se encuentra enfocado en personas con conocimiento en construcción de cadenas de búsqueda de Scopus
- Indicar visualmente que los artículos se encuentran ordenados dentro de cada grupo dependiendo de la relevancia de ese artículo hacia el grupo
- Indicar visualmente que ya se realizó la evaluación de un grupo
- Indicar dentro de la aplicación para que sirve esta
- Obtener los resultados de la búsqueda y agrupamiento de inmediato

Teniendo en cuenta las sugerencias de los evaluadores y analizando el alcance del proyecto se aplicaron las siguientes modificaciones en la aplicación:

- Se agregó en la página inicial una sección donde se explica el objetivo de la aplicación, cómo se utiliza y cómo observar los resultados (ver Figura 20). Además, se agregó un Tour o visita guiada (ver Figura 21 y Figura 22) en el componente de resultados con el objetivo de mejorar la usabilidad de la página. Este tour se activa la primera vez que ingresa el usuario y también se encuentra disponible desde el botón de ayuda (signo de pregunta) que se puede utilizar en cualquier momento.

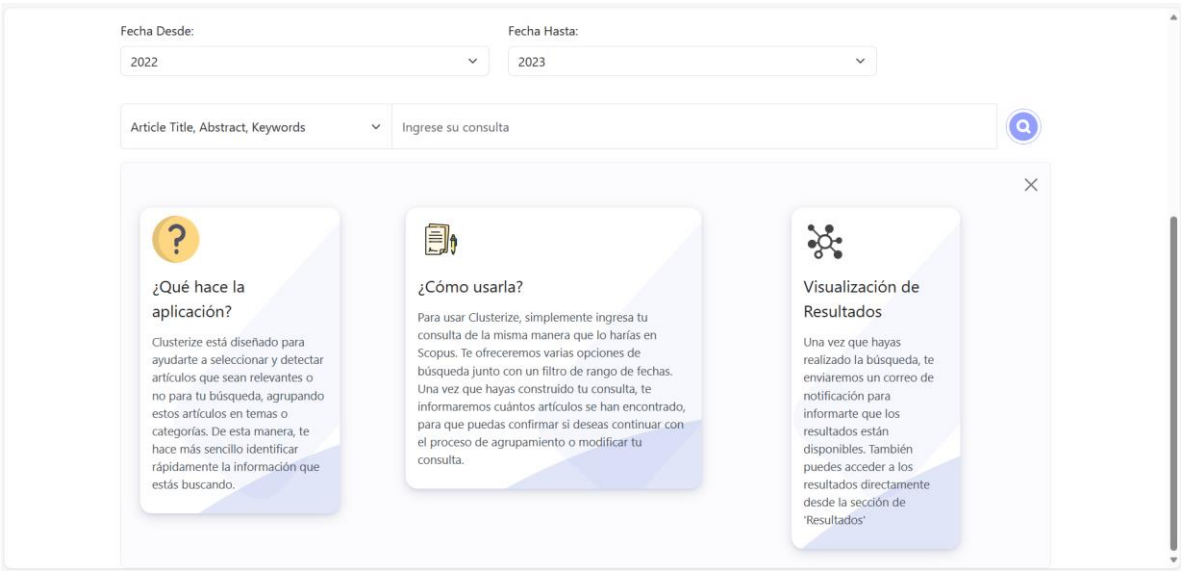


Figura 20. Vista de la página inicial con la sección de explicación agregada

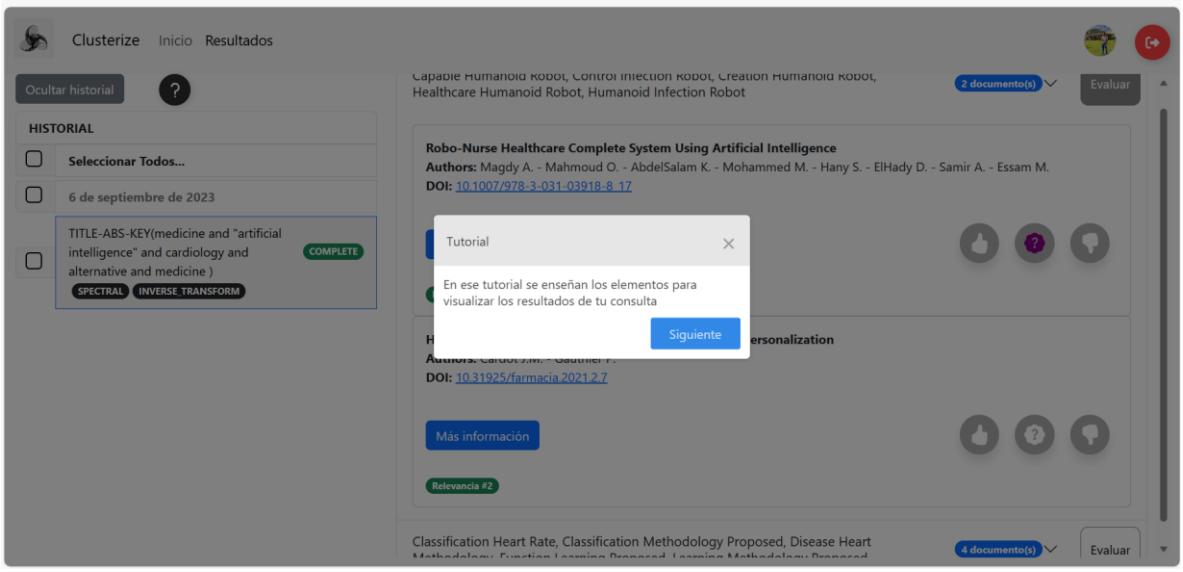


Figura 21. Vista de la página de resultados con el inicio del tour o visita guiada

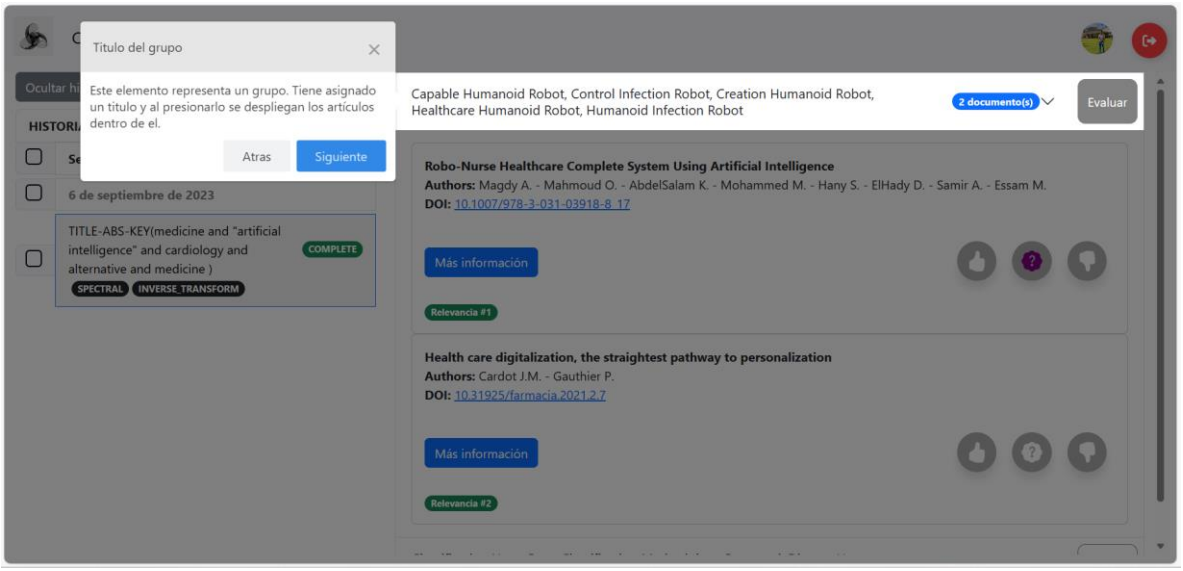


Figura 22. Vista de la página de resultados con visita guiada en explicación del título del grupo

- Se indicó visualmente que los artículos se encuentran ordenados por relevancia hacia el grupo mediante un distintivo que indica el ranking de relevancia (ver Figura 23).

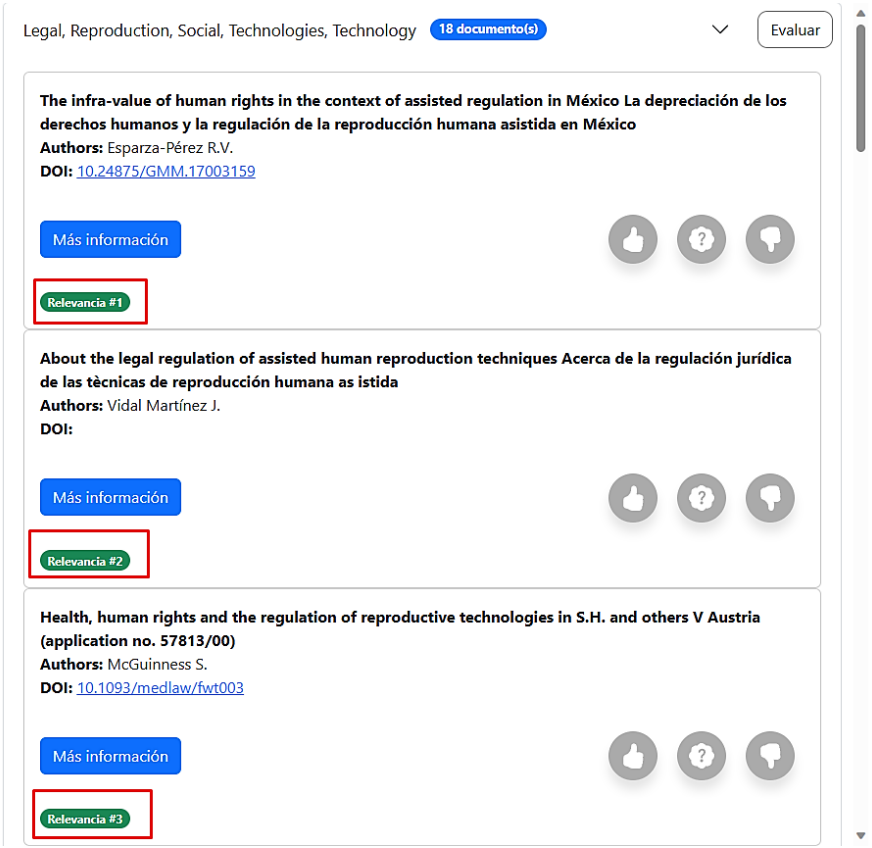


Figura 23. Vista de la página de resultados donde se enseña el orden de los artículos

- Se modificó la opción de respuesta de la evaluación del grupo “No sabe” por “Regular” en las preguntas ¿Cómo considera el título del grupo? Y ¿Cómo considera el orden de los artículos dentro de este grupo? Ahora las opciones de respuesta son Bueno, Regular o Malo. Esto con el objetivo de dar mayor claridad y que sea evidente la respuesta intermedia, es decir, que no está del todo bien pero tampoco mal (ver Figura 24).

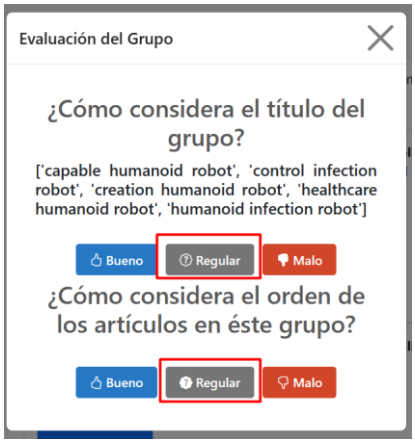


Figura 24. Vista de la evaluación de un grupo con el cambio de respuesta

- Se indico visualmente cuando se termina de realizar la evaluación de un grupo mediante un cambio de color. Si no ha dado respuesta a las dos preguntas el botón de evaluación estará de color blanco, en caso de que responda a las dos pasará a estar de color gris (ver Figura 25).

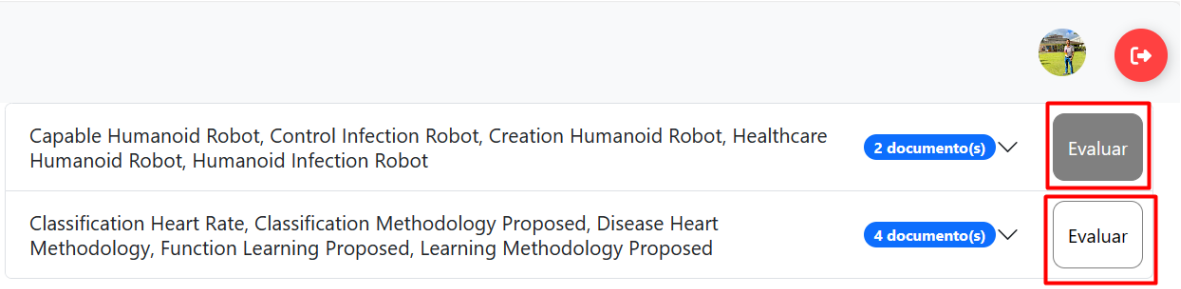


Figura 25. Vista de la página de resultados con el botón de evaluar resaltado cuando se completa la evaluación

Es importante mencionar que las siguientes sugerencias no se realizaron y se dejan como trabajo futuro del proyecto:

1. Agregar la funcionalidad para exportar los resultados obtenidos con la consulta
2. Implementar una forma de asistencia al momento de realizar la consulta
3. Realizar un filtrado o mejorar la calidad de los artículos obtenidos con la búsqueda de Scopus antes de realizar el agrupamiento
4. Mejorar el mecanismo de evaluación utilizando formas de respuesta en escala que permitieran respuestas intermedias
5. Mejorar la usabilidad para considerar diferentes tipos de usuarios debido a que por el momento se encuentra enfocado en personas con conocimiento en construcción de cadenas de búsqueda de Scopus
6. Obtener los resultados de la búsqueda y agrupamiento de inmediato

Las sugerencias 1, 2 y 3 no se realizaron porque rebasaban el alcance y los objetivos del proyecto.

La sugerencia 4 se tomó en cuenta en cuanto a las respuestas intermedias para realizar el cambio de la Figura 24, pero no se aplicó un formato de respuesta en forma escalar debido a que al final sólo se obtendría un promedio y no un porcentaje por cada uno de los posibles valores, que era lo que se quería para identificar más fácilmente la inclinación de los usuarios.

La sugerencia 5 también se aplicó parcialmente debido a que a pesar de que se intenta mejorar a través de la guía de las Figura 22 y la Figura 23. Aun se presentan dificultades de usabilidad sobre todo en la construcción de la consulta, ya que, requiere del usuario un conocimiento previo.

La sugerencia 6 no se implementó porque hay 3 cuellos de botella que implican que la aplicación no pueda responder de manera inmediata, el primer cuello de botella es la recuperación de los datos desde Scopus porque por cada artículo de

una consulta se debe hacer una petición para recuperar su información, la API que se uso es gratis y su rendimiento no es el mejor. El segundo cuello de botella se tiene al momento de determinar el número de grupos en los cuales se realizará el agrupamiento porque para ello se hace agrupamiento de antemano con diferentes valores para el numero de grupos lo que conlleva un tiempo considerable y finalmente el ultimo cuello de botella fue realizar el agrupamiento con la cantidad de grupos previamente determinada. Además, es de resaltar que al alojar la aplicación en un espacio gratuito de Google Cloud no se disponía de mucha capacidad de procesamiento.

5.2.2 EVALUACIÓN FINAL

Con los cambios realizados en la aplicación se procedió a realizar una evaluación final, de la misma forma que se describió en la evaluación inicial. La diferencia es que para esta evaluación los usuarios fueron 2 docentes, 1 egresada de la universidad y 3 estudiantes. Además, se incluyó un formato de encuesta de satisfacción como soporte de las sugerencias y observaciones realizadas por los encuestados. Este formato consta de dos preguntas: ¿A nivel general cómo considera la aplicación desarrollada?, ¿Qué sugerencias tiene para mejorar la aplicación? (ver **Anexo 7**).

A continuación, se presenta un resumen del perfil de los evaluadores:

Evaluador 1 (Ver Figura 26)**:**

Nombre: Ricardo Antonio Zambrano Segura

Ocupación: Docente de la FIET

Consulta realizada: TITLE("speech recognition" or "speech-to-text" or "voice recognition" or "voice analysis" and "electronic health records" or "natural language processing" or "nlp" or "ehr")&date=2020-2023

Cantidad de Artículos encontrados: 19

Cantidad de Grupos generados: 2

Títulos asignados a los grupos:

- Application, Artificial, Intelligence, Neural, Text
- Learning, Machine, Model, Performance, System

Evaluador 2 (Ver Figura 27)**:**

Nombre: María Isabel Vidal Caicedo

Ocupación: Docente de la FIET

Consulta realizada: TITLE-ABS-KEY(((imbalanced or balanc) and data) and (protein and structure))&date=2019-2023

Cantidad de Artículos encontrados: 45

Cantidad de Grupos generados: 2

Títulos asignados a los grupos:

- Imbalanced, Patients, Sleep
- Learning, Prediction, Structure

Evaluador 3 (Ver Figura 28)**:**

Nombre: Nesbi Jhoana Campo Campo

Ocupación: Estudiante de Economía

Consulta realizada: TITLE-ABS-KEY("time poverty" and gender)&date=2022-2023

Cantidad de Artículos encontrados: 23

Cantidad de Grupos generados: 2

Títulos asignados a los grupos:

- Income
- Rural

Evaluador 4 (Ver Figura 29)**:**

Nombre: Jefferson Eduardo Campo Yule

Ocupación: Estudiante de Ingeniería de Sistemas

Consulta realizada: TITLE-ABS-KEY(machine and learning and psychoactive and substance)&date=2018-2023

Cantidad de Artículos encontrados: 29

Cantidad de Grupos generados: 3

Títulos asignados a los grupos:

- Abuse, Alcohol, Disorders, Individual, Interactions
- Cannabis, Classification, Mass, Models, Substances
- Data, Drug, Nps, Substances, Synthetic

Evaluador 5 (Ver Figura 30)**:**

Nombre: Julio Cesar Mellizo Hurtado

Ocupación: Estudiante de Ingeniería de Sistemas

Consulta realizada: TITLE-ABS-KEY(smart and cities and in and tourism and education)&date=2022-2023

Cantidad de Artículos encontrados: 22

Cantidad de Grupos generados: 3

Títulos asignados a los grupos:

- Development, Digital, Economy, Tourists, Transformation
- City, Development, Future, Health, Sustainable
- Architecture, City, Data, Studies, Technologies

Evaluador 6 (Ver Figura 31)**:**

Nombre: Jisele Guacheta Campo

Ocupación: Egresada de la Universidad del Cauca

Consulta realizada: ALL("ethnic activism" and gender and media and literacy)&date=2010-2023

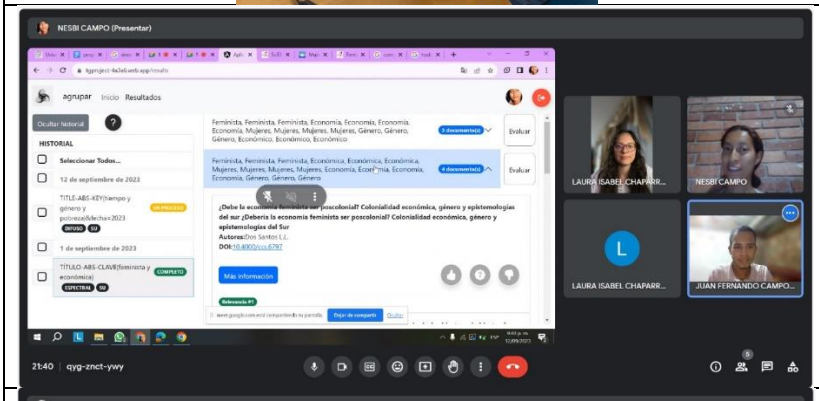
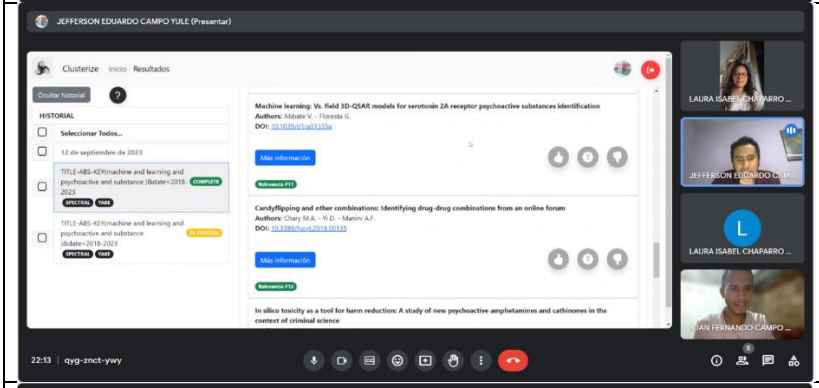
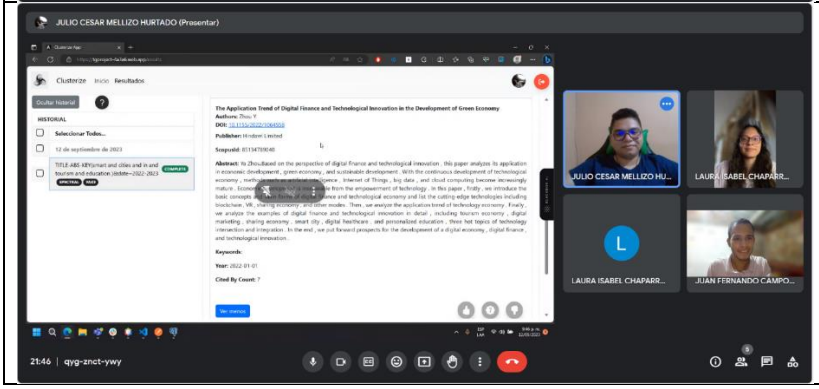
Cantidad de Artículos encontrados: 8

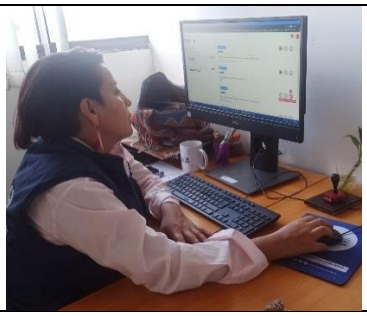
Cantidad de Grupos generados: 2

Títulos asignados a los grupos:

- Education, Ethnography, Nepal, Thangmi
- Cultural, Immigrants, Skills, Women

	Figura 26. Evaluación con Ricardo Antonio Zambrano Segura

	Figura 27. Evaluación con María Isabel Vidal Caicedo
	Figura 28. Evaluación con Nesbi Jhoana Campo Campo
	Figura 29. Evaluación con Jefferson Eduardo Campo Yule
	Figura 30. Evaluación con Julio Cesar Mellizo Hurtado

	Figura 31. Evaluación con Jisele Guacheta Campo

Los resultados de la evaluación final fueron los siguientes:

En esta evaluación se obtuvieron 14 grupos, se recuperaron 146 artículos y se realizaron 6 consultas. Los resultados para cada una de las preguntas son los siguientes:

1. ¿Cómo considera el título del grupo?

- Bueno – 10 evaluaciones
- Regular – 3 evaluaciones
- Malo – 1 evaluación

2. ¿Cómo considera el orden de los artículos dentro de este grupo?

- Bueno – 13 evaluaciones
- Regular – 1 evaluación
- Malo – 0 evaluaciones

3. ¿Considera que el articulo pertenece a este grupo?

- Si – 96 evaluaciones
- Inseguro – 24 evaluaciones
- No – 26 evaluaciones

En la Figura 32 se puede observar que los evaluadores consideran que el 71.4% de los grupos tienen un buen título, el 21.4% los títulos son regulares y el 7.1% tienen un mal título. Adicionalmente la forma en cómo se ordenaron los artículos dentro de cada grupo se considera que 92.9% de los grupos tenían un buen orden de los artículos y un 7.1% no estaban del todo bien ordenados, y para ninguno de los grupos se consideró mal ordenados los artículos que los conformaban.

Por otro lado, de acuerdo con los datos de la Figura 33 se puede apreciar que los evaluadores consideraron que el 65.8% de los artículos si estaban bien agrupados, es decir, si pertenecen al grupo asignado. Además, se encuentran inseguros sobre la pertenencia del 16.4% de los artículos y agregan que el 17.8% de los artículos no pertenecen al grupo asignado.

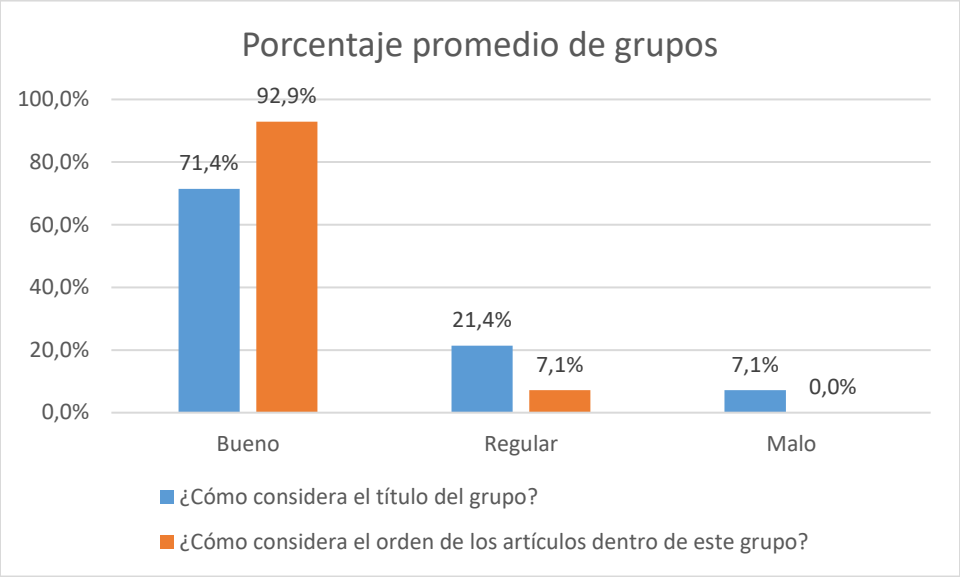


Figura 32. Porcentaje promedio de grupos para la evaluación final

Es de resaltar que, debido al cambio de opciones de respuesta para las dos primeras preguntas de la evaluación, los evaluadores pudieron expresar de una mejor manera cuando el titulo y el orden del grupo no estaban del todo bien pero tampoco estaban mal, esto gracias a la opción de respuesta “Regular”. Además, gracias a que se enseña gráficamente como están ordenados los artículos dentro de cada grupo, los evaluadores pudieron dar respuesta de una forma más consiente a la pregunta ¿Cómo considera el orden de los artículos dentro de este grupo?

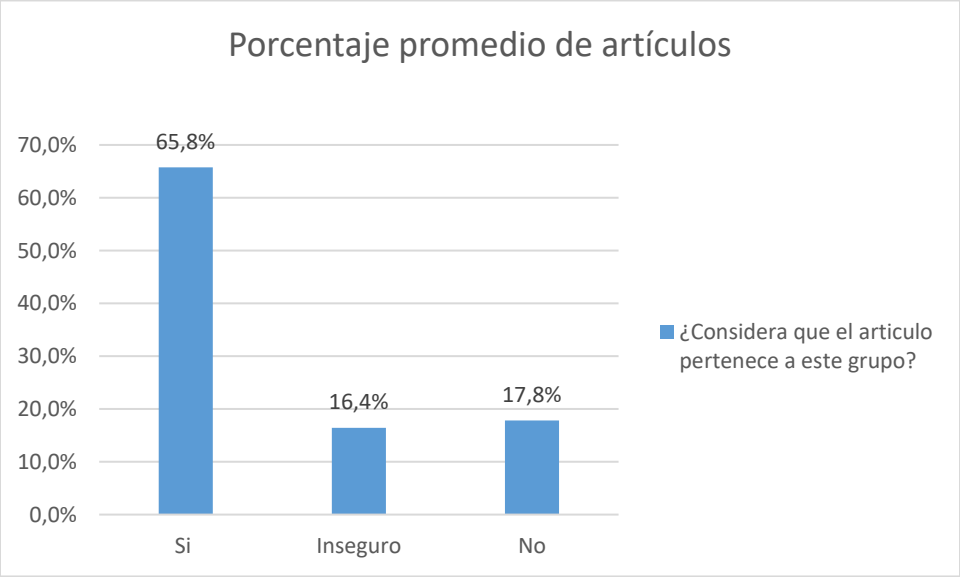


Figura 33. Porcentaje promedio de artículos para la evaluación final

Aunque los cambios realizados para esta evaluación tenían el objetivo de mejorar la experiencia y usabilidad de la aplicación web, los resultados de la evaluación final presentan un comportamiento similar a los obtenidos en la evaluación inicial.

De forma general las reacciones de los evaluadores fueron positivas debido a que le hallaron valor y funcionalidad a la aplicación desarrollada, además indican que la curva de aprendizaje de la aplicación es rápida y el nivel de usabilidad es bueno.

Las sugerencias realizadas fueron las siguientes:

- Mejorar la construcción de los títulos de los grupos debido a que muchas veces no se entregan títulos que representen el contenido del grupo o que son muy generales, además el hecho de que se repita una palabra muchas veces en un artículo no quiere decir que deba estar como título de un grupo.
- Resaltar de mejor manera las respuestas marcadas en la evaluación
- Para hacer uso de la aplicación se debe tener conocimiento sobre la construcción de cadenas de búsqueda de Scopus lo cual no permite que cualquier persona la pueda utilizar
- Permitir realizar búsquedas en español para facilitar la selección de artículos relevantes
- Permitir que la aplicación sea compatible con diferentes dispositivos y tamaños de pantalla, es decir, responsive.
- Notificar los resultados no solo en el correo sino dentro de la aplicación También.
- Consultar más bases de datos para tener diversidad en los artículos encontrados por cada consulta.
- Realizar un diseño más colorido y llamativo de la aplicación.

Esta página ha sido dejada intencionalmente en blanco.

CAPÍTULO 6

6 CONCLUSIONES, RECOMENDACIONES Y TRABAJO FUTURO

Con respecto al primer objetivo específico, se implementaron 5 algoritmos de agrupamiento: K-means, Spectral, Fuzzy C-means, Lingo y STC, y 5 algoritmos de etiquetado o construcción de títulos: Inverse transform, Noun phrases, Semantic frequency, Yake y Graph Topic Rank. A los algoritmos de agrupamiento K-means, Spectral y Fuzzy C-means se les modifico el código fuente para utilizar distancia de cosenos en lugar de distancia euclidiana, esto debido a que esta distancia no es sensible a la longitud de los vectores lo que permite obtener mejores resultados de agrupamiento con documentos de diferentes tamaños o longitudes. Posteriormente, se realizó la evaluación de los algoritmos de agrupamiento mediante el uso de métricas clásicas (precisión, recuerdo, medida f y exactitud) y la utilización de 4 conjuntos de datos reales, obteniendo que los algoritmos K-means y Spectral poseen mejor funcionamiento con respecto a los otros algoritmos.

En relación con el segundo objetivo específico, se desarrolló la aplicación web llamada Clusterize en Angular la cual fue desplegada de Firebase Hosting, esta cuenta con los componentes: Login, Home, Results y Profile. El backend se desarrolló en Django el cual consta de los siguientes módulos: Gestión de artículos, Agrupamiento, Gestión de la evaluación, Gestión de consultas y Gestión de usuarios; este está desplegado en Google Cloud. Además, se creó un trabajador de celery, una base de datos MySQL, y un servicio de carrot2, todo lo anterior desplegado sobre la nube de Google Cloud. Es de resaltar que, debido a que la aplicación es non-real time se realiza una notificación a través del correo electrónico al usuario cuando los resultados de la consulta se encuentren disponibles.

Finalmente, en el tercer objetivo específico se obtuvieron resultados prometedores, donde se aprecia que la aplicación genera un aporte de valor para los investigadores en el desarrollo de los proyectos de investigación, debido a que ayuda en el proceso de depuración de artículos en la revisión de literatura de forma intuitiva y con una curva de aprendizaje corta, mediante el agrupamiento de estos artículos, permitiendo descartar de manera consciente grupos de artículos no relevantes a sus necesidades de información. Además, se determinó que de forma general la aplicación agrupa correctamente los artículos haciendo uso de Spectral, aunque se debe mejorar la forma en la cual se obtiene la cantidad de grupos, ya que, a partir de esto se generan inconsistencias tanto en el agrupamiento como en el ordenamiento de los artículos hacia el título del grupo al cual pertenecen. En cuanto a la generación de los títulos con el algoritmo Yake se

obtuvo un buen funcionamiento, ya que genera títulos cortos y fáciles de entender, sin embargo, se presentan títulos muy genéricos o que contienen palabras que en el contexto de la búsqueda no aportan significado al grupo.

Debido a las dificultades presentadas al momento de asignar los títulos a los grupos por la complejidad de encontrar títulos representativos, comprensibles y disyuntivos, como trabajo futuro se propone obtener las etiquetas más importantes primero y a partir de ellas realizar el agrupamiento de los documentos que van dentro de las etiquetas halladas inicialmente. Además, como alternativa para mejorar el etiquetado de títulos se sugiere evaluar el uso de modelos a gran escala (LLM), realizar la búsqueda de herramientas o algoritmos de etiquetado o construir una propuesta nueva de un algoritmo que entregue títulos disyuntos, es decir, que los títulos sean mutuamente excluyentes y por lo tanto se pueda diferenciar más fácilmente un grupo de otro; y que representen de forma clara la temática de los artículos pertenecientes a cada grupo. De igual forma, también se puede incorporar el uso de LLM para que el usuario cuente la posibilidad de realizar preguntas en lenguaje natural y el LLM pueda entregar información detallada a partir de los resultados obtenidos con los algoritmos de agrupamiento.

Por otra parte, la construcción de la matriz TF-IDF puede mejorar con el uso de tesauros y diccionarios o incluso se puede pensar en representaciones que capturen las relaciones semánticas entre las palabras como lo hace Doc2Vec. Lo anterior con el fin de mejorar los resultados del agrupamiento.

Se recomienda buscar metodologías para equilibrar la cantidad de grupos de tal manera que la cantidad de documentos por cada grupo no sea excesiva y tampoco se presenten casos donde hay muchos grupos con muy pocos documentos. Esto con el fin de ayudar al usuario a descartar los artículos que no son relevantes para su búsqueda, leyendo la menor cantidad de artículos posibles.

Con el objetivo de aportar más valor, teniendo en cuenta las sugerencias obtenidas en la evaluación con los investigadores se propone un asistente de consultas que permita ayudar a los usuarios en la construcción de cadenas de búsqueda para obtener mejores documentos, además de realizar un filtrado a los resultados para identificar y excluir aquellos artículos que hablan de áreas diferentes a las consultadas. También se sugiere consultar más bases de datos bibliográficas con el fin de tener mayor variedad de artículos. Además, con el objetivo de ampliar las posibilidades de búsqueda se aconseja adicionar dentro de la implementación el procesamiento de palabras en español.

Finalmente, se recomienda realizar mejoras u optimizaciones en los algoritmos de agrupamiento de tal forma que se pueda mejorar la calidad de agrupamiento entregadas por estos. Una forma de realizar esto es investigar en el estado del arte la utilización de algoritmos de optimización que permitan encontrar parámetros óptimos para la configuración de los algoritmos de agrupamiento.

● CAPÍTULO 7

7 BIBLIOGRAFÍA

[1] Institut Teknologi dan Bisnis, Institute of Electrical and Electronics Engineers. Indonesia Section, and Institute of Electrical and Electronics Engineers, *2019 1st International Conference on Cybernetics and Intelligent System (ICORIS) : Institut Teknologi dan Bisnis (ITB) STIKOM Bali, Indonesia, 22nd-23rd August 2019*.

[2] D. Hanyurwimfura, L. Bo, D. Njagi, and J. P. Dukuzumuremyi, "A centroid and relationship based clustering for organizing research papers," *International Journal of Multimedia and Ubiquitous Engineering*, vol. 9, no. 3, pp. 219–233, 2014, doi: 10.14257/ijmue.2014.9.3.21.

[3] Rachel McCullough, "Scopus Roadmap: What's New in 2022?," https://blog.scopus.com/posts/scopus-roadmap-whats-new-in-2022.

[4] S. W. Kim and J. M. Gil, "Research paper classification systems based on TF-IDF and LDA schemes," *Human-centric Computing and Information Sciences*, vol. 9, no. 1, Dec. 2019, doi: 10.1186/s13673-019-0192-7.

[5] Y. Liang, Q. Li, and T. Qian, "Finding Relevant Papers Based on Citation Relations," 2011. doi: 10.1007/978-3-642-23535-1_35.

[6] A. Sesagiri Raamkumar, S. Foo, and N. Pang, "Using author-specified keywords in building an initial reading list of research papers in scientific paper retrieval and recommender systems," *Inf Process Manag*, vol. 53, no. 3, pp. 577–594, May 2017, doi: 10.1016/j.ipm.2016.12.006.

[7] T. R. P. M. Rúbio and C. A. S. J. Gulo, "Enhancing Academic Literature Review through Relevance Recommendation Using Bibliometric and Text-based Features for Classification," 2016. doi: 10.1109/CISTI.2016.7521620.

[8] Z. Yu and T. Menzies, "FAST2: An intelligent assistant for finding relevant papers," *Expert Syst Appl*, vol. 120, pp. 57–71, Apr. 2019, doi: 10.1016/j.eswa.2018.11.021.

[9] X. Bai, M. Wang, I. Lee, Z. Yang, X. Kong, and F. Xia, "Scientific paper recommendation: A survey," *IEEE Access*, vol. 7. Institute of Electrical and Electronics Engineers Inc., pp. 9324–9339, 2019. doi: 10.1109/ACCESS.2018.2890388.

[10] J. Chen and Z. Ban, “Academic Paper Recommendation Based on Clustering and Pattern Matching,” Aug. 2019, pp. 171–182. doi: https://doi.org/10.1007/978-981-32-9298-7.

[11] K. Rinartha and L. G. Surya Kartika, “Scientific Article Clustering Using String Similarity Concept,” in *2019 1st International Conference on Cybernetics and Intelligent System (ICORIS)*, IEEE, Aug. 2019, pp. 13–17. doi: 10.1109/ICORIS.2019.8874879.

[12] R. F. M. Ahmed, C. Salama, and H. Mahdi, “Clustering Research Papers Using Genetic Algorithm Optimized Self-Organizing Maps,” in *Proceedings of ICCES 2020 - 2020 15th International Conference on Computer Engineering and Systems*, Institute of Electrical and Electronics Engineers Inc., Dec. 2020. doi: 10.1109/ICCES51560.2020.9334573.

[13] D. Gaikwad, V. Yelnoorkar, A. Jadhav, and Y. Haribhakta, “Clustering Research Papers: A Qualitative Study of Concatenated Power Means Sentence Embeddings over Centroid Sentence Embeddings,” vol. 2, 2021, pp. 311–325. doi: 10.1007/978-981-33-6987-0_26.

[14] A. A. Jalal and B. H. Ali, “Text documents clustering using data mining techniques,” *International Journal of Electrical and Computer Engineering*, vol. 11, no. 1, pp. 664–670, Feb. 2021, doi: 10.11591/ijece.v11i1.pp664-670.

[15] B. Probierz, J. Kozak, and A. Hrabia, “Clustering of scientific articles using natural language processing,” in *Procedia Computer Science*, Elsevier B.V., 2022, pp. 3443–3452. doi: 10.1016/j.procs.2022.09.403.

[16] A. Solem, “Celery.” Accessed: Aug. 14, 2023. [Online]. Available: https://docs.celeryq.dev/en/stable/index.html

[17] K. S. Pratt and H. R. Bright, “Design Patterns for Research Methods: Iterative Field Research,” 2009. doi: 10.1.1.535.345.

[18] K. Haruna *et al.*, “Research paper recommender system based on public contextual metadata,” *Scientometrics*, vol. 125, no. 1, pp. 101–114, Oct. 2020, doi: 10.1007/s11192-020-03642-y.

[19] R. van Dinter, B. Tekinerdogan, and C. Catal, “Automation of systematic literature reviews: A systematic literature review,” *Information and Software Technology*, vol. 136. Elsevier B.V., Aug. 01, 2021. doi: 10.1016/j.infsof.2021.106589.

[20] S. Gonzalez-Toral, R. Freire, R. Gualan, and V. Saquicela, “A ranking-based approach for supporting the initial selection of primary studies in a Systematic Literature Review,” in *Proceedings - 2019 45th Latin American*

Computing Conference, CLEI 2019, Institute of Electrical and Electronics Engineers Inc., Sep. 2019. doi: 10.1109/CLEI47609.2019.235079.

[21] C. K. Kreutz and R. Schenkel, “Scientific paper recommendation systems: a literature review of recent publications,” *International Journal on Digital Libraries*, vol. 23, no. 4, pp. 335–369, Dec. 2022, doi: 10.1007/s00799-022-00339-w.

[22] W. M. Lim, A. Gunasekara, J. L. Pallant, J. I. Pallant, and E. Pechenkina, “Generative AI and the future of education: Ragnarök or reformation? A paradoxical perspective from management educators,” *International Journal of Management Education*, vol. 21, no. 2, 2023, doi: 10.1016/j.ijme.2023.100790.

[23] F. Qasem, “ChatGPT in scientific and academic research: future fears and reassurances,” *Library Hi Tech News*, vol. 40, no. 3, pp. 30–32, 2023, doi: 10.1108/LHTN-03-2023-0043.

[24] M. Haman and M. Školník, “Using ChatGPT to conduct a literature review,” *Account Res*, 2023, doi: 10.1080/08989621.2023.2185514.

[25] D. Goldenberg, H. Merlino, and E. Fernandez, “CATEGORIZACION AUTOMATICA DE DOCUMENTOS CON MAPAS AUTO-ORGANIZADOS DE KOHONEN,” Instituto Tecnológico de Buenos Aires - Universidad Politécnica, Madrid , 2007. Accessed: Apr. 14, 2023. [Online]. Available: https://ri.itba.edu.ar/server/api/core/bitstreams/c72fcf46-b0fe-4c87-a5e1-aefcb6eaf766/content

[26] V. C. J. Rijsbergen, *INFORMATION RETRIEVAL*. 1979.

[27] G. Mustafa, M. Usman, L. Yu, M. T. afzal, M. Sulaiman, and A. Shahid, “Multi-label classification of research articles using Word2Vec and identification of similarity threshold,” *Sci Rep*, vol. 11, no. 1, Dec. 2021, doi: 10.1038/s41598-021-01460-7.

[28] Q. Zhou, X. Chen, and C. Chen, “Authoritative Scholarly Paper Recommendation Based on Paper Communities,” in *2014 IEEE 17th International Conference on Computational Science and Engineering*, IEEE, Dec. 2014, pp. 1536–1540. doi: 10.1109/CSE.2014.284.

[29] Q. Yang *et al.*, “A novel hybrid publication recommendation system using compound information,” *World Wide Web*, vol. 22, no. 6, pp. 2499–2517, Nov. 2019, doi: 10.1007/s11280-019-00687-9.

[30] S. Tahvili and L. Hatvani, “Transformation, vectorization, and optimization,” *Artificial Intelligence Methods for Optimization of the Software Testing Process*, pp. 35–84, 2022, doi: 10.1016/B978-0-32-391913-5.00014-2.

[31] A. Amalia, M. S. Lydia, S. D. Fadilla, M. Huda, and D. Gunawan, “Document clustering optimization with synonym dictionary check function,” in *Proceedings - 2017 International Conference on Electrical Engineering and Informatics: Advancing Knowledge, Research, and Technology for Humanity, ICELTICs 2017*, 2017, pp. 286–291. doi: 10.1109/ICELTICS.2017.8253285.

[32] A. Amalia, O. S. Sitompul, E. B. Nababan, and T. Mantoro, “A comparison study of document clustering using DOC2VEC versus TFIDF combined with LSA for small corpora,” *J Theor Appl Inf Technol*, vol. 98, no. 17, pp. 3644–3657, 2020.

[33] S. Theodoridis and K. Koutroumbas, “Clustering Algorithms I: Sequential Algorithms,” *Pattern Recognit*, pp. 627–652, 2009, doi: 10.1016/B978-1-59749-272-0.50014-1.

[34] I. D. Baruah, “Cheat sheet for implementing 7 methods for selecting the optimal number of clusters in Python.” [Online]. Available: https://towardsdatascience.com/cheat-sheet-to-implementing-7-methods-for-selecting-optimal-number-of-clusters-in-python-898241e1d6ad

[35] T. Sterling, M. Anderson, and M. Brodowicz, “MapReduce,” *High Performance Computing*, pp. 579–589, Jan. 2018, doi: 10.1016/B978-0-12-420158-3.00019-8.

[36] A. Huang, “Similarity measures for text document clustering,” in *New Zealand Computer Science Research Student Conference, NZCSRSC 2008 - Proceedings*, 2008, pp. 49–56.

[37] A. Amine, Z. Elberrichi, M. Simonet, and M. Malki, “WordNet-Based and N-Grams-Based document clustering: A comparative study,” in *Proceedings - 3rd International Conference on Broadband Communications, Informatics and Biomedical Applications, BroadCom 2008*, 2008, pp. 394–401. doi: 10.1109/BROADCOM.2008.7.

[38] S. Chaudhary, “Why ‘1.5’ in IQR Method of Outlier Detection?,” Medium. [Online]. Available: https://towardsdatascience.com/why-1-5-in-iqr-method-of-outlier-detection-5d07fdc82097

[39] R. Davies, U. Ghosh-Dastidar, J. Knisley, and W. Samyono, “Toward Revealing Protein Function: Identifying Biologically Relevant Clusters With Graph Spectral Methods,” *Algebraic and Combinatorial Computational Biology*, pp. 375–409, Jan. 2019, doi: 10.1016/B978-0-12-814066-6.00012-X.

[40] A. Kumar and H. Daumé III, “A co-training approach for multi-view spectral clustering,” in *Proceedings of the 28th International Conference on Machine Learning, ICML 2011*, 2011, pp. 393–400.

[41] T. H. Sardar and Z. Ansari, “MapReduce-based Fuzzy C-means Algorithm for Distributed Document Clustering,” *Journal of The Institution of Engineers (India): Series B*, vol. 103, no. 1, pp. 131–142, 2022, doi: 10.1007/s40031-021-00651-0.

[42] D. Weiss and S. Osiński, “Carrot2 Docs.” [Online]. Available: https://carrot2.github.io/release/4.2.0/doc/choosing-clustering-algorithm/

[43] Y.-H. Tseng, “Generic title labeling for clustered documents,” *Expert Syst Appl*, vol. 37, no. 3, pp. 2247–2254, 2010, doi: 10.1016/j.eswa.2009.07.048.

[44] M. A. Al-Betar, A. K. Abasi, G. Al-Naymat, K. Arshad, and S. N. Makhadmeh, “Optimization of scientific publications clustering with ensemble approach for topic extraction,” *Scientometrics*, vol. 128, no. 5, pp. 2819–2877, 2023, doi: 10.1007/s11192-023-04674-w.

[45] Cambridge University Press & Assessment, “Cambridge Dictionary.” [Online]. Available: https://dictionary.cambridge.org/us/grammar/british-grammar/noun-phrases

[46] Heka.ai, “Labeling text clusters with keywords,” Medium. [Online]. Available: https://heka-ai.medium.com/labeling-text-clusters-with-keywords-b5b5b6c1a89e

[47] R. Campos, V. Mangaravite, A. Pasquali, A. Jorge, C. Nunes, and A. Jatowt, “YAKE! Keyword extraction from single documents using multiple local features,” *Inf Sci (N Y)*, vol. 509, pp. 257–289, Jan. 2020, doi: 10.1016/J.INS.2019.09.013.

[48] S. Brown, “The C4 model for visualising software architecture.” [Online]. Available: https://c4model.com/

[49] S. Brown, “The C4 Model for Software Architecture,” InfoQ. [Online]. Available: https://www.infoq.com/articles/C4-architecture-model/

[50] DemšarJanez, “Statistical Comparisons of Classifiers over Multiple Data Sets,” *The Journal of Machine Learning Research*, Dec. 2006, doi: 10.5555/1248547.1248548.

Anexo 7: Encuestas de satisfacción

ENCUESTA DE SATISFACCIÓN CLUSTERIZE APP

Nombre: Fecha:

__Jefferson Eduardo Campo Yule_____ _____12/09/2023_____

Email:

___jeffersoncy@unicauca.edu.co_____

[x] Estudiante [ ] Docente

¡Gracias por utilizar Clusterize! Estamos interesados en conocer tu opinión para mejorar nuestra aplicación. Por favor, tómate un momento para responder a las siguientes preguntas.

¿A nivel general cómo considera la aplicación desarrollada?

La aplicación a mi consideración tiene muy buena precisión de búsqueda ya que muestra resultados precisos relacionados con la cadena de búsqueda que ingresé.

También me gusta la interfaz de usuario ya que es intuitiva y fácil de usar puesto que ingresé la consulta de manera sencilla y pude interactuar con los resultados de manera organizada y agrupada.

Finalmente, en temas de privacidad y seguridad me parece importante que el inicio de sesión sea con la cuenta institucional, puesto que esto permite ganar confianza en sus usuarios y respetar su privacidad.

¿Qué sugerencias tiene para mejorar la aplicación?

En cuanto a la calidad de los resultados, si bien la aplicación brinda muy buenos resultados, podría permitir también el reconocer palabras en español para obtener y evaluar más resultados relevantes.

Otra sugerencia es que la aplicación sea compatible con diferentes dispositivos y tamaños de pantalla, un ejemplo de ello es la adaptabilidad a dispositivos móviles.

ENCUESTA DE SATISFACCIÓN CLUSTERIZE APP

Nombre: Julio Cesar Mellizo Hurtado

Email: mellizohurt@unicauca.edu.co

Fecha: 12/09/2023

[X] Estudiante [ ] Docente

¡Gracias por utilizar Clusterize! Estamos interesados en conocer tu opinión para mejorar nuestra aplicación. Por favor, tómate un momento para responder a las siguientes preguntas.

¿A nivel general cómo considera la aplicación desarrollada?

Es una buena aplicación que brinda una ayuda optima, sencilla y de utilidad para las personas que vayan a utilizarla. Manejar un "filtrado" apoya mucho la búsqueda de artículos los cuales en muchas ocasiones suelen ser difíciles de encontrar.

¿Qué sugerencias tiene para mejorar la aplicación?

En términos generales la aplicación la está bien, es intuitiva para el usuario y de fácil manejo. Sugeriría que la aplicación en el momento que notifica al correo si la persona tiene activa la pestaña de búsqueda también se le notifique ahí.

CLUSTERIZE APP

ENCUESTA DE SATISFACCIÓN CLUSTERIZE APP

Nombre: María Isabel Vidal

Fecha: 15 sept / 2023

Email: isaca98@gmail.com maria.vidal@uaicauca.edu.co

☐ Estudiante ☒ Docente

¡Gracias por utilizar Clusterize! Estamos interesados en conocer tu opinión para mejorar nuestra aplicación. Por favor, tómate un momento para responder a las siguientes preguntas.

¿A nivel general cómo considera la aplicación desarrollada?

En general es una buena aplicación, facil de usar util,

¿Qué sugerencias tiene para mejorar la aplicación?

1. Enviar articulos al correo - doi, titulo

ENCUESTA DE SATISFACCIÓN CLUSTERIZE APP

Nombre: Fecha:

__Nesbi Johana Campo _____ _12-09-2023_____

Email:

nesbijocampo@unicauca.edu.co____

[X] Estudiante [ ] Docente

¡Gracias por utilizar Clusterize! Estamos interesados en conocer tu opinión para mejorar nuestra aplicación. Por favor, tómate un momento para responder a las siguientes preguntas.

¿A nivel general cómo considera la aplicación desarrollada?

___considero que es una buena herramienta para buscar artículos científicos, al agruparlos por títulos facilita la búsqueda para descartar conceptos o temas poco relevantes para la investigación.

_____

_____

_____

_____

_____

_____

_____

¿Qué sugerencias tiene para mejorar la aplicación?

_ sin embargo puede mejorar en la diversidad de artículos mostrados, incluyendo más artículos de revistas indexadas, con más opciones de agrupación y el diseño podría ser más colorido

_____

CLUSTERIZE APP

ENCUESTA DE SATISFACCIÓN CLUSTERIZE APP

Nombre: Ricardo Antonio Zambrano Segura

Fecha: Septiembre 12 /2023

Email: rzambran@unicauca.edu.co

☐ Estudiante ☑ Docente

¡Gracias por utilizar Clusterize! Estamos interesados en conocer tu opinión para mejorar nuestra aplicación. Por favor, tómate un momento para responder a las siguientes preguntas.

¿A nivel general cómo considera la aplicación desarrollada?

Me parece una excelente herramienta para el desarrollo y formulación de proyectos de investigación.

¿Qué sugerencias tiene para mejorar la aplicación?

- Revisar el algoritmo de nombramiento de grupos.
- En la herramienta resaltar mejor las respuestas dadas al evaluar los grupos

Exige buen conocimiento en la construcción de cadenas de búsqueda.

CLUSTERIZE

ENCUESTA DE SATISFACCIÓN CLUSTERIZE APP

Nombre: Jisele Guacheta campo

Fecha: 15-09-2023

Email: guaseke@gmail.com

☐ Estudiante ☒ Docente

¡Gracias por utilizar Clusterize! Estamos interesados en conocer tu opinión para mejorar nuestra aplicación. Por favor, tómate un momento para responder a las siguientes preguntas.

¿A nivel general cómo considera la aplicación desarrollada?

Me parece que puede ser útil, pues la clasificación temática que posibilita y darle una jerarquización a los artículos le ayuda al investigador con esos procesos iniciales de depuración de información y clasificación para la construcción de estado del arte

¿Qué sugerencias tiene para mejorar la aplicación?

¿Que busque en otras bases diferentes a scopus?
¿Que busque en otros idiomas?
¿Que emita un informe de resultados?