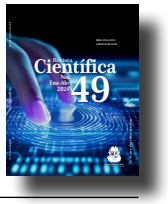
UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS

Revista Científica

http://revistas.udistrital.edu.co/ojs/index.php/revcie/index



ARTÍCULO DE REVISIÓN

Selección de artículos de investigación relevantes y no relevantes con base en resultados de Scopus y visualización por grupos de documentos

Selection of Relevant and Non-Relevant Research Articles based on Scopus Results and Visualization by Document Groups

Seleção de artigos de investigação relevantes e não relevantes com base nos resultados do Scopus e visualização por agrupamento de documentos

Juan-Fernando Campo-Mosquera[1]
Laura-Isabel Chaparro-Navia[2]
Carlos-Alberto Cobos-Lozada[3]

Recibido: octubre de 2023 **Aceptado:** diciembre de 2023

Para citar este artículo: Campo-Mosquera J. F., Chaparro-Navia, L. I. y Cobos-Lozada, C. A. (2024). Selección de artículos de investigación relevantes y no relevantes con base en resultados de Scopus y visualización por grupos de documentos. *Revista Científica*, *49*(1), 28-43. https://doi.org/10.14483/23448350.21439

Resumen

Este artículo presenta una aplicación web que busca facilitar la selección de artículos de investigación relevantes o no para una temática. El proceso inicia cuando un investigador escribe una cadena de búsqueda y esta se envía a la API de Scopus. Con los resultados obtenidos, se realiza un proceso de agrupamiento para generar una visualización por grupos o tópicos en lugar de las clásicas listas ordenadas de resultados, facilitando al usuario descartar grupos de artículos irrelevantes a su consulta. La propuesta utiliza cinco algoritmos de agrupamiento, entre los cuales *Spectral* y *K-means* obtuvieron el mejor rendimiento en métricas clásicas de recuperación de información sobre cuatro conjuntos de datos del estado del arte. La aplicación fue evaluada en dos rondas por investigadores de la Universidad del Cauca, quienes consideraron en la ronda final que el 71.4 % de los grupos tenían un buen título, el 92.9 % de los grupos tenían un buen orden de los documentos y el 65.8 % de los artículos estaban bien agrupados. Se destaca la implementación del solapamiento en el agrupamiento, pues permite a los artículos pertenecer a varios tópicos. Finalmente, los resultados son prometedores, y la aplicación constituye una valiosa contribución para los investigadores en el desarrollo de sus proyectos. Sin embargo, los resultados no son generalizables, y se evidencia la necesidad de crear mejores algoritmos de etiquetado para generar títulos más descriptivos, así como el uso de herramientas para asistir al usuario en la construcción de las consultas.

Palabras clave: agrupamiento; agrupamiento de artículos científicos; búsqueda de artículos; etiquetado de grupos; selección de artículos relevantes; solapamiento.

1. Universidad del Cauca (Popayán-Cauca, Colombia). juancamm@unicauca.edu.co.
2. Universidad del Cauca (Popayán-Cauca, Colombia). lauraich@unicauca.edu.co.
3. Ph. D. Universidad del Cauca (Popayán-Cauca, Colombia). ccobos@unicauca.edu.co.

Abstract
This paper presents a web application that seeks to facilitate the selection of research articles that are relevant or not to a topic. The process starts when a researcher writes a search string, which is sent to the Scopus API. With the results obtained, a grouping process is carried out to generate a visualization by groups or topics instead of the traditional ordered lists of results, making it easier for users to discard groups of articles irrelevant to their query. The proposal uses five clustering algorithms, among which *Spectral* and *K-means* exhibited the best performance in classical information retrieval metrics on four state of the art datasets. The application was assessed in two rounds by researchers of Universidad del Cauca, who, in the final round, considered that 71.4% of the clusters had a good title, 92.9% of the clusters had a good document order, and 65.8% of the articles were well clustered. The implementation of overlapping in grouping stands out since it allows articles to belong to several topics. Finally, the results are promising, and the application constitutes a valuable contribution for researchers in developing their projects. However, the results are not generalizable, and the need to create better labeling algorithms to generate more descriptive titles is evident, along with the use of tools to assist the user in query construction.
Keywords: article search; cluster labeling; clustering; clustering of scientific articles; overlapping; selection of relevant articles.

Resumo
Este artigo apresenta um aplicativo da Web que visa a facilitar a seleção de artigos de pesquisa relevantes ou não para um tópico. O processo começa quando um pesquisador escreve uma string de pesquisa e ela é enviada para a API do Scopus. Com os resultados obtidos, é realizado um processo de agrupamento para gerar uma visualização por grupos ou tópicos em vez das clássicas listas ordenadas de resultados, facilitando ao usuário descartar grupos de artigos irrelevantes para sua consulta. A proposta usa cinco algoritmos de agrupamento, mas o *Spectral* e o *K-means* tiveram o melhor desempenho em métricas clássicas de recuperação de informações em quatro conjuntos de dados de última geração. O aplicativo foi avaliado em duas rodadas por pesquisadores da Universidad del Cauca, onde os pesquisadores na rodada final consideraram que 71,4% dos clusters tinham um bom título, 92,9% dos clusters tinham uma boa ordem de documentos e 65,8% dos artigos estavam bem agrupados. A implementação de clustering sobreposto é destacada, pois permite que os artigos pertençam a vários tópicos. Por fim, os resultados são promissores e o aplicativo gera uma contribuição valiosa para os pesquisadores no desenvolvimento de seus projetos. No entanto, os resultados não são generalizáveis e há necessidade de melhores algoritmos de marcação para gerar títulos mais descritivos, bem como o uso de ferramentas para auxiliar o usuário na construção de consultas.
Palavras-chaves: agrupamento; agrupamento de artigos científicos; etiquetagem de grupos; pesquisa de artigos; seleção de artigos relevantes; sobreposição.

INTRODUCCIÓN

Los proyectos de investigación dan a conocer sus resultados de diferentes maneras y por diversos medios. Entre ellos, uno de los más conocidos y divulgados corresponde a los artículos científicos, donde se resumen los principales aportes de las investigaciones en diferentes áreas del conocimiento. Estos artículos se publicitan en revistas, seminarios, conferencias y congresos, por nombrar algunos (Institut Teknologi dan Bisnis et al., 2019). Por otro lado, cuando plantean proyectos o los ejecutan, los investigadores deben asegurarse de utilizar información de fuentes confiables acerca de los temas concretos de sus investigaciones, y para ello deben buscar, leer, estudiar y analizar múltiples documentos, que pueden ser

artículos de investigación o capítulos de libros, entre otros, en el marco de un proceso que generalmente implica mucho tiempo (Hanyurwimfura *et al.*, 2014). Esto, dado que la cantidad de información disponible en las bases de datos es gigante; tan solo en Scopus, al mes de junio del 2022, se reportaban más de 87 millones de documentos, y esta base de datos diariamente indexa aproximadamente 11 000 nuevos escritos (Rachel McCullough, 2022). Un ejemplo de la gran cantidad de información que se puede encontrar al realizar una consulta es el área de las ciencias de la salud, con miles o decenas de miles de resultados (Rachel McCullough, 2022).

En aras de apoyar a los investigadores, se han propuesto diversas formas de soportar un mejor proceso de búsqueda y selección de documentos. En 2011 se presentó una metodología basada en las relaciones de citación, utilizando un grafo de distancias y relaciones para comparar la relevancia de dos artículos (Liang *et al.*, 2011). En 2016 se presentó una técnica llamada *AKR*, que proporciona una lista inicial de artículos relevantes basada en palabras clave especificadas por el autor (Sesagiri Raamkumar *et al.*, 2017). En el mismo año, Rúbio y Gulo (2016) presentaron un enfoque basado en modelos de contenido y medidas bibliométricas que utiliza minería de datos y aprendizaje automático.

En 2019, se presentó el asistente inteligente FAST2, que emplea métodos basados en resúmenes y aprendizaje automático para apoyar la selección de artículos relevantes (Yu & Menzies, 2019). Además, en el mismo año se realizó un estudio sobre la recomendación de artículos científicos, encontrando que se pueden aplicar diferentes métodos como el filtrado basado en contenido, el filtrado colaborativo, métodos basados en grafos y métodos de recomendación híbrida (Bai *et al.*, 2019). También en 2019, Chen y Ban 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. Rinartha y Surya Kartika (2019) propusieron una aplicación web para el agrupamiento de artículos empleando 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 (Ahmed *et al.*, 2020). En 2021 se propuso un nuevo enfoque de agrupamiento no supervisado de artículos usando incrustación de frases, encontrando un incremento de 47.94 % en la calidad de los resultados cuando se utilizó la distancia de coseno (Gaikwad *et al.*, 2021). También en 2021, Jalal y Ali presentaron el uso de *document clustering*, TF-IDF y la similitud de cosenos para agrupar artículos en categorías significativas. En 2022 se propuso el agrupamiento de artículos usando NLP, *K-means* y ponderación basada en TF y TF-IDF (Probierz *et al.*, 2022). El uso de *K-means* en estas investigaciones resulta en grupos sin solapamiento, *i.e.*, un documento pertenece a un solo grupo, y, en el agrupamiento de documentos científicos o no, esto puede ser inadecuado, puesto que, por ejemplo, una revisión sistemática puede estar relacionada con diversos tópicos de un área de interés.

Asimismo, existen numerosas herramientas que hacen uso de inteligencia artificial y se enfocan en facilitar la búsqueda y selección de artículos científicos, *e.g.*, consensus.app, Elicit.org, Scite.ai y Research Rabbit, 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 tarda más tiempo en realizar la búsqueda de los artículos relevantes para su consulta, pues no puede desechar grupos de documentos no relevantes de manera consciente.

A pesar de los avances mencionados, aún persiste el problema para los investigadores, debido a que, al momento de consultar los artículos de investigación en bases de datos científicas, se les dificulta poder descartar de manera consciente aquellos documentos que no son relevantes y ubicar aquellos que sí lo son. Por lo tanto, en este trabajo de investigación se desarrolló una solución expresada en una aplicación

web que permite la búsqueda y selección de artículos con base en una consulta construida por el usuario. La 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. Estos grupos permiten el solapamiento de documentos –los documentos multitema como las revisiones sistemáticas pueden pertenecer a varios grupos– y contienen títulos o etiquetas que ayudan a identificar su contenido, ayudando a que el usuario pueda obtener una mejor comprensión de los grupos y su relevancia.

A continuación, se presenta un resumen de los algoritmos de agrupamiento y etiquetado que se implementaron, así como su evaluación con conjuntos de datos reales haciendo uso de métricas tradicionales del área de la recuperación de información y la opinión de algunos investigadores en relación con su uso y los resultados que brindan. Finalmente, se presentan las conclusiones de este estudio y algunas propuestas para trabajos futuros.

METODOLOGÍA

En este trabajo se usó el patrón de investigación iterativa (PII) propuesto por Pratt (2009), el cual consta de cuatro etapas principales: 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 cuatro iteraciones incrementales. En la primera iteración se analizaron algoritmos de agrupamiento, implementando cinco de ellos. Estos algoritmos se evaluaron mediante pruebas con cuatro conjuntos de datos reales y métricas de recuperación de información. La segunda iteración se centró en mejorar lo previamente desarrollado, mientras que la tercera incorporó algoritmos de etiquetado de grupos. En la última iteración se logró una versión final de la propuesta, que posteriormente se evaluó. Para la implementación de la aplicación web (desarrollo de la solución) se usó la metodología SCRUM, el *backend* se programó con Python y Django y el *frontend* se creó con Angular. La evaluación o prueba de la solución desarrollada contó con la participación de once investigadores mediante una encuesta dispuesta en el mismo aplicativo web.

Proceso de agrupamiento y etiquetado. El agrupamiento de documentos es un proceso no supervisado que tiene como objetivo agrupar un conjunto de textos sin etiquetar, de modo que los textos de un mismo grupo sean similares entre sí (Tahvili & Hatvani, 2022) y que cada grupo cuente con una etiqueta (título) que ayude al usuario a identificar lo que los documentos tienen en común. Para lograr esta tarea, se siguieron una serie de pasos que se agrupan en los siguientes macropasos: preprocesamiento de documentos, agrupamiento de documentos y etiquetado de grupos (Figura 1).

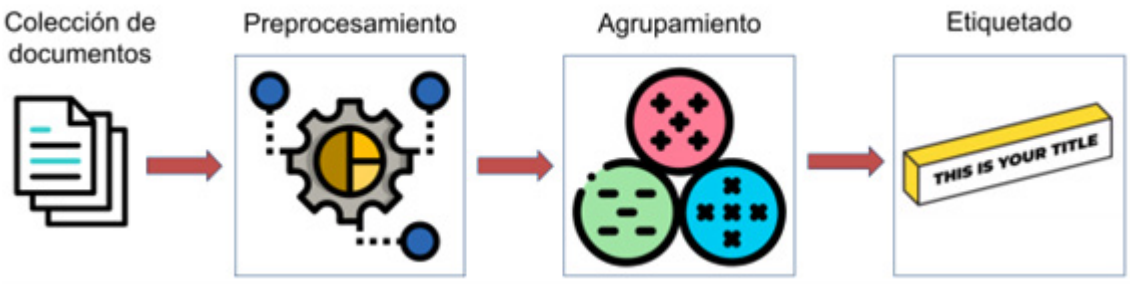


Figura 1. *Proceso de agrupamiento y etiquetado.*

Preprocesamiento de documentos. Los pasos del preprocesamiento de los documentos para los algoritmos de agrupamiento, denominados *K-means*, *spectral* y *fuzzy C-means*, se describen en la Tabla 1. Este procesamiento busca transformar la lista de documentos no estructurados o semiestructurados en una tabla estructurada que los represente de tal manera que puedan ser utilizados por los algoritmos de agrupamiento.

Además de los tres algoritmos de agrupamiento previamente mencionados, también se usaron los algoritmos *STC* y *Lingo* (Weiss & Osiński, n.d.). El preprocesamiento correspondiente a estos dos algoritmos solo tuvo en cuenta los pasos 2 y 3 de la Tabla 1, ya que estos algoritmos requieren, como entrada, la lista de documentos en su forma original.

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	División del documento en palabras o términos (tokenización)
Paso 3	Eliminación de palabras vacías o no deseadas (filtrado) (Amalia *et al.*, 2017)
Paso 4	Transformación de cada *token* (palabra en este caso) a su palabra raíz, quitando o reemplazando prefijos y sufijos (*stemming*)
Paso 5	Construcción de la matriz TF-IDF con los textos preprocesados incluyendo unigramas, bigramas y trigramas de los *tokens*
Paso 6	Representación semántica basada en LSA (*latent semantic analysis*), haciendo uso de la descomposición truncada por valores singulares (*truncated SVD*) y la matriz TF-IDF previamente construida

Agrupamiento de documentos. Los pasos de la implementación de los algoritmos de agrupamiento *spectral*, *K-means* y *fuzzy C-means* se describen en la Tabla 2. Para estos pasos, se tomó como entrada la matriz TF-IDF transformada semánticamente con LSA.

Tabla 2. *Resumen del procedimiento para los algoritmos de agrupamiento K-means, spectral y fuzzy C-means*

Paso 1	Definición de la cantidad máxima de grupos ($\max\_k$) que se van a evaluar usando una regla empírica ampliamente reportada en el estado del arte, que define $\max\_k = \sqrt{N/2}$, donde N representa el número total de documentos recuperados de Scopus.
Paso 2	Definición del número de grupos que se van a crear y mostrar al usuario (K). Este paso implica la creación de varias soluciones con dos, tres o cuatro grupos –y así sucesivamente hasta con $\max\_k$ grupos– para luego escoger cuál es el mejor valor de K con base en la calidad de las soluciones de agrupamiento obtenidas (medida a través de un índice o coeficiente). En esta investigación se utilizaron el coeficiente de Silhouette, el criterio de información bayesiano (BIC) y el índice Davies-Bouldin. Entre estos métodos, el BIC se destacó por proporcionar una cantidad de grupos equilibrada y se estableció como medida predeterminada.
Paso 3	Agrupamiento mediante *K-means* (Sterling *et al.*, 2018), *spectral* (Davies *et al.*, 2019; Kumar & Daumé III, 2011) y *fuzzy C-means* (Sardar & Ansari, 2022). Estos algoritmos fueron adaptados para utilizar la similitud del coseno en lugar de la distancia euclidiana, ya que en trabajos previos se demostró ser más efectiva para el agrupamiento de documentos (Amine *et al.*, 2008; Huang, 2008).
Paso 4	Solapamiento (solo aplica para *K-means* y *spectral*) mediante la detección de documentos atípicos dentro de cada grupo, *i.e.*, los documentos que están por fuera de la zona de pertenencia: encima del cuartil 3 (Q3) más 1.5 veces el rango intercuartil o que están por debajo del cuartil 1 (Q1) menos 1.5 veces el rango intercuartil, todo medido en similitud de cosenos al centroide de cada grupo. Se evalúa si estos documentos están en la zona de pertenencia de los otros grupos y, si es así, se incluyen en los mismos. Esto permite obtener agrupamientos con solapamiento con estos dos algoritmos.
Paso 5	Filtrado del solapamiento (solo aplica para *fuzzy C-means*). Se toma la matriz que muestra la medida en la que cada documento pertenece a los diferentes grupos. Luego, se ordenan las pertenencias de cada documento con todos los grupos y se seleccionan únicamente los grupos con los que un documento tiene mayor pertenencia, es decir, donde la sumatoria de las pertenencias con los grupos seleccionados sea de al menos el 95 % de la pertenencia total del documento.
Paso 6	Ordenamiento de los documentos en cada grupo. Para todos los algoritmos, los artículos se ordenan dentro del grupo calculando la similitud de coseno entre el artículo con el centroide del grupo al que pertenece (de mayor a menor similitud). Adicionalmente, se asigna un orden a los grupos de acuerdo con la suma de relevancias de los artículos dentro de ellos. Las relevancias de los artículos se asignan de acuerdo con el orden en que los artículos fueron retornados por Scopus.

El procedimiento correspondiente al algoritmo de agrupamiento *STC* (Weiss & Osiński, n.d.) solo incluyó los pasos 1 y 3 de la Tabla 2, teniendo en cuenta que en el paso 3 se hace uso de *STC*, mientras que, para el algoritmo *Lingo* (Weiss & Osiński, n.d.) de Carrot2, solo se requirió el paso 3. Es de notar que, para estos pasos, se toma como entrada la lista de documentos recuperados de Scopus.

Etiquetado de grupos. Asignar títulos adecuados a los grupos tiene un alto grado de dificultad, pues dichos títulos deben representar el tema principal o transversal y deben ser diferentes a los de otros grupos. Aunque a la fecha existen diferentes maneras de hacer esto, la mayoría de los métodos de extracción automática de títulos se basan en extraer los términos más significativos de los documentos de cada grupo, sin considerar la similitud que estos puedan tener con los otros títulos (Tseng, 2010). Los algoritmos seleccionados para generar los títulos o etiquetas de los grupos fueron *graph topic rank* (Heka.ai, 2023), *Yake* (Campos *et al.*, 2020), *inverse transform*, *semantic frequency* y *noun phrases*. Es de resaltar que los últimos tres fueron incorporados en la investigación debido a su relevancia en el estado del arte, pues tienen en cuenta el análisis semántico y la frecuencia de los términos en la construcción de títulos.

Los pasos del proceso de etiquetado con los algoritmos *Yake*, *semantic_frequency*, *graph topic rank* y *noun_phrases* se describen en la Tabla 3. La entrada para estos algoritmos corresponde a la lista de los documentos sin procesar.

El algoritmo *Yake* usa un enfoque no supervisado que extrae palabras clave de manera automática y con base en estadísticas del texto (Campos *et al.*, 2020). El algoritmo *semantic_frecuency* combina la frecuencia de palabras con su importancia semántica, utiliza la librería Spacy para calcular la norma del vector de cada palabra y luego normaliza los valores; finalmente, devuelve las cinco palabras más relevantes por grupo.

El algoritmo *graph topic rank* se implementó usando NetworkX y utiliza una estructura de grafo, donde los nodos representan palabras o grupos de palabras, conectados según su relevancia semántica. Estos nodos temáticos se generan mediante el agrupamiento aplicado a los grupos iniciales, y las frases clave se obtienen de los nodos mejor clasificados en términos de su peso y relación semántica (Heka.ai, 2023).

El algoritmo *noun_phrases* utiliza frases sustantivas como títulos de grupos, combina los resúmenes de los artículos en un solo texto y procesa las frases sustantivas utilizando la librería Spacy. Este algoritmo no se incorporó en la aplicación web debido a la falta de claridad de los títulos generados y a que estos eran demasiado extensos.

Tabla 3. *Resumen del procedimiento implementado para los algoritmos de etiquetado Yake, semantic_frequency, graph topic rank y noun_phrases*

Paso 1	Filtrado de los documentos por cada grupo.
Paso 2	Unión de los documentos de cada grupo en un solo documento.
Paso 3	Eliminación de las palabras usadas en la consulta de Scopus en cada documento unificado.
Paso 4	Ejecución de los algoritmos (*Yake*, *semantic_frequency*, *graph topic rank* o *noun_phrases*).
Paso 5	Ordenamiento de las palabras o frases generadas para el título de cada grupo de acuerdo con su frecuencia en los documentos correspondientes.
Paso 6	Post-procesamiento de los títulos generados por el algoritmo, eliminando palabras repetidas en los títulos de un grupo y entre los grupos.

El algoritmo *inverse_transform* usa TF-IDF y reducción de dimensionalidad con base en LSA para generar palabras clave. Calcula centroides, revierte la transformación y extrae los términos más relevantes, retornando las 10 palabras principales de cada grupo. Los pasos que se siguieron durante el proceso de

etiquetado con el algoritmo de *inverse_transform* se describen en la Tabla 4. La entrada para este algoritmo también corresponde a la lista de los documentos sin procesar.

Tabla 4. *Resumen del procedimiento para el algoritmo de etiquetado inverse_transform*

Paso 1	Eliminación de las palabras usadas en la consulta de Scopus en cada documento unificado.
Paso 2	Construcción de una matriz TF-IDF con la lista de documentos usando solamente trigramas.
Paso 3	Ejecución del algoritmo *inverse_transform*.
Paso 4	Ordenamiento de las palabras o frases generadas para el título de cada grupo de acuerdo con su frecuencia en los documentos correspondientes.
Paso 5	Post-procesamiento de los títulos generados por el algoritmo, eliminando palabras repetidas en los títulos de un grupo y entre grupos.

Métricas de evaluación con conjuntos de datos del estado del arte. Los algoritmos de agrupamiento se evaluaron utilizando métricas del área de recuperación de información, a saber: precisión, recuerdo, exactitud promedio ponderada (*accuracy*) y medida F. Para ello, se promediaron los resultados obtenidos de 31 ejecuciones independientes de los algoritmos, con el fin de obtener un valor promedio apropiado con base en el teorema del límite central.

Para llenar la matriz de confusión, primero se calculó la similitud de los centroides de las clases predichas y los centroides de las clases reales (*centroid linkage*), y se asignó a cada clase predicha la clase real con la que tuvo mayor similitud. Luego, se procedió a contar los aciertos y errores en una adaptación de la matriz de confusión tradicional. Una vez construida esta matriz, se procedió a calcular las métricas mencionadas anteriormente.

La matriz de confusión tradicional se adaptó para contar solapamientos teniendo en cuenta que un artículo puede pertenecer a varios grupos. Para lograr esto, primero se creó una matriz de n*m, donde n corresponde a la cantidad de clases predichas (grupos asignados a un artículo dado por el algoritmo de agrupamiento) +1 (esta dimensión adicional se denomina *huérfana*) y m corresponde a la cantidad de clases reales (grupos asignados a un artículo, dados por el conjunto de datos) +1 (también huérfana). Es de resaltar que las clases reales huérfanas se presentan al intersecar las clases reales con las clases predichas para un artículo, y queda una asignación de clase real fuera de la intersección (+1 en la coordenada [última fila (cantidad de clases predichas)] [clase real]). En caso contrario, las clases predichas huérfanas se dan cuando, al hacer la intersección, quedan clases predichas fuera de la misma, mas no clases reales (+1 en la coordenada [clase predicha] [última columna (cantidad de clases predichas)]). Para ilustrar este concepto, la Tabla 5 presenta un ejemplo con las Clases Reales = [[1, 2], [3], [4, 5]] y las Clases Predichas = [[1, 2], [2, 3], [4]] para tres documentos.

Tabla 5. *Construcción matriz de confusión*

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

En este ejemplo se aprecia la matriz de confusión construida. Aquí, se tomaron los valores de cada una de las clases (predichas y reales) y se compararon para contar los aciertos y errores resultantes. Para ello, de la primera posición (primer documento), se tomaron las clases reales [1, 2] junto con la primera posición de las clases predichas [1, 2], y se compararon los valores que estas contenían, confirmando dos aciertos, ya que en ambas clases coinciden los grupos asignados al artículo. Por lo tanto, la posición [1, 1] de la matriz se incrementó en 1, al igual que la posición [2, 2]. Después se procedió a la siguiente posición de los vectores de clases reales y predichas, *i.e.*, [3] y [2, 3], observando solo un acierto para ese documento, dado que ambas clases coincidían en el grupo 3. Por lo tanto, se aumentó en 1 en la posición [3, 3] de la matriz. En ese mismo documento se puede observar que el artículo fue erróneamente asignado al grupo 2 respecto a las clases predichas, por lo que se contabilizó el error en la posición 2 de las clases predichas y el huérfano en las clases reales (posición [2, H]). De la misma manera, se siguieron contando los aciertos y errores para el tercer documento.

Conjuntos de datos para la evaluación y comparación. Para la selección de los conjuntos de datos, se verificó que estos contaran con artículos científicos y, en especial, que contaran por lo menos con el resumen del artículo y la clase (grupo ideal) a la cual pertenecía cada artículo. Para realizar la evaluación, se seleccionaron cuatro conjuntos de datos, dos de los cuales se obtuvieron a partir de un artículo del estado del arte (Ahmed *et al.*, 2020). Los otros dos se hallaron a través de una búsqueda en la web.

Los conjuntos de datos utilizados fueron AAAI 2014 Accepted Papers y AAAI 2013 Accepted Papers, 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, y el segundo se llama arXiv Dataset.

Implementación de la REST API y la aplicación web. Para el desarrollo de la REST API, primero se crearon diferentes diagramas, comenzando por el de base de datos y los del modelo C4 (Brown, n.d.). La Figura 2 muestra el diagrama de componentes, que ilustra la estructura completa de la API REST.

Para la implementación de la API REST, se utilizaron Django y Django Rest Framework, y para la ejecución de tareas asíncronas se usó Celery y Redis. El proyecto se dividió en cinco módulos: gestión de los artículos, agrupamiento, gestión de la evaluación, gestión de consultas y gestión de usuarios. 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 el desarrollo del *frontend*, se usó Angular estructurado en seis componentes: *navbar, loader, login, home, results* y *profile*.

Es de resaltar que los resultados de una consulta se obtienen mediante peticiones a la API de Scopus, donde la información de cada artículo corresponde al título, el resumen y las palabras clave. Esta base de datos se seleccionó porque que es comúnmente usada para la búsqueda de información por parte de diversos investigadores en el mundo y porque su contenido tiene filtros de calidad. Además, esta investigación está enfocada en usuarios de la Universidad del Cauca, una institución con suscripción a Scopus, lo cual facilita el acceso.

Para usar *Clusterize* (nombre dado a la aplicación web desarrollada), luego del proceso de autenticación y autorización de ingreso, se debe ingresar una consulta igual que en Scopus (Figura 3). La aplicación tiene varias opciones de búsqueda, así como un filtro de rango de fechas. Una vez que se ha construido la consulta y se presiona la lupa, se ejecuta la búsqueda en Scopus. Acto seguido, la aplicación informa cuántos artículos se han encontrado, y el usuario confirma si desea continuar con el proceso de agrupamiento o modificar la consulta. Una vez se ha confirmado la búsqueda, se realiza el proceso de agrupamiento y, cuando los resultados están disponibles, se envía un correo electrónico al usuario para informarle. El usuario puede acceder a los resultados directamente desde la sección de Resultados (ver Figura 4).

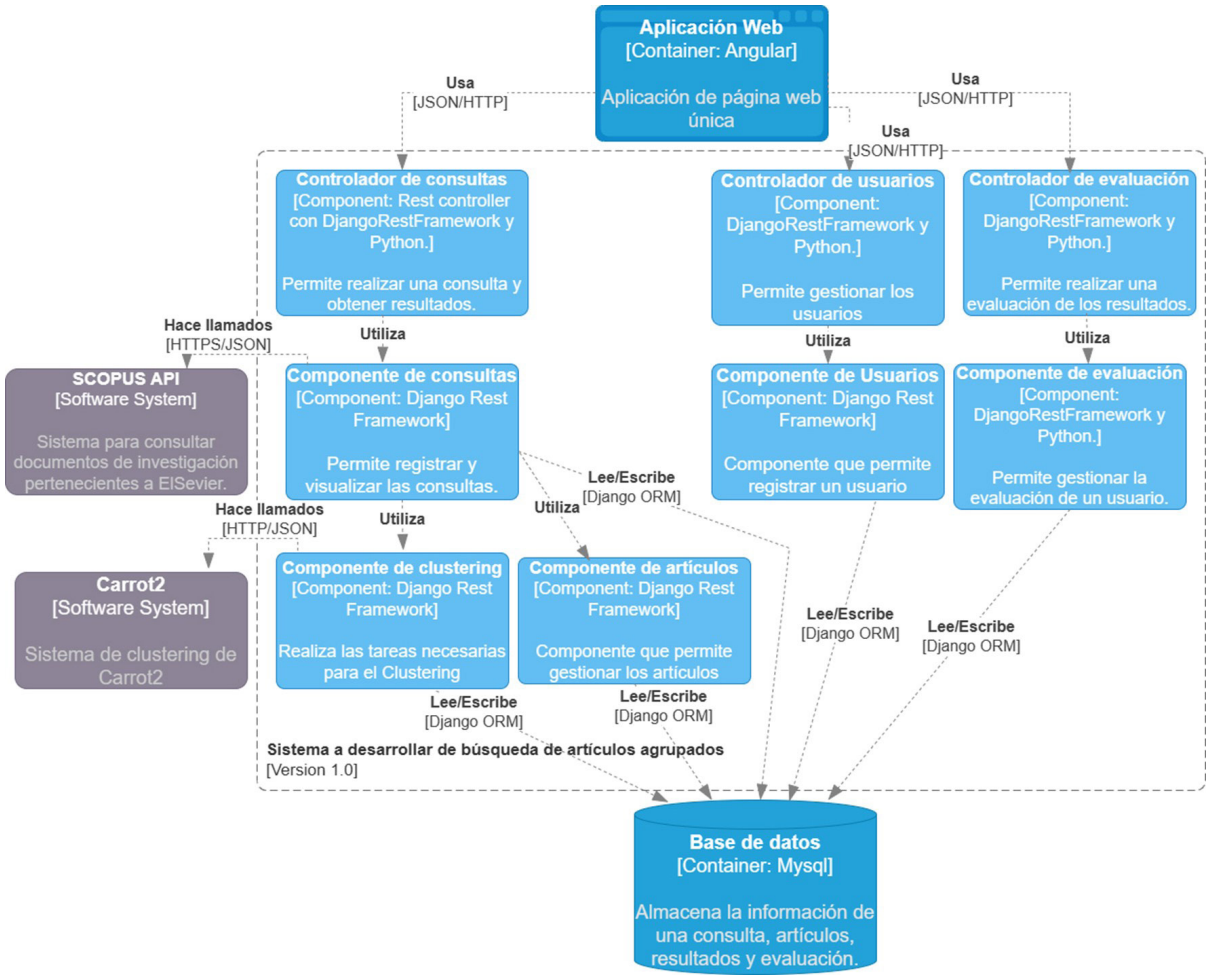


Figura 2. *Diagrama de componentes*

RESULTADOS

Se utilizaron dos formas para evaluar la propuesta desarrollada. La primera consistió en evaluar el comportamiento de los algoritmos de agrupamiento con métricas clásicas como la precisión, el recuerdo, la medida F y la exactitud. La segunda forma de evaluación consistió en determinar el nivel de satisfacción de un conjunto de investigadores de la Universidad del Cauca mediante tres preguntas que se encuentran en el aplicativo web, a saber: *¿cómo considera el título del grupo?*, *¿cómo considera el orden de los artículos dentro de este grupo?*, *¿considera que el artículo pertenece a este grupo?*

Evaluación con métricas clásicas. Para realizar esta evaluación, se usaron los cuatro conjuntos de datos previamente mencionados: AAAI13 con 150 artículos, AAAI14 con 396 artículos, Arxiv con 10 000 artículos y Topic Modeling con 14 004 artículos.

Los resultados de la métrica de precisión (Tabla 6) muestran que, en dos de los cuatro conjuntos de datos (AAAI14 y Arxiv), el algoritmo *spectral* es el mejor. Para el conjunto de datos AAAI13, el mejor valor de precisión se obtuvo con *K-means* y, para Topic Modeling, el mejor valor se obtuvo con *fuzzy C-means*.

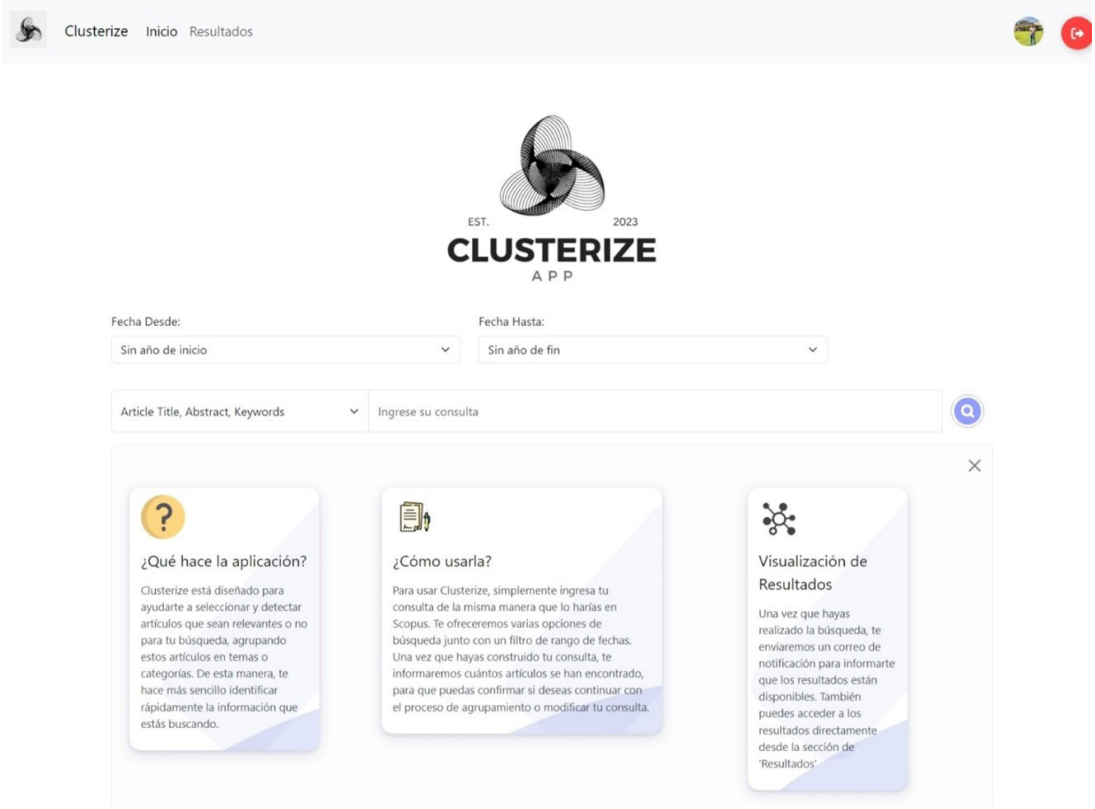


Figura 3. *Vista de la página principal*

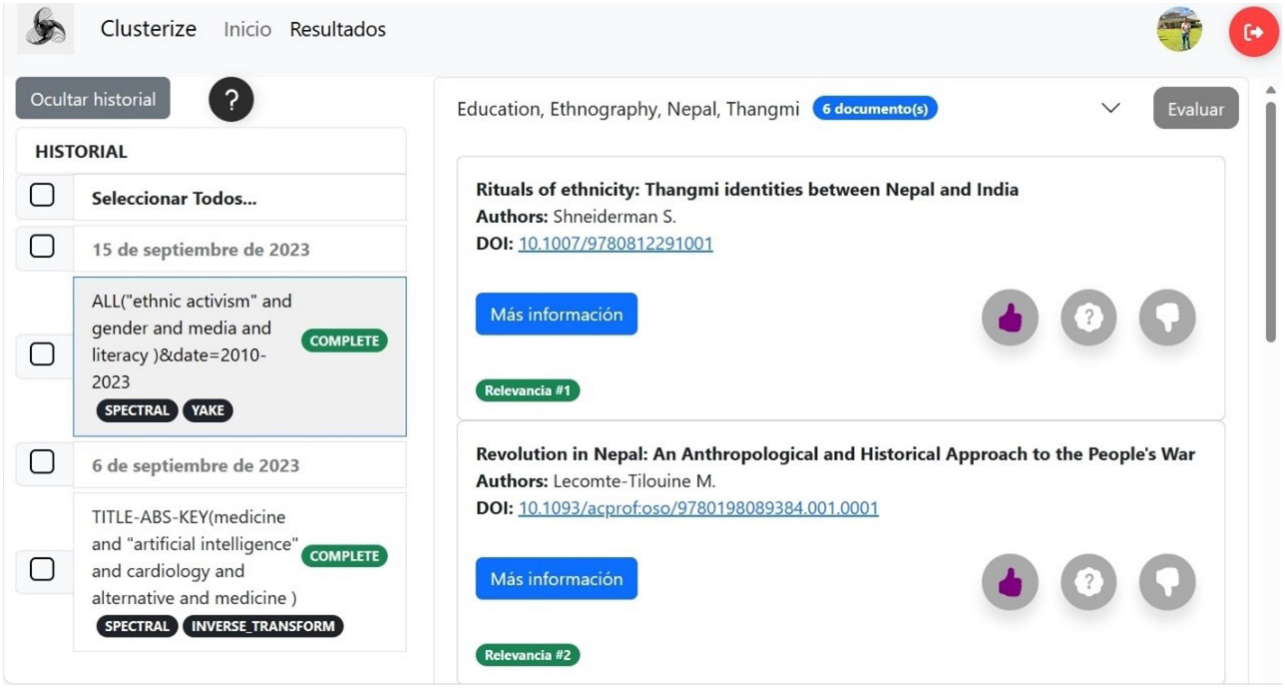


Figura 4. *Vista de resultados*

Tabla 6. *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

Los resultados de la métrica de recuerdo (Tabla 7) muestran que el algoritmo *spectral* obtuvo los valores más altos en tres de los conjuntos de datos (AAAI13, AAAI14 y Arxiv), mientras que *fuzzy C-means* reporta el valor más alto para Topic Modeling.

Tabla 7. *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

Los resultados de la medida F (Tabla 8) muestran que el algoritmo *spectral* obtuvo mejores valores en dos de los cuatro conjuntos de datos (AAAI14 y Arxiv). Para el conjunto de datos AAAI13, el mejor valor corresponde a *K-means* y, para Topic Modeling, el algoritmo *fuzzy C-means* reporta el mejor desempeño.

Tabla 8. *Resultados de la Métrica de medida F*

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

Los resultados de la métrica de exactitud (Tabla 9) muestran que, en dos de los conjuntos de datos (AAAI13 y Arxiv), el algoritmo *spectral* obtuvo los valores más altos. En AAAI14, *K-means* obtuvo el valor más alto, y *fuzzy C-means* reporta los mejores valores en Topic Modeling.

Tabla 9. *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 términos generales, *spectral* y *K-means* destacan por su buen rendimiento en términos de precisión, recuerdo, medida F y exactitud en varios conjuntos de datos, mostrando fortaleza con diferentes tamaños de datos. *Fuzzy C-means* y *Lingo* también ofrecen resultados competitivos, mientras que *STC* muestra un desempeño inferior en comparación con los demás algoritmos. Además, se puede observar que *fuzzy C-means* presenta los valores más altos en todas las métricas para el conjunto de datos más grande (Topic Modeling).

Por otro lado, debido a que *STC* y *Lingo* son algoritmos que se utilizan desde una API de Carrot2 y no se modificó su funcionamiento, su calidad se comparó con la del resto de los algoritmos implementados (*spectral, K-means, fuzzy C-means*), cuyo código fuente sí se modificó: se incluyó un índice de calidad para la selección de la mejor solución, y los datos que usaron habían sido procesados. Al realizar esta comparación, se observa que *spectral*, *K-means* y *fuzzy C-means* presentan un mejor comportamiento, pues tienen los valores más altos en todas las métricas evaluadas.

Al realizar un análisis puntual de 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. Por otro lado, 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*, todos los valores son bajos, lo que significa que en los grupos faltaron artículos. 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, en términos generales, clasifican correctamente la mayoría de las instancias, aunque puede que los valores altos representen solo a la clase mayoritaria (*i.e.*, desbalanceo de datos en la variable de clase).

Con los resultados de cada una de las métricas (promedios de las 31 repeticiones independientes de cada experimento) se realizó la prueba estadística no paramétrica de Friedman usando la herramienta Keel (*knowledge extraction based on evolutionary learning*) (Demšar, 2006). Con ella, se pudo observar que en relación con la precisión hay un empate por el primer puesto entre *spectral* y *K-means*. La siguiente posición es de *fuzzy C-means*, seguido de *Lingo* y *STC*, si bien este hallazgo es solo informativo y no es estadísticamente significativo.

Por otro lado, respecto a la métrica de recuerdo, el primer puesto del *ranking* pertenece a *spectral*. Este dato sí es estadísticamente concluyente y, según el test de Holm, se puede afirmar que *spectral* es mejor que *STC* y *Lingo* con un 95 % de confianza. En la medida F se pudo observar que hay un empate entre *K-means*, *spectral* y *fuzzy C-means*, siendo este *ranking* informativo mas no estadísticamente significativo. Finalmente, se pudo observar que en la métrica de exactitud, *spectral* tiene el primer lugar del *ranking*, siendo este estadísticamente significativo. Según el *post hoc* de Holm, este algoritmo es mejor que *STC* con un 95% de confianza.

Evaluación con investigadores. Se realizaron dos rondas de evaluación del comportamiento de la aplicación, con un total de 11 usuarios. Con los resultados y las sugerencias obtenidos en la primera ronda con cinco evaluadores (evaluación inicial), se aplicaron mejoras que estuvieran dentro del alcance del proyecto, y, con ellas, se realizó una ronda de evaluación final, que contó con otros seis usuarios (evaluación final). Las evaluaciones se realizaron haciendo uso únicamente del algoritmo de agrupamiento *spectral* y del algoritmo de etiquetado *Yake*. La elección del algoritmo de agrupamiento se basó en los resultados de las métricas clásicas, y *Yake* se seleccionó a partir de un análisis visual, donde se observó la longitud de los títulos entregados, cuán significativos eran y si eran entendibles. A criterio subjetivo de los autores de la investigación, *Yake* entregó títulos cortos y más informativos que los otros algoritmos. A continuación, se muestran los resultados de la evaluación final.

Según la Figura 5, los evaluadores consideraron que el 71.4 % de los grupos tiene un buen título, el 21.4 % tiene títulos regulares y el 7.1 % tiene un mal título. Respecto al orden de los artículos dentro de cada grupo, se consideró que el 92.9 % de los grupos tenía un buen orden, solo un 7.1 % de los artículos no estaba bien ordenado y ninguno de los grupos se consideró mal ordenado.

En la Figura 6 se puede apreciar que, según los evaluadores, el 65.8 % de los artículos sí estaban bien agrupados, es decir, sí pertenecían al grupo asignado. Además, los participantes estuvieron inseguros sobre la pertenencia del 16.4 % de los artículos y, para ellos, el 17.8 % de los artículos no pertenecía al grupo asignado.

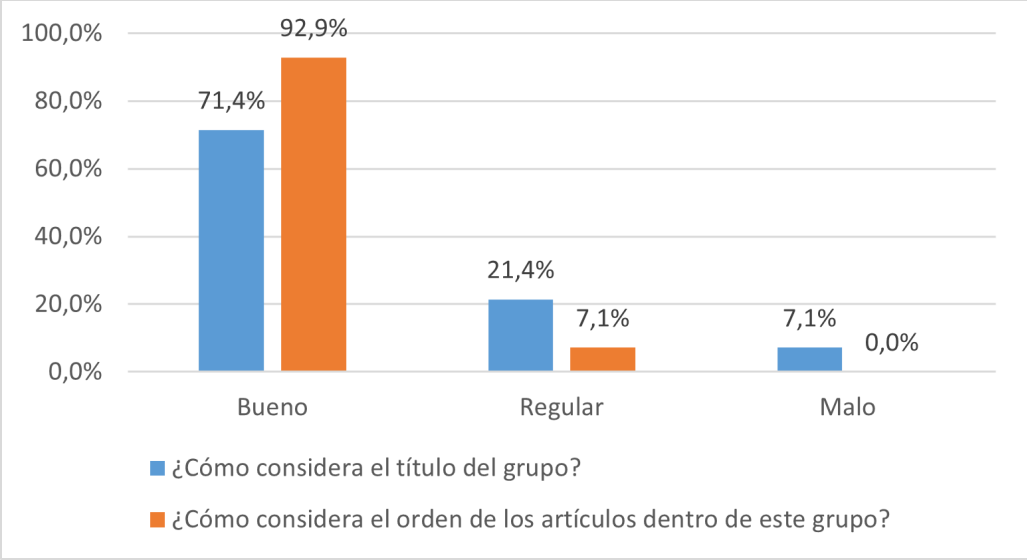


Figura 5. *Porcentaje promedio de grupos para la evaluación final*

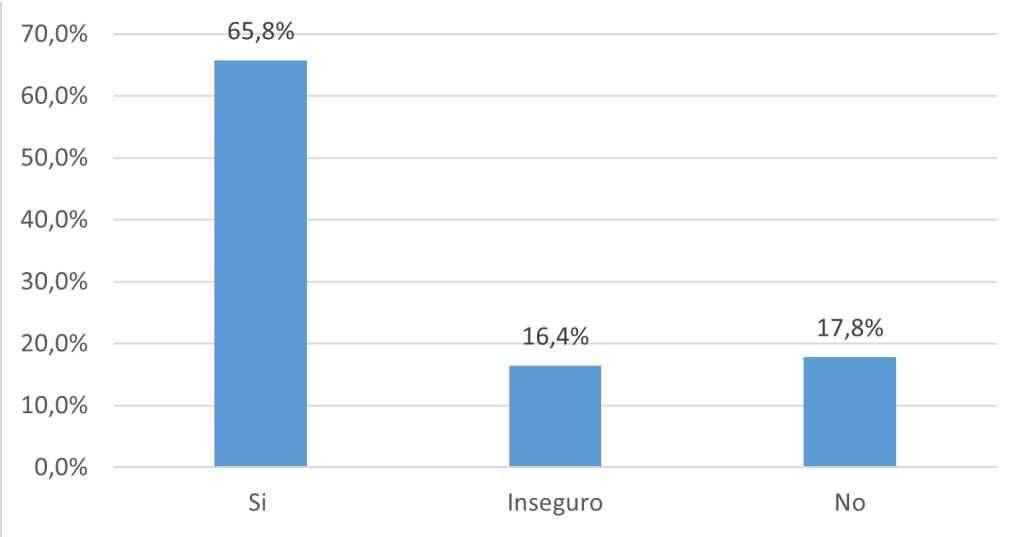


Figura 6. *Porcentaje promedio de artículos para la evaluación final*

CONCLUSIONES

Los resultados de la evaluación de la aplicación desarrollada, y en especial el uso de *spectral* como algoritmo de agrupamiento y de *Yake* como algoritmo de etiquetado, muestran que la aplicación desarrollada (*Clusterize*) constituye un aporte valioso para los investigadores, específicamente en el desarrollo de revisiones de literatura y en la actualización de trabajos relacionados al momento de elaborar una publicación. Esta aplicación ayuda a seleccionar grupos de documentos relevantes y a descartar los no relevantes de manera consciente y con una curva de aprendizaje corta.

Se determinó, mediante una evaluación por conjuntos de datos del estado del arte y la valoración de usuarios investigadores, que la aplicación agrupa apropiadamente los artículos haciendo uso de *spectral*, aunque se debe mejorar en términos del número de grupos que se crean. En cuanto a la generación de títulos con el algoritmo *Yake*, se reporta un buen funcionamiento; se generan títulos cortos y fáciles de entender. Sin embargo, los títulos son muy genéricos o contienen palabras que no aportan claridad al contenido de cada grupo en el contexto de la búsqueda.

Debido a las dificultades que se presentaron al momento de asignar títulos a los grupos, por la complejidad de encontrar títulos representativos, comprensibles y disyuntivos, como trabajo futuro se propone obtener primero las etiquetas más importantes en el conjunto de documentos y definir de estas cuáles son las más apropiadas para cada grupo. Además, como alternativa para mejorar el etiquetado, se sugiere probar el uso de modelos de lenguaje grandes (*large language models*, LLM) como ChatGPT, o desarrollar algoritmos que suministren títulos que sean mutuamente excluyentes entre los grupos generados, de modo que se pueda distinguir más fácilmente su contenido.

De igual manera, también se sugiere evaluar el uso de LLM para que el usuario realice preguntas en lenguaje natural sobre los documentos encontrados en un grupo y que el LLM responda basándose únicamente en la información de ese grupo. Para ello, también se hace necesario recolectar el texto completo de los artículos.

Con base en las sugerencias obtenidas en la evaluación con los investigadores, se propone la integración de un componente que asista a los usuarios en la construcción de las consultas, haciendo uso, por ejemplo, de la estrategia PICOT (*population, intervention, comparison, outcome, time period*), que se utiliza ampliamente en el campo de la salud pero que reciente se ha utilizado en otras áreas del conocimiento. Esto, además de incluir un proceso iterativo de mejora de la consulta con base en la visualización de los resultados de cada iteración.

Se necesita desplegar y evaluar la aplicación propuesta en un entorno con un mayor número de usuarios, y, con base en resultados positivos, posibilitar su crecimiento mediante la integración de otras bases de datos, no solo Scopus. Es preciso comentar que, en Colombia, un gran número de universidades tienen acceso a Scopus y otras bases de datos científicas por medio de Consortia, un consorcio nacional que busca potenciar el acceso a los artículos y productos de investigación e innovación, entre otros.

AGRADECIMIENTOS

Agradecemos a la Universidad del Cauca por financiar parcialmente el desarrollo de esta investigación.

CONTRIBUCIÓN DE AUTORÍA

Juan-Fernando Campo-Mosquera: investigación, metodología, escritura-borrador original.
Laura-Isabel Chaparro-Navia: investigación, metodología, escritura-borrador original.
Carlos-Alberto Cobos-Lozada: investigación, metodología, escritura-revisión y edición.

REFERENCIAS

Ahmed, R. F. M., Salama, C., Mahdi, H. (2020). *Clustering research papers using genetic algorithm optimized self-organizing maps* [Presentación en conferencia]. En 15th International Conference on Computer Engineering and Systems, Cairo, Egipto. https://doi.org/10.1109/ICCES51560.2020.9334573

Amalia, A., Lydia, M. S., Fadilla, S. D., Huda, M., Gunawan, D. (2017). *Document clustering optimization with synonym dictionary check function* [Presentación en conferencia]. En International Conference on Electrical Engineering and Informatics: Advancing Knowledge, Research, and Technology for Humanity, Banda Aceh, Indonesia. https://doi.org/10.1109/ICELTICS.2017.8253285

Amine, A., Elberrichi, Z., Simonet, M., Malki, M. (2008). *WordNet-based and N-Grams-based document clustering: A comparative study* [Presentación en conferencia]. En 3rd International Conference on Broadband Communications, Informatics and Biomedical Applications, Pretoria, Sudáfrica. https://doi.org/10.1109/broadcom.2008.7

Bai, X., Wang, M., Lee, I., Yang, Z., Kong, X., Xia, F. (2019). Scientific paper recommendation: A survey. *IEEE Access*, *7*, 9324-9339. https://doi.org/10.1109/access.2018.2890388

Brown, S. (n.d.). *The C4 model for visualising software architecture*. https://c4model.com/

Campos, R., Mangaravite, V., Pasquali, A., Jorge, A., Nunes, C., Jatowt, A. (2020). YAKE! Keyword extraction from single documents using multiple local features. *Information Sciences*, *509*, 257-289. https://doi.org/10.1016/j.ins.2019.09.013

Chen, J., Ban, Z. (2019). Academic paper recommendation based on clustering and pattern matching. En K. Knight, C. Zhang, G. Holmes & M.-L. Zhang (Eds.), *Second CCF International Conference, ICAI 2019* (pp. 171-182). Springer. https://doi.org/https://doi.org/10.1007/978-981-32-9298-7

Davies, R., Ghosh-Dastidar, U., Knisley, J., Samyono, W. (2019). Toward revealing protein function: Identifying biologically relevant clusters with graph spectral methods. En R. Robeva & M. Macauley (Eds.), *Algebraic and Combinatorial Computational Biology* (pp. 375-409). Elsevier. https://doi.org/10.1016/B978-0-12-814066-6.00012-X

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research, 7*, 1-30.

Gaikwad, D., Yelnoorkar, V., Jadhav, A., Haribhakta, Y. (2021). Clustering research papers: A qualitative study of concatenated power means sentence embeddings over centroid sentence embeddings. En S. M. Thampi, E. Gelenbe, M. Atiquzzaman, V. Chaudhary, & K. C. Li (Eds.), *Advances in Computing and Network Communications* (pp. 311-325). Springer. https://doi.org/10.1007/978-981-33-6987-0_26

Hanyurwimfura, D., Bo, L., Njagi, D., Dukuzumuremyi, J. P. (2014). A centroid and relationship based clustering for organizing research papers. *International Journal of Multimedia and Ubiquitous Engineering*, *9*(3), 219-233. https://doi.org/10.14257/ijmue.2014.9.3.21

Heka.ai. (2023). *Labeling text clusters with keywords*. https://heka-ai.medium.com/labeling-text-clusters-with-keywords-b5b5b6c1a89e

Huang, A. (2008). *Similarity measures for text document clustering* [Presentación en conferencia]. En New Zealand Computer Science Research Student Conference, Nueva Zelanda.

Intitut Teknologí dan Bisnis et al. (2019). Proceedings, International Conference on Cybernetics and Intelligent System. https://doi.org/10.1109/ICORIS46391.2019

Jalal, A. A., Ali, B. H. (2021). Text documents clustering using data mining techniques. *International Journal of Electrical and Computer Engineering*, *11*(1), 664-670. https://doi.org/10.11591/ijece.v11i1.pp664-670

Kumar, A., Daumé III, H. (2011). *A co-training approach for multi-view spectral clustering*. [Presentación en conferencia]. En 28th International Conference on Machine Learning, Bellevue, WA, USA.

Liang, Y., Li, Q., Qian, T. (2011). Finding relevant papers based on citation relations. En H. Wang, S. Li, S. Oyama, X. Hu & T. Qian (Eds.) *Web-Age Information Management, WAIM 2011* (pp. 403-414. Springer. https://doi.org/10.1007/978-3-642-23535-1_35

Pratt, K. S. (2009). *Design patterns for research methods: Iterative field research*. https://www.kpratt.net/wp-content/uploads/2009/01/research_methods.pdf

Probierz, B., Kozak, J., Hrabia, A. (2022). Clustering of scientific articles using natural language processing. *Procedia Computer Science*, *207*, 3443-3452. https://doi.org/10.1016/j.procs.2022.09.403

Rachel M. (2022). *Scopus Roadmap: What's New in 2022?* https://blog.scopus.com/posts/scopus-roadmap-whats-new-in-2022.

Rinartha, K., Surya Kartika, L. G. (2019). *Scientific article clustering using string similarity concept* [Presentación en conferencia]. En 1st International Conference on Cybernetics and Intelligent System, Denpasar, Indonesia. https://doi.org/10.1109/icoris.2019.8874879

Rúbio, T. R., Gulo, C. A. (2016). *Enhancing academic literature review through relevance recommendation using bibliometric and text-based features for classification* [Presentación en conferencia]. En 11th Iberian Conference on Information Systems and Technologies, Gran Canaria, España. https://doi.org/10.1109/cisti.2016.7521620

Sardar, T. H., Ansari, Z. (2022). MapReduce-based fuzzy C-means algorithm for distributed document clustering. *Journal of The Institution of Engineers (India): Series B*, *103*(1), 131-142. https://doi.org/10.1007/s40031-021-00651-0

Sesagiri Raamkumar, A., Foo, S., Pang, N. (2017). Using author-specified keywords in building an initial reading list of research papers in scientific paper retrieval and recommender systems. *Information Processing and Management*, *53*(3), 577-594. https://doi.org/10.1016/j.ipm.2016.12.006

Sterling, T., Anderson, M., Brodowicz, M. (2018). MapReduce. En T. Sterling, M. Anderson & M. Brodowicz (Eds.), *High Performance Computing* (pp. 579-589). Elsevier. https://doi.org/10.1016/B978-0-12-420158-3.00019-8

Tahvili, S., Hatvani, L. (2022). *Artificial intelligence methods for optimization of the software testing process*. Elsevier. https://doi.org/10.1016/B978-0-32-391913-5.00014-2

Tseng, Y.-H. (2010). Generic title labeling for clustered documents. *Expert Systems with Applications*, *37*(3), 2247-2254. https://doi.org/10.1016/j.eswa.2009.07.048

Weiss, D., Osiński, S. (n.d.). *Carrot2 Docs*. https://carrot2.github.io/release/4.2.0/doc/choosing-clustering-algorithm/

Yu, Z., Menzies, T. (2019). FAST2: An intelligent assistant for finding relevant papers. *Expert Systems with Applications*, *120*, 57-71. https://doi.org/10.1016/j.eswa.2018.11.021