UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CARRERA DE INGENIERÍA EN SOFTWARE

TEMA

DESARROLLO DE UNA APLICACIÓN WEB BASADA EN APRENDIZAJE AUTOMÁTICO PARA LA AUTOMATIZACIÓN DE LA SELECCIÓN DE LITERATURA Y SÍNTESIS DE DATOS EN LA ELABORACIÓN DE ARTÍCULOS DE REVISIÓN SISTEMÁTICA.

TRABAJO DE GRADO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERÍA EN SOFTWARE

AUTOR: Rivaldo Alexander Sánchez Reyes

DIRECTOR: MSc. Julio Esteban Guerra Masson

Ibarra – Ecuador

2025

UNIVERSIDAD TÉCNICA DEL NORTE

DIRECCIÓN DE BIBLIOTECA

1. IDENTIFICACIÓN DE LA OBRA

En cumplimiento del Art. 144 de la Ley de Educación Superior, hago la entrega del presente trabajo a la Universidad Técnica del Norte para que sea publicado en el Repositorio Digital Institucional, para lo cual pongo a disposición la siguiente información:

DATOS DE CONTACTO			
CÉDULA DE IDENTIDAD:	1004281174		
APELLIDOS Y NOMBRES:	Sánchez Reyes Rivaldo Alexander		
DIRECCIÓN:	Tanguarín, calle Alejandro López		
EMAIL:	rasanchezr@utn.edu.ec		
TELÉFONO FIJO:	062933331	**TELÉFONO MÓVIL:**	0939702502

DATOS DE LA OBRA	
TÍTULO:	Desarrollo de una aplicación web basada en aprendizaje automático para la automatización de la selección de literatura y síntesis de datos en la elaboración de artículos de revisión sistemática.
AUTOR (ES):	Sánchez Reyes Rivaldo Alexander
FECHA: DD/MM/AAAA	31/07/2025
SOLO PARA TRABAJOS DE GRADO	
PROGRAMA:	■ **PREGRADO** ☐ **POSGRADO**
TITULO POR EL QUE OPTA:	Ingeniero en Software
ASESOR /DIRECTOR:	MSc. Julio Esteban Guerra Masson

2. CONSTANCIAS

El autor manifiesta que la obra objeto de la presente autorización es original y se la desarrolló, sin violar derechos de autor de terceros, por lo tanto, la obra es original y que es el titular de los derechos patrimoniales, por lo que asume la responsabilidad sobre el contenido de la misma y saldrá en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 04 días del mes de agosto de 2025

AUTOR:

.....

Sánchez Reyes Rivaldo Alexander

C.C. 1004281174

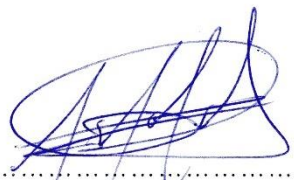
CERTIFICACIÓN DEL DIRECTOR

Ibarra, 31 de julio de 2025

Por medio del presente, yo MSc. Julio Esteban Guerra Masson, certifico que el Sr. Rivaldo Alexander Sánchez Reyes, portador de la cédula de identidad Nro. 1004281174 ha trabajado en el desarrollo del proyecto de grado "DESARROLLO DE UNA APLICACIÓN WEB BASADA EN APRENDIZAJE AUTOMÁTICO PARA LA AUTOMATIZACIÓN DE LA SELECCIÓN DE LITERATURA Y SÍNTESIS DE DATOS EN LA ELABORACIÓN DE ARTÍCULOS DE REVISIÓN SISTEMÁTICA.", previo a la obtención del título de Ingeniero en Software, lo cual ha realizado en su totalidad con responsabilidad.

Es todo cuanto puedo certificar en honor a la verdad.

Atentamente,



MSc. Julio Esteban Guerra Masson

DIRECTOR DE TRABAJO DE GRADO

DEDICATORIA

Dedico este trabajo de grado a mis padres, cuyo amor incondicional, apoyo inquebrantable y guía fueron los pilares para mi crecimiento personal y académico. El ejemplo que me dieron sobre el esfuerzo y la perseverancia han sido una fuente constante de inspiración en mi vida.

A mi hermano y familia que me dieron su confianza y me dieron su apoyo y me acompañaron en este largo camino.

También dedico este logro a mis profesores y amigos con quienes compartí altos y bajos nos dimos la mano para poder mejorar personal y académicamente.

AGRADECIMIENTO

Con profunda gratitud, quiero expresar mi más sincero agradecimiento a todas las personas que hicieron posible la culminación de este camino y mi trabajo de grado.

En primer lugar, a mis padres, cuyo amor incondicional, apoyo inquebrantable y sabios consejos han sido fundamentales en cada paso de mi vida. Su ejemplo de esfuerzo y perseverancia me han inspirado de seguir adelante incluso en mis momentos más difíciles.

A mi hermano y familia, por su confianza en mí, por alentarme cuando más lo necesitaba y por acompañarme en este largo camino.

A mis profesores, quienes me brindaron sus conocimientos con paciencia y dedicación, motivándome a dar siempre lo mejor de mí.

A mis amigos, con quien compartimos retos alegrías y momentos de aprendizaje.

A todos, gracias por ser parte de este logro.

ÍNDICE DE CONTENIDO

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

RESUMEN

Este proyecto está centrado en la creación de una aplicación web para el filtrado de textos científicos sirviendo de apoyo a los investigadores para realizar las indagaciones. El objetivo es optimizar la búsqueda y selección de textos científicos en la creación de documentos de revisión sistemática, siendo una muestra de lo que se puede lograr con la tecnología actual.

Para el desarrollo de la aplicación web se utilizó tecnologías como: Python y React usando la norma ISO 25010. El desarrollo de la aplicación se basó en la metodología Kanban y el proceso metodológico KDD, usando Fly.io y Vercel como plataformas de despliegue para el api y aplicación web.

Tiene un enfoque integral, mediante una encuesta se busca evaluar la usabilidad y funcionalidad del proyecto web. Este proyecto contribuye a la innovación en la creación de aplicaciones utilizando la inteligencia artificial, al mismo tiempo cumple con el Plan Nacional de Desarrollo, mediante el objetivo 8 con la política 8.1 donde se pretende mejorar la conectividad digital y el acceso a nuevas tecnologías para la población.

Palabras clave: Inteligencia Artificial, Revisión Sistemática, ISO/IEC 25010, Python, React-Vite, Plan Nacional de Desarrollo.

ABTRACT

This project is focused on the creation of a web application for filtering scientific texts, supporting researchers to carry out research. The objective is to optimize the search and selection of scientific texts in the creation of systematic review documents, being a sample of what can be achieved with current technology.

For the development of the web application, technologies such as: Python and React were used using the ISO 25010 standard. The development of the application was based on the Kanban methodology and the KDD methodological process, using Fly.io and Vercel as deployment platforms for the API and web application.

It has a comprehensive approach, through a survey it seeks to evaluate the usability and functionality of the web project. This project contributes to innovation in the creation of applications using artificial intelligence, at the same time it complies with the National Development Plan, through objective 8 with policy 8.1, which aims to improve digital connectivity and access to new technologies for the population.

Keywords: Artificial Intelligence, Systematic Review, ISO/IEC 25010, Python, React-Vite, National Development Plan.

Introducción

Antecedentes

Este proyecto se realizó con la idea de ayudar a optimizar el análisis de bases de datos masivas de forma automática, para reducir la carga que tienen los investigadores al momento de analizar documentación, debido a que es un trabajo arduo que implica la búsqueda, filtrado, evaluación, selección de literatura científica.

Con base en lo expuesto, se identifica el problema y la oportunidad de utilizar Inteligencia Artificial para la creación de la aplicación web para filtrado de textos científicos de forma automática, que permita a los usuarios reducir el tiempo en la elaboración de los trabajos investigativos aumentando la productividad con eficiencia al seleccionar datos relevantes para el investigador.

Al realizar la investigación se llegó a concluir que existen estudios similares, pero no iguales como: “Automated medical literature screening using artificial intelligence: a systematic review and meta-analysis” enfocado en un análisis exhaustivo de la precisión en el filtrado de textos científicos de diferentes Inteligencias Artificiales.

Situación Actual

En la actualidad en el Ecuador no se ha fundamentado la existencia de una investigación sobre la IA para el filtrado de textos científicos que permitan facilitar la redacción de artículos de revisión sistemática; lo que da lugar a que exista un vacío en el desarrollo de este tipo de herramientas tecnológicas para la optimización de estos procesos investigativos.

Prospectiva

Este tipo de soluciones tiene la capacidad de mejorar el nivel, velocidad y precisión de la producción de textos científicos y revolucionar la investigación académica en

Ecuador. Este proyecto da un pequeño vistazo de lo que se puede hacer con esta tecnología.

Planteamiento del Problema

La selección inadecuada en la literatura científica al realizar estudios del arte, de la técnica o artículos científicos de revisión sistemática representa un gran desafío por la cantidad de textos existentes en las diversas bases de datos académicas y científicas como lo muestra la Figura 1. De no hacerse correctamente, puede producirse un sesgo de contenido en la producción analizada por el investigador, que puede ser de relevancia para el autor. En el mundo se publican alrededor de 8.000-10.000 artículos científicos diarios [1], lo que lleva a complicar la evaluación de los criterios de exclusión e inclusión necesarios para una revisión sistemática, es decir, la amplia disponibilidad de información dificulta la identificación de estudios pertinentes, mientras que la diversidad de enfoques y hallazgos dificulta la integración coherente de los resultados. Estas dificultades pueden llevar a la omisión de estudios relevantes y a errores en la interpretación de datos, comprometiendo así la calidad y validez de la revisión sistemática, lo que impacta negativamente en la generación de conocimiento científico y en la toma de decisiones informadas en diversos campos de estudio. Es fundamental abordar esta dificultad para garantizar la calidad y rigor de las revisiones sistemáticas en la investigación académica.

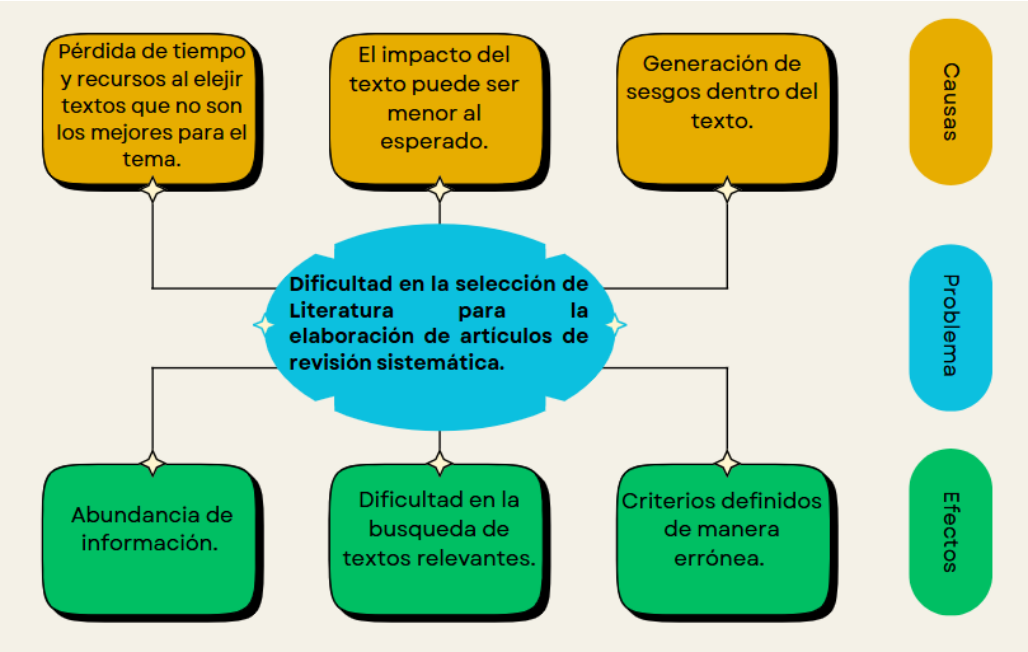


Figura 1. Árbol de Problemas

Objetivos

Objetivo General

Desarrollar una Aplicación Web basada en Aprendizaje Automático (AA) el cual automatice la selección de literatura y la síntesis de datos para mejorar la eficiencia en la elaboración de artículos de revisión sistemática.

Objetivos Específicos

- Redactar un marco teórico que explique los sistemas de aprendizaje automático para el procesamiento del lenguaje natural (NLP).
- Desarrollar una aplicación web para selección de literatura utilizando Python, el framework Flask y React-Vite.
- Evaluar el desempeño de la aplicación web comparando sus resultados con otras IAs, considerando la facilidad de uso de la norma ISO 25010.

Alcance

La presente investigación se enfocará en la aplicación de técnicas de Aprendizaje Automático (AA) ya creadas para el desarrollo de una Aplicación Web para la Automatización de la selección de Literatura y Síntesis de Datos en la elaboración de artículos de revisión sistemática. Se utilizarán técnicas y modelos ya entrenados disponibles en librerías de Python, como por ejemplo spaCy, SciBert, all-MiniLM-L6-v2, para abordar la automatización de las tareas mencionadas. El alcance específico del anteproyecto incluirá los siguientes aspectos:

La identificación de métodos y modelos previamente entrenados de los cuales se seleccionarán técnicas y modelos previamente entrenados disponibles en bibliotecas de ML (por ejemplo, all-MiniLM-L6-v2) que sean adecuados para seleccionar automáticamente literatura relevante siendo más compacto. Esta librería fue desarrollada con la arquitectura sentence-transformers que es de las más poderosa cuando se trata de procesamiento de lenguaje natural, pues ayuda a la máquina a entender, interpretar y generar lenguaje humano[2]. La librería all-MiniLM-L6-v2 funciona usando embeddings, su funcionamiento se indica en la Figura 2.

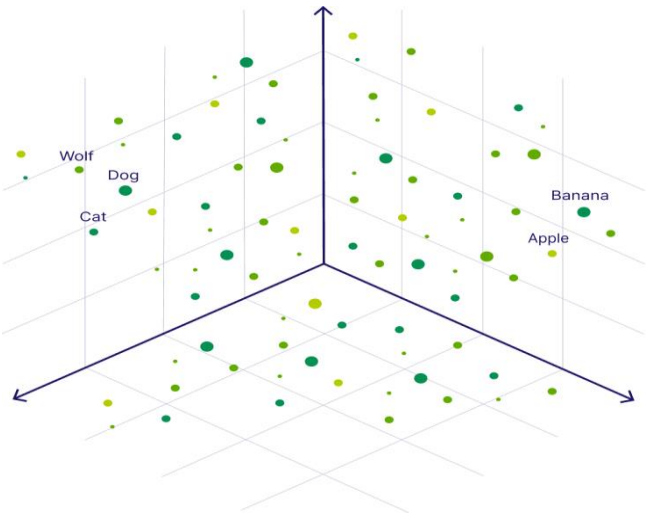


Figura 2. Embeddings *[3]*.

La aplicación esta desarrollada con la metodología Kanban usando el framework Flask, una extensión de Python. La capa de presentación (Front-End) la cual será la encargada de toda la interacción con el usuario, será desarrollada usando React-Vite. La capa de lógica de negocio (Back-End) será creada en Python y haciendo uso de la herramienta Flask. También, está basado en el modelo KDD (Knowledge Discovery in Databases), contenedores Docker para garantizar la compatibilidad de la aplicación web estando alojada en los servidores Fly.io y Vercel. La aplicación funcionará recibiendo un archivo Excel con metadatos, realizará un proceso de normalización y filtrado. Además de recibir criterios de exclusión e inclusión, retornando como resultado una lista con los documentos más relevantes según los criterios seleccionados. Para medir su efectividad se buscará comparar los resultados de la IA con resultados humanos o de otras IAs, así mismo, se planea usar una encuesta sobre la facilidad de uso basada en de la norma ISO 25010, esto lo podemos ver más a detalle en la Figura 3.

Figura 3. Flujo de Desarrollo

Metodología

La investigación es de tipo aplicada, y está orientada al desarrollo de una aplicación web para el filtrado de textos científicos y así facilitar la redacción de artículos de revisión sistemática. Para lograr este objetivo, haremos uso de librerías basadas en transformes como por ejemplo las librerías spaCy, SciBert, all-MiniLM-L6-v2, junto a una técnica de investigación aplicada.

Los instrumentos usados para esta investigación serán el gestor bibliográfico Mendeley y bases de datos indexadas.

Para completar los objetivos, se deben realizar las siguientes actividades:

Objetivo 1: Investigar como las librerías basadas en Transformes tales como spaCy o Bert, podrían ayudar a realizar el filtrado de textos científicos.

Objetivo 2: Desarrollar el Back-End, y una vez que esté listo se realizan pruebas para determinar su efectividad, para proceder con el desarrollo del Front-End.

Objetivo 3: Probar el funcionamiento de la aplicación web comparando sus resultados de una revisión tradicional u otras IAs además, de hacer uso de las métricas propuestas por la ISO 25010 en una encuesta.

Justificación

El proyecto puede contribuir significativamente al objetivo nueve de los Objetivos del Desarrollo Sostenible (ODS), que buscan la creación de nuevas tecnologías innovadoras, agregando un valor extra a los textos de revisión sistemáticas redactados con la ayuda de esta tecnología [4].

Esta aplicación ayudará a mejorar el tiempo en el que los textos de revisión sistemática son escritos, ahorrando recursos y aumentando su calidad.

Justificación Tecnológica. - El aumento masivo en la publicación de documentos científicos dificulta a los profesionales poder seleccionar textos que sean relevantes. Por lo tanto, esta aplicación reducirá el tiempo necesario para el desarrollo de los artículos y aumentará su impacto [5].

Con estos datos el mayor beneficiario de este proyecto es La Universidad Técnica del Norte, los docentes investigadores y estudiantes en proceso de titulación.

Capítulo 1

Marco Teórico

1.1. Revisión Sistemática

La revisión sistemática se trata de un proceso riguroso usado para poder identificar, evaluar y crear una investigación con el objetivo de responder una pregunta en específico. Siguiendo una serie de pasos preestablecidos para poder disminuir los sesgos a la hora de elegir estudios basados en el tema que estamos tratando, solo siendo elegidos los estudios que mayor cercanía tengan con dicho tema [6].

Utilidad

Estas revisiones son una herramienta eficiente cuando se trata de consolidar conocimientos existentes, descubrir brechas que puedan existir, ayudar a tomar decisiones con bases sólidas, ayudar a cualquier persona a enriquecer el conocimiento sobre cualquier tema que esta necesite.

Proceso

Para poder realizar una revisión sistemática existen diferentes pasos que se deben seguir los cuales se muestran en la Tabla 1:

Tabla 1. Pasos para una Revisión Sistemática

Paso	Descripción
Pregunta	En este paso se debe crear la pregunta que servirá como base para toda la investigación, esta pregunta debe ser clara y enfocada al tema [7].
Protocolo	En este paso se define una guía para todos los involucrados en la revisión donde se definen en que lugares van a buscar información, como van a hacerlo, que criterios van a usar para elegir los artículos [7].

Búsqueda	En este paso se realiza la búsqueda con las métricas dadas en el protocolo con el fin de encontrar textos que sean relevantes [7].
Selección	Usando la guía los textos encontrados se someten a un filtrado usando los criterios de elegibilidad que han designado para esto [7].
Extracción	Los datos relevantes son extraídos y gestionados [7].
Evaluación	En este paso se vuelve a evaluar a los textos con el objetivo de reducir cualquier tipo de sesgo que se haya cometido en los anteriores pasos [7].
Síntesis	Ahora se analizan los textos y se discuten cuáles fueron los hallazgos que se tuvo escribiendo un borrador de todo lo descubierto durante esta investigación [7].
Redacción	El paso último en el que se redacta el informe final con toda la información recopilada en la síntesis dejando el documento listo para ser publicado [7].

Tipos

En la revisión sistemática existen algunos tipos de revisiones como se muestra en la Tabla 2:

Tabla 2. Tipos de Revisiones Sistemáticas

Tipo	Descripción
Cuantitativa	Este tipo de revisión se enfoca en sintetizar datos numéricos utilizando técnicas estadísticas. Un ejemplo común es el meta-análisis, que combina los resultados de varios estudios cuantitativos para proporcionar una estimación general del efecto de una intervención o tratamiento [8].

Cualitativa	Se dedica a integrar y comparar hallazgos de estudios cualitativos. Este tipo de revisión busca identificar temas, patrones y construcciones a través de los datos cualitativos de múltiples estudios, proporcionando una comprensión más profunda de fenómenos complejos [8].
Integrativa	Combina datos cuantitativos y cualitativos en una sola síntesis. Es útil para abordar preguntas de investigación que requieren múltiples perspectivas y para proporcionar una visión más completa de la evidencia disponible sobre un tema [8].
De Alcance	Mapea la literatura existente sobre un tema amplio para identificar brechas en la investigación y evaluar la viabilidad de realizar una revisión sistemática completa. Este tipo de revisión es particularmente útil para explorar áreas emergentes o poco estudiadas [8].
Rápida	Se realiza en un período de tiempo reducido utilizando métodos simplificados. Este tipo de revisión es ideal para contextos en los que se necesitan respuestas rápidas, como en situaciones de emergencia sanitaria [8].
De Revisiones	Integra los resultados de múltiples revisiones sistemáticas previas para proporcionar una visión general de la evidencia acumulada sobre un tema amplio. Es útil para resumir un cuerpo extenso de literatura de manera comprensible [8].

Evaluación Económica	Evalúa los costos y beneficios económicos de intervenciones, tratamientos o procedimientos específicos. Proporciona información crucial para la toma de decisiones en políticas de salud y la gestión de recursos [8].
Prevalencia e Incidencia	Mide la carga de enfermedades en diferentes niveles geográficos y temporales. Ayuda en la planificación y asignación de recursos en el sector salud al proporcionar datos sobre la distribución y frecuencia de enfermedades [8].
Exactitud Diagnóstica	Evalúa el rendimiento de pruebas diagnósticas en términos de su precisión y utilidad clínica. Es esencial para determinar la efectividad de diferentes métodos diagnósticos en la práctica clínica [8].
Riesgo y Etiología	Examina la relación entre factores de riesgo específicos y resultados de salud. Este tipo de revisión es valioso para la planificación de la salud pública y la prevención de enfermedades al identificar y cuantificar los riesgos asociados con diferentes exposiciones [8].

1.2. Filtrado de Textos

El filtrado de texto es un método que aumenta la eficiencia a la hora de encontrar información relevante en bases de datos bibliográficas. Este método utiliza filtros metodológicos para minimizar la recopilación de información innecesaria y maximizar la

especificidad y sensibilidad de la búsqueda. Los filtros incluyen combinaciones de términos y descripciones que le permiten encontrar tipos específicos de investigación.

El primer filtro fue desarrollado en 1993 por K. Dickersin, R. Scherer y Carol Lefebvre se fundaron para superar las limitaciones de MEDLINE en la indexación de ensayos clínicos. Desde entonces, se han desarrollado muchos filtros diferentes para diferentes tipos de publicaciones y bases de datos [9].

1.2.1. Técnicas de Filtrado

Filtrado Basado en Palabras Clave

Este método utiliza un conjunto de palabras clave predefinidas para identificar y seleccionar textos adecuados. Se trata de una técnica sencilla y eficaz para encontrar información específica, lo que permite a los investigadores acceder rápidamente a documentos que contienen términos importantes relacionados con el campo de estudio. Según Zhang et al [10], este método es eficaz y se utiliza ampliamente para extraer características destacadas de documentos de texto.

Filtrado Semántico

El filtrado semántico utiliza técnicas avanzadas de procesamiento del lenguaje natural (PLN) para comprender el contexto y el significado del texto. A diferencia del filtrado de palabras clave, el filtrado semántico analiza la relación entre las palabras y el contexto en el que se utilizan, proporcionando una comprensión más profunda y precisa del documento que está viendo. Guo et al., [11], destacan que el uso de algoritmos como TextRank y modelos basados en redes neuronales mejora significativamente la precisión del filtrado semántico.

Filtrado Basado en Modelos de Aprendizaje Automático

Este método utiliza algoritmos de aprendizaje automático, especialmente aquellos basados en la arquitectura Transformers como BERT y GPT, para clasificar y filtrar texto

automáticamente. Los modelos de aprendizaje automático pueden aprender de grandes conjuntos de datos y mejorar continuamente la capacidad para identificar y clasificar texto relevante, lo que los hace extremadamente eficientes y precisos al realizar tareas de filtrado. En el estudio de Ma et al., [12], muestran que los modelos basados en aprendizaje profundo son muy eficientes en la extracción de palabras clave y la clasificación de documentos.

Beneficios

En el tema que manejado hacer uso de estos puntos puede ayudar de manera positiva a cumplir el objetivo como se muestra en la Tabla 3:

Tabla 3. Beneficios de las Técnicas de Filtrado

Eficiencia de la Investigación	Mejora de Calidad
Al automatizar el proceso de revisión y filtrado, estas técnicas aceleran el trabajo de los investigadores. La eficiencia en la identificación de textos relevantes permite a los investigadores dedicar más tiempo y recursos al análisis profundo del contenido, mejorando la productividad y la rapidez en la obtención de resultados.	El uso de técnicas de filtrado de textos asegura que las revisiones sistemáticas sean más completas y precisas. Al seleccionar únicamente los documentos que cumplen estrictamente con los criterios de relevancia, se mejora la validez y la fiabilidad de los resultados, contribuyendo a la calidad general de la investigación.

1.3. Inteligencia Artificial

La inteligencia artificial (IA) se puede definir de varias maneras. En términos generales, la IA es un conjunto de tecnologías que permiten a las computadoras realizar una variedad de funciones avanzadas, incluida la capacidad de ver, comprender y traducir lenguaje hablado y escrito, analizar datos, hacer recomendaciones y mucho más. La IA es la columna vertebral de la innovación en la computación moderna, lo que genera valor para las personas y las empresas [13].

La IA también se puede considerar como una rama de la informática que tiene como objetivo crear máquinas capaces de realizar tareas que tradicionalmente requerían inteligencia humana. Sin embargo, la IA es una ciencia interdisciplinaria con múltiples enfoques y, por lo tanto, no existe una definición única [14].

Algunos expertos definen la IA como el estudio de agentes que reciben percepciones del entorno y realizan acciones. Otros la describen como máquinas que responden a simulaciones como los humanos, con capacidad de contemplación, juicio e intención. Estos sistemas son capaces de tomar decisiones que normalmente requieren un nivel humano de conocimiento [14].

Peligros

Si bien la inteligencia artificial (IA) ofrece un enorme potencial para el progreso humano, también presenta una serie de peligros que es fundamental considerar. A continuación, se exponen algunos de los riesgos más destacados:

Sesgos: La IA puede llegar a presentar sesgos ya existentes en el ser humando, lo que abre la posibilidad de una discriminación, desigualdad de algunos grupos;

Desigualdad: La IA puede agravar las divisiones y desigualdades existentes en el mundo, tanto dentro de los países como entre ellos. Es fundamental defender la justicia, la confianza y la equidad para garantizar que nadie se quede atrás, ya sea en el acceso a las tecnologías de la IA o en la protección contra sus consecuencias negativas;

Medio ambiente: Si bien la IA puede ser beneficiosa para el medio ambiente, también puede ocasionar daños y repercusiones negativas que no deben pasarse por alto. Es necesario considerar el impacto ambiental de los sistemas de IA, incluida su huella de carbono y el consumo de energía;

Desafíos éticos: Los sistemas de IA pueden realizar tareas que antes solo podían hacer los seres vivos, lo que plantea nuevos desafíos éticos relacionados con la toma de

decisiones, el empleo, la interacción social, la atención de la salud, la educación, los medios de comunicación y el acceso a la información. También pueden surgir problemas relacionados con la protección del consumidor y de los datos personales, la democracia, el estado de derecho, la seguridad, los derechos humanos y las libertades fundamentales[15].

1.4. Librerías Basadas en Transformers

Para poder hablar de este tipo de librerías tenemos que hablar de la base que es la arquitectura Transformers este tipo de arquitectura se diferencia mucho de las demás debido a que transforma las secuencias de entrada a salida lo que le permite encontrar relaciones entre los datos que entran siendo extremadamente eficiente además de una excelente opción cuando se trata de comprender el lenguaje natural [16]. En la Tabla 7 tenemos algunos ejemplos de librerías creadas usando como base los Transformers.

Tabla 4. Librerías Basadas en Transformers

Librería	Desarrollador	Descripción
Bert	Google	BERT es una biblioteca de código abierto creada por Google en 2018. Utiliza un enfoque bidireccional que considera tanto el contexto previo como el siguiente de una palabra en el texto. Permitiendo una comprensión exacta del lenguaje natural. BERT es capaz de manejar tareas como: clasificación de texto, análisis de sentimientos y respuesta a preguntas [17].
GPT	OpenAI	GPT es una serie de modelos de lenguaje desarrollados por OpenAI, conocidos por la capacidad para generar texto consistente y apropiado en una variedad de contextos. Estos modelos utilizan aprendizaje profundo basado en una arquitectura transformadora que puede procesar grandes cantidades de texto de manera eficiente; una de las versiones más avanzadas, GPT-3, destaca por la capacidad para realizar tareas complejas como:

		traducción automática, resumen de texto y edición de contenidos, todo ello basado en una amplia formación previa [18].
SciBERT	Allen Institute for AI	Es una variante de BERT (Representación de codificador bidireccional de Transformers) específicamente capacitada en literatura científica, gracias a esta optimización, SciBERT puede ser muy eficiente en la búsqueda y filtrado de problemas en bases de datos académicas. Este modelo tiene como objetivo mejorar la precisión y relevancia de la búsqueda en contextos científicos al facilitar la identificación y síntesis de información relevante en grandes corpus académicos [19].
all-MiniLM-L6-v2	Sentence-transformers	Se trata de un modelo compacto creado a partir de sentence-transformers que transforma las palabras en vectores haciendo que pueda entender perfectamente su significado, gracias a su tamaño ofrece un equilibrio entre rendimiento y precisión en la búsqueda de textos científicos relevantes[2].

1.5. Herramientas de Desarrollo

Las herramientas de desarrollo juegan un papel fundamental en la creación y mantenimiento de software. Estas herramientas no solo facilitan el proceso de codificación, sino que también mejoran la eficiencia y calidad del software desarrollado. A continuación, se describen algunas de las herramientas más destacadas en el ámbito del desarrollo de software.

1.5.1. Flask

Flask es un framework web minimalista que proporciona las herramientas esenciales para construir aplicaciones web, pero permite a los desarrolladores personalizar y

extender las funcionalidades según sea necesario [20]. Lo que lo convierte en una excelente opción tanto para principiantes como para desarrolladores.

Características Principales

- **Minimalismo:** Flask se centra en proporcionar solo lo esencial para construir aplicaciones web, lo que lo hace muy eficiente para APIs, donde la velocidad y el bajo consumo de recursos son cruciales;
- **Enrutamiento Flexible:** Flask te permite definir fácilmente las rutas de tu API y los métodos HTTP que manejan cada una (GET, POST, PUT, DELETE, etc.);
- **Manejo de Solicitudes y Respuestas:** Flask, a través de Werkzeug, te proporciona herramientas para manejar las solicitudes entrantes y generar respuestas en varios formatos, como JSON, que es común en las APIs;
- **Extensiones para Funcionalidades Adicionales:** Puedes utilizar extensiones de Flask para añadir funcionalidades específicas a tu API, como autenticación, autorización, manejo de bases de datos, etc [20].

1.5.2. React,js

Es una biblioteca de JavaScript ideal para desarrollar interfaces de usuario (UI) interactivas y eficientes, lo cual es crucial en el desarrollo de aplicaciones web complejas como: las basadas en aprendizaje automático para la automatización de la selección de literatura y síntesis de datos en artículos de revisión sistemática.

Características de React

Se describe las principales características que tiene React, que facilitan la integración con las APIs en la Tabla 4:

Tabla 5. Características Relevantes de React

Característica	Descripción

JSX	Facilita la integración de HTML y JavaScript, mejorando la claridad del código y la productividad del desarrollo;
Virtual DOM	Optimiza la eficiencia de la aplicación, crucial para manejar grandes volúmenes de datos y actualizaciones frecuentes en tiempo real;
Componentes Reutilizables	Permite crear componentes modulares y reutilizables, lo que facilita la construcción y el mantenimiento de aplicaciones complejas. Por ejemplo, componentes para visualización de datos, formularios de entrada y paneles de control;
Unidirectional Data Flow	Simplifica la gestión del estado y el flujo de datos, crucial en aplicaciones donde la consistencia y la predictibilidad del estado son esenciales para el correcto funcionamiento de los algoritmos de aprendizaje automático;
Integración con Herramientas de Gestión de Estado	Redux y Context API ayudan a manejar de manera eficiente el estado global de la aplicación, garantizando que los datos se sincronizan adecuadamente entre los diferentes componentes;
Renderizado del Lado del Servidor (SSR)	Mejora la velocidad de carga y la optimización en motores de búsqueda (SEO), beneficioso para aplicaciones que requieren una rápida accesibilidad y visibilidad en la web

Fuente: [21].

1.5.3. Vite

Es una herramienta de construcción de código fuente para proyectos web modernos que ofrece una experiencia de desarrollo rápida y flexible.

Características Clave: Tiene dos características que permiten la fácil utilización y flexibilidad al momento de integrarse las cuales se muestran en la Figura 4:

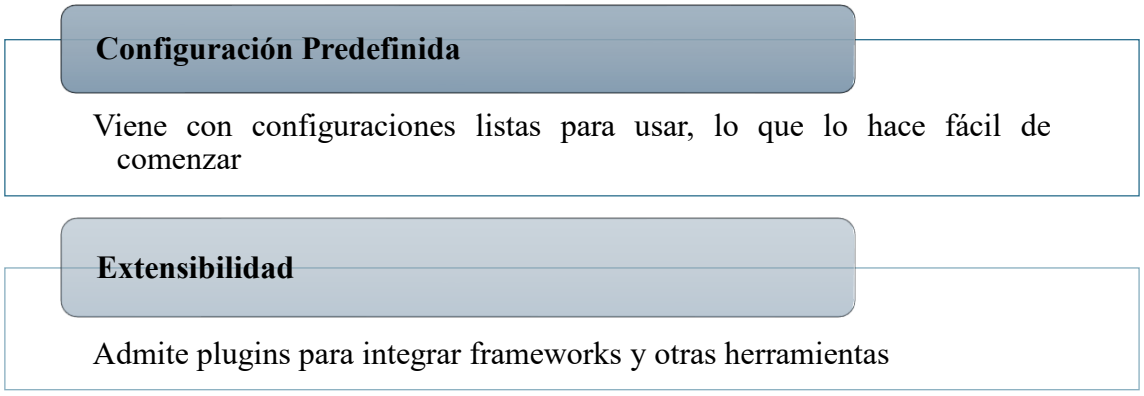


Figura 4. Características de Vite [22].

1.5.4. Docker

Docker es una plataforma de contenedor de software que permite a los desarrolladores empaquetar aplicaciones y dependencias en un contenedor estandarizado.

Estos contenedores pueden ejecutarse en cualquier entorno que admita Docker, lo que garantiza que las aplicaciones se comporten igual sin importar dónde se implementen. Docker se ha convertido en una herramienta esencial para crear e implementar aplicaciones gracias a la capacidad para mejorar el rendimiento y la coherencia del software.

Características

Las características que hacen que la herramienta Docker sea utilizada con mayor frecuencia en el desarrollo de software se muestran en la Tabla 5:

Tabla 6. Características de Docker

Característica	Descripción
Portabilidad	Los contenedores Docker pueden ejecutarse en cualquier sistema operativo que admita Docker, lo que facilita mover aplicaciones entre entornos de desarrollo, pruebas y producción sin cambiar el código;
Aislamiento	Docker proporciona un entorno aislado para cada aplicación, lo que garantiza que no haya conflictos entre las dependencias de diferentes aplicaciones; esto es extremadamente importante para mantener la estabilidad y seguridad del sistema;
Eficiencia de Recursos	A diferencia de las máquinas virtuales, los contenedores Docker comparten el mismo sistema operativo, lo que reduce el consumo de recursos y mejora el rendimiento;

Escalabilidad	Docker facilita la creación y gestión de múltiples versiones de aplicaciones, proporcionando una escalabilidad horizontal sencilla. Los desarrolladores pueden manejar de manera eficiente diferentes cargas de trabajo mediante la replicación de contenedores;
Automatización	Docker se integra con herramientas de automatización y CI/CD (integración continua/implementación continua) para agilizar el proceso de desarrollo y permitir implementaciones más rápidas y confiables;

Fuente: [23].

1.5.5. Entornos de Desarrollo Integrados (IDEs)

Visual Studio:

Visual Studio es un Entorno de Desarrollo Integrado (IDE) completo para desarrolladores .NET y C++ en Windows10. Es una plataforma creativa que se puede utilizar para editar, depurar, compilar código y publicar aplicaciones. Visual Studio va más allá de las funciones básicas de un editor y depurador al incluir compiladores, herramientas de completado de código, diseñadores gráficos y muchas otras características para mejorar el proceso de desarrollo de software [24].

1.5.6. Similitud de coseno

Elegir una IA adecuada es fundamental para el éxito de la aplicación; sin embargo, definir la forma en que se utiliza resulta crucial. En ese contexto la “similitud del coseno” se presenta como una herramienta altamente efectiva, ya que permite calcular el grado de semejanza entre vectores en un espacio multidimensional. Esta propiedad la convierte en una opción ideal para comparar representaciones vectoriales de textos.

La similitud de coseno se calcula como el coseno del ángulo entre dos vectores lo que permite medir su proximidad semántica sin verse afectada sin verse afectada por diferencias de escala o longitud del texto[25].

1.6. Librerías y Paquetes

Las librerías y paquetes de Visual Studio Code ayudan a que se desarrolle de forma eficiente el software y se mencionan en la Tabla 6:

Tabla 7. Librerías y Paquetes de Visual Studio

Librerías y Paquetes	Descripción
Transformación y Análisis de Datos	Realizar operaciones complejas de transformación de datos, necesarias para alimentar algoritmos de aprendizaje automático
Servidor Web	Flask se utiliza para exponer una API
Procesamiento de Texto	Transformers y Torch procesan y generan embeddings
Similitud Coseno	Sklearn.metrics.pairwise.cosine_similarity calcula la relevancia de texto
Gestión de Datos	Pandas organiza y filtra los datos en tablas
Manipulación Numérica	Numpy calcula promedios y convierte tensores a matrices numpy
React	Construcción de la interfaz de usuario y manejo del estado con useState
sweetalert2	Mostrar alertas personalizadas para errores, confirmaciones o mensajes informativos
Xlsx	Manipulación de archivos Excel, para leer y escribir datos
Axios	Realizar solicitudes HTTP al servidor backend.
@mui/material	Componentes de diseño Material-UI para construir la interfaz

1.7. Plataforma de Despliegue

Vercel

Es una plataforma en la nube para desarrolladores diseñada para construir e implementar aplicaciones web. Ofrece herramientas para crear productos más rápido. Por ejemplo, v0 genera un punto de partida para una aplicación, incluyendo el código de su apariencia y funcionamiento, y crea una URL para compartir.

Características y ventajas clave:

Implementación e iteración rápidas: Automatiza la infraestructura, examinando el código, instalando dependencias, construyendo la aplicación, generando la infraestructura en la nube y asignando un dominio para acceder a la aplicación a través de una URL. Vercel crea automáticamente URLs para previsualizar los cambios antes de la fusión en un entorno de preview, permitiendo iterar y realizar cambios de forma segura sin afectar a la aplicación. En caso de errores, se puede volver instantáneamente a la versión anterior.

Soporte de frameworks: Soporta más de 30 frameworks con configuración cero. Es el creador y mantenedor de Next.js, un framework para construir aplicaciones React. También financia el desarrollo de Svelte y apoya otros frameworks de código abierto. Los frameworks simplifican patrones comunes como el enrutamiento entre páginas o la obtención y visualización de datos;

Optimización y seguridad automáticas: Optimiza y asegura las aplicaciones automáticamente, gestionando aspectos como redes, dominios (incluyendo DNS, certificados SSL y nameservers), almacenamiento (caché y almacenamiento de objetos), computación (plataforma de computación autoescalable, distribuida y segura) y CI/CD (implementación automática al hacer push al repositorio Git);

Observabilidad: Incluye herramientas para ver registros y trazas, medir el rendimiento y analizar el tráfico. Permite ver, buscar y filtrar registros de compilación/tiempo de ejecución para investigar problemas y monitorizar la aplicación. Soporta integraciones con herramientas de tracing como OpenTelemetry para un análisis de rendimiento más profundo. Ofrece analíticas propias, respetuosas con la privacidad, para comprender cómo interactúan los usuarios con la aplicación;

Seguridad: Protege las aplicaciones web y previene el tráfico no deseado. Ofrece un firewall de plataforma que bloquea automáticamente las solicitudes maliciosas y los bots

no deseados. Proporciona protección contra ataques DDoS y permite definir reglas personalizadas para protegerse de ataques comunes, web scrapers y otro tráfico no deseado a través de un Web Application Firewall;

Integraciones: Se integra con bases de datos, proveedores de infraestructura en la nube como AWS y otras herramientas [26]

Fly.io

Es una plataforma que ofrece una variedad de servicios y características para el despliegue y la gestión de aplicaciones.

Características y ventajas clave:

Fly Machines: Permite ejecutar código de usuario;

Fly Volumes: Ofrece soluciones de almacenamiento;

Redes: Proporciona herramientas para conectar a un servicio de aplicación, redes públicas y privadas, enrutamiento dinámico de solicitudes, dominios personalizados y soporte TLS;

Bases de datos y almacenamiento: Incluye opciones como Tigris Object Storage, Fly Postgres, SQLite & LiteFS, Upstash for Redis®, Upstash Kafka y Upstash Vector.

Seguridad: Ofrece características como SSO para organizaciones, terminación TLS y seguridad de aplicaciones por Arcjet;

Monitoreo: Incluye métricas, seguimiento de errores Sentry y registro (logs).

Fly Kubernetes: Permite crear y conectar a un clúster FKS, configurar servicios FKS y usar GPUs y volúmenes con FKS;

Autoscaling: Permite el escalado automático de máquinas;

Arquitectura Compartida: Implementa una arquitectura "Shared Nothing";

Soporte Multi-región: Ofrece bases de datos multi-región y fly-replay;

Aislamiento: Permite el aislamiento entre staging y producción[27].

1.8. Marco de Trabajo KDD

Es un proceso sistemático utilizado para descubrir patrones y conocimiento útil a partir de grandes conjuntos de datos. Este proceso es fundamental en la investigación y el análisis de datos, especialmente en contextos académicos e industriales.

Etapas

El marco de trabajo KDD consta de cinco etapas que se muestran y describen a continuación en la Figura 5:

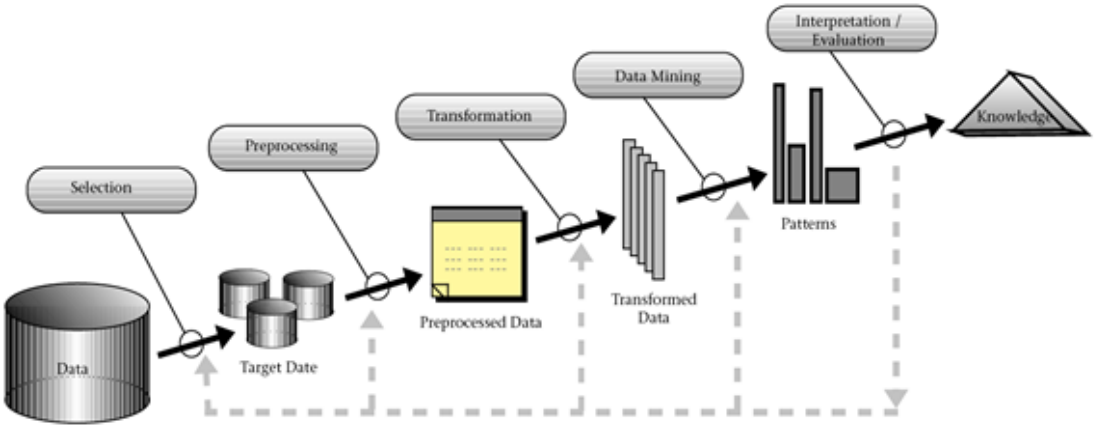


Figura 5. Proceso KDD [28]

La primera etapa, **Selección de Datos**, implica la identificación y recolección de datos relevantes para el análisis, creando un conjunto de datos objetivos que puede ser el conjunto completo o una muestra representativa. La selección se realiza en función de los objetivos del negocio o del estudio [28].

En la fase de **Preprocesamiento**, se lleva a cabo la limpieza y preparación de los datos, eliminando datos ruidosos, gestionando datos faltantes y duplicados, y aplicando técnicas estadísticas para asegurar la calidad de los datos. La interacción con el usuario o analista es crucial para garantizar que los datos sean precisos y útiles [28].

La **Transformación** busca convertir los datos a formatos adecuados para el análisis, utilizando métodos de reducción de dimensiones y transformación para disminuir la cantidad de variables y encontrar representaciones invariantes de los datos [28].

La **Minería de Datos** es el núcleo del proceso KDD, donde se aplican algoritmos de aprendizaje automático para extraer patrones y modelos de los datos. Las técnicas utilizadas incluyen clasificación, agrupamiento, patrones secuenciales y asociaciones, pudiendo los modelos ser predictivos o descriptivos según los objetivos del análisis [28].

Finalmente, en la fase de **Interpretación y Evaluación**, los patrones descubiertos se interpretan y evalúan para asegurar la relevancia y utilidad. Se visualizan los resultados, se eliminan patrones redundantes y se traduce el conocimiento obtenido en términos comprensibles para los usuarios finales. Además, se documentan los resultados y se preparan para el uso futuro o para informar a las partes interesadas [28].

1.9. ISO 25010

1.9.1. Características de Calidad

La norma ISO posee algunas pautas que ayudan a mejorar la calidad del software haciendo que este sea mucho más rentable y económico de cara al futuro que este puede tener, además de ayudarlo a sobrellevar futuros problemas que se puedan presentar, las características se encuentran en la Tabla 8:

Tabla 8. Características de Calidad

Característica	Desarrollador

Adecuación Funcional	El grado en que el software proporciona funciones que satisfacen las necesidades especificadas [29].
Eficiencia de Desempeño	Medición de la capacidad del software para proporcionar tiempos de respuesta y procesamiento adecuados [29].
Compatibilidad	Capacidad del software para operar con otros sistemas.
Usabilidad	Facilidad con que los usuarios pueden aprender y utilizar el software [29].
Fiabilidad	Capacidad del software para mantener el nivel de desempeño bajo condiciones específicas durante un período de tiempo [29].
Seguridad	Protección del software contra el acceso no autorizado [29].
Mantenibilidad	Que tan fácil es modificar el software [29].
Portabilidad	Capacidad del software para cambiar entre entornos [29].

1.9.2. Aplicación en el Proyecto

La norma ISO 25010 define la usabilidad como una de las principales características de calidad del software, para poder evaluar esta característica, se utilizó la encuesta SUS, una herramienta que nos permite medir la facilidad de uso desde la perspectiva de los usuarios finales.

1.9.3. Encuesta SUS

Aplicando la norma ISO se desarrolló una encuesta para determinar la facilidad de uso de un software, mediante el System Usability Scale (SUS), una herramienta estándar para evaluar la usabilidad de sistemas interactivos, con preguntas basadas en una escala de Likert de 5 puntos (1 = Totalmente en desacuerdo, 5 = Totalmente de acuerdo) [30].

Preguntas [30]**:**

1. “Creo que me gustaría utilizar este sistema con frecuencia”

2. “Encontré el sistema innecesariamente complejo”

3. “Pienso que el sistema era fácil de usar”

4. “Creo que necesitaría el apoyo de un técnico para poder utilizar este sistema”

5. “Encontré que las diversas funciones de este sistema estaban bien integradas”

6. “Pensé que había demasiada inconsistencia en este sistema”

7. “Me imagino que la mayoría de la gente aprendería a utilizar este sistema muy rápidamente”

8. “Encontré el sistema muy complicado de usar”

9. “Me sentí muy seguro usando el sistema”

10. “Necesitaba aprender muchas cosas antes de empezar con este sistema”

Con los resultados obtenidos solo falta calcular la puntuación final de la aplicación que se hace de la siguiente manera [30]:

“Suma las respuestas de los enunciados impares y después resta 5”

“Sumas las respuestas de los enunciados pares y después resta ese total a 25”

“Suma ambos resultados y multiplica por 2.5”

Los dos primeros resultados se obtienen aplicando la encuesta de forma individual a cada encuestado, una vez obtenidos el total de los datos se sacará la media aritmética de estos para obtener el resultado final.

1.10. Metodologías Ágiles

1.10.1. Kanban

Kanban es una metodología de gestión de trabajo que ayuda a los equipos a visualizar y limitar el trabajo en progreso y maximizar la eficiencia. El origen se remonta a la década de 1940, cuando Taiichi Ohno, un ingeniero de Toyota, lo desarrolló como un sistema para mejorar la eficiencia de la producción. Kanban se basa en la idea de un sistema de "extracción", donde las tareas se "extraen" de una lista de tareas pendientes a medida que los recursos están disponibles [31].

1.10.2. Principios de Kanban

Estos principios son fundamentales al realizar los proyectos, permiten ejecutarlos de forma eficiente, lo cual se muestra la Figura 6:

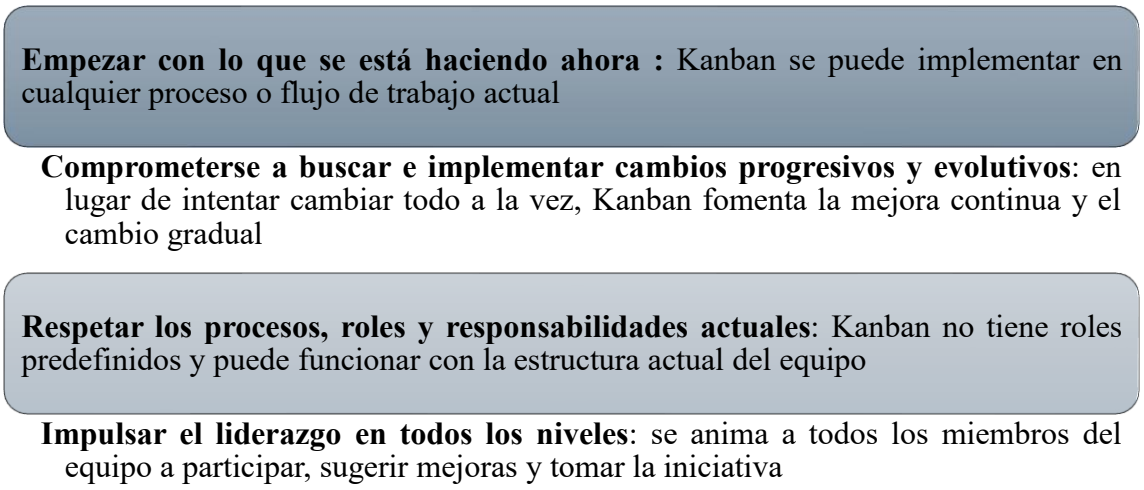


Figura 6. Principios de Kanban [31].

1.10.3. Métricas de Evaluación

Estas métricas son claves en la evaluación del rendimiento del trabajo, permiten al usuario determinar los cuellos de botella, y tomar decisiones acertadas, algunas se muestran en la Figura 7:

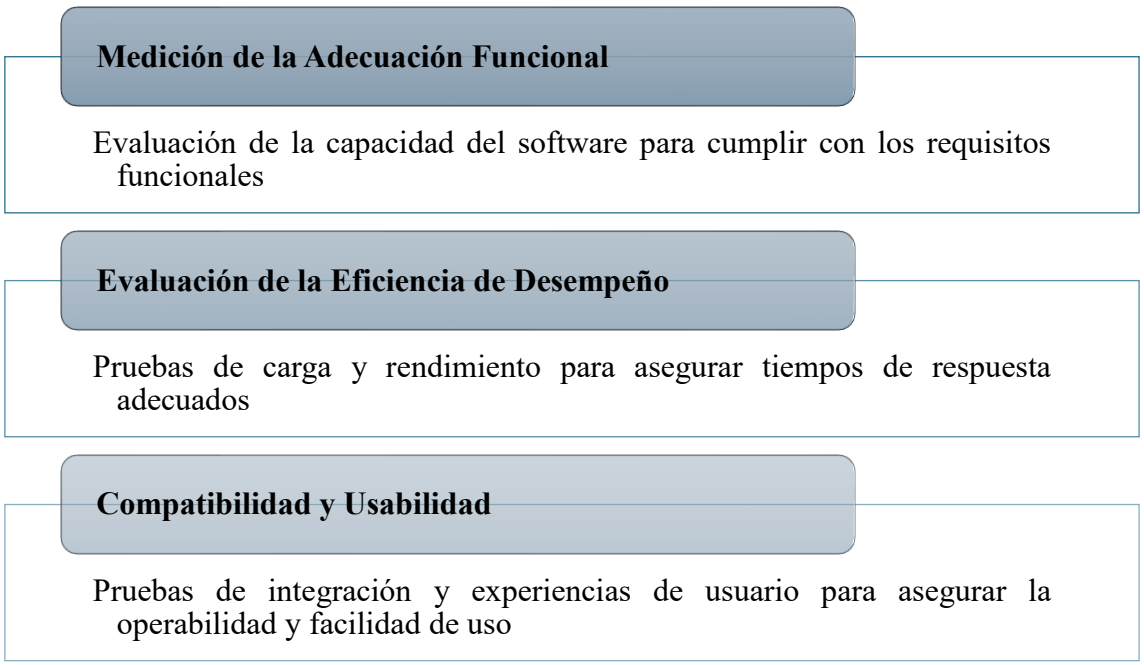


Figura 7. Métricas de Evaluación [29].

1.11. Trabajos Relacionados

Una investigación realizada en la universidad de Oxford [32], propuso la implementación de diferentes métodos basados en inteligencia artificial para automatizar la selección de literatura médica que permita revisiones sistemáticas. Se aplicó una búsqueda en diferentes bases de datos como: *PubMed, Embase y IEEE Xplore Digital Library* para extraer datos que serían usados en diferentes IAs para poder evaluar la efectividad como *SVM (máquinas de vectores de soporte), NB (Naive Bayes), K-NN (k-nearest neighboor)* haciendo uso de métricas como: *WSS (Trabajo Ahorrado sobre el Muestreo)* el recall y la precisión. En la métrica WSS existe una gran variabilidad dependiendo del estudio, los resultados arrojados sugieren que todas las IAS tienen un recall demasiado bajo, si se hace uso de la precisión como valor principal pueden perderse textos relevantes para el estudio. La falta de datos y resultados en algunos estudios, el error humano al seleccionar datos.

La investigación realizada por Bannach et al., [33] exploró la idea de usar IA para poder reducir la carga de trabajo en revisiones sistemáticas. Con el objetivo de poder llegar al nivel de un humano reduciendo así recursos en este paso que es tan importante. Se empleó algunos algoritmos de los cuales Random Forest y Logistic Regresion obtuvieron el mejor puntaje siendo evaluados usando métricas como la sensibilidad, especificidad y precisión, estos modelos fueron entrenados y probados con datos que fueron filtrados por humanos. Los dos algoritmos lograron un porcentaje superior a los esperados haciendo que la carga de trabajo de las personas se redujera de manera drástica. Los modelos pueden ser muy generalizados desentendiendo del entrenamiento y conjunto de datos para poder resolver algunas preguntas, la existencia de errores humanos puede afectar negativamente el rendimiento de estos algoritmos.

Según la investigación realizada por Cohen [34] el procesamiento de lenguaje natural se puede aplicar a la minería de datos biomédicos haciendo uso de diferentes técnicas NLP para poder identificar textos que tengan relevancia. Para lograr esto se emplearon diferentes enfoques para extraer la información: Segmentación, Tokenización, Etiquetas, Análisis sintáctico, Precisión, Detección de Negaciones, Extracción de Relaciones. Al culminar la investigación se demostró que estos modelos ayudaron de manera efectiva el funcionamiento de los sistemas NLP coadyuvando a la labor de los doctores. Los desafíos que enfrentan los sistemas es la ambigüedad del lenguaje natural, falta de datos públicos.

En la investigación realizada por Cierco et al., [35] con el objetivo de poder identificar cuales herramientas hechas a partir de IA podrían ser de ayuda en la creación de revisiones sistemáticas. Se realizaron búsquedas en lugares diferentes uno de publicaciones sobre el tema como: serian PubMed, EMBASE, Web of Science, la segunda fue en repositorios como: GitHub CPAN, CRAN, por mencionar algunos en donde buscaron las herramientas para luego mediante el uso de un formulario tratar de extraer toda la información posible sobre cada proyecto con el fin de poder clasificarlo. De los pocos proyectos que quedaron se determinó que la gran mayoría eran de fácil uso y software libre logrando darle una idea a los autores sobre que herramienta deberían usar. A pesar de ser una revisión exhaustiva siempre existirá un sesgo en las búsquedas y esto debido a los mismos motores de búsqueda.

Capítulo 2

Desarrollo del Módulo

2.1. Diseño de Proceso

El desarrollo de una aplicación implica un proceso estructurado y meticuloso que abarca desde la planificación inicial hasta la implementación de cada una de las funcionalidades. Este capítulo detalla el desarrollo de la aplicación diseñada, abordando las metodologías, herramientas y decisiones técnicas que permitieron convertir los requerimientos en un sistema funcional, eficiente y adaptado a las necesidades del usuario.

El sistema fue concebido bajo un enfoque basado en APIs, lo que permitió separar claramente la lógica del backend y la experiencia del frontend. Esta arquitectura modular no solo asegura la escalabilidad y la flexibilidad del sistema, sino que también facilita futuras integraciones o mejoras. Para llevar a cabo el desarrollo, se consideraron varios aspectos clave, descritos a continuación:

- **Diseño de la Arquitectura:** La aplicación se construyó usando la metodología Kanban, donde el backend gestiona la lógica de negocio y el procesamiento de datos, mientras que el frontend interactúa con el usuario a través de una interfaz dinámica y responsiva.
- **Construcción del Backend:** La API se diseñó para cubrir las funcionalidades principales del sistema, incluyendo la validación y procesamiento de archivos, análisis de datos basado en criterios personalizados, y la integración de inteligencia artificial para mejorar la precisión de los resultados. Para ello, se implementaron endpoints específicos que gestionan estas tareas de manera modular, garantizando una comunicación fluida entre el cliente y el servidor.

- **Diseño e Implementación del Frontend:** La interfaz de usuario se desarrolló utilizando React-Vite y Material-UI, tecnologías que destacan por la capacidad para construir interfaces modernas, intuitivas y responsivas. El frontend incluye módulos específicos para cargar archivos, visualizar datos procesados y generar gráficos que permiten al usuario interpretar fácilmente la información.
- **Integración de la Inteligencia Artificial:** Una de las características distintivas del sistema es el uso de modelos de IA para realizar un análisis avanzado de los datos. Estos modelos permiten identificar patrones, clasificar información y ofrecer predicciones que enriquecen la experiencia del usuario y añaden un valor significativo a los resultados.
- **Pruebas y Validación:** Para asegurar la calidad y confiabilidad del sistema, se llevaron a cabo pruebas exhaustivas en cada etapa del desarrollo. Estas pruebas incluyeron validaciones de funcionalidad, pruebas unitarias e integrales, así mismo, evaluaciones de la experiencia de usuario para garantizar que la aplicación cumpla con los estándares de calidad y las expectativas del cliente.

2.1.1 Diagrama de Procesos

En el diagrama se describe los procesos que se van a ejecutar en el software a desarrollar, con la finalidad de tener una guía que permita al usuario identificar los procedimientos a seguir en la realización de este. En la Figura 8, se muestra cómo se efectúo la aplicación de revisión sistemática.

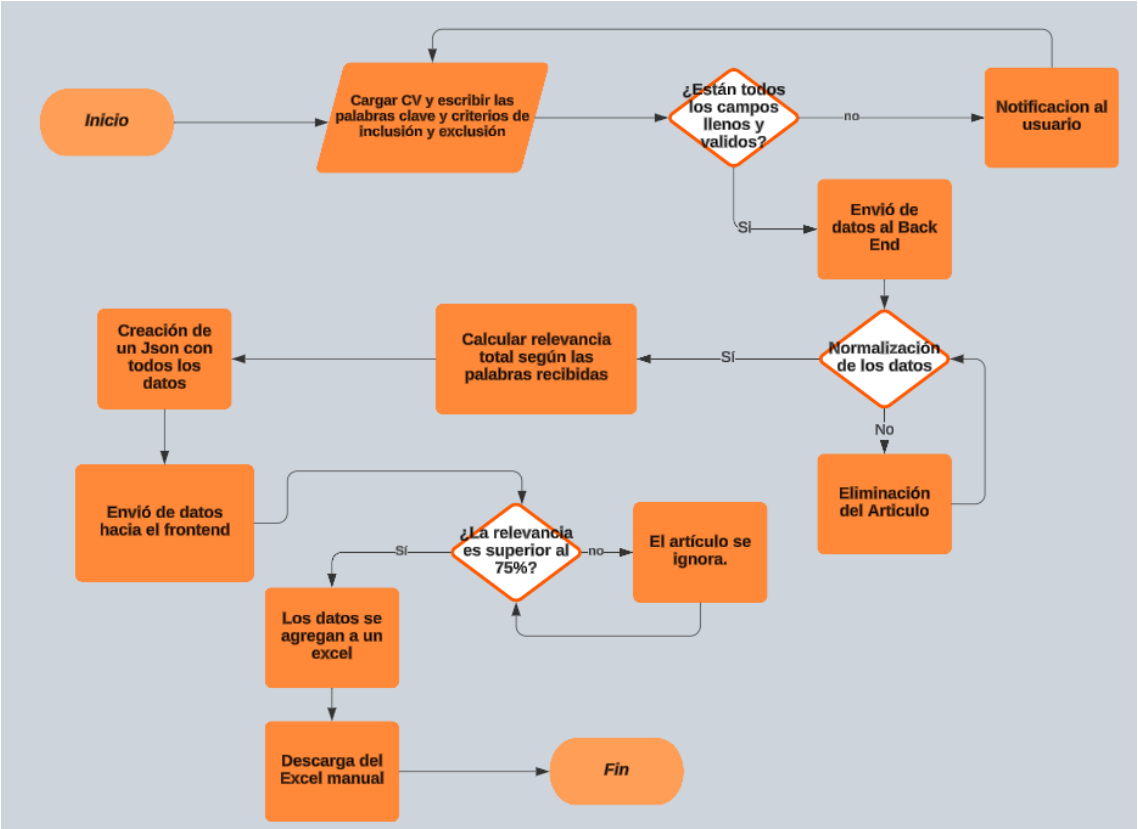


Figura 8. Diagrama de Procesos

2.2. Inicio

2.2.1. Diseño de Actividades Kanban

Usando la metodología Kanban se crearon los siguientes pasos para poder llegar al resultado deseado tratándose de una página web:

Backend

- Definir el propósito de la IA (clasificación, predicción, etc.).
- Investigar y seleccionar el modelo adecuado, en este caso fue elegido all-MiniLM-L6-v2 para el análisis de texto.
- Cargar el modelo preexistente en el backend y agregar el cálculo de similitud del coseno.
- Documentar cómo se integra la IA en el flujo del proyecto.
- Crear el endpoint /analyze para procesar datos CSV.
- Implementar el endpoint /predict con el modelo de IA.

- Probar el modelo con datos de prueba.

Frontend

I. **Diseño de la interfaz**

- Crear una página principal con las siguientes secciones:
  - Cargar archivo CSV (<input> para seleccionar archivos).
  - Campos de entrada para palabras clave, criterios de inclusión y exclusión (<TextField>).
  - Botón para iniciar el análisis (<Button>).
- Diseñar un modal (<Modal>) para confirmar los datos antes de enviarlos al backend.
- Crear una sección de resultados que muestre:
  - Resumen general (número total, aceptados, rechazados).
  - Gráfico de líneas y gráfico de pastel.

II. **Conexión con la API**

- Implementar la lógica de carga de archivo:
  - Enviar el archivo CSV al endpoint POST /upload.
  - Mostrar mensajes de error si el archivo no es válido.
- Implementar la funcionalidad de análisis:
  - Enviar los datos (archivo, palabras clave, criterios) al endpoint POST /analyze.
  - Mostrar una barra de progreso o un estado de carga mientras se realiza el análisis.
- Mostrar resultados:
  - Consultar el endpoint GET /statistics para obtener datos generales.
  - Consultar el endpoint GET /pie-data para obtener datos del gráfico de pastel.

- Renderizar los resultados usando Material-UI (<Typography> y gráficos).

III. **Mejoras de experiencia de usuario (UX)**

- Añadir validación a los campos de entrada (por ejemplo, no permitir enviar datos vacíos).
- Implementar notificaciones para errores o procesos completados (por ejemplo, con SweetAlert2).
- Hacer que la interfaz sea responsiva usando las utilidades de Material-UI.

IV. **Pruebas del Frontend**

- Probar la carga de diferentes archivos CSV.
- Validar que los gráficos se actualicen correctamente con los datos del backend.
- Verificar que los mensajes de error sean claros y específicos.

2.3. Desarrollo

Backend

Selección de IA

Para este trabajo se usó la IA all-MiniLM-L6-v2 por estar creado en sentence-transformers una variante de Transformers especializado en precisión y rendimiento.

Para comenzar el Backend una de las partes más importantes, es donde se utiliza la IA, como se muestra en la siguiente Figura 9.

```
# Modelo ligero para embeddings rápidos y de bajo consumo de recursos
model = SentenceTransformer('all-MiniLM-L6-v2')
```

Figura 9. IA Elegida

Se cargo el modelo preentrenado all-MiniLM-L6-v2 para procesar el texto y obtener las representaciones vectoriales (embeddings), lo que lo hace adecuado para tareas de procesamiento de lenguaje natural en ese dominio.

Cálculo de la relevancia usando similitud del coseno

Esta función usa la similitud del coseno entre el embedding de un texto y un conjunto de embeddings (por ejemplo, palabras clave, inclusiones o exclusiones) el funcionamiento se muestra en la Figura 10.

```
def calcular_relevancia(texto_emb, conjunto_emb):
    if conjunto_emb.size == 0:
        return 0.0
    sims = cosine_similarity(texto_emb.reshape(1, -1), conjunto_emb)
    return float(sims.mean())
```

Figura 10. Similitud de Coseno

Análisis de relevancia total del texto

La función recibe los embedding calculados anteriormente y los usa para calcular la relevancia total que va a tener ese artículo, el funcionamiento está en la Figura 11.

```
resultados = []
for idx, row in enumerate(rows):
    emb = abstracts_emb[idx]
    r_kw  = calcular_relevancia(emb, kw_emb)
    r_inc = calcular_relevancia(emb, inc_emb)
    r_exc = calcular_relevancia(emb, exc_emb)
    relev = r_kw + r_inc - r_exc
    print(f"Artículo {idx}: '{row['Title']}' → relevancia {relev:.4f}")
```

Figura 11. Cálculo de Relevancia

Endpoint

Muestra el proceso de la solicitud Post y los pasos para poder entregar un json que será usado por el frontend, la Figura 12 indica todo ese proceso.

```python
@app.route('/analyze', methods=['POST'])
def analyze():
    # Medición de tiempo total
    start_total = time.perf_counter()
    print("Request received")

    file = request.files.get('file')
    if not file:
        return jsonify({"error": "No se recibió archivo CSV"}), 400

    keywords   = [kw.strip() for kw in request.form.get('keywords','').split('.')    if kw.strip()]
    inclusions = [inc.strip() for inc in request.form.get('inclusions','').split('.') if inc.strip()]
    exclusions = [exc.strip() for exc in request.form.get('exclusions','').split('.') if exc.strip()]

    df = pd.read_csv(file, usecols=['Title','Year','DOI','Abstract'])
    df = df.dropna(subset=['Title','Abstract'])

    abstracts = []
    rows = []
    for _, row in df.iterrows():
        txt = str(row['Abstract']).strip()
        if txt and txt != '[No abstract available]':
            abstracts.append(txt)
            rows.append(row)
    print(f"Abstracts a procesar: {len(abstracts)}")

    abstracts_emb = get_embedding(abstracts)
    print(f"Shape embeddings abstracts: {abstracts_emb.shape}")

    dim = abstracts_emb.shape[1]
    kw_emb  = get_embedding(keywords)   if keywords   else np.zeros((0, dim))
    inc_emb = get_embedding(inclusions) if inclusions else np.zeros((0, dim))
    exc_emb = get_embedding(exclusions) if exclusions else np.zeros((0, dim))
    print(f"kw_emb={kw_emb.shape}, inc_emb={inc_emb.shape}, exc_emb={exc_emb.shape}")

    resultados = []
    for idx, row in enumerate(rows):
        emb = abstracts_emb[idx]
        r_kw  = calcular_relevancia(emb, kw_emb)
        r_inc = calcular_relevancia(emb, inc_emb)
        r_exc = calcular_relevancia(emb, exc_emb)
        relev = r_kw + r_inc - r_exc
        print(f"Artículo {idx}: '{row['Title']}' → relevancia {relev:.4f}")

        resultados.append({
            'Title':      row['Title'],
            'Relevancia': relev,
            'Year':       row['Year'],
            'DOI':        row['DOI'],
            'Abstract':   row['Abstract'],
        })
    end_total = time.perf_counter()
    print(f"Total processing time: {end_total - start_total:.3f} seg")
    return jsonify(resultados)
```

Figura 12. Endpoint

Frontend

Se recibe una solicitud POST para analizar un archivo CSV con artículos científicos.

- **Entrada**: Se recibe un archivo CSV, junto con las palabras clave, inclusiones y exclusiones.
- **Proceso**:
  - Se filtran las columnas necesarias del CSV (Title, Year, DOI, Abstract).
  - Se calculan los embeddings para las palabras clave, inclusiones y exclusiones.
  - Se analiza cada abstract utilizando la función analizar_texto para calcular su relevancia total.
- **Salida**: Se devuelve un JSON con el título, año, DOI, abstract y relevancia calculada de cada artículo.

2.4. Implementación

En la interfaz, el investigador es el responsable de cargar los archivos CSV con los artículos a analizar. Además, debe ingresar las palabras clave, criterios de inclusión y exclusión, que servirán como parámetros para el análisis de los artículos. La página tiene la tarea de asegurarse de que todos los datos ingresados sean correctos y completos antes de proceder con el análisis, garantizando que el proceso sea preciso y efectivo. Además, podrá visualizar los resultados y descargar los archivos generados en formato Excel para ser utilizado posteriormente.

La Figura 13 muestra el menú principal en la que se pueden ingresar todos los datos necesarios para el filtrado y de existir alguna duda puede usar los botones de ayuda.

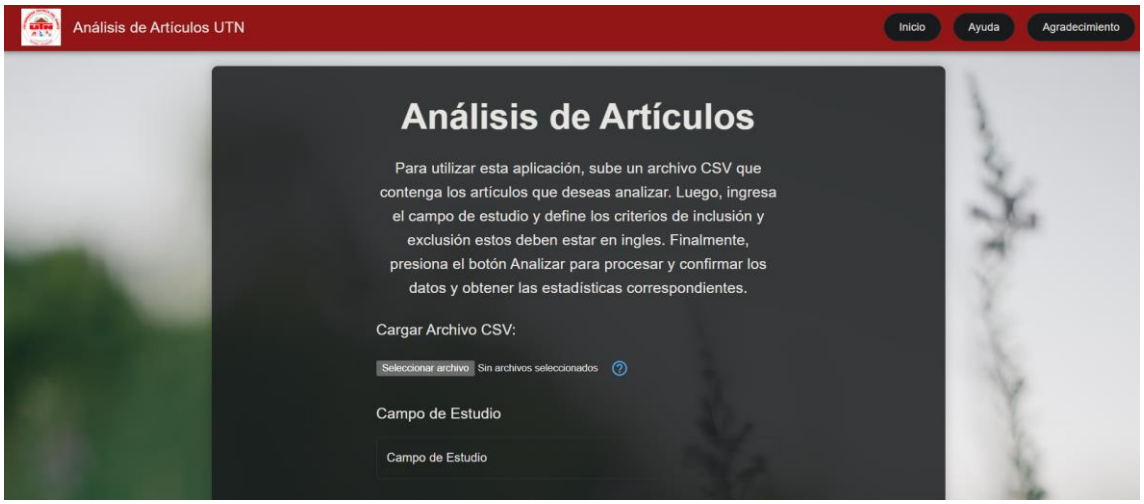


Figura 13. Vista general de la Aplicación

En la Figura 14, el botón indica cómo debe estar estructurado el archivo .csv para la correcta manipulación.

Figura 14. Cuadro de Ayuda 1

Según la Figura 15 el segundo botón muestra que se necesita para separar cada criterio de inclusión y exclusión por si llega a existir algún problema con ellos.

Figura 15. Cuadro de Ayuda 2

Como medida de seguridad se mostrará una pantalla con todos los datos para poder asegurar que sean correctos antes de comenzar el filtrado, como se muestra en la Figura 16.

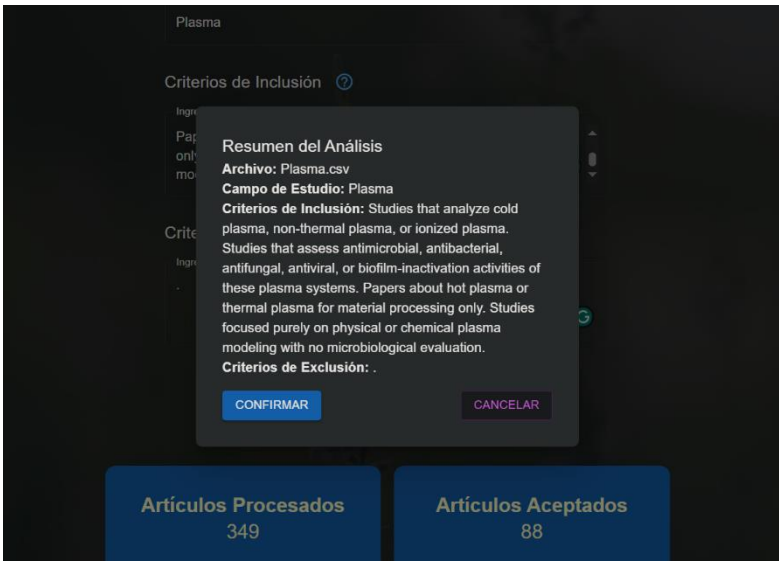


Figura 16. Cuadro de Verificación de Datos

Una vez realizado el filtrado la página nos permite descargar un archivo que contiene los textos que pasaron la prueba, como se muestra en la figura 17.

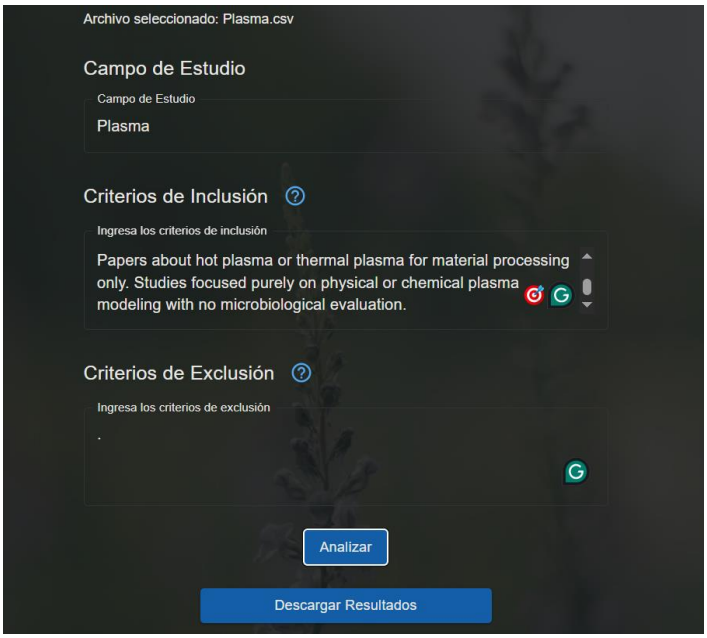


Figura 17. Botón de Descarga

Finalmente, de esta vista se muestra datos sobre el total de los artículos aceptados, rechazados y negados, además de los resultados para cada artículo aceptado, indicando en una gráfica en que rango se encuentran, ejemplo tenemos la Figura 18.

Figura 18. Presentación de Resultado

Seguidamente aparecerá otra vista más con una pequeña guía sobre cómo funciona la aplicación, dando una serie de pasos como indica la Figura 19.

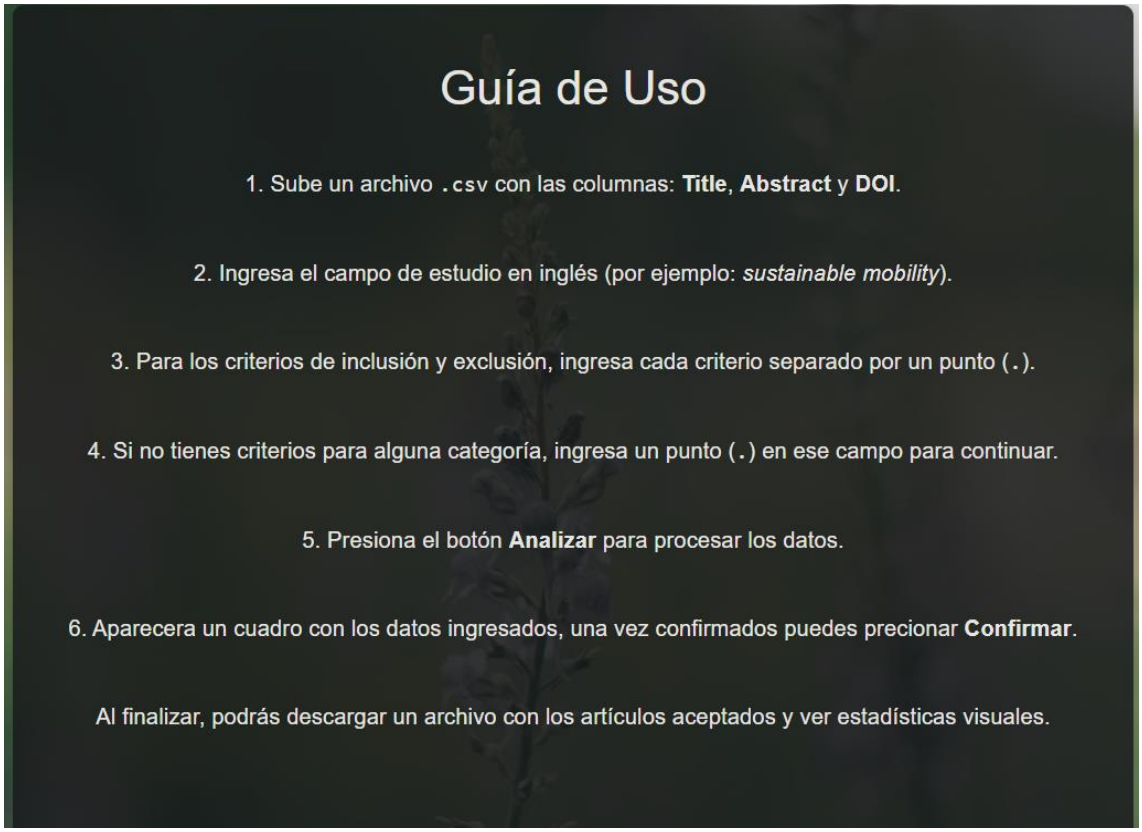


Figura 19. Ayuda

Capítulo 3

Resultados

3.1. Introducción

El capítulo está centrado en la recolección y análisis de los datos obtenidos en la evaluación de la aplicación web desarrollada para filtrar textos científicos para asistir en el proceso de revisión sistemática. Estos resultados estarán separados en dos secciones principales:

- Evaluación de la facilidad de uso mediante una encuesta basada en la norma ISO 25010.
- Evaluación de la eficacia del filtrado de textos científicos.

3.2. Proceso de la Encuesta

- Se utilizó el cuestionario System Usability Scale (SUS).
- La encuesta fue aplicada a usuarios de la aplicación, seleccionados independientemente de la profesión que ejerzan, con el objetivo de obtener una evaluación diversa y representativa.
- Detalle del número total de participantes.

3.2.1. Resultados de la Encuesta

A continuación, se muestra los resultados de todas las preguntas, dadas por 13 usuarios que probaron el proyecto y participaron en la encuesta:

La mayoría de los usuarios muestran una actitud positiva hacia el uso continuo del sistema. Este es un buen indicador de usabilidad general y de valor percibido, según muestra la Figura 20.

Figura 20. Frecuencia de Uso

En la Figura 21 más de la mitad de los encuestados (70%) están de acuerdo en que el sistema es fácil de usar, mientras que otro porcentaje (30%) se mantiene neutro lo que indica que perciben el sistema en su mayoría simple.

Figura 21. Complejidad

La mayoría de los usuarios (90%) perciben que el sistema es fácil de usar, lo cual es un indicador fuerte de una buena experiencia de usuario (UX). En general, el resultado sugiere que el diseño y funcionalidad del sistema son intuitivos y accesibles para la mayoría de los usuarios como muestra la Figura 22.

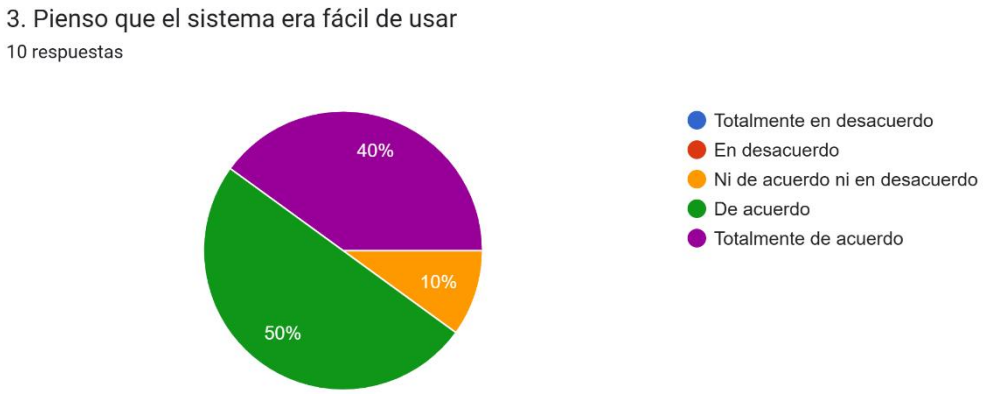


Figura 22. Facilidad de Uso

La Figura 23 muestra que las opiniones muestran que la gran mayoría de los usuarios consideran que no es necesario la ayuda de un técnico, lo que sugiere un diseño efectivo.

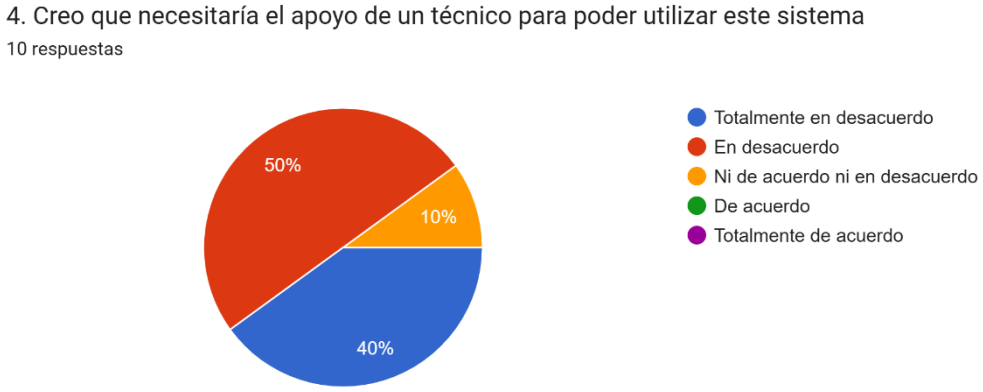


Figura 23. Necesidad de Soporte

Todos los usuarios consideran que las funciones del sistema están bien integradas, lo cual es un excelente indicador de coherencia funcional y fluidez en la experiencia de usuario, como indica la Figura 24.

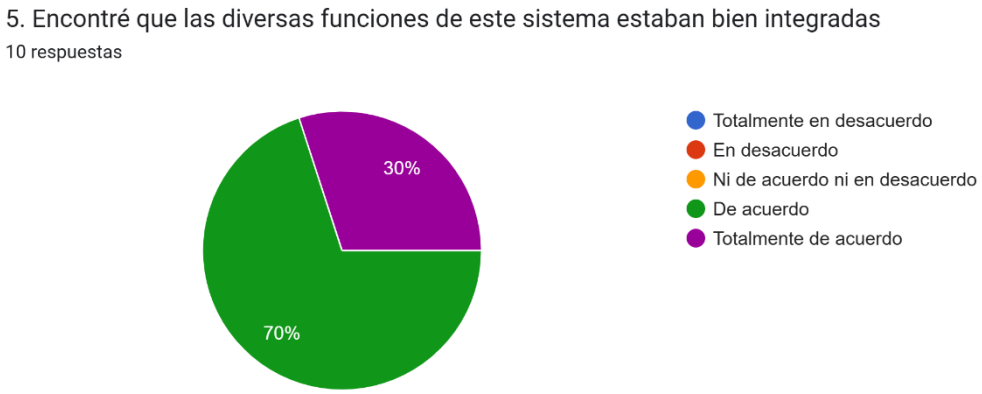


Figura 24. Integración de Funciones

Los usuarios (100%) considera que el sistema no presenta inconsistencias, lo cual es muy positivo, estos datos se ven de forma clara en la Figura 25.

Figura 25. Inconsistencia de la aplicación

La Figura 26 muestra que la gran mayoría (90%) considera que el sistema puede ser aprendido rápidamente por otras personas, lo cual es un fuerte indicador de usabilidad y diseño intuitivo.

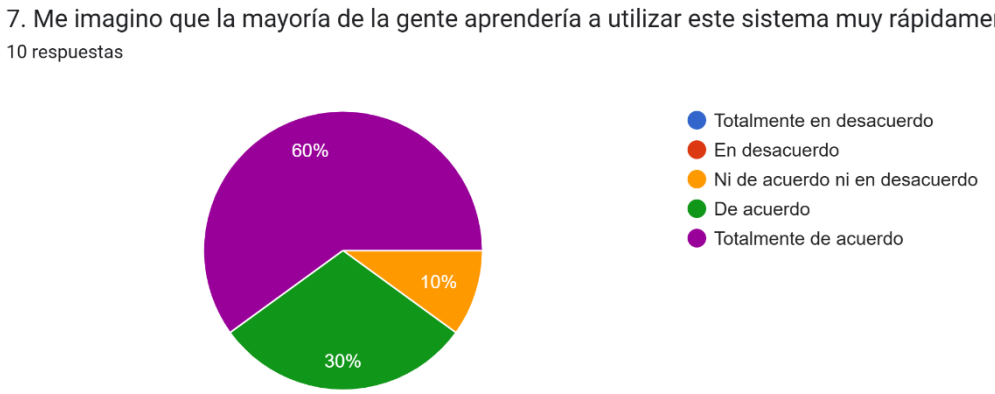


Figura 26. Facilidad de Aprendizaje

Un 90% de los usuarios no encontró el sistema complicado de usar, lo que refleja una percepción muy positiva sobre la simplicidad y accesibilidad del sistema. Las respuestas indicaron una tendencia hacia la facilidad de uso, ya que la mayoría de los usuarios se sintieron cómodos al interactuar con la aplicación, como indica la Figura 27.

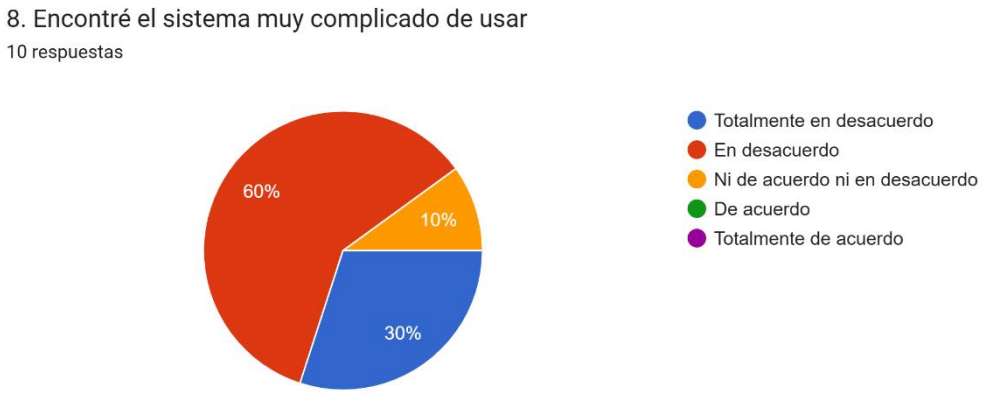


Figura 27. Complejidad

Según la Figura 28 el 70% de los usuarios se sintió seguro usando el sistema, lo que indica una percepción positiva general sobre la seguridad. Sin embargo, un 30% de los usuarios se mantuvo neutral, lo cual es aceptable, pero podría ser un área de mejora para fortalecer la confianza del usuario.

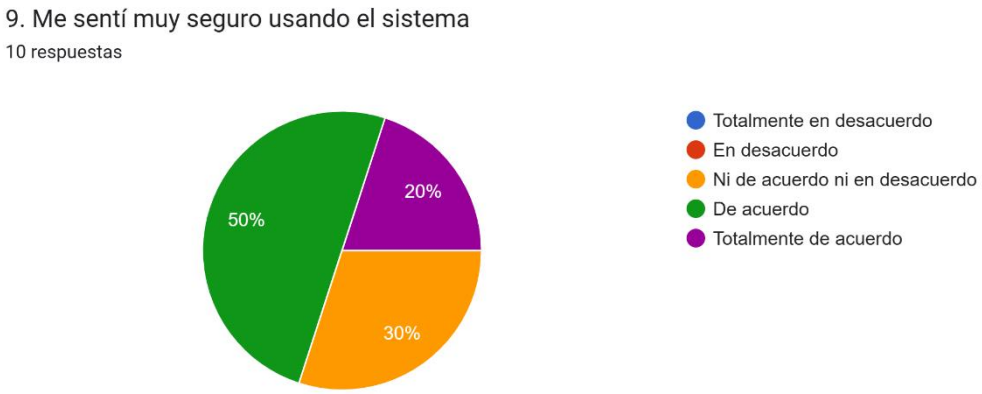


Figura 28. Seguridad de la aplicación

La mayoría de los usuarios (un 90%) no percibió que se necesitara un aprendizaje extenso antes de comenzar a usar el sistema, lo que refleja una buena accesibilidad y facilidad de uso del sistema desde el inicio. Sin embargo, un pequeño porcentaje (10%) indicó que no está seguro, como se indica en la Figura 29.

Figura 29. Aprendizaje

3.2.2. Interpretación de los Resultados

La encueta cuenta con 10 enunciados, para medir cada uno se hace uso de la escala de Likert conteniendo 5 opciones quedando de la siguiente:

- **Totalmente en desacuerdo** = 1
- **En desacuerdo** = 2
- **Ni de acuerdo ni en desacuerdo** = 3
- **De acuerdo** = 4
- **Totalmente de acuerdo** = 5

Los enunciados se dividen en positivos (1, 3, 5, 7, 9) y negativos (2, 4, 6, 8, 10).

Puntaje:

Considerando la información proporcionada para la puntuación, se presenta los resultados obtenidos en este proceso en la Tabla 9:

Tabla 9 Resultados de las Preguntas

Preguntas	Totalmente de acuerdo	De acuerdo	Neutro	En desacuerdo	Totalmente en desacuerdo
Pregunta 1	2	8	0	0	0
Pregunta 2	0	0	3	6	1
Pregunta 3	4	5	1	0	0
Pregunta 4	0	0	1	5	4
Pregunta 5	3	7	0	0	0
Pregunta 6	0	0	0	6	4
Pregunta 7	6	3	1	0	0
Pregunta 8	0	0	1	6	3
Pregunta 9	2	5	3	0	0
Pregunta 10	0	0	1	7	2

Ahora usando un Excel especializado en esta encuesta podemos ver los resultados de cada pregunta en la Tabla 10:

Tabla 10 Total por Investigador

Investigadores	Totalmente
Investigador 1	92.5
Investigador 2	80
Investigador 3	82.5
Investigador 4	72.5
Investigador 5	82.5
Investigador 6	85
Investigador 7	90
Investigador 8	72.5
Investigador 9	70
Investigador 10	72.5
Total	**800**

Finalmente, calculamos la puntuación final usando la formula dada anteriormente y obtenemos como resultado 80, como se muestra a continuación:

(800/10) = 80

3.2.3. Análisis

Con los resultados obtenidos podemos ver la usabilidad que tiene el sistema, este tiene una puntuación de 80 sobre 100 el cual según los parámetros está dentro de los aceptable.

Una puntuación superior a 68 indica que el sistema posee una usabilidad por encima del promedio, mientras que una puntuación inferior puede señalar áreas potenciales de mejora. En este caso, aunque el sistema evaluado supera la media, el resultado de 80 lo que demuestra que los usuarios lo consideran funcional y aceptable.

3.3. Evaluación del Filtrado de Textos Científicos

En este punto se muestra los resultados que proporcionados por los expertos al comparar sus resultados manuales con los de la aplicación y escoger que tan bien filtra la aplicación en una escala del 1 al 5 como se muestra en la Tabla 11:

Tabla 11. Índice de Aceptación

1	No utilizable
2	Filtrado deficiente
3	Filtrado aceptable
4	Buen nivel de filtrado
5	Excelente

3.3.1. Resultados de Funcionamiento

Caso 1

La temática de este caso es “Open Science”, con un corpus de 3.187 artículos científicos. Tras aplicar el sistema de filtrado automático, se procesaron 2.750 artículos, utilizando como criterios de inclusión: “open science in universities, open science regulations, open science politics.” y criterios de exclusión: “-”. Como resultado, se aceptaron 688 artículos, lo que representa una diferencia de 263 artículos más en comparación con el filtrado manual realizado por el investigador que son 425 artículos. Según la evaluación, el investigador asignó a la aplicación un nivel de aceptación 3, lo que indica que el filtrado es aceptable. Cabe destacar que los niveles de aceptación dentro de los artículos aceptados varían, como se muestra en la Figura 30:

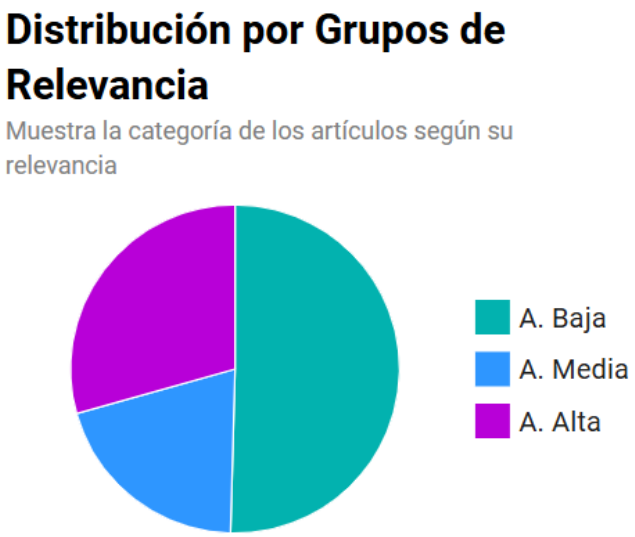


Figura 30. Resultado Open Science

Caso 2

La temática de este caso es "Electric Vehicles", con un corpus de 4.699 artículos científicos. Tras aplicar el sistema de filtrado automático, se procesaron 4.680 artículos, utilizando como criterios de inclusión: "Studies that address electric vehicles (BEVs, PHEVs, EVs) operating in high-altitude or mountainous cities. Studies that analyze the performance, energy consumption, thermal management, or technical adaptation of EVs in high-altitude conditions. Studies addressing the conversion, retrofitting, or repowering of internal combustion vehicles to electric drive systems, explicitly in high-altitude or mountain environments." Y criterios de exclusión: "Studies not related to electric vehicles or their conversion. Studies focused exclusively on hydrogen fuel cell vehicles. Studies focused on low-altitude or sea-level cities, without mention of altitude/mountain conditions." Como resultado, se aceptaron 1.170 artículos, lo que representa una diferencia de 269 artículos más en comparación con el filtrado manual realizado por el investigador que son 901 artículos. Según la evaluación, el investigador asignó a la aplicación un nivel de aceptación 3, lo que indica que el filtrado es aceptable.

Cabe destacar que los niveles de aceptación dentro de los artículos aceptado varían, como se muestra en la Figura 31:

Figura 31. Resultados Electric Vehicles

Caso 3

La temática de este caso es "Plasma", con un corpus de 350 artículos científicos. Tras aplicar el sistema de filtrado automático, se procesaron 349 artículos, utilizando como criterios de inclusión: "Studies that analyze cold plasma, non-thermal plasma, or ionized plasma. Studies that assess antimicrobial, antibacterial, antifungal, antiviral, or biofilm-inactivation activities of these plasma systems." Y criterios de exclusión: "Papers about hot plasma or thermal plasma for material processing only. Studies focused purely on physical or chemical plasma modeling with no microbiological evaluation." Como resultado, se aceptaron 88 artículos, lo que representa una diferencia de 17 artículos menos en comparación con el filtrado manual realizado por el investigador que son 105. Según la evaluación, el investigador asignó a la aplicación un nivel de aceptación 5, lo que indica que el filtrado es excelente. Cabe destacar que los niveles de aceptación dentro de los artículos aceptado varían, como se muestra en la Figura 32:

Figura 32. Resultados Plasma

Caso 4

La temática de este caso es "Turismo", con un corpus de 371 artículos científicos. Tras aplicar el sistema de filtrado automático, se procesaron 368 artículos, utilizando como criterios de inclusión: "Tourist behavior changes due to climate change. Tourism demand shifts linked to climate variability. Impact of extreme weather events or temperature changes on tourism destinations. Economic effects of climate change on tourism markets. Tourism management or development strategies in response to climate change. Adaptation measures in the tourism industry facing climate risks." y criterios de exclusión: "-." Como resultado, se aceptaron 92 artículos, lo que representa una diferencia de 2 artículos más en comparación con el filtrado manual realizado por el investigador que son 90. Según la evaluación, el investigador asignó a la aplicación un nivel de aceptación 5, lo que indica que el filtrado es excelente. Cabe destacar que los niveles de aceptación dentro de los artículos aceptado varían, como se muestra en la Figura 33:

Figura 33. Resultados Turismo

Caso 5

La temática de este caso es "Baterías ion litio", con un corpus de 141 artículos científicos. Tras aplicar el sistema de filtrado automático, se procesaron 123 artículos, utilizando como criterios de inclusión: "Application of nanotechnology or nanomaterials in lead-acid batteries. Use of nano-additives, nano-coatings, electrodes with nanostructures, or nanocomposite materials in the battery's internal structure." y criterios de exclusión: "Articles that mention nanotechnology and lead-acid batteries without providing experimental results or simulation data." Como resultado, se aceptaron 31 artículos, lo que representa una diferencia de 3 artículos más en comparación con el filtrado manual realizado por el investigador que son 28. Según la evaluación, el investigador asignó a la aplicación un nivel de aceptación 5, lo que indica que el filtrado es excelente. Cabe destacar que los niveles de aceptación dentro de los artículos aceptado varían, como se muestra en la Figura 34:

Figura 34. Resultados Baterías ion litio

Caso 6

La temática de este caso es "twitter", con un corpus de 1.595 artículos científicos. Tras aplicar el sistema de filtrado automático, se procesaron 1.432 artículos, utilizando como criterios de inclusión: "The article must address the use of Twitter by government officials, public institutions, political leaders, or governance bodies, either at local, national, or international levels. Twitter (or X) must be a central focus of the study, not just mentioned in passing or grouped with other platforms without specific analysis." y criterios de exclusión: "-" Como resultado, se aceptaron 358 artículos, lo que representa una diferencia de 15 artículos menos en comparación con el filtrado manual realizado por el investigador que son 373. Según la evaluación, el investigador asignó a la aplicación un nivel de aceptación 4, lo que indica que el filtrado es mayormente aceptable. Cabe destacar que los niveles de aceptación dentro de los artículos aceptado varían, como se muestra en la Figura 35:

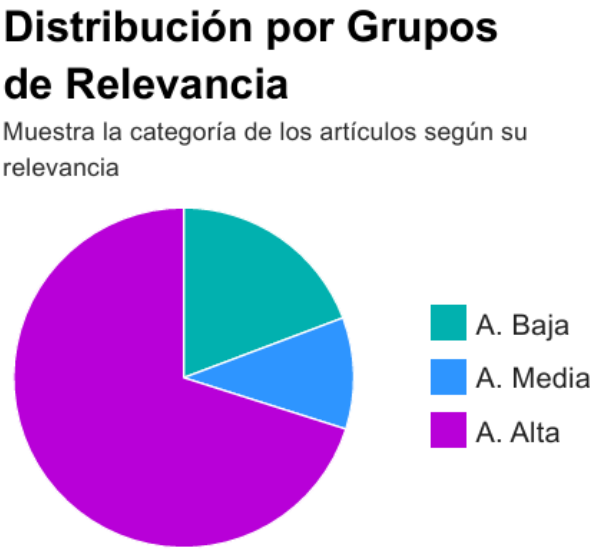


Figura 35. Resultados Twitter

Caso 7

La temática de este caso es "Bibliotecas", con un corpus de 2.488 artículos científicos. Tras aplicar el sistema de filtrado automático, se procesaron 2.471 artículos, utilizando como criterios de inclusión: "Research conducted by academic/university libraries. Scientific or scholarly publications authored by library professionals. information literacy, open access, digital preservation, library innovation, data curation, research support, etc." y criterios de exclusión: "Articles focusing on public, school, or national libraries without mention of academic or university libraries. Articles that do not analyze research production, scholarly publishing, or research support roles of libraries." Como resultado, se aceptaron 618 artículos, lo que representa una diferencia de 20 artículos más en comparación con el filtrado manual realizado por el investigador que son 598. Según la evaluación, el investigador asignó a la aplicación un nivel de aceptación 4, lo que indica que el filtrado es mayormente aceptable. Cabe destacar que los niveles de aceptación dentro de los artículos aceptado varían, como se muestra en la Figura 36:

Figura 36. Resultados Bibliotecas

Caso 8

La temática de este caso es "Educación e IA", con un corpus de 1.204 artículos científicos. Tras aplicar el sistema de filtrado automático, se procesaron 1.061 artículos, utilizando como criterios de inclusión: "Pedagogical changes, instructional transformations, or teaching innovations. Caused by or closely related to the use of artificial intelligence in education." y criterios de exclusión: "Articles that do not discuss teaching strategies, learning methods, instructional design, or educational innovation. Articles about digital tools like virtual reality, blockchain, robotics, etc., where AI is not the core focus." Como resultado, se aceptaron 266 artículos, lo que representa una diferencia de 52 artículos más en comparación con el filtrado manual realizado por el investigador que son 214. Según la evaluación, el investigador asignó a la aplicación un nivel de aceptación 4, lo que indica que el filtrado es mayormente aceptable. Cabe destacar que los niveles de aceptación dentro de los artículos aceptado varían, como se muestra en la Figura 37.

Figura 37. Resultados Educación e IA

Caso 9

La temática de este caso es "TENGS", con un corpus de 2.476 artículos científicos. Tras aplicar el sistema de filtrado automático, se procesaron 2.452 artículos, utilizando como criterios de inclusión: "Open-circuit voltage (Voc). Short-circuit current (Isc). Optionally, power density, internal resistance, or charge density. The article must provide detailed information about the triboelectric materials used in the construction of the TENG." y criterios de exclusión: "Articles that do not mention Voc or Isc or only refer to them conceptually without reporting values or measurements." Como resultado, se aceptaron 613 artículos, lo que representa una diferencia de 3 artículos más en comparación con el filtrado manual realizado por el investigador que son 610. Según la evaluación, el investigador asignó a la aplicación un nivel de aceptación 5, lo que indica que el filtrado es excelente. Cabe destacar que los niveles de aceptación dentro de los artículos aceptado varían, como se muestra en la Figura 38.

Figura 38. Resultados TENGS

Caso 10

La temática de este caso es "Estadística y nanotecnología", con un corpus de 1.689 artículos científicos. Tras aplicar el sistema de filtrado automático, se procesaron 1.585 artículos, utilizando como criterios de inclusión: "The article must explicitly apply at least one statistical method or technique in the analysis, modeling, interpretation, or validation of data related to nanoscience or nanotechnology." y criterios de exclusión: "-". Como resultado, se aceptaron 397 artículos, lo que representa una diferencia de 30 artículos menos en comparación con el filtrado manual realizado por el investigador que son 427. Según la evaluación, el investigador asignó a la aplicación un nivel de aceptación 4, lo que indica que el filtrado es mayormente aceptable, cabe destacar que los niveles de aceptación dentro de los artículos aceptado varían, como se muestra en la Figura 39.

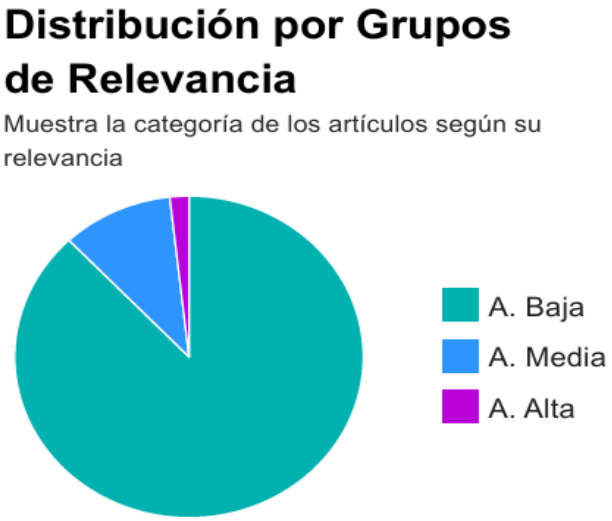


Figura 39. Resultados Estadística y Nanotecnología

A continuación, se presenta todos los resultados obtenidos, como se puede ver en la Tabla 12.

Tabla 12. Resultados de Investigado

Campo de Estudio	Cantidad de Datos	Artículos Procesados	Artículos Filtrados	Artículos filtrados por investigador de manera manual	Comparación de resultados	Aceptación de resultados (escala del 1 - 5) de acuerdo con el criterio del investigador
Open Science	3187	2750	688	425	263	3
Electric Vehicles	4699	4680	1170	901	269	3
Plasma	350	349	88	105	-17	5
Turismo	371	368	92	90	2	5
Baterías ion litio	141	123	31	28	3	5
Twitter	1595	1432	358	373	-15	4
Bibliotecas	2488	2471	618	598	20	4
Educación e IA	1204	1061	266	214	52	4
TENGS	2476	2452	613	610	3	5
Estadística y nanotecnología	1689	1585	397	427	-30	4

3.3.2. Interpretación de los Resultados

Los resultados muestran una variabilidad considerable entre los diferentes campos de estudio en cuanto en volumen de datos procesados, artículos filtrados y nivel de aceptación de los resultados.

En las áreas “Electric Vehicles”, “Open Science” tiene un corpus grande (4699 y 3187) y un gran número de artículos filtrados (1170 y 688), la diferencia con los filtrados manuales es considerable (269 y 253), y la aceptación de estos resultados es moderada.

Por otro lado, las áreas “Plasma”, “Turismo”, “Baterías ion litio” tiene un corpus más reducido y la concordancia con los resultados es más estable (-17, 2, 3), la aceptación es muy positiva.

3.3.3. Discusión sobre los Resultados

Los resultados indican que la eficiencia del sistema de filtrado puede variar dependiendo de la cantidad de datos a filtrar y la complejidad de los criterios de búsqueda, aquí tenemos lo campos “Electric Vehicles” o “Bibliotecas” que poseen un corpus grande, donde el sistema presenta una desviación alta frente a los criterios humanos, en contraste tenemos a “Plasma” y “TENGS” que poseen un corpus más reducido, donde los resultados de la aplicación y humanos son más cercanos. Además, la aceptación de los investigadores hacia los resultados respalda esto, porque los valores más altos se asignaron a los campos donde la diferencia en los resultados fue mínima.

3.3.4. Sugerencias de los Investigadores

Durante la evaluación de la herramienta se efectuaron observaciones por parte de los investigadores con la finalidad de mejorar la nota del filtrado final:

Primero se recomendó incorporar una funcionalidad que permita resumir brevemente los motivos por los cuales se seleccionó determinado articulo como relevante. Esto ayudaría al investigador a facilitar la comprensión de los resultados.

Conclusiones

- El marco teórico fue redactado de manera correcta explicando el tema de las Inteligencias Artificiales, lo que ayudó a comprender el tema de las Inteligencias Artificiales y los puntos importantes para el desarrollo de este proyecto.
- La aplicación fue desarrollada exitosamente demostrando una buena funcionalidad y capacidad de filtrado, esto demostrado por las pruebas aplicadas a los investigadores y usuarios más generales.
- La aplicación demostró ser una gran herramienta a la hora de automatizar el filtrado de textos científicos, reduciendo el tiempo y esfuerzo empleado en la escritura de textos de revisión sistemática.

Recomendaciones

- Con el creciente avance de la tecnología cada vez se inventan mejores IAs que poseen más potencia con menos procesamiento, siendo ese el caso se puede considerar probar nuevas IAs emergentes para mejorar la precisión y capacidades del programa.
- Explorar la idea de integrar nuevas y mejores tecnologías con el fin de aumentar las funcionalidades del software y facilitar aún más la tarea de crear textos científicos.
- Considerar la opción de poner este software como código abierto de forma que cualquiera pueda tener acceso a él y aprender o mejorarlo.

Referencias Bibliográficas

[1] Álvaro Lorite, “Papers y más papers: las sombras en la industria de las publicaciones científicas,” El Salto . Accessed: May 28, 2024. [Online]. Available: https://www.elsaltodiario.com/universidad/papers-y-mas-papers-las-sombras-en-la-industria-de-las-publicaciones-cientificas

[2] R. Tiwari, “Unlocking the Power of Sentence Embeddings with all-MiniLM-L6-v2 ,” Medium. Accessed: May 22, 2025. [Online]. Available: https://medium.com/@rahultiwari065/unlocking-the-power-of-sentence-embeddings-with-all-minilm-l6-v2-7d6589a5f0aa

[3] D. Dascalescu and Z. Hasan, “Vector Embeddings Explained,” Weaviate. Accessed: May 29, 2025. [Online]. Available: https://weaviate.io/blog/vector-embeddings-explained

[4] Naciones Unidas, “Objetivo 9: Construir infraestructuras resilientes, promover la industrialización sostenible y fomentar la innovación,” Naciones Unidas . Accessed: Apr. 29, 2024. [Online]. Available: https://www.un.org/sustainabledevelopment/es/infrastructure/

[5] Iain J. Marshall and Byron C. Wallace, “Toward systematic review automation: a practical guide to using machine learning tools in research synthesis,” Systematic Reviews. Accessed: Apr. 29, 2024. [Online]. Available: https://systematicreviewsjournal.biomedcentral.com/articles/10.1186/s13643-019-1074-9#Sec4

[6] BIBLIOGUÍAS, “Revisiones sistemáticas: Definición: ¿qué es una revisión sistemática?,” BIBLIOGUÍAS. Accessed: Jul. 12, 2024. [Online]. Available: https://biblioguias.unav.edu/revisionessistematicas/que-es-una-revision-sistematica

[7] BIBLIOGUÍAS, “Revisiones sistemáticas: Pasos o Etapas para realizar una revisión sistemática,” BIBLIOGUÍAS. Accessed: Jul. 12, 2024. [Online]. Available: https://biblioguias.unav.edu/revisionessistematicas/pasos-realizar-revisionsistematica

[8] Z. Munn, C. Stern, E. Aromataris, C. Lockwood, and Z. Jordan, “What kind of systematic review should I conduct? A proposed typology and guidance for

systematic reviewers in the medical and health sciences," *BMC Med Res Methodol*, vol. 18, no. 1, p. 5, Dec. 2018, doi: 10.1186/s12874-017-0468-4.

[9] Campos, "¿Qué son los filtros de búsqueda? ," BiblioGETAFE. Accessed: Jul. 10, 2024. [Online]. Available: https://bibliogetafe.com/2013/08/22/que-son-los-filtros-de-busqueda/

[10] M. Zhang, X. Li, S. Yue, and L. Yang, "An Empirical Study of TextRank for Keyword Extraction," *IEEE Access*, vol. 8, pp. 178849–178858, Sep. 2020, doi: 10.1109/ACCESS.2020.3027567.

[11] W. Guo, Z. Wang, and F. Han, "Multifeature Fusion Keyword Extraction Algorithm Based on TextRank," *IEEE Access*, vol. 10, pp. 71805–71813, Jul. 2022, doi: 10.1109/ACCESS.2022.3188861.

[12] J. Ma, "Research on Keyword Extraction Algorithm in English Text Based on Cluster Analysis," *Comput Intell Neurosci*, vol. 2022, pp. 1–8, Mar. 2022, doi: 10.1155/2022/4293102.

[13] Google, "¿Qué es la inteligencia artificial o IA? | Google Cloud," Google Cloud . Accessed: Dec. 16, 2024. [Online]. Available: https://cloud.google.com/learn/what-is-artificial-intelligence?hl=es-419

[14] Anonimo, "Inteligencia artificial : definición, historia, usos, peligros," DataScientest. Accessed: Dec. 16, 2024. [Online]. Available: https://datascientest.com/es/inteligencia-artificial-definicion

[15] UNESCO, "Recommendation on the Ethics of Artificial Intelligence," 2021, Accessed: Apr. 16, 2025. [Online]. Available: https://unesdoc.unesco.org/ark:/48223/pf0000380455

[16] AWS, "¿Qué son los transformadores en la inteligencia artificial?," AWS. Accessed: Jul. 12, 2024. [Online]. Available: https://aws.amazon.com/es/what-is/transformers-in-artificial-intelligence/#:~:text=Los%20transformadores%20son%20un%20tipo,color%20es%20el%20cielo%3F%E2%80%9D.

[17] Fernando Cardellino, "Tutorial de Google BERT para PNL con aprendizaje automático," freeCodeCamp. Accessed: Jul. 10, 2024. [Online]. Available:

https://www.freecodecamp.org/espanol/news/tutorial-de-google-bert-para-pnl-con-aprendizaje-automatico/#:~:text=BERT%20es%20una%20biblioteca%20de,de%20codificador%20bidireccional%20de%20Transformer

[18] Z. Kasneci et al., “GPT (Generative Pre-trained Transformer) – A Comprehensive Review on Enabling Technologies, Potential Applications, Emerging Challenges, and Future Directions,” ar5iv.org. Accessed: Jul. 10, 2024. [Online]. Available: https://ar5iv.labs.arxiv.org/html/2305.10435

[19] I. Beltagy, K. Lo, and A. Cohan, “SciBERT: A Pretrained Language Model for Scientific Text,” in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Stroudsburg, PA, USA: Association for Computational Linguistics, Mar. 2019, pp. 3613–3618. doi: 10.18653/v1/D19-1371.

[20] Flask, “Welcome to Flask — Flask Documentation (3.1.x).” Accessed: Dec. 19, 2024. [Online]. Available: https://flask.palletsprojects.com/en/stable/

[21] Deyimar A., “Qué es React: definición, características y funcionamiento,” Glosario Jun 29, 2023 Deyimar A. 9min de lectura Qué es React: definición, características y funcionamiento. Accessed: Jul. 10, 2024. [Online]. Available: https://www.hostinger.es/tutoriales/que-es-react

[22] Vite, “Getting Started | Vite.” Accessed: Dec. 25, 2024. [Online]. Available: https://vite.dev/guide/

[23] RedHat, “¿Qué es Docker y cómo funciona?,” RedHat. Accessed: Jul. 10, 2024. [Online]. Available: https://www.redhat.com/es/topics/containers/what-is-docker

[24] Microsoft, “Visual Studio: IDE y Editor de código para desarrolladores de software y Teams.” Accessed: Dec. 19, 2024. [Online]. Available: https://visualstudio.microsoft.com/es/

[25] Vercel, “Vercel Documentation.” Accessed: Apr. 16, 2025. [Online]. Available: https://vercel.com/docs

[26] Fly, “Fly.io developer documentation.” Accessed: Apr. 16, 2025. [Online]. Available: https://fly.io/docs/

[27] L. Velásquez and B. Hitpass, *El nivel de Actividad en el Proceso Educativo como Indicador de Riesgo de Deserción Estudiantil medido en tiempo real con apoyo de tecnología BAM*. Chile: Universidad Técnica Federico Santa María Santiago, 2014. Accessed: Jul. 12, 2024. [Online]. Available: http://dx.doi.org/10.13140/2.1.3217.8880

[28] ISO25000, “ISO/IEC 25010,” ISO25000. Accessed: Jul. 10, 2024. [Online]. Available: https://iso25000.com/index.php/en/iso-25000-standards/iso-25010?start=5

[29] C. Budquets, “Medir la usabilidad con el Sistema de Escalas de Usabilidad (SUS),” uiFromMars. Accessed: Jul. 23, 2025. [Online]. Available: https://www.uifrommars.com/como-medir-usabilidad-que-es-sus/

[30] Martins Julia, “¿Qué es la metodología Kanban y cómo funciona? [2024] • Asana,” Asana. Accessed: Dec. 24, 2024. [Online]. Available: https://asana.com/es/resources/what-is-kanban

[31] Y. Feng *et al.*, “Automated medical literature screening using artificial intelligence: a systematic review and meta-analysis,” *Journal of the American Medical Informatics Association*, vol. 29, no. 8, pp. 1425–1432, Jul. 2022, doi: 10.1093/jamia/ocac066.

[32] A. Bannach-Brown *et al.*, “Machine learning algorithms for systematic review: reducing workload in a preclinical review of animal studies and reducing human screening error,” *Syst Rev*, vol. 8, no. 1, p. 23, Dec. 2019, doi: 10.1186/s13643-019-0942-7.

[33] K. B. Cohen, “Biomedical Natural Language Processing and Text Mining,” in *Methods in Biomedical Informatics*, Elsevier, 2014, pp. 141–177. doi: 10.1016/B978-0-12-401678-1.00006-3.

[34] R. Cierco Jimenez *et al.*, “Machine learning computational tools to assist the performance of systematic reviews: A mapping review,” *BMC Med Res Methodol*, vol. 22, no. 1, p. 322, Dec. 2022, doi: 10.1186/s12874-022-01805-4.