

# **Optimización de estrategias de búsquedas científicas médicas utilizando técnicas de inteligencia artificial**

Daniel Mauricio Nieto González

PONTIFICIA UNIVERSIDAD  
JAVERIANA FACULTAD DE  
INGENIERÍA  
MAESTRÍA EN INGENIERÍA DE SISTEMAS Y COMPUTACIÓN  
BOGOTÁ, D.C.

2021

# **Optimización de estrategias de búsquedas científicas médicas utilizando técnicas de inteligencia artificial**

**Autor:**

**Daniel Mauricio Nieto González**

**MEMORIA DEL TRABAJO DE GRADO REALIZADO PARA CUMPLIR UNO DE LOS  
REQUISITOS PARA OPTAR AL TÍTULO DE  
MAGÍSTER EN INGENIERÍA DE SISTEMAS Y COMPUTACIÓN**

**Director**

**Cesar Julio Bustacara Medina**

**Página web del Trabajo de Grado**

**<https://livejaverianaedu.sharepoint.com/sites/Ingsis/TGMISC/213006>**

**PONTIFICIA UNIVERSIDAD JAVERIANA FACULTAD DE INGENIERÍA  
MAESTRÍA EN INGENIERÍA DE SISTEMAS Y COMPUTACIÓN BOGOTÁ, D.C.**

**DICIEMBRE, 2021**

**PONTIFICIA UNIVERSIDAD JAVERIANA**  
**FACULTAD DE INGENIERÍA**  
**MAESTRÍA EN INGENIERÍA DE SISTEMAS Y COMPUTACIÓN**

**Rector Magnífico**

**Jorge Humberto Peláez, S.J.**

**Decano Facultad de Ingeniería**

**Ingeniero Lope Hugo Barrero Solano**

**Director Maestría en Ingeniería de Sistemas y Computación**

**Ingeniera Ángela Carrillo Ramos**

**Director Departamento de Ingeniería de Sistemas**

**Ingeniero Cesar Julio Bustacara**

**Artículo 23 de la Resolución No. 1 de Junio de 1946**

*“La Universidad no se hace responsable de los conceptos emitidos por sus alumnos en sus proyectos de grado. Sólo velará porque no se publique nada contrario al dogma y la moral católica y porque no contengan ataques o polémicas puramente personales. Antes bien, que se vean en ellos el anhelo de buscar la verdad y la Justicia”*

## **AGRADECIMIENTOS**

Agradezco enormemente a mi hermosa novia Carolina Pantoja, quien en cada momento del difícil recorrido que fue diseñar y traer a la realidad un proyecto tan complejo, estuvo pacientemente acompañándome y apoyándome, a pesar de los incontables días sin descansar.

A mi familia, a mis padres Gilberto Nieto y Doris González por el gran apoyo a lo largo de toda la carrera, y por la formación que me brindaron que permitió conseguir este logro.

También agradezco a mi director de tesis César Julio Bustacara, quien a pesar de los constantes retos que representó el proyecto, de los inconvenientes que iban surgiendo a lo largo de él, y de la impaciencia que me iba tomando hacia el final del proyecto, no perdió la paciencia y la confianza en que podría lograrlo, y en que con las ideas que ambos habíamos pensado, todo se iba a resolver.

## Tabla de Contenido

<b>1. DESCRIPCION GENERAL</b>	<b>12</b>
1.1. Oportunidad y problemática	12
1.1.1. Descripción del flujo de trabajo	12
1.1.2. Descripción de la solución de inteligencia artificial	13
<b>2. DESCRIPCION DEL PROYECTO</b>	<b>14</b>
2.1. Objetivo general	14
2.2. Objetivos específicos	14
2.3. Fases de desarrollo	14
<b>3. MARCO TEÓRICO</b>	<b>16</b>
3.1. La importancia de las revisiones sistemáticas	16
3.2. Marcos metodológicos para la elaboración de revisiones sistemáticas	16
3.3. Pasos para elaborar una revisión sistemática utilizando PRISMA y PICOS	17
3.4. Limitaciones y dificultades al realizar revisiones sistemáticas	20
3.5. ¿Por qué son necesarias las herramientas de automatización de revisiones sistemáticas?	20
3.6. Características y fases de las revisiones sistemáticas de interés para la automatización	21
3.7. Automatización de búsquedas, análisis de textos y PNL	21
3.8. ¿Qué debemos tener en cuenta para desarrollar un software que mejore el flujo de trabajo para la elaboración de revisiones sistemáticas?	22
<b>4. TRABAJOS RELACIONADOS</b>	<b>24</b>
4.1. Herramientas de automatización de revisiones sistemáticas y tareas automatizables	24
4.2. Descripción y comparación de las herramientas que cubren un mayor número de fases de la elaboración de revisiones sistemáticas	24
4.3. Herramientas de automatización de revisiones sistemáticas y características deseadas por los usuarios	26
4.4. Diferencias y puntos en común de las herramientas que buscan incluir las características deseadas por los usuarios	27
4.5. Diferencias y puntos en común de los marcos metodológicos existentes	28
4.6. Técnicas de inteligencia artificial utilizadas en otras herramientas	29
4.7. ¿Qué se requiere, además de conocer las características requeridas por todos los Stakeholders para construir una herramienta de automatización de elaboraciones sistemáticas potente?	31
<b>5. DESARROLLO DEL PROYECTO</b>	<b>32</b>
5.1 Diseño de la solución	32
5.1.1 Selección de componentes a implementar en la versión 1.0	32
5.1.2 Requerimientos funcionales	33
5.1.2.1 Requerimientos funcionales desde la perspectiva de las características deseadas por el usuario	33
5.1.2.2 Requerimientos funcionales desde la perspectiva de los marcos metodológicos	34

---

5.1.2.3 Componentes funcionales del sistema basados en Harrison (2020)	34
5.1.3 Requerimientos no funcionales:	35
5.1.4. Diseño arquitectónico	39
5.1.4.1 ADD	39
5.1.4.2 Patrones Arquitecturales	39
5.1.4.2.1 Primera iteración	40
5.1.4.2.2 Segunda iteración	41
5.1.4.3 Vistas 4 + 1	43
5.1.4.3.1 Vista de Casos de Uso	43
5.1.4.3.2 Vista Lógica	43
5.1.4.3.3 Vista de Despliegue	43
5.1.4.3.4 Vista física	43
5.1.5. ATAM	47
5.2. Módulo de Inteligencia Artificial	49
5.2.1 Marco metodológico: CRISP-DM	49
5.2.2 Entendimiento de negocio	50
5.2.3 Entendimiento de los datos	50
5.2.3.1 Datos de entradas a los módulos de inteligencia artificial.	50
5.2.3.2 Datos de salida de los módulos de inteligencia artificial	51
5.2.3.3 Fuentes de información.	51
5.2.3.4 Variables de interés.	52
5.2.4 Procesamiento de los datos	52
5.2.5 CRISP-DM: Modelado	53
5.2.6 CRISP-DM: Evaluación	53
5.2.7 CRISP-DM: Despliegue	53
5.3. Implementación de la solución	53
5.4 Implementación del modulo de Inteligencia Artificial	54
5.4.1 Procesamiento de Lenguaje Natural	54
5.4.2 Information Retrieval.	55
5.4.3 Clasificación utilizando Machine Learning	57
<b>6. PRUEBAS Y RESULTADOS</b>	<b>59</b>
6.1. Pruebas de funcionalidad	59
6.2. Garantía de alta calidad de los resultados	64
6.3 Resultados de la optimización	64
6.3.1 Optimización en tiempo	69
6.3.2 Optimización en calidad	64
<b>7. CONCLUSIONES</b>	<b>69</b>

7.1 Conclusiones del proyecto	69
<b>8. TRABAJO FUTURO</b>	<b>70</b>
8.1 Optimización en calidad	70
<b>9. REFERENCIAS</b>	<b>71</b>
<b>10. ANEXOS</b>	<b>79</b>



## RESUMEN

Este proyecto diseñó e implementó un prototipo de software funcional, capaz de optimizar el proceso y los resultados de las estrategias de búsqueda y la selección de artículos, al elaborar una revisión sistemática médica. La aplicación utilizó inteligencia artificial para apoyar la toma de decisiones en las fases más críticas del proceso. Se logró sugerir qué artículos incluir o excluir, utilizando un modelo Naive Bayes que logró clasificar con un 79,66% de precisión, y un AUC de 91,13% los manuscritos con base en clasificaciones previa realizadas por el usuario. Se utilizó para la extracción de información un algoritmo TF-IDF regularizado con L2, y procesamiento de lenguaje natural incluyendo stemming. Se logró también sugerir qué palabras clave incluir o excluir con el algoritmo TD-IDF. Al comparar los resultados con las métricas de otras herramientas, los AUC y precisiones fueron buenos. Queda pendiente explorar más a fondo WSS@95. Para el diseño arquitectónico se identificaron los RNF y RF, también se utilizaron las metodologías ADD y ATAM, elaborando vistas 4+1, lo cual permitió diseñar e implementar una arquitectura cliente-servidor con un servidor en capas, en donde el módulo de inteligencia artificial se desplegó como servidor backend en una de las 4 aplicaciones construidas. Se incluyeron las mejores prácticas de seguridad tales como encriptación, whitelisting y uso de tokens JWT. La arquitectura construida utilizó servicios de la infraestructura de la nube de AWS, tales como DynamoDB, S3 Buckets y RDS Postgres, que junto con el diseño RESTful de las aplicaciones permiten una alta escalabilidad.

## ABSTRACT

This project designed and implemented a functional software prototype, capable of optimizing the process and results of search strategies and the selection of articles when preparing a medical systematic review. The application used artificial intelligence for decision-making support in the most critical phases of the process. It was possible to suggest which articles to include or exclude, using a Naive Bayes model that managed to classify the manuscripts with 79.66% precision, and an AUC of 91.13% based on previous classifications made by the user. A TF-IDF algorithm regularized with L2, and natural language processing including stemming, was used for information extraction. It was also possible to suggest which keywords to include or exclude with the TD-IDF algorithm. When comparing the results with the metrics of other tools, the AUC and accuracies were good. For the architectural design, the RNF and RF were identified, the ADD and ATAM methodologies were also used, developing 4 + 1 views, which allowed the design and implementation of a client-server architecture with a layered server, where the artificial intelligence module it was deployed as a backend server in one of the 4 applications built. The best security practices such as encryption, whitelisting and use of JWT tokens were included. The architecture built used services of the AWS Cloud infrastructure such as DynamoDB, S3 Buckets and RDS Postgres which altogether to the RESTful design of the applications permitted a high scalability.

## RESUMEN EJECUTIVO

En la investigación biomédica se utilizan las revisiones sistemáticas como fuentes de información, son de gran relevancia debido a que sintetizan toda la literatura científica sobre un tópico, permitiendo la elaboración de lineamientos como las guías de prácticas clínicas. Desafortunadamente, la mayoría de las revisiones no cumplen los estándares de calidad, dificultando esto.

Dado el potencial que tienen las soluciones tecnológicas para mejorar los resultados y reducir el riesgo de errores, se han construido más de 200 herramientas que buscan facilitar la elaboración de revisiones sistemáticas. Este proyecto se basó en los requerimientos y necesidades identificados en la literatura, tanto desde la perspectiva de los usuarios como desde la de los marcos metodológicos

El objetivo de este proyecto fue diseñar e implementar un prototipo de software funcional que fuera capaz de optimizar el proceso y los resultados de las fases de búsqueda y selección de artículos durante la elaboración de una revisión sistemática médica, utilizando un sistema de soporte en la toma de decisiones potenciado por técnicas de inteligencia artificial. Se implementó un módulo capaz de sugerir al investigador biomédico qué palabras clave podría incluir, y cuáles podría eliminar, durante la elaboración de la estrategia de búsqueda. Y qué manuscritos podría incluir o no, durante la fase de selección de artículos PRISMA.

Tras identificar los requerimientos funcionales y no funcionales, se utilizó la metodología Attribute Driven Design (ADD) para diseñar la arquitectura de la aplicación y Architecture Tradeoff Analysis Method (ATAM) para evaluar los tradeoffs de las decisiones arquitectónicas tomadas. También se elaboraron las vistas 4+1 incluyendo las vistas de casos de uso, lógica, de implementación, y física. Todo esto permitió identificar que los patrones más importantes en la arquitectura de Digital Medtools fueron el patrón de arquitectura en capas y el patrón cliente-servidor.

En la capa de presentación, la más exterior y la cual es el único punto de acceso hacia las demás capas, se construyó una aplicación frontend utilizando Angular, en la capa lógica se construyeron 3 aplicaciones tipo servidor RESTful backend utilizando Python y Django, y en la capa de Cloud Services, se construyeron diferentes infraestructuras de AWS como Parameter Stores, S3 Buckets y bases de datos DynamoDB y RDS PostgreSQL.

El módulo de inteligencia logró clasificar los artículos utilizando machine learning (SVM, Bayes Naive) además de PLN y TF-IDF. Se obtuvo una precisión de 79%, y un AUC. De 91%, los cuales son resultados muy buenos para un módulo de procesamiento natural con capacidades de inteligencia artificial. Adicionalmente vale la pena aclarar que se utilizó la regularización L2, validaciones cruzadas 5-fold, y 20 ejecuciones por cada combinación de hiperparámetros por modelo, para garantizar la máxima calidad y veracidad de estos resultados. Se ejecutaron un total de 3300 modelos, combinando hiperparámetros para los algoritmos de aprendizaje de máquina SVM y Bayes Naive.

## INTRODUCCIÓN

Las revisiones sistemáticas, metaanálisis y *scoping reviews*, son algunos de los tipos de estudio más importantes en la investigación biomédica ya que sintetizan y compilan los hallazgos de múltiples fuentes científicas sin que los lectores tengan que acceder directamente a ellas, permitiendo la elaboración de documentos de síntesis de literatura, como lineamientos hospitalarios y guías de práctica clínica.

Para que una revisión sistemática sea de alto valor se requiere que colecte la mayor cantidad de artículos relevantes sobre un tópico, y que excluya aquellos no relevantes, de baja calidad, o que no se ajustan a los criterios definidos durante la fase de elaboración del protocolo. Desafortunadamente, se ha identificado que la mayoría de las revisiones sistemáticas presentan baja calidad tras evaluarlas con los marcos metodológicos de referencia. Estos contienen listas de verificación que puntúan el diseño de una revisión sistemática.

El proyecto Digital Medtools, en el marco del cual se desarrolla este proyecto, cuenta con varios años de experiencia interactuando con investigadores biomédicos. Mediante esta experiencia se ha podido identificar que dos de los procesos más críticos al construir una revisión sistemática, donde se invierten más recursos y se corre mayor riesgo de cometer errores, son la construcción de la estrategia de búsqueda y la selección de artículos. Con el fin de contrarrestar esta situación, en este proyecto se logró diseñar e implementar una solución tecnológica capaz de:

1. Sugerir cuáles de las palabras clave utilizadas en una estrategia de búsqueda deberían removerse y cuales podrían incluirse, con el objetivo de mejorar los resultados.
2. Sugerir cuáles artículos científicos podrían incluirse o excluirse, para facilitar el proceso de selección de manuscritos.

Este manuscrito está compuesto por 7 capítulos: 1. Descripción General: Se expone la oportunidad identificada y la solución propuesta para abordarla. 2. Descripción del proyecto: Incluye los objetivos generales y específicos, y las fases de desarrollo del proyecto. 3. Marco Teórico: Explora qué son las revisiones sistemáticas, su importancia, características, y fases para la elaboración. Incluye también conceptos sobre procesamiento de lenguaje natural (PLN) e inteligencia artificial (IA), que serán abordados con más detalle en la sección de inteligencia artificial. 4. Trabajos relacionados: Expone las herramientas para automatización de revisiones sistemáticas más completas y mejor valoradas, sus características y una comparación entre ellas. También presenta los marcos metodológicos utilizados para evaluar la calidad de revisiones sistemáticas. 5. Desarrollo del proyecto: Incluye, por un lado, el diseño de la solución, donde se abordan los requerimientos funcionales y no funcionales, las vistas arquitectónicas, los patrones arquitectónicos escogidos, y, por otro lado, el módulo de inteligencia artificial, donde se presenta la forma en la que se utilizó la inteligencia artificial para lograr los objetivos del proyecto, esto, en el marco de la metodología CRISP-DM. Incluye diagramas para representar los flujos de información más complejos. 6. Resultados: donde se presenta el Portal web construido, se demuestra el flujo de trabajo y los resultados de la inteligencia artificial aplicada. 7. Conclusiones. 8. Trabajos futuros: donde se presenta qué puede realizarse a futuro en el marco de este proyecto. 9. Referencias bibliográficas. 10. Anexos.

Aunque este proyecto es presentado bajo la modalidad de profundización, no se limitó a construir únicamente un archivo 1-file en jupyter notebooks, como la mayoría de los proyectos de inteligencia artificial. En este proyecto se logró construir una aplicación backend RESTful de inteligencia artificial integrada y consumida por otras aplicaciones RESTful, también construidas durante este proyecto. Se utilizaron igualmente tecnologías de Amazon Web Services (AWS) como S3, DynamoDB, RDS, y se siguieron las metodologías ADD y ATAM para diseño y evaluación de arquitecturas de software y CRISP-DM para proyectos de ciencia de datos. Fue clave para el éxito del proyecto haber podido identificar todos los requerimientos funcionales y no funcionales de relevancia, a partir de la literatura.

## **1. DESCRIPCION GENERAL**

### **1.1. Oportunidad y problemática**

Digital Medtools es un emprendimiento orientado a ofrecer servicios a los investigadores biomédicos. Uno de sus portafolios consiste en asesorar al cliente en la construcción de las estrategias de búsqueda para sus proyectos de elaboración de revisiones sistemáticas, lo cual es realizado por expertos en el área de bibliotecología que utilizan las soluciones tecnológicas construidas dentro de Digital Medtools para optimizar el proceso.

A partir de la experiencia del proyecto Digital Medtools interactuando con investigadores biomédicos por más de 4 años, y de la evidencia encontrada en la literatura, como se presentará posteriormente, fue posible identificar dos procesos críticos al construir revisiones sistemáticas donde se invierten la mayor cantidad de recursos y esfuerzo y existe la mayor probabilidad de cometer errores que reduzcan la calidad del trabajo. Estos son la construcción de la estrategia de búsqueda y la clasificación de artículos según los criterios de exclusión. Por un lado, para construir una excelente estrategia de búsqueda se requiere el conocimiento técnico y formación adecuada para entender las operaciones lógicas que ocurren al realizar las conjunciones, o en su defecto una herramienta tecnológica que permita facilitar la construcción de las mismas. Por otro lado, para realizar la clasificación de artículos más fácilmente, se requiere una solución tecnológica que permita facilitar el proceso, ya que esto se realiza de forma manual, haciéndolo propenso a errores y demorado. Para una versión 1.0 de la aplicación que se expondrá en este proyecto, Digital Medtools estará enfocado en estos dos procesos.

Este proyecto se diferencia de los más de 200 que construyeron aplicaciones similares, plenamente identificados y caracterizados, en que:

1. Busca la forma más costo eficiente de realizar la revisión sistemática en términos de tiempo, calidad y recursos
2. Su diseño estuvo enfocado a objetivos desde el primer momento. Se identificaron las características más prioritarias desde la perspectiva de los investigadores biomédicos y desde la perspectiva de los marcos metodológicos que evalúan la calidad de las revisiones sistemáticas
3. Ofrece el servicio de asesoría en la construcción de estrategias de búsqueda, en vez del uso de la aplicación manejada totalmente por el usuario.

#### **1.1.1. Descripción del flujo de trabajo**

Digital Medtools consta de un portal digital dividido en un portal para clientes y otro para staff. A través del portal de clientes, los investigadores biomédicos pueden solicitar que les sea creado un proyecto de revisión sistemática.

Una vez este proyecto es creado por el asesor, entre asesor y cliente se coordina una reunión donde se comparte pantalla y el asesor accede al portal de staff, para crear la estrategia de búsqueda utilizando el panel de revisiones sistemáticas. Los asesores, con base en las indicaciones del cliente, construyen la estrategia, y la guarda cuando considere necesario. El cliente no tiene acceso a crear o modificar la estrategia de búsqueda ya que el asesor, gracias a su formación en bibliotecología, puede hacer esta tarea de forma más eficiente.

Una vez entre cliente y asesor han construido la primera versión de la estrategia de búsqueda se procede a mejorar el set de palabras clave escogidas. Para lograrlo se requiere que el algoritmo de inteligencia artificial sea capaz de reconocer que palabras clave se encuentran o se omiten en unos manuscritos encontrados al realizar una búsqueda preliminar, y que el cliente deberá clasificar como “incluir” “neutro” y “excluir”.

Para que el usuario pueda visualizar la lista de referencias que deberá clasificar, el asesor previamente debe haber completado la metadata faltante de estas referencias, en una pantalla diseñada específicamente para facilitar el proceso. Los elementos más importantes de la metadata faltante son los abstracts, a los cuales no es posible acceder automáticamente debido a las restricciones legales de SCOPUS, y a que el proyecto actualmente no se encuentra suscrito a ninguna dependencia de la Pontificia Universidad Javeriana, sino al proyecto Digital Medtools.

Para intentar obtener toda la metadata disponible el proyecto se conecta a SCOPUS y Crossref. También intenta buscar si el abstract ya existe en la base de datos de Digital Medtools. En caso que el resumen no sea encontrado, el asesor con solo presionar un botón accederá a una página web donde podrá copiar y pegar el abstract para la respectiva referencia. No es posible hacer web scrapping sobre este enlace dado que la mayoría de ellos se encuentran protegidos por CloudFare, haciendo difícil que el host no reconozca al cliente Web Scraper de Digital Medtools como un robot.

Una vez el usuario ha clasificado los artículos, se puede visualizar el puntaje de las palabras clave, y mejorar la estrategia de búsqueda utilizando esta información.

Finalmente, cuando el cliente reconoce que la estrategia de búsqueda es lo suficientemente buena, se continua a la fase de screening, donde el cliente deberá realizar una clasificación definitiva sobre que manuscritos desea incluir o excluir para la elaboración del contenido de su revisión sistemática.

Para facilitar este proceso, la aplicación tiene pantallas diseñadas para hacerlo más sencillo, y además, para cada referencia presenta la sugerencia de si ese artículo se podría incluir o excluir, basado en los patrones que el algoritmo de inteligencia artificial identificó en los artículos previamente clasificados por el cliente para este proyecto.

Una vez los artículos se han clasificado termina el flujo de trabajo para la versión 1.0 de Digital Medtools. En ese momento el cliente recibe un set de resultados extremadamente útil para escribir el contenido de su revisión sistemática.

### **1.1.2. Descripción de la solución de inteligencia artificial**

Para sugerir al usuario que palabras clave debería eliminar y cuales podría incluir para mejorar su estrategia de búsqueda, existe un módulo capaz de convertir el texto en lenguaje natural a un set de características numéricas, de modo que cada artículo científico se puede representar según la frecuencia inversa de las palabras que lo componen.

Para la fase de mejorar las palabras clave se busca hacer el match entre las palabras clave y la presencia en la representación del artículo, de modo que esta relación se puede representar numéricamente. A partir de eso, aquellas palabras clave que tengan una mayor presencia en los artículos calificados como incluido serán clasificadas como de utilidad y aquellas que tengan más presencia en los artículos calificados negativamente serán clasificadas como “por eliminar”.

Las palabras conectoras más comunes como “the”, “with”, “and”, etc., que podrían eventualmente puntuar alto en ambos factores, son eliminadas en la fase previa del módulo de procesamiento de lenguaje natural, de modo que el ruido se reduce.

En cuanto a la clasificación de artículos, aquellos que tengan más palabras clasificadas positivamente serán puntuados mejor, y aquellos que tengan más palabras puntuadas negativamente recibirán una penalización.

## 2. DESCRIPCION DEL PROYECTO

### 2.1. Objetivo general

El objetivo de este proyecto es diseñar e implementar un prototipo de software funcional que sea capaz de optimizar el proceso y los resultados de las fases de búsqueda y selección de artículos durante la elaboración de una revisión sistemática médica, utilizando un sistema de soporte en la toma de decisiones potenciado por técnicas de inteligencia artificial.

### 2.2. Objetivos específicos

- Revisar y analizar la bibliografía existente.
- Diseñar la arquitectura de las plataformas del proyecto.
- Procesar los datos y entrenar y evaluar los modelos de inteligencia artificial en el marco de CRISP-DM.
- Construir el prototipo funcional.

### 2.3. Fases de desarrollo

El proyecto se divide en 5 fases, cada una de ellas permite lograr uno de los objetivos específicos propuestos. A continuación, se describe cada fase:

- **Fase 1. Investigación preliminar.** Revisión bibliográfica de información. Se busca identificar el set de técnicas de inteligencia artificial a utilizar, soportar los requerimientos funcionales y no funcionales del proyecto, entender los datos y el negocio (CRISP-DM) [1-3]. Se elaborará el estado del arte del documento de grado. En esta fase también se realiza el entendimiento de negocio y de los datos, lo cual permite en la siguiente fase construir el módulo de Inteligencia Artificial.
- **Fase 2. Diseño arquitectónico.** Dado que no se encontró una arquitectura de referencia, se construirá desde cero la solución. Se utilizará Attribute Driven Design (ADD) para su diseño [6,7]. Se elaborará el documento de arquitectura de software (SAD) donde también estarán contenidas las vistas arquitecturales 4+1 y los casos de uso [8-10]. Se utilizará la metodología Architecture Tradeoff Analysis Method (ATAM) para mitigar los riesgos de la arquitectura escogida [7,11,100]. Finalmente se utilizará Bootstrap 4 para construir los mockups. Un punto importante de esta fase es construir la aplicación donde va a estar alojado el algoritmo de inteligencia artificial. También se buscará construir los componentes del módulo de inteligencia artificial de modo que sea posible realizar el proceso de inicio a fin sin entrar en detalles avanzados de la implementación de la inteligencia artificial. Estos componentes hacen parte del ETL (Extract, Transform, Load).
- **Fase 3. Procesamiento de los datos, entrenamiento y evaluación de los modelos de inteligencia artificial.** En la fase 2 se logrará establecer un flujo básico de trabajo al construir el módulo ETL. En esta fase se buscará ajustarlo. Se construirá una pregunta problema con unos criterios básicos de inclusión y exclusión, se realizará la búsqueda en SCOPUS y se elegirán manualmente los resultados considerados aptos. Este set de resultados será considerado el “gold standard” o la fuente de veracidad.

Para el módulo de inteligencia artificial que sugiere que artículos incluir y excluir, se buscara que el algoritmo se aproxime tanto como sea posible a sugerir estos resultados. En cuanto al módulo de

inteligencia artificial que sugiere las palabras clave, se buscara que al cambiar las palabras clave, el algoritmo sea capaz de sugerir incluir palabras nuevas o excluir palabras de poco valor para la estrategia, de modo que, al realizar la búsqueda, en los resultados se encuentren tantos manuscritos del gold standard como sea posible y la menor cantidad posible de los que no se encuentren en el “gold standard”. La métrica de interés para elegir el mejor modelo es el AUC (Area Under the ROC Curve), al comparar contra el Gold Standard [4].

Para procesar el texto en lenguaje natural, en el que se encuentran los manuscritos científicos, y obtener una matriz numérica que pueda ser entendible por los algoritmos, se utilizarán técnicas de Information Retrieval. Se obtendrá la frecuencia inversa de las características que representan al set de manuscritos utilizando TF-IDF. Para realizar la partición de datos y posterior validación, cuando sea requerido por los algoritmos, se utilizará 5-fold cross validation, una técnica con excelente desempeño [5].

- **Fase 4. Implementación del prototipo funcional.** Se utilizará la metodología de desarrollo ágil Lean Software Development, adaptándola a un equipo de trabajo compuesto por un solo desarrollador, con el fin de construir un producto mínimo viable funcional e iterativamente mejorarlo [12,13].
- **Fase 5. Elaboración del documento de trabajo de grado.** Se terminará de escribir y corregir tanto el documento de trabajo de grado como la presentación.

### 3. MARCO TEÓRICO

En esta sección se implementa la primera fase de desarrollo del presente documento, asociado a la revisión y análisis del estado del arte. En primer lugar, se incluye el contexto sobre revisiones sistemáticas, su propósito, fases, limitaciones, restricciones, inconvenientes. Se aborda el flujo de trabajo para crear una revisión sistemática, que fases se pueden automatizar, de qué forma se pueden aplicar las técnicas de procesamiento de lenguaje natural y de inteligencia artificial y qué algoritmos se conocen.

#### 3.1. La importancia de las revisiones sistemáticas

Las revisiones sistemáticas, metaanálisis y scoping reviews son tipos de estudio claves en la investigación biomédica en el contexto de la medicina basada en evidencia (EBM) [14]. Estos estudios buscan recopilar los hallazgos de todos los manuscritos científicos de un tópico en específico, para sintetizar de forma concisa, analítica y lógica conclusiones generales de interés para ese tópico [14, 15]. Dado que las revisiones sistemáticas, meta-análisis y scoping reviews permiten sintetizar la información de múltiples fuentes, sin tener que acceder directamente a todas ellas, y sin requerir conocer a profundidad cada uno de los manuscritos, sus hallazgos, su metodología y sus limitaciones, son una base importante para construir lineamientos hospitalarios y guías de práctica clínica, las cuales dictan las decisiones que se toman día a día por todos los profesionales de la salud alrededor del mundo [14-18].

Para hacer revisiones sistemáticas de calidad es fundamental lograr acceder a la mayor cantidad de manuscritos científicos del área de interés, esto se logra diseñando a detalle los queries de búsqueda, mejor conocidos en el campo de la investigación biomédica como estrategias de búsqueda [19]. Por otro lado, la búsqueda no debe ser demasiado genérica, abierta o desenfocada, ya que esto aumenta considerablemente el tiempo y esfuerzo invertido en filtrar los resultados para encontrar los artículos elegibles. Esto cobra más importancia cuando el número de manuscritos científicos publicados incrementa anualmente, y se hace cada año más difícil recopilarlos todos, y entender el contenido, hallazgos y metodología de cada uno [17-19].

Dada la importancia de las revisiones sistemáticas como puente entre la experimentación científica y el conocimiento médico práctico, se hace indispensable buscar mejorar la calidad de las mismas [17-19]. Por tanto, se han propuesto diversos marcos metodológicos que buscan además de mejorar la calidad, facilitar el flujo de trabajo y reducir la dificultad de los proyectos [20-22].

#### 3.2. Marcos metodológicos para la elaboración de revisiones sistemáticas

Entre los marcos metodológicos más conocidos se encuentran PRISMA y PICOS. PRISMA revisa la calidad del estudio a través de un checklist, calificando la información sobre fuentes de datos, criterios de inclusión y exclusión, estrategia de búsqueda, selección de estudios, proceso de recolección de datos, elaboración de la búsqueda, entre otros [23,24]. Por otro lado, PICOS segmenta de forma entendible para los investigadores la construcción de la estrategia de búsqueda en: Population, Intervention, Comparison, Outcome, Study type [25,26].

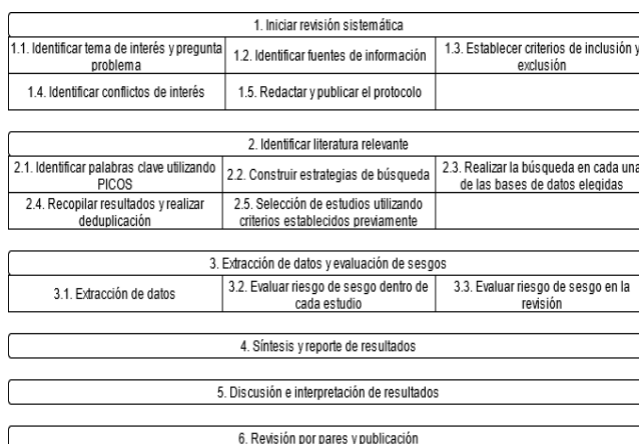
Adicionalmente, existen otros marcos o herramientas metodológicas. AMSTAR, acrónimo para A MeaSurement Tool to Assess Systematic Reviews, permite evaluar la calidad metodológica de una revisión sistemática a través de un checklist de 16 ítems [20-22, 27]. ROBIS por otro lado, es una herramienta que permite identificar el riesgo de sesgo en revisiones sistemáticas. Tiene 3 fases: 1. Identificar relevancia (Opcional), 2. Identificar preocupaciones con el proceso de revisión y 3. Juzgar el riesgo de sesgo. ROBIS evalúa el riesgo de sesgo en 4 dominios: 1. Especificación de criterios de elegibilidad de estudios. 2. Métodos usados para identificar y/o seleccionar estudios. 3. Métodos usados para coleccionar datos y evaluar estudios. 4. Síntesis y hallazgos [28, 29]. Estas no son herramientas tecnológicas sino metodológicas.



Con base en la evidencia de la eficacia de estas metodologías y herramientas, se podría decir que para elaborar una revisión sistemática de alta calidad se debería, desde un primer momento, pensar en cumplir la mayor cantidad de elementos posibles de las 4 metodologías.

### 3.3. Pasos para elaborar una revisión sistemática utilizando PRISMA y PICOS

La elaboración de una revisión sistemática está compuesta por múltiples fases. Inicialmente a partir de la pregunta problema se realiza la construcción del esquema de palabras clave y se establecen unos criterios de inclusión y exclusión de manuscritos. A partir de estas palabras clave se construye un esquema de estrategia de búsqueda, para posteriormente realizar la búsqueda en varias bases de datos científicas. Luego se recopilan los resultados de todas las bases de datos científicas y se realiza la deduplicación, un proceso que consiste en eliminar las referencias que se repiten entre los resultados de distintas bases de datos. Posteriormente, los revisores clasifican cada uno de los manuscritos hallados utilizando los criterios de inclusión y exclusión, generando un punto de filtro que suele reportarse en el diagrama PRISMA. Luego, se hace un filtro final, leyendo a detalle cada uno de los manuscritos para excluir aquellos que no se ajustan. Finalmente, se elabora el documento con base en el contenido de cada uno de los manuscritos incluidos [30].



**Figura 1.** Flujo de trabajo en una revisión sistemática [30,31]

- **Paso 1.1. Identificar tema de interés y pregunta problema**

Se debe formular una pregunta problema que se pueda responder consultando y analizando la literatura existente y sus métodos y hallazgos [31].

- **Paso 1.2. Identificar fuentes de información**

Se deben identificar las bases de datos, fuentes de literatura gris, revistas de interés, donde se va a realizar la búsqueda de manuscritos científicos. También definir si se va a realizar estrategia Snowball [30].

- **Paso 1.3. Establecer criterios de inclusión y exclusión**

Se establecen criterios como año del estudio, tipo de estudio, lenguajes, región donde fue realizado, tipo de metodologías usadas, contexto del estudio, entre otros [31].

- **Paso 1.4. Identificar conflictos de interés**

Identificar y reportar situaciones de conflicto de interés entre los miembros del equipo, el área de estudio y los estudios y/o intervenciones estudiadas [31].

- **Paso 1.5. Redactar y publicar el protocolo**

El protocolo contiene la pregunta problema, la metodología a usar y un amplio marco teórico que servirá de base para realizar la revisión sistemática [31].

- **Paso 2.1. Identificar palabras clave utilizando PICOS**

En esta fase se suele utilizar la metodología PICOS para transformar un objetivo de proyecto o pregunta problema en un set de palabras clave que representa a la temática de búsqueda. Esta metodología permite segmentar una pregunta problema en cinco elementos cuya conjunción describe la temática de interés [31]. Los segmentos de una búsqueda PICO incluyen:

**Paciente / Problema / Población:** Hace referencia al contexto, situación problema, región, características sociodemográficas o de interés médico de los participantes del estudio [25,26,32].

**Intervención:** Intervenciones, situaciones y/o tratamientos a los que se exponen los sujetos del elemento anterior. Si el objetivo del estudio está más orientado a presentar cuáles existen y/o qué diferencias y similitudes tienen que, al compararlos, se pueden colocar todos en esta sección [25,26,32].

**Comparación:** Otras intervenciones, situaciones y/o tratamientos a los que se exponen los sujetos del elemento anterior. Se debe tener presente que PICO hace una conjunción (operación AND) entre sus elementos, por lo que, si se colocan elementos tanto en esta sección como en intervención, cada uno de los artículos encontrados deberá contener palabras de ambas secciones [25,26,32].

**Outcome:** En esta sección se hace referencia a qué impactos, desenlaces, o métricas son de interés. También, qué posibles desenlaces o consecuencias se desean evaluar en la población [25,26,32].

**Study:** Tipos de estudio permitidos. Es común que se excluyan otras revisiones sistemáticas o que se incluyan tipos de estudio como ensayos controlados aleatorizados [25,26,32].

Las palabras clave escogidas en esta fase deben estar optimizadas para búsquedas, es decir en vez de usar lenguaje natural, se debe procurar usar lenguaje controlado, o lenguaje científico conciso.

- **Paso 2.2. Construir estrategias de búsqueda**

A partir de las palabras clave de la fase 1.3.1 se elabora una estrategia de búsqueda que haga conjunción de los segmentos del esquema PICO utilizando los operadores AND, OR y NOT, y agrupaciones en paréntesis [33]. Idealmente la búsqueda debería refinarse lo suficiente para aumentar la sensibilidad, es decir, encontrar más artículos que tengan información relevante del tema, y la especificidad, es decir, evadir artículos que no aportan información al tópico [30, 33].

- **Paso 2.3. Realizar la búsqueda en cada una de las bases de datos elegidas**

Se debe construir una estrategia de búsqueda específica para cada base de datos. Se debe tener en cuenta que las bases de datos presentan diferentes formatos de operadores de campo (Para buscar solo en ciertas secciones del documento) y de la estructura de las estrategias en sí. Las estrategias no deben ser complejas o anidadas ya que los buscadores menos potentes no las permiten [30,33].

- **Paso 2.4. Recopilar resultados y realizar deduplicación**

En cada una de las bases de datos, se exportan los resultados encontrados para la búsqueda, por lo general en formato RIS, el cual contiene toda la información de la metadata de cada artículo. Esta información se recopila para cada una de las bases de datos en un repositorio central, donde se hace referencia a la fuente de origen de cada artículo [34-36].

Posteriormente, se realiza la deduplicación, donde se remueven las referencias duplicadas entre las bases de datos. Este es un proceso de cuidado ya que referencias con nombres similares, incluso a veces con el mismo nombre, pueden no pertenecer a un mismo manuscrito, por lo que es recomendable elegir herramientas adecuadas para realizar este proceso [30,35,36].

Para este paso se requiere tener una buena documentación que permita dar soporte de la validez del estudio ante los requerimientos de la metodología PRISMA [23,24].

- ***Paso 2.5 Selección de estudios utilizando criterios establecidos previamente***

Dos investigadores, por separado, evalúan cada una de las referencias teniendo en cuenta los criterios de inclusión y exclusión. Idealmente, la herramienta o metodología usada debería permitir que los investigadores no tuvieran acceso a la calificación del otro, lo cual se conoce como evaluación ciega.

Luego, en cada una de las referencias donde no hubo un match entre la decisión de ambos investigadores, entra un tercer investigador que se encarga de desempatar el puntaje, añadiendo su voto, de modo que cada referencia queda incluida o excluida. Adicionalmente se construye un diagrama de flujo para cumplir los requisitos de la metodología PRISMA [23,24,30].

- ***Paso 3.1. Extraer datos***

En esta fase se extraen todos los datos que permitan responder la pregunta de estudio y/o que estén incluidos en las variables de estudio seleccionadas. Estos datos por lo general están resumidos en tablas y/o gráficos, dentro de un documento PDF, por lo que es necesario, primero, transcribirlos a mano o usar herramientas de extracción de datos, y segundo, verificar que la transcripción fue correcta. En esta etapa también es importante entender el contexto y limitaciones de la información e interpretarla adecuadamente [30].

- ***Paso 3.2. Evaluar riesgo de sesgo dentro de cada estudio***

Para que los hallazgos y datos encontrados en un manuscrito se puedan usar en la elaboración de una revisión sistemática se deben evaluar el sesgo y las limitaciones dentro de su metodología y hallazgos. También es importante excluir estudios cuya metodología no sea clara o apropiada ya que pueden perjudicar la calidad de la interpretación del investigador. Los sesgos y limitaciones encontrados se deben reportar para darle claridad al peso del estudio dentro de la temática [30].

- ***Paso 3.3. Evaluar riesgo de sesgo en la revisión***

Adicionalmente, es necesario evaluar el sesgo y las limitaciones de la metodología y hallazgos de la revisión sistemática que está realizando el investigador. Esto teniendo en cuenta factores como la disponibilidad de información, la calidad de los estudios incluidos, la heterogeneidad, el contexto de cada estudio respecto a la temática de interés, entre otros [30].

- ***Paso 4. Síntesis y reporte de resultados***

Para realizar la síntesis de resultados se deben tener en cuenta los datos extraídos de cada estudio, su contexto, sus sesgos y sus limitaciones. Se le debe dar una adecuada interpretación a los datos que se van a usar. Se suelen utilizar medidas de estadística clínica como RR, OR [30]

- ***Paso 5. Síntesis de documento***

Posteriormente, los investigadores leen con detalle cada texto completo de cada uno de los artículos que pasaron el filtro, para decidir cuáles serán los manuscritos finales que se utilizarán para construir la revisión sistemática, con base en los criterios de inclusión y exclusión, la calidad de su metodología, entre otros. Luego, se busca responder la pregunta problema planteada inicialmente, identificando hallazgos y/o puntos

en común de distintos artículos que permitan inferir información para responder la pregunta. El investigador debe tener en cuenta la heterogeneidad de cada estudio, sus limitaciones y además inferir correctamente a qué hacen referencia los hallazgos de cada estudio [34-41].

#### • **Paso 6. Revisión por pares y publicación**

Una vez se ha recopilado la evidencia, el investigador terminará de elaborar el manuscrito científico, procurando completar los requisitos exigidos por alguna de las metodologías de referencia, como PRISMA, para finalmente proceder a someter a la revisión de pares su manuscrito para que sea publicado [37].

### **3.4. Limitaciones y dificultades al realizar revisiones sistemáticas**

La elaboración de revisiones sistemáticas no está libre de sesgo. Entre los problemas más comunes que se suelen presentar están:

- Uso de estrategias de búsqueda de bajo poder o desenfocadas: Ocasionalmente que los manuscritos encontrados no correspondan al área de interés y que no se pueda llegar tan cerca como sea posible al total de manuscritos disponibles sobre el tema.
- Selección de artículos mal realizada: Se pueden incluir manuscritos cuya información errónea o poco reproducible afecte la calidad de las conclusiones de la revisión sistemática.
- Ausencia de evaluación de la heterogeneidad: Comparar los resultados de los estudios sin tener en cuenta la varianza dentro de cada uno puede inducir a conclusiones erróneas.
- Interpretación errónea de los hallazgos: Interpretar de forma inadecuada los hallazgos de un estudio y su contexto afectará las conclusiones de la revisión sistemática. [34,35]

Esto es un problema mayor ya que se ha identificado que la mayoría de las revisiones sistemáticas presentan baja calidad. Por ejemplo, Storman et al. (2020) [36], encontró que alrededor del 99% de 4236 estudios entre 2016 y 2017, clasificados como revisiones sistemáticas en el campo de la cirugía bariátrica, fueron clasificados como con “una metodología críticamente pobre” por AMSTAR-2 y ROBIS. Yu et al. (2020) [38] advirtió que la mayoría de las revisiones sistemáticas, de un set de 49 evaluadas en el área de COVID-19, presentaron una pobre metodología y recomendó usar y adoptar las conclusiones obtenidas de estos estudios con cautela. Pollock et al. (2020) [20], de la misma forma, encontró que la calidad de las revisiones sistemáticas en el área de la investigación In Vitro fue baja, con un promedio de 68% de adherencia a los criterios PRISMA. Opheim et al. (2019) [40] encontró que, de un total de 20 revisiones en el área de estrés postraumático, solo el 70% presentaban un diagrama de flujo y solo 3 lo adaptaron al estamento PRISMA, y advirtió que los investigadores deberían explorar de qué forma las estrategias de búsqueda afectan sus resultados. Y Layton (2017) [41] resaltó que más del 95% de las revisiones sistemáticas en el campo de prostodoncia e implantes fallaron en usar estrategias de búsqueda sistemáticas, lo cual debilita sus conclusiones.

### **3.5. ¿Por qué son necesarias las herramientas de automatización de revisiones sistemáticas?**

A pesar de que los marcos metodológicos no son nuevos, y que su documentación es clara y concisa, los investigadores biomédicos, como se mencionó previamente, fallan al aplicarlas lo cual resulta en revisiones de baja calidad [22,23,25,29,41]. En parte esto se debe a que el proceso de elaboración de una revisión sistemática contiene varias tareas repetitivas, o muy intensivas en documentación. Un ejemplo es la construcción de las estrategias de búsqueda, un paso crítico que demanda un nivel de atención continuo, ya que errores pequeños como un paréntesis mal ubicado podría terminar en un resultado erróneo con gran probabilidad. Las estrategias de búsqueda deben ser lo suficientemente precisas para que se incluyan todos los artículos de interés, a la vez que se excluyan todos los manuscritos potencialmente no deseados [18,26,42-44]. Debido a estos retos han surgido decenas de proyectos que mediante la implementación de soluciones tecnológicas buscan automatizar partes del proceso de la elaboración de revisiones sistemáticas,

tanto para reducir el tiempo y esfuerzo invertidos, como para mejorar la calidad de la metodología, los hallazgos y la revisión sistemática en sí.

### **3.6. Características y fases de las revisiones sistemáticas de interés para la automatización**

Dado que la elaboración de revisiones sistemáticas está compuesta por fases que pueden ser divididas en tareas más pequeñas, existe una infinidad de herramientas que técnicamente podrían aportar a facilitar el proceso o mejorar la calidad de los resultados, sin embargo, solo unas cuantas de ellas aportan de forma integral a la automatización de varias fases de la elaboración de una revisión sistemática. Es por esto que para que una herramienta tecnológica pueda ser considerada una herramienta de automatización para la elaboración de revisiones sistemáticas, debería poder permitir automatizar varios de los procesos clave. A continuación, se mencionan las fases que suelen automatizar las herramientas en la actualidad, según Marshal y Sutton (2021) [45,46]:

- **Elaboración del protocolo:** Consiste en construir un borrador del documento que contiene el marco metodológico que será la guía para realizar la revisión sistemática. Esto a partir de metadata y sobre el proyecto y datos específicos de metodología que el usuario introduce de forma manual a lo largo del proceso [31,45,46].
- **Automatización de búsquedas:** Consiste en automatizar la elaboración de las estrategias de búsqueda, orientándolas a incluir tantos manuscritos relevantes y excluir tantos no relevantes como sea posible. Es decir, aumentar sensibilidad y especificidad. Las herramientas también suelen traducir la búsqueda al formato de las diferentes bases de datos más utilizadas [30,45,46].
- **Selección de estudios:** Consiste en brindar herramientas e interfaces gráficas que permitan al usuario cumplir de forma efectiva los requerimientos del estamento PRISMA en cuanto a selección de estudio. De esta forma se reduce el riesgo de sesgo al incluir o excluir estudios [30,45,46].
- **Aseguramiento de calidad:** Consiste en brindar herramientas que permitan validar la heterogeneidad y el riesgo de sesgo dentro de cada estudio y en la revisión sistemática en general, conforme a los requerimientos del estamento PRISMA [30,45,46].
- **Extracción de datos:** Extracción de datos, tablas, información de PDF, de archivos de citas bibliográficas RIS y de otras fuentes [30,45,46].
- **Análisis de texto:** Uso de técnicas de procesamiento de texto que permitan mejorar la calidad y el contenido de la revisión sistemática. Un ejemplo puede ser el procesamiento de los abstracts encontrados en los resultados de una búsqueda para mejorar la forma en la que fue construida [30,45,46].
- **Metaanálisis:** Brindar herramientas estadísticas para realizar el análisis siguiendo los requerimientos del estamento PRISMA. Generar reporte de la metodología. Construir diagramas de bosque. Validar riesgos, limitaciones y heterogeneidad [31,45,46].
- **Escritura del reporte:** Generación automática de un borrador de los resultados. Incluyendo gráficos y estadísticas de resumen [37,45,46].
- **Colaboración:** La herramienta debería tener características que permitan cooperar entre los miembros del proyecto y potenciar el trabajo en equipo sin que requieran estar juntos de forma presencial [37,45,46].
- **Gestión del documento:** Construcción de un documento borrador a partir de la metadata del proyecto y de la información introducida por el usuario a lo largo de todo el proceso [37,45,46].

### **3.7. Automatización de búsquedas, análisis de textos y PNL**

El componente de análisis de texto es uno de los más importantes que puede poseer una herramienta de automatización de revisiones sistemática. Incluso para tareas que no parecen directamente relacionadas, como realizar una búsqueda automática este módulo se hace necesario [47]. Esto se debe a que los

algoritmos no son capaces de entender el texto en formato científico, y en general ningún tipo de texto, por lo que estos componentes realizan la tarea de extraer información procesable del texto en lenguaje natural.

Este tipo de situaciones suelen abordarse utilizando técnicas de procesamiento de lenguaje natural (PNL) para transformar el texto de su formato nativo a un formato que los algoritmos de procesamiento de información puedan entender. En este paso se suelen eliminar caracteres que no son ni alfanuméricos ni espacios, incluyendo puntos y comas. Posteriormente se identifican las stopwords, que son las palabras como “the” y “with” que actúan como conectores pero que no brindan información acerca del contenido del texto. También se suele aplicar stemming para transformar cada palabra en su raíz semántica, lo cual puede ser de utilidad para encontrar un mayor número de asociaciones, teniendo en cuenta la complejidad de algunos términos del texto de tipo científico [48,49]. El resultado de esta fase es la lista de expresiones de texto o palabras que componen cada documento.

Posteriormente se utilizan técnicas de Information Retrieval (IR) para convertir la información obtenida del elemento anterior en variables numéricas. Las palabras obtenidas del elemento anterior se pueden dejar libres, o agruparse en bigramas u otro tipo de asociaciones que sean de interés. Cada característica corresponde a una variable por lo que se suelen escoger las  $n$  mejores variables, es decir, las  $n$  mejores palabras, bigramas y/o asociaciones que mejoren describen la distribución de la información a lo largo del corpus o set de documentos [50-52].

Entre las técnicas de IR más usadas se encuentran: TF-IDF que busca identificar qué tan relevante es una palabra dentro de un documento en relación a qué tan abundante es en la colección de documentos [50]. Okapi BM25 que permite organizar de mayor a menor importancia los documentos según su relevancia para una palabra clave, y que se usa en buscadores como Google [51]. Bag of Words (BoW), que busca crear una estructura tipo diccionario donde las llaves son cada palabra clave y los valores el conteo a lo largo del texto [14].

Previo a enviar los datos al modelo se debe realizar una normalización. Existen principalmente dos formas de realizarla. La normalización L1 o lasso, que utiliza la distancia de manhattan, donde a cada valor se le añade como penalización la suma total de los valores absolutos del vector, y la normalización L2 o normalización de ridge, que utiliza la distancia euclideana, donde a cada valor se le añade como penalización el valor cuadrado de la desviación del esperado. La normalización L2 es más efectiva para encontrar outliers [15].

Finalmente, para realizar la clasificación, se puede utilizar una gran variedad de técnicas de inteligencia artificial. SVM construye un clasificador efectivo cuando existen numerosas dimensiones o variables y donde la cantidad de dimensiones o variables es mayor a la cantidad de registros. Gradient Boosted Decision Trees (GBDT) es un ensamble de algoritmos que consiste en una generalización de funciones de penalización y que es efectivo y preciso en múltiples campos del conocimiento. Random Forest Classifier es una técnica que consiste en un ensamble de árboles de decisiones de decisiones y busca compensar la varianza u overfitting que se obtiene normalmente con los árboles tradiciones [53,54].

### **3.8. ¿Qué debemos tener en cuenta para desarrollar un software que mejore el flujo de trabajo para la elaboración de revisiones sistemáticas?**

A lo largo de este capítulo pudimos apreciar qué se debe tener en cuenta para elaborar revisiones sistemáticas y metaanálisis desde el punto de vista científico, qué dificultades y limitaciones existen en la elaboración de este tipo de estudios, qué tareas y pasos se deben realizar desde su concepción hasta su publicación, y cuáles de estas tareas se pueden y suelen automatizar.

Para poder desarrollar un software que permita mejorar la calidad, repetibilidad, tiempo de elaboración y precisión de una revisión sistemática, los próximos pasos son: primero, poder conocer qué se debe tener en

cuenta para realizar una revisión sistemática de alta calidad desde el punto de vista científico, lo cual abordaremos identificando y comparando los marcos metodológicos más eficaces y conocidos. Y segundo, poder conocer qué tareas de la elaboración de una revisión sistemática y de qué forma son automatizadas por otras herramientas tecnológicas, comparando sus puntos en común y diferencias. Esto lo abordaremos en el próximo capítulo.

## 4. TRABAJOS RELACIONADOS

Para poder conocer qué se debe tener en cuenta para realizar una revisión sistemática de alta calidad desde el punto de vista científico, se identificará y comparará los marcos metodológicos más eficaces y conocidos. Por otro lado, se explorarán qué tareas son automatizadas por las herramientas tecnológicas que buscan mejorar el flujo de trabajo en la elaboración de una revisión sistemática, y de qué forma. También se compararán los puntos en común y diferencias de estas herramientas.

### 4.1. Herramientas de automatización de revisiones sistemáticas y tareas automatizables

Marshall & Sutton (2021) [46] identificaron 11 módulos de la elaboración de una revisión sistemática que se pueden abordar a través de soluciones tecnológicas. También identificó que existen 245 herramientas que han sido catalogadas, ya sea por sus autores, por los autores del repositorio o por los investigadores que las han usado, como de utilidad a la hora de automatizar revisiones sistemáticas.

La gran mayoría de estas herramientas logran automatizar 1 o 2 subprocesos de la elaboración de una revisión sistemática, y son herramientas que más que estar orientadas a facilitar y mejorar el flujo de trabajo y calidad de la elaboración de una revisión sistemática, están orientadas a realizar el procesamiento de una tarea específica de forma genérica dentro del marco de múltiples áreas del conocimiento. Por otro lado, hay herramientas que abordan de forma integral el proceso de elaboración de una revisión sistemática, cubriendo varias fases del ciclo y evitando a los investigadores el esfuerzo de utilizar múltiples herramientas que no están integradas entre sí. En la tabla 1, de forma comparativa se presenta cuáles de los 11 módulos de automatización de revisiones sistemáticas identificados por Marshall & Sutton (2021) son incluidos en las 38 herramientas que cubren 4 o más de estas fases, y que, por tanto, se puede considerar que están orientadas a automatizar la elaboración de revisiones sistemáticas.

### 4.2. Descripción y comparación de las herramientas que cubren un mayor número de fases de la elaboración de revisiones sistemáticas

A continuación, se presenta una descripción corta de las herramientas mencionadas junto con un análisis de los puntos comunes y diferencias observadas entre ellas.

**Buhos:** Es una herramienta web desarrollada en Ruby. Se puede desplegar localmente e integrar con otros servicios en línea. Busca apoyar el proceso de búsqueda y selección de artículos, cuenta también con la generación de resultados [55].

**SRDB.PRO:** Herramienta en línea de acceso libre. Existen 2 versiones, instalada o alojada. Tiene capacidades para realizar tareas de minería de datos y consultas, y también para generar reportes automáticos [56].

**SLuRp (Systematic Literature Unified Review Program):** Herramienta open source construida en Java y disponible al público para ser adaptada y usada. A través de su uso, se registran y manejan los datos necesarios para una revisión sistemática [57].

**DistillerSR:** Aplicación web. Tiene la capacidad para administrar, manejar y optimizar los procesos de selección y extracción de datos. También es capaz de generar informes y resultados automáticos [58].

**JBISUMARI:** Desarrollada en Angular. Es una aplicación web que asiste al usuario a la hora de formular una pregunta de revisión, también apoya en la fase de protocolo, y al igual que las demás herramientas, en la selección de estudios. Permite extraer datos y evaluar la calidad de los estudios. También construye resultados, y apoya en la síntesis y elaboración del reporte final y de los reportes de resultados [59].



**Tabla 1.** Herramientas de automatización para la elaboración de revisiones sistemáticas y fases del ciclo que abarcan.

	Desarrollo de protocolo	Búsqueda automatizada	Selección de estudios	Aseguramiento de calidad	Extracción de datos	Análisis automatizado	Análisis de texto	Meta-análisis	Escritura del reporte	Colaboración	Manejo de documentación	Puntaje Total
Buhos												10
SRDB.PRO												10
SLuRp												9
DistillerSR												8
JB1-SUMARI												8
Nested-Knowledge												8
StArt												8
TaskExchange												8
CADIMA												7
EPPI-Reviewer												7
Parsifal												7
ReLiS												7
SESRA												7
Rax												6
Review Manager (RevMan)												6
SLR-Tool												6
Systematic Review Accelerator												6
Thoth												6
ASReview												5
Covidence												5
PICO Portal												5
Rayyan												5
SLRTOOL												5
Trialstreamer												5
2dSearch												4
BADERI												4
dmeter												4
DoctorEvidence (DOC Data)												4
Health Assessment Workspace Collaborative (HAWC)												4
KGen: a knowledge graph generator from biomedical scientific literature												4
Kumu												4
Lingo3G: text document clustering engine												4
MAGICapp												4
MeSHSIM												4
PredicTER												4
Qiqqa												4
SLR.qub												4
SyRF: Systematic Review Facility												4

Sintetizado y procesado a partir de la información encontrada en [46]

**Nested Knowledge:** Permite filtrar y etiquetar la información, lo cual se logra con extracción de datos. Permite apoyar en la elaboración de meta-análisis cuantitativos. La información extraída la presenta de forma resumida en fomato cualitativo. Adicionalmente, realiza un manejo de datos de forma tal que aquellos nuevos que entren se puedan incorporar al pipeline sin inconvenientes. También tiene características de elaboración de documentos y reportes finales [60].

**StArt (State of the art through systematic review):** Permite la colaboración entre revisores, incluir nuevas referencias a través de la técnica bola de nieve, mejorar la estrategia de de búsqueda a través del análisis de la frecuencia de las palabras clave y del uso de BoW, bag of words, vectorizador para el conteo

de palabras. También permite la selección automática de citas utilizando la puntuación de ellas para un tópico dado (SCAS, por sus siglas en inglés) [61].

**TaskExchange:** Plataforma web desarrollada en Ruby y Javascript, con capacidad para trabajar de forma colaborativa. Permite reunir en los proyectos personas que necesiten ayuda para la revisión de referencias de tu revisión sistemática con personas que cuenten con el tiempo y las habilidades para asistir en este proceso. Esto permite construir de forma eficiente revisiones sistemáticas de alta calidad. Permite seguir los lineamientos para realizar síntesis de la evidencia relevante y construir documentos como guías de práctica clínica [62].

**CADIMA:** Aplicación open source de tipo-cliente servidor, desarrollada en PHP. Está diseñada para revisiones sistemáticas más allá del ámbito médico. Sus fortalezas se encuentran en las capacidades de extracción de datos. También cuenta con un modulo de cooperación lo suficientemente maduro para tener un alto desempeño [63].

**EPPI-Reviewer:** Aplicación web. Permite mejorar la selección de referencias utilizando minería de texto y aprendizaje de máquina. Se utiliza la información de cómo los revisores califican a cada referencia, de forma continua, para mejorar la precisión predictiva [64].

**ReLiS (Revue littéraire systématique):** Herramienta instalable que se puede reconfigurar a lo largo de plazo. Permite realizar revisiones sistemáticas de forma cooperativa e iterativa tanto local como en la nube. Diseñada para utilizar un editor de modelo diseñado para investigadores que evalúan revisiones sistemáticas. Es una herramienta para instalar y configurar automáticamente proyectos de revisiones [65].

**SESRA (Software engineering systematic literature review application):** Aplicación web. Permite al investigador manejar el ciclo de vida de las revisiones sistemáticas, planificando su diseño, ejecutándolas y finalmente elaborando el informe. El flujo de trabajo sigue la metodología de Kitchenham y Charters (2007) [66,67].

#### 4.3. Herramientas de automatización de revisiones sistemáticas y características deseadas por los usuarios

Otra aproximación para encontrar las diferencias y puntos en común entre las herramientas de automatización para revisiones sistemáticas es identificar cuáles son las características más deseadas por los usuarios, y en qué grado fueron implementadas por las herramientas mejor valoradas. A partir de la infografía construida por Harrison et al. (2020), se construyó la tabla 2 que se presenta a continuación, en donde se muestra en qué grado de detalle fueron implementadas las características más deseadas por los usuarios en una herramienta de automatización de revisiones sistemáticas. Los features están ordenados según su relevancia de mayor a menor. En relevancia del feature se puede apreciar el puntaje que los usuarios le otorgaron a cada feature [68].

**Tabla 2.** Características más deseadas por los usuarios de revisiones sistemáticas y su grado de implementación a lo largo de las herramientas de automatización mejor valoradas.

Features	Relevancia del feature																														
	Múltiples usuarios	Exportación de referencias	Exportación de decisiones	Gratuita	Fácil instalación	El usuario no necesita escribir código	Selección de doble pantalla	Asignación de trabajo	Herramienta flexible a varios flujos de trabajo	Monitoreo de progreso	Categorizar referencias	Toma ciega de decisiones	Resolución de conflictos	Seguridad del sitio web	Sección ayuda	App móvil	Deduplicación	Screening de fulltext	Múltiples proyectos	Manejo de roles	Importación de pdfs	Aprendizaje rápido	Resaltado de palabras clave	Filtrado de citaciones por categoría	Búsqueda de referencias	Extracción de datos	Otras etapas de revisión	Herramienta de ranking de referencias	Screening de título y resumen	Importación de referencias	Incluir/Excluir
	5	5	5	4	4	4	4	4	4	4	4	4	4	4	3	3	3	3	3	3	3	3	3	3	3	2	2	2	1	1	1
Rayyan	3	3	3	3	3	3	2	1	3	3	3	3	3	3	2	3	3	3	3	3	2	3	3	3	3	1	1	3	3	3	3
Covidence	2	2	2	2	2	3	3	3	3	3	3	3	3	3	3	2	3	3	2	1	2	2	3	3	2	3	3	3	3	3	3
DRAGON	3	3	3	1	2	3	3	3	3	3	3	3	3	3	1	1	3	3	3	3	1	2	3	3	3	3	3	3	3	3	3
Abstrackr	3	2	3	3	3	3	3	3	3	3	3	3	3	1	3	2	3	1	3	3	1	3	3	1	1	1	1	3	3	3	3
Colandr	3	2	3	3	2	3	3	1	2	3	3	2	2	3	2	1	3	3	3	3	2	2	3	2	3	3	3	3	3	3	3
EPPI-Reviewer	2	2	2	1	2	3	3	3	3	3	3	3	3	3	3	2	3	3	2	1	2	2	3	3	2	3	3	3	3	3	3

Modificado y procesado a partir de la información encontrada en [68].

#### 4.4. Diferencias y puntos en común de las herramientas que buscan incluir las características deseadas por los usuarios

A continuación, se presenta una descripción corta de las herramientas mencionadas en la tabla junto con un análisis de los puntos comunes y diferencias observadas entre ellas.

**Rayyan:** Herramienta gratuita y open source desarrollada en Ruby. Está orientada a optimizar y acelerar el proceso de realizar una revisión sistemática, en especial en la clasificación de referencias. Incluye un módulo de soporte de toma de decisiones que facilita la inclusión y exclusión de referencias utilizando técnicas de aprendizaje de máquina. Cada usuario puede participar en ilimitadas revisiones [69,70].

**Covidence:** Aplicación web. Permite selección e importación de referencias, análisis de texto completo, valoración de la calidad, extracción de datos, y exportación de datos [71].

**DRAGON:** Aplicación web que incluye tareas como extracción de datos, valoración de la calidad, visualización en gráficos, trabajo colaborativo, soporte de toma de decisiones, selección de referencias, evaluación del riesgo de sesgo e integración de datos [72].

**Abstrackr:** Es una aplicación de código abierto y en línea para la selección de citas para revisiones sistemáticas, además proporciona una interfaz para aplicar aprendizaje automático [73].

**Colandr:** Aplicación web gratuita y open source que contiene características de aprendizaje automático. Se utiliza para investigación y síntesis de la evidencia. Tiene cuatro módulos importantes: planificación, extracción de datos, selección de texto completo y selección de citas [74].

**EPPI-Reviewer:** Ver sección 4.2

#### 4.5. Diferencias y puntos en común de los marcos metodológicos existentes

Como se mencionó previamente, PRISMA es el marco metodológico de referencia para realizar una revisión sistemática, mientras que AMSTAR y ROBIS son herramientas metodológicas que buscan evaluar la calidad de una revisión sistemática, una vez ésta se ha realizado. PRISMA está enfocado en los elementos que debe contener el manuscrito final, mientras que AMSTAR y ROBIS están enfocadas en evaluar la calidad de los hallazgos, las limitaciones y el sesgo.

PRISMA está compuesto por 27 elementos que son necesarios para construir una revisión sistemática de calidad. AMSTAR, por otro lado, está compuesto por 16 elementos que son requeridos para que la calidad del flujo de trabajo de una revisión sistemática sea considerada adecuada Y ROBIS está orientada hacia valorar la calidad del manejo del sesgo en 4 dominios distintos.

En la tabla 3 se presentan los requisitos metodológicos para alta calidad, ordenados por sección del entregable.

**Tabla 3.** Puntos en común de los frameworks metodológicos

Criterios de las herramientas o marcos metodológicos			
Sección del entregable	Criterios PRISMA	Criterios AMSTAR	Criterios ROBIS
Título	1. Es revisión sistemática o meta-análisis, o ambos		-
Abstract	2. Abstract adecuadamente estructurado		-
Introducción	3. Justificación de la revisión		-
	4. Estrategia PICOS	1. Estrategia PICO	F1
	5. URL y metadata del documento de protocolo	2. Selección de metodología	
	6. Características del estudio y criterios de elegibilidad	3. Selección de diseño	F2 - D1: Criterios de elegibilidad: D1.1, D1.2, D1.3, D1.4, D1.5
	7. Descripción de bases de datos usadas, fuentes y fecha de búsquedas. Incluir Estrategia de búsqueda	7. Razones de exclusión y manuscritos excluidos	
	8. Descripción del proceso de selección de manuscritos	3. Selección de diseño	F2 - D2: Identificación y selección de estudios. D2.1, D2.2, D2.3
	9. Método de extracción de datos de los estudios y procesos para confirmar información	4. Búsqueda detallada	
	10. Variables para las que se buscaron datos	5. Selección de estudios con más de 1 revisor	
Metodología	9. Método de extracción de datos de los estudios y procesos para confirmar información	6. Extracción de datos por varios revisores	
	11. Métodos para evaluar riesgo de sesgo y cómo se abordará esto en la síntesis de resultados	3. Selección de diseño	
	12. Estadísticos de resumen utilizados como RR	9. Metodología de evaluación de riesgo de sesgo (RoB)	
	13. Métodos de manejo de datos y combinación de resultados	11. Métodos apropiados para combinación estadística	
	14. Identificación de riesgos de sesgo	2. Selección de metodología	
	15. Verificación de veracidad. Descripción de métodos de análisis adicionales	11. Métodos apropiados para combinación estadística	
		9. Metodología de evaluación de riesgo de sesgo (RoB)	
Resultados	16. Diagrama de flujo. Número de	2. Selección de metodología	
		7. Razones de exclusión y lista de	

	estudios excluidos en cada etapa e incluidos finalmente	excluidos	
	17. Características para las que se extrajeron los datos en cada estudio, con referencias	8. Descripción de estudios incluidos	
	18. Presentar riesgo de sesgo en cada estudio, analizar punto 12	9. Metodología de evaluación de riesgo de sesgo: Risk of Biases (RoB) 13. Redacción de resultados teniendo en cuenta RoB	
	19. Datos estadísticos, estimación de efecto y diagrama de bosque para riesgos y beneficios encontrados	12. Metaanálisis: Evaluación de impacto de RoB en estudios individuales	F2 - D2: D2.4, D2.5
	20. Presentar resultados principales. Si es metaanálisis intervalos de confianza también	13. Redacción de resultados teniendo en cuenta RoB	F2 - D3: Colección de datos y revisión de estudios: D3.3, D3.4, D3.5
	21. Presentar resultados de evaluación de riesgo. Ítem 15	9. Metodología de evaluación de riesgo de sesgo (RoB)	
	22. Otros resultados incluyendo información de la solidez de la evidencia	13. Redacción de resultados teniendo en cuenta RoB	F2 - D4: Síntesis y hallazgos: D4.1, D4.2
Discusión	23. Principales hallazgos, solidez de la evidencia. Limitaciones a nivel del estudio y de la revisión. Interpretación de la evidencia e implicaciones	13. Redacción de resultados teniendo en cuenta sesgos.	
		14. Explicación y análisis de heterogeneidad	F2 - D4: D4.3, D4.4, D4.5, D4.6
		15. Síntesis cuantitativa: Análisis de sesgo de estudio	
	24. Registro y protocolo		
Otra información	25. Fuentes de financiación	10. Información de fuentes de financiación	
	26. Conflictos de intereses	16. Información de conflicto de intereses	
	27. Códigos y otros medios		

Sintetizado y procesado a partir de la información encontrada en [21, 22, 23, 24, 27, 28, 29, 36, 43].

#### 4.6. Técnicas de inteligencia artificial utilizadas en otras herramientas

Jin *et al.* (2021) [95] identificaron las técnicas de inteligencia artificial utilizadas por algunas de las herramientas de automatización de revisiones sistemáticas más populares para los pasos de selección de estudios y automatización de la búsqueda. Se puede observar en la tabla 4 que todas ellas incluyen las categorías “incluido” y “excluido” para clasificar, y algunas incluyen “neutro”

**Tabla 4.** Resumen de las técnicas de inteligencia artificial más usadas por las herramientas de automatización de elaboración de revisiones sistemáticas, y su propósito dentro del proceso.

Herramienta	Input	Técnica de IA	Extracción de características	Label (opciones de clasificación)
AbstrackR	Palabras clave Referencias	SVM	TF-IDF	<ul style="list-style-type: none"> <li>• Incluir</li> <li>• Excluir</li> <li>• Tal vez</li> </ul>

ASReview	Fragmentos de texto	Naive Bayes, SVM, DNN, LR, LSTM-base, LSTM-pool, RF	Doc2Vec, IDF, TF-IDF, sBERT	<ul style="list-style-type: none"> <li>• Incluir</li> <li>• Excluir</li> </ul>
Colandr	Palabras clave Título, abstract, Palabras clave	SVM con SGD	Word2Vec	<ul style="list-style-type: none"> <li>• Incluir</li> <li>• Excluir</li> </ul>
Rayyan	Palabras clave Títulos y abstracts	SVM	Unigramas, Bigramas, Términos MeSH	<ul style="list-style-type: none"> <li>• Incluir</li> <li>• Excluir</li> <li>• Tal vez</li> </ul>
RobotAnalyst	Título Abstract	SVM	TF-IDF Regularizado, BOW para abstract, Topic Model Proportion	<ul style="list-style-type: none"> <li>• Incluir</li> <li>• Excluir</li> <li>• Tal vez</li> </ul>

Modificado y procesado a partir de la información encontrada en [95]

Se exponen los fundamentos matemáticos detrás de algunas de las técnicas de inteligencia artificial.

**Máquinas de soporte de vectores:** SVM (Support vector machine) son métodos de aprendizaje de máquina supervisado. Se utilizan para clasificar o categorizar elementos y se caracterizan por utilizar kernels o hiperplanos para realizar esta tarea. Debido a la naturaleza multidimensional en la que se encuentran distribuidos los datos, el algoritmo SVM busca trazar un kernel o hiperplano que logre dividir adecuadamente el set de datos. Los algoritmos SVM se pueden modificar para utilizar distintas configuraciones de hiperplanos, para castigar el ajuste excesivo, y para adaptarlos a una gran cantidad de situaciones específicas como clasificaciones multiclase o series de tiempo. La representación matemática de SVM consiste en una función que busca maximizar el margen de separación entre las diferentes categorías, de modo que se minimice la posibilidad de que se trace el límite de forma que se clasifiquen incorrectamente muchos datos. En la ecuación 1, a continuación, se representa cómo el algoritmo busca maximizar la distancia del hiperplano a los datos en las categorías [76,77].

$$\text{margin} = \arg \min_{\mathbf{x} \in D} d(\mathbf{x}) = \arg \min_{\mathbf{x} \in D} \frac{|\mathbf{x} \cdot \mathbf{w} + b|}{\sqrt{\sum_{i=1}^d w_i^2}}$$

**Ecuación 1.** Representación matemática del algoritmo SVM, tomada de [76].

**Naive Bayes:** Es un algoritmo de aprendizaje simple basado en la regla de Bayes bajo el supuesto que los atributos son independientes dada la clase, lo cual en ciertos escenarios puede ser inconveniente. A pesar de que esta asunción no se cumple a menudo, dada la simplicidad y forma intuitiva como este maximiza la función objetivo, suele tener un alto desempeño. Además, adicionalmente es computacionalmente eficiente en comparación con otros algoritmos. Bayes es un algoritmo que se puede implementar de múltiples formas y con múltiples objetivos, la implementación Naive Bayes, utilizada frecuentemente en el contexto de clasificación de términos obtenidos tras extraer características, busca realizar una clasificación Crisp, donde cada elemento se busca clasificar dentro de una categoría. Por lo que finalmente, la probabilidad final de clasificar todo el dataset estará dada por la conjunción de la probabilidad de clasificar cada término en la categoría especificada. Esto se puede apreciar en la ecuación 2, donde se observa como la función se representa en términos de la probabilidad de clasificación de cada uno de los términos [78,79].

$$\hat{y} = \operatorname{argmax}_{y_j} \prod_{k=1}^p P(x_k | y_j) P(y_j)$$

**Ecuación 2.** Representación matemática del algoritmo Naive Bayes, tomada de [79].

**4.7. ¿Qué se requiere, además de conocer las características requeridas por todos los Stakeholders para construir una herramienta de automatización de elaboraciones sistemáticas potente?**

Explorar los marcos metodológicos nos permitió tener una aproximación hacia qué se requiere para lograr construir una herramienta de alta calidad, y explorar las características más deseadas por los usuarios nos permitió entender qué se requiere para construir una herramienta capaz de facilitar el flujo de información. Sin embargo, para lograr construir una solución tecnológica de automatización en la elaboración de revisiones sistemáticas que sea lo suficientemente potente, se requiere ayudar al usuario a tomar decisiones. Por ejemplo, a elegir qué palabras clave incluir o no en su búsqueda, o qué artículos incluir o excluir, etc. Este requerimiento se satisface a través de los módulos de soporte de toma de decisiones potenciados por los módulos de inteligencia artificial que se construirán en este proyecto.

## 5. DESARROLLO DEL PROYECTO

Este proyecto tiene como objetivo optimizar el proceso y los resultados de las fases de búsqueda y selección de artículos durante la elaboración de una revisión sistemática médica, a través del diseño e implementación de una plataforma digital capaz de aportar sugerencias en las fases más críticas del proceso, utilizando técnicas de inteligencia artificial. Se busca sugerir al investigador biomédico qué palabras clave podría incluir, y cuáles podría eliminar, para mejorar los resultados durante la elaboración de la estrategia de búsqueda. Del mismo modo, se proyecta sugerirle qué manuscritos podría incluir y cuáles no, durante la fase de selección de artículos que serán utilizados para escribir el contenido de la revisión sistemática. A lo largo de este capítulo se busca diseñar e implementar un prototipo de software funcional, que sea capaz de cumplir con este objetivo.

### 5.1 Diseño de la solución

Para lograr construir una solución tecnológica que realmente sea capaz de satisfacer los requerimientos de los usuarios, se incluyó esta sección, donde a partir de la literatura encontrada y de los trabajos relacionados se logran identificar los requerimientos funcionales por parte de los usuarios, y por parte de las metodologías de puntuación de la calidad de las revisiones sistemáticas. También se identificaron los requerimientos no funcionales para que el sistema funcione correctamente.

Dado que no se encontró una arquitectura de referencia para los intereses de este proyecto, se construirá desde cero la solución. Se utilizará Attribute Driven Design (ADD) para el diseño [6,7]. Se elaborará el documento de arquitectura de software (SAD) donde también estarán contenidas las vistas arquitecturales 4+1 y los casos de uso [8-10]. Se utilizará la metodología Architecture Tradeoff Analysis Method (ATAM) para mitigar los riesgos de la arquitectura escogida [7,11,100].

Un elemento importante de esta fase es diseñar la aplicación donde va a estar alojado el algoritmo de inteligencia artificial, de modo que sea posible realizar el proceso de inicio a fin sin errores en el flujo de trabajo, y sin entrar en detalles avanzados de la implementación de la inteligencia artificial.

#### 5.1.1 Selección de componentes a implementar en la versión 1.0

Es importante aclarar que para la versión 1.0 de la aplicación, solo se implementarán aquellos componentes funcionales que se identifiquen como prioritarios. A continuación, se presentan los componentes elegidos y los criterios de inclusión.

##### **Componentes incluidos para la versión 1.0:**

- *Búsqueda automatizada*: Alto valor. Permite optimizar el tiempo y la calidad del proceso.
- *Selección de estudios*: Alto valor. Permite optimizar el tiempo y la calidad del proceso.
- *Análisis de texto y Análisis automatizado*: Alto valor. Ambos son indispensables para el funcionamiento de Búsqueda automatizada y selección de estudios. Hacen parte de los objetivos del proyecto. En el diseño de la aplicación, ambos elementos constituyen un único componente funcional.
- *Extracción de datos*: Mediano valor. Aumenta las capacidades del análisis de texto.
- *Colaboración*: Mediano valor. Se implementaron los casos de uso que permitían la cooperación básica entre asesores y clientes.

##### **Componentes postergados para versiones futuras:**

- *Desarrollo de protocolo*: Se incluirá únicamente el paso donde se especifican los criterios de inclusión y exclusión.
- *Meta análisis*: Se requiere una mayor madurez y precisión en los componentes para realizar de forma



acertada este paso.

- *Manejo de documentación y Escritura del reporte:* No es imperativo y fundamental para la elaboración de una revisión crear o gestionar los documentos requeridos a través de la aplicación.
- *Aseguramiento de calidad:* Se requiere una mayor madurez y entendimiento del flujo de trabajo para poder construir exitosamente este módulo.

### 5.1.2 Requerimientos funcionales

Para que una herramienta de automatización de elaboración de revisiones sistemáticas permita mejorar el flujo de trabajo y facilitar el proceso, y a la vez cumpla con los altos estándares de calidad requeridos debe cumplir con dos tipos de requerimientos funcionales. Los primeros, basados en los marcos metodológicos de referencia, y los segundos, en las características más deseadas por los usuarios.

#### 5.1.2.1 Requerimientos funcionales desde la perspectiva de las características deseadas por el usuario

En la tabla 5, presentada a continuación, se listan los requerimientos funcionales desde la perspectiva de las características requeridas por el usuario, obtenidos a partir de comparar 9 distintos artículos que exploraban los requerimientos más solicitados por los usuarios en una aplicación de construcción de revisiones sistemáticas. Solo se describen aquellos que serán implementados en la versión 1.0, tal como fue especificado en la sección 5.1.1. La lista completa se encuentra en la tabla 19 del anexo 2.

**Tabla 5.** Requerimientos funcionales desde la perspectiva de las características deseadas por los usuarios a ser tenidos en cuenta en la versión 1.0

Requerimiento funcional	Descripción
RF-SCR-1	Screening: Permite la selección de títulos y resúmenes
RF-SCR-2	Screening: Permite la detección de texto completo
RF-SCR-3	Screening: Permite categorizar más referencias
RF-SCR-4	Screening: Análisis de texto
RF-SCR-5	Screening: Vista ciega a las decisiones de otros.
RF-SCR-7	Screening: Enfoque en Sensibilidad y especificidad de la búsqueda
RF-SCR-8	Screening: Criterios de inclusión/exclusión
RF-SR-1	Soporte de revisión sistemática: Permite la de-duplicación
RF-SR-2	Soporte de revisión sistemática: Facilita la interacción de las partes interesadas
RF-SR-3	Soporte de revisión sistemática: Elementos PICO especificados
RF-SR-4	Soporte de revisión sistemática: Permite la extracción de datos
RF-REF-2	Referencias: Exportación de referencias
RF-REF-3	Referencias: Aleatorización de referencias
RF-REF-4	Referencias: Exportación de decisiones
RF-REF-5	Referencias: Importación de PDF
RF-BUS-1	Búsqueda: Puede buscar referencias (Ideal: Integrado con otras DB)
RF-BUS-5	Búsqueda: Resaltado de palabras clave

RF-GES-2	Gestión: Soporte para múltiples proyectos: Usuario puede crear proyectos, editarlos
RF-GES-4	Gestión: Localización de trabajo
RF-GES-5	Gestión: Gestión de roles

---

Sintetizado y procesado a partir de la información encontrada en [63, 68, 90, 91, 92, 93, 94, 95, 96].

### 5.1.2.2 Requerimientos funcionales desde la perspectiva de los marcos metodológicos

A continuación, se presentan los requerimientos funcionales desde la perspectiva de los marcos metodológicos, con su respectiva descripción. Estos requerimientos buscan cumplir con los estándares de alta calidad buscados por las revistas científicas. Solo se describen aquellos que serán implementados en la versión 1.0, tal como fue especificado en la sección 5.1.1. La lista completa se encuentra en la tabla 19 del anexo 3.

**Tabla 6.** Requerimientos funcionales desde la perspectiva de los requerimientos de los marcos metodológicos a ser tenidos en cuenta en la versión 1.0

Requerimiento funcional	Descripción
RF-PR-IN-2	Introducción: Mostrar Estrategia PICOS
RF-PR-ME-2	Metodología: Se deben mostrar características del estudio y criterios de elegibilidad y validar la calidad de los criterios de elegibilidad
RF-PR-ME-3	Metodología: Se deben describir de bases de datos usadas, fuentes de información y fecha de búsquedas. Se debe explicar la estrategia de búsqueda utilizada. Se deben utilizar estrategias de búsqueda optimizadas y realizar una búsqueda exhaustiva
RF-PR-ME-4	Metodología: Se debe describir el proceso de selección de estudios. Debe incluir más de un revisor. Se deben incluir restricciones de fecha, idioma y formato. Se deben recopilar todos los estudios relevantes para la temática. Deben existir esfuerzos para minimizar el sesgo en selección
RF-PR-RE-1	Resultados: Incluir diagrama de exclusión PRISMA, razones de exclusión e inclusión y número de estudios excluidos en cada etapa

---

Sintetizado y procesado a partir de la información encontrada en [21, 22, 23, 24, 27, 28, 29, 36, 43]

### 5.1.2.3 Componentes funcionales del sistema basados en Harrison (2020)

En la tabla 7 se presenta la forma en que se agruparon los requerimientos funcionales, tanto desde la perspectiva del usuario, como desde la perspectiva de los marcos metodológicos, dentro de los componentes funcionales identificados por Harrison (2020), los cuales corresponden a los módulos que se suelen encontrar a lo largo de las diferentes herramientas de automatización de revisiones sistemáticas. Aquellos requerimientos funcionales que se encuentran mencionados en esta tabla pero que no hayan sido mencionados previamente, se pueden encontrar en las tablas en los anexos 3, 4 y 5. Ver sección 5.1.1

De forma complementaria, en la tabla 8, se presentan otros requerimientos funcionales secundarios, que son necesarios para garantizar el correcto funcionamiento de la aplicación.

### 5.1.3 Requerimientos no funcionales:

En esta sección se presentan los drivers arquitectónicos, o requerimientos no funcionales, los cuales son requerimientos orientados hacia las características del sistema. En la tabla 9 se presentan aquellos identificados en la literatura, resaltando la importancia de la usabilidad, facilidad de aprendizaje y operatividad. Por otro lado, en la tabla 10 se presentan los drivers arquitectónicos de interés para lograr el correcto funcionamiento de esta aplicación. Se especifican los drivers arquitectónicos correspondientes, y los subatributos y métricas de interés.

**Tabla 7.** Componentes funcionales del sistema.

Componente funcional	Se implementará en la versión 1.0? *	Requerimientos funcionales
Desarrollo de protocolo	No	· RF-PR-ME-1, RF-PR-ME-2, RF-PR-ME-6
Búsqueda automatizada	Sí	· RF-BUS-1, RF-BUS-2, RF-BUS-3, RF-BUS-4, RF-BUS-5, RF-BUS-6 · RF-PR-ME-3 · RF-SR-3
Selección de estudios	Sí	· RF-REF-1, RF-REF-2, RF-REF-3, RF-REF-4, RF-REF-5, RF-REF-6 · RF-SCR-1, RF-SCR-2, RF-SCR-3, RF-SCR-4, RF-SCR-5, RF-SCR-6, RF-SCR-7, RF-SCR-8 · PR-ME-2, PR-ME-4, · PR-RE-1 · RF-SR-1 · RF-GES-3
Aseguramiento de calidad	No	· RF-SR-5 · RF-MET-7, RF-MET-10, RF-MET-11
Extracción de datos	Sí	· RF-SR-4 · RF-PR-MET-5, RF-PR-MET-6, RF-PR-MET-8, RF-PR-MET-9
Análisis automatizado	Sí	
Análisis de texto	Sí	· RF-SCR-4: Soporte para la selección de palabras clave & Soporte de toma de decisiones para la clasificación
Metaanálisis	No	· F-MET-1, F-MET-2
Escritura del reporte	No	· F-RES-1, F-RES-2, F-RES-3, F-RES-4 · PR-TI, PR-AB, PR-IN-1, PR-IN-2, PR-ME-1, PR-ME-2, PR-ME-3, PR-ME-4, PR-ME-5, PR-ME-6, PR-ME-7, PR-ME-8, PR-ME-9, PR-ME-10, PR-ME-11, PR-RE-1, PR-RE-2, PR-RE-3, PR-RE-4, PR-RE-5, PR-RE-6, PR-RE-7, PR-DI, PR-OT-1, PR-OT-2, PR-OT-3, PR-OT-4
Colaboración	No	· F-GES-1, F-GES-2, F-GES-3, F-GES-4, F-GES-5

El contenido fue sintetizado y procesado a partir de la información encontrada en [21, 22, 23, 24, 27, 28, 29, 36, 43, 63, 68, 90, 91, 92, 93, 94, 95, 96].

**Tabla 8.** Componentes funcionales adicionales

Componente funcional	Requerimiento funcional	Descripción
Usuario	F-BA-1	Usuario puede registrarse, hacer login, logout, modificar su perfil
Manejo de documentación	F-BA-2	Registro, acceso de documentos subidos
Módulo de integración	F-BA-3	Integración con otras herramientas de revisiones

**Tabla 9.** Requerimientos no funcionales deseados por los usuarios.

Driver arquitectónico (Atributo)	Descripción	Relevancia
Otros	Otros: Soporta distintos tipos de caracteres	1
Portabilidad: Adaptabilidad	Facilidades: Tiene una aplicación móvil	1
Funcionalidad: Interoperabilidad	Facilidades: Se puede integrar con otras herramientas	2
Funcionalidad: Adecuación	Facilidades: Contiene todo el flujo de trabajo sin requerir otras herramientas	1
Usabilidad: Aprendizaje	Facilidades: Requisitos de sistema sencillos y facilidad para instalar y configurar	1.5
Usabilidad: Atractividad	Facilidades: No se requiere que el usuario escriba código	1.5
Usabilidad: Atractividad	Facilidades: Experiencia de usuario	1
Usabilidad: Aprendizaje	Facilidades: Facilidad de aprendizaje	1
Usabilidad: Aprendizaje	Corto tiempo para poder usarse y para aprender	1
Funcionalidad: Seguridad	Seguridad: El sitio web es seguro	1.5
Usabilidad: Aprendizaje	Soporte: Contiene ayudas y soporte técnico	1
Usabilidad: Operatividad	Flujo de trabajo: Se pueden realizar distintos flujos de trabajo	2.5
Usabilidad: Aprendizaje	Flujo de trabajo: Se puede configurar rápidamente antes de iniciar	1
Usabilidad: Operatividad	Flujo de trabajo: El progreso se retroalimenta al usuario	1.5
Usabilidad: Operatividad	Flujo de trabajo: Monitoreo del estado del proyecto	1

El contenido fue sintetizado y procesado a partir de la información encontrada en [63, 68, 90, 91, 92, 93, 94, 95, 96]

**Tabla 10.** Requerimientos y drivers arquitectónicos.

Componente funcional	ID	Requerimiento no funcional	Atributo (Driver arquitectural)	Subatributo	Métrica de interés
Plataforma	RNF-1	Cumplir con la mayor cantidad de requerimientos deseados por el usuario	Usabilidad	Aprendizaje	Facilidad de aprender Frecuencia y accesibilidad de ayuda

	RNF- Cumplir con la mayor cantidad de requerimientos establecidos por los marcos metodológicos	Funcionalidad	Operatividad	De acuerdo a expectativas del usuario
	2		Adecuación	Adecuada funcionalidad Completitud de la implementación funcional
General	RNF- Seguridad en el manejo de la información y en la protección de los niveles de acceso a los recursos	Funcionalidad	Seguridad	Seguridad
	3			
	RNF- La plataforma debe estar disponible online más del 99% del tiempo	Fiabilidad	Recuperabilidad	Disponibilidad
	4			
	RNF- Se requiere que la funcionalidad de la búsqueda y traducción de la búsqueda al formato de otras bases de datos sea realizada por completo	Portabilidad	Adaptabilidad	Adaptabilidad al ambiente de hardware y facilidad de instalación
	5			
	RNF- La plataforma debe estar disponible online más del 99% del tiempo	Mantenibilidad	Facilidad de cambio	Capacidad de control de cambios de software
	6			
	RNF- Se requiere guardar los estados de algunas de las aplicaciones internas	Fiabilidad	Recuperabilidad	Restaurabilidad
	7			
Búsqueda automatizada	RNF- Se requiere que la funcionalidad de la búsqueda y traducción de la búsqueda al formato de otras bases de datos sea realizada por completo	Funcionalidad	Adecuación	Adecuada funcionalidad Completitud de la implementación funcional
	A-1			
	RNF- Se debe garantizar que los resultados de la búsqueda correspondan a lo esperado y el algoritmo debe ser computacionalmente eficiente		Exactitud	Precisión a esperado Precisión computacional
	A-2			
	RNF- La conexión con terceros, como los Apis de lenguaje controlado y de conexión con las bases de datos de búsqueda gratuitas debe tener una tasa alta de éxito		Interoperabilidad	Tasa de éxito
	A-3			
	RNF- Dado que es una operación computacional de carga pesada se requiere que el procesamiento sea eficiente	Eficiencia	Comportamiento en el tiempo	Tiempo de respuesta promedio
	A-4			
	RNF- Se requiere manejo de excepciones que prevenga que se entregue una respuesta nula o no funcional	Fiabilidad	Tolerancia a fallos	Evitamiento de operaciones incorrectas
	A-5			
Selección de estudios	RNF- Se requiere que la funcionalidad de selección de estudios sea realizada por completo	Funcionalidad	Adecuación	Adecuada funcionalidad Completitud de la implementación funcional
	B-1			

	RNF- Se debe garantizar que el output B-2 corresponda a lo esperado		Exactitud	Precisión a esperado
	RNF- Se debe hacer un manejo adecuado de las B-3 excepciones y evitar que el flujo de trabajo se vea interrumpido		Tolerancia a fallos	Evitamiento de operaciones incorrectas
	RNF- El módulo debe poder conectarse con B-4 AWS, SCOPUS, Crossref. Lo debe hacer de tal forma que la integración sea exitosa y la aplicación se adapte a los datos de entrada.		Interoperabilidad	Tasa de éxito
	RNF- El tiempo de respuesta desde la perspectiva B-5 del usuario debe ser relativamente corto	Eficiencia	Comportamiento en el tiempo	Tiempo de respuesta promedio
Extracción de datos	Dado que los datos se deben extraer de RNF- diferentes fuentes y formatos, es probable C-1 que existan errores. e debe procurar importar los resultados correctamente y que siempre haya un resultado que permita que no se bloquee el flujo de trabajo	Fiabilidad	Tolerancia a fallos	Evitamiento de colapso Evitamiento de fallo
	RNF- Se debe lograr que se puedan extraer datos C-2 tanto de pdfs, como de textos planos	Funcionalidad	Interoperabilidad	Tasa de éxito
	RNF- Se debe lograr que el módulo sea capaz de C-3 realizar operaciones de procesamiento de lenguaje natural (PNL), que pueda realizar operaciones de Information Retrieval. Desplegar este módulo de inteligencia artificial representa un reto tecnológico y por tanto debe procurarse que el resultado sea correcto	Funcionalidad	Exactitud	Precisión a esperado
Análisis automatizado y análisis de texto (Incluye módulos de IA)	RNF- Dado que es una operación computacional D-1 de carga pesada se requiere que el procesamiento sea eficiente	Eficiencia	Comportamiento en el tiempo	Tiempo de respuesta promedio
	RNF- El módulo debe poder conectarse con D-2 AWS, SCOPUS, Crossref. Lo debe hacer de tal forma que la integración sea exitosa y la aplicación se adapte a los datos de entrada.	Funcionalidad	Interoperabilidad	Tasa de éxito
	RNF- Se debe hacer un manejo adecuado de las D-3 excepciones y evitar que el flujo de trabajo se vea interrumpido	Fiabilidad	Tolerancia a fallos	Evitamiento de operaciones incorrectas
	RNF- Se debe garantizar que los resultados de los D-4 módulos de ciencia de datos correspondan a lo esperado y el algoritmo debe ser computacionalmente eficiente	Funcionalidad	Exactitud	Precisión a esperado Precisión computacional
Usuario	RNF- La información guardada en este módulo no E-1 debe perderse. La recuperabilidad es muy importante. Además, es importante que este módulo tenga una muy alta disponibilidad ya que es usado por todos los demás	Fiabilidad	Recuperabilidad	Restaurabilidad Disponibilidad
Manejo de documentación	RNF- Se debe garantizar que se retorne el F-1 documento adecuado	Funcionalidad	Exactitud	Precisión a esperado

	RNF- Es importante que los documentos no se pierdan, sino que persistan en la arquitectura	Fiabilidad	Recuperabilidad	Restaurabilidad
Desarrollo de protocolo	Se implementará en la versión 2.0			
Meta-análisis	Se implementará en la versión 2.0			
Escritura del reporte	Se implementará en la versión 2.0			
Colaboración	Se implementará en la versión 2.0			
Aseguramiento de calidad	Se implementará en la versión 2.0			

#### 5.1.4. Diseño arquitectónico

En esta sección se explicará cómo se utilizaron los requerimientos funcionales para identificar y construir los casos de uso, cómo se utilizaron los requerimientos no funcionales para elegir los patrones arquitecturales más importantes y asignarles responsabilidades siguiendo la metodología ADD, qué consideraciones se tuvieron para resolver las desventajas encontradas en los patrones arquitectónicos escogidos siguiendo la metodología ATAM y de qué forma a partir de los elementos anteriores se construyeron las vistas de escenario, lógica, despliegue y procesos.

##### 5.1.4.1 ADD

Este proyecto utiliza la metodología Attribute Driven Design (ADD) con el fin de identificar qué requerimientos no funcionales tiene el sistema, qué drivers arquitectónicos o atributos no funcionales sería necesario priorizar para cumplir con las métricas requeridas por los requerimientos no funcionales y lograr la estabilidad del sistema, y qué modificaciones es necesario realizar sobre estos requerimientos.

Esta metodología permite obtener el valor agregado de garantizar la calidad del diseño del sistema y reducir la cantidad de modificaciones a realizar a lo largo de la implementación debido a que se tiene una idea mucho más clara de cómo se va a diseñar la aplicación [6].

La metodología tiene como punto de entrada los requerimientos funcionales, los componentes funcionales, y los requerimientos no funcionales. Todos ellos se han presentado a lo largo de ese capítulo. A continuación, se presentarán los elementos que abordara esta sección.

1. Identificar patrones arquitecturales candidatos que puedan satisfacer el driver arquitectural, o atributo no funcional requerido.
2. Instanciar cada patrón, asignándole responsabilidades para resolver cada driver arquitectural.

En las próximas secciones se abordarán las interfaces para cada patrón instanciado, mediante las vistas 4+1 y la valoración de los requerimientos, a través de la metodología ATAM.

##### 5.1.4.2 Patrones Arquitecturales

En esta sección se presentan los patrones arquitecturales escogidos para resolver los drivers arquitectónicos identificados previamente. Esto se hará de forma iterativa, empezando por alto nivel para después entrar en detalle en los componentes.

### 5.1.4.2.1 Primera iteración

Para la primera iteración se presentan los patrones arquitectónicos escogidos a nivel general para la arquitectura de Digital Medtools. En la tabla a continuación, se presenta de qué forma los patrones elegidos resuelven los drivers arquitectónicos identificados. Como se puede observar, los dos patrones arquitectónicos que lograron resolver más drivers arquitectónicos fueron el patrón Arquitectura en capas y el patrón Cliente Servidor, por lo cual estos serán los dos patrones principales de la arquitectura de Digital Medtools.

**Tabla 11.** Patrones arquitecturales propuestos, primera iteración

Driver Arquitectónico	Cliente servidor	Arquitectura	Suite de pruebas	Arquitectura en	Parameter store	ECS	Single Container	SoA	Code Repository	S3 buckets	NoSQL Storage	RESTful APIs	SQL Storage	Responsabilidades
Arquitectura general														
Usabilidad (RNF-1)	X	X												El frontend construido utilizando MVVM mejorara la usabilidad
Funcionalidad (RNF-2)	X	X	X											Además del frontend en MVVM, contar con suites de pruebas, siguiendo la metodología BDD para identificar que componentes implementar, sería de gran utilidad.
Seguridad (RNF-3)	X			X	X									Contar con un cliente y un servidor, y que además la arquitectura del servidor estuviera diseñada en capas, donde solo la capa más externa se puede acceder públicamente, mejoraría mucho la seguridad. Usar AWS Parameter Stores también permitiría hacer un mejor manejo de los datos sensibles.
Disponibilidad (RNF-4)				X		X								Una arquitectura en capas permitirá que cada capa se escale por separado, lo cual aumenta la disponibilidad. También, usar ECS para desplegar los servicios que más requieren disponibilidad podría ser factible, como un patrón de arquitectura secundario.
Adaptabilidad (RNF-5)							X							Los contenedores de docker son extremadamente versátiles. Cada uno de los módulos o capas de la arquitectura se podría desplegar en su propio contenedor, haciéndolos resistentes a cualquier entorno donde se requieran desplegar
Facilidad de cambio (RNF-6)				X				X	X					Una arquitectura en capas, donde los módulos están orientados a servicios es más fácil de modificar ya que los repositorios se encuentran separados. El versionamiento de cada una de las apps se manejará desde Github.
Recuperabilidad (RNF-7)										X	X	X	X	Utilizando RESTful APIs, se garantiza que el backend no contiene estados por lo que es más difícil que la información se pierda. Los estados pueden ser guardados tanto en recursos SQL como no SQL,



incluyendo AWS RDS Postgres, AWS DynamoDB, AWS S3 Buckets, etc.

#### 5.1.4.2.2 Segunda iteración

En la tabla a continuación se presentan los patrones arquitectónicos escogidos, y sus responsabilidades, para la segunda iteración, la cual explora los requerimientos dentro de cada uno de los componentes funcionales.

**Tabla 12.** Patrones arquitecturales propuestos, segunda iteración

Driver arquitectural	Responsabilidades					
	BDD	Adaptador	BlackBoard	Manejo de excepciones	ETL	ECS
<b>Búsqueda automatizada</b>						
Funcionalidad: Adecuación (RNF-A-1)	X					
Funcionalidad: Exactitud (RNF-A-2)	X					
Funcionalidad: Interoperabilidad (RNF-A-3)		X				
Eficiencia: Comportamiento en el tiempo (RNF-A-4)			X			
Fiabilidad: Tolerancia a fallos (RNF-A-5)				X		
<b>Selección de estudios</b>						
Funcionalidad: Adecuación (RNF-B-1)	X					
Funcionalidad: Exactitud (RNF-B-2)	X					
Funcionalidad: Tolerancia a fallos (RNF-B-3)				X		
Funcionalidad: Interoperabilidad (RNF-B-4)		X				
Eficiencia: Comportamiento en el tiempo (RNF-B-5)			X			
<b>Extracción de datos</b>						
Fiabilidad: Tolerancia a fallos (RNF-C-1)				X		

BDD Permite identificar que funcionalidades es necesario implementar y si están correctas. El patrón adaptador permite conectarse con los servicios de AWS, tales como S3, dynamoDB y RDS, que son requeridos para guardar o cargar estados de los modelos. El patrón Blackboard busca hacer un flujo eficiente del guardado y cargado de modelos, incluyendo las estrategias de búsqueda, palabras clave utilizadas en el proyecto, etc., lo cual además permite que no sea necesario volver a procesar toda una petición si el recurso ya existe. El manejo de excepciones permite reducir el número de excepciones fatales y evitar que el flujo de trabajo se detenga.

Este componente tiene drivers arquitectónicos y patrones seleccionados similares al anterior. La diferencia es que el patrón blackboard de este módulo gestiona el guardado y cargado de los modelos de abstracts y clasificación por parte del usuario.

El patrón ETL (Extract, Transform, Load) es un patrón arquitectónico para ciencia de datos, que en este

Funcionalidad: Interoperabilidad (RNF-C-2)	X		componente se encargará de transformar un PDF a texto plano y de extraer data de archivos RIS. Es muy importante en este módulo el manejo de excepciones, que garantice la continuidad del flujo de trabajo. También es de importancia el patrón adaptador para recibir los distintos formatos aceptados. Por último, BDD permite identificar que funcionalidades es necesario implementar.
Funcionalidad: Exactitud (RNF-C-3)		X	
	X		
Análisis de texto			
Eficiencia: Comportamiento en el tiempo (RNF-D-1)		X	Este componente corresponde a los módulos de inteligencia artificial. El patrón arquitectónico ETL (Extract, Transform, Load) de este módulo busca transformar texto en lenguaje natural utilizando los algoritmos de PNL previamente mencionados. Adicionalmente, busca utilizar TF-IDF para hacer Information Retrieval. Es muy importante en este módulo el manejo de excepciones, que garantice la continuidad del flujo de trabajo. También, contar con el total de características, para lo cual está la suite de pruebas de BDD.
Funcionalidad: Interoperabilidad (RNF-D-2)	X		
		X	
Fiabilidad: Tolerancia a fallos (RNF-D-3)			
	X		
Funcionalidad: Exactitud (RNF-D4)		X	
Usuario			
Restaurabilidad (RNF-E-1)			X
Disponibilidad (RNF-E-1)			X
Manejo de documentación			
Funcionalidad: Exactitud (RNF-F-1)	X		Gestión del Repositorio / Almacén no relacional de documentos. Se utilizarán AWS S3 y AWS DynamoDB. También PostreSQL para almacenar las relaciones e IDs.
Restaurabilidad (RNF-F-2)		X	

#### *5.1.4.3 Vistas 4 + 1*

En esta sección se presentan las vistas 4+1. En ellas se representa la forma en la que los drivers arquitectónicos y los diferentes casos de uso se unieron para construir un diseño de la arquitectura de la implementación de la aplicación. En el documento expuesto en este link se pueden apreciar con detalle cada una de las vistas. Las vistas se pueden observar con detalle en los documentos anexos al trabajo de grado en:

<https://livejaverianaedu.sharepoint.com/:f:/r/sites/Ingsis/TGMISC/213006/Documentos%20compartidos/7.%20documentos/e.%20vistas?csf=1&web=1&e=r6fhdJ>

##### *5.1.4.3.1 Vista de Casos de Uso*

Inicialmente, en la vista de casos de uso, se presentan las acciones que tanto el investigador (usuario) como el asesor (staff) realizan al interactuar con el sistema, y de qué forma estas acciones incluyen las funcionalidades de otros componentes funcionales. En la figura 2 se presenta este diagrama.

##### *5.1.4.3.2 Vista Lógica*

Por otro lado, en la figura 3 se presenta la vista lógica, que representa la forma en la que los componentes funcionales se conectan entre sí, a lo largo de los diferentes subsistemas para lograr cumplir con los casos de uso identificados y teniendo en cuenta los drivers arquitectónicos escogidos utilizando la metodología ADD y las consideraciones señaladas al utilizar la metodología ATAM.

##### *5.1.4.3.3 Vista de Despliegue*

En la figura 4 se presenta la vista de despliegue. En ella se presenta la forma en la que los diferentes subsistemas son desplegados en el entorno físico. Podemos observar que se tienen 4 diferentes aplicaciones en el entorno On-Premise y se tienen estructuras en AWS Cloud, tales como Buckets S3, DynamoDB y AWS RDS Postgres. A continuación, podemos apreciar la vista de despliegue.

##### *5.1.4.3.4 Vista física*

En la vista física, se representa la forma en la que los componentes funcionales se conectan entre si, una vez están desplegados e instanciados. Básicamente es la forma en la que se conectan los componentes de la vista lógica en el entorno representado en la vista de despliegue. A continuación, se puede apreciar la vista de despliegue.

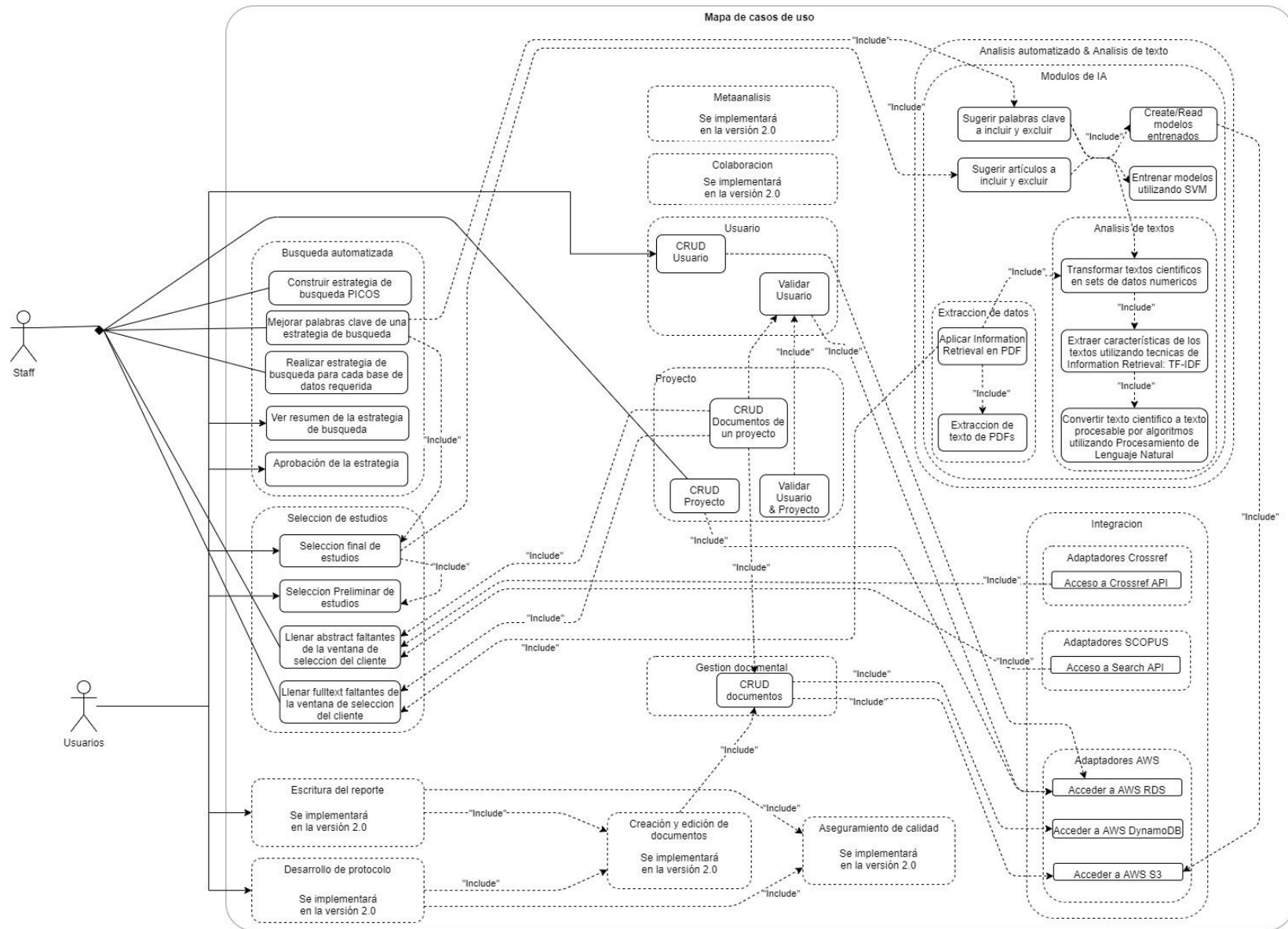
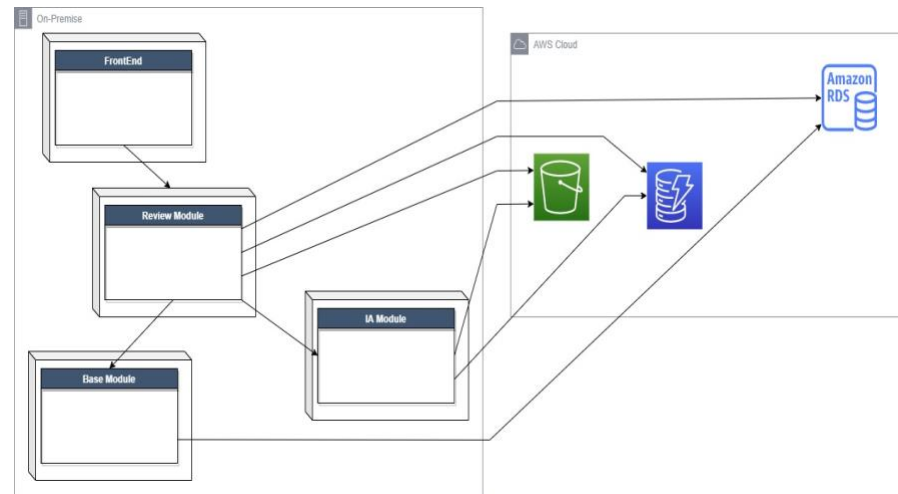
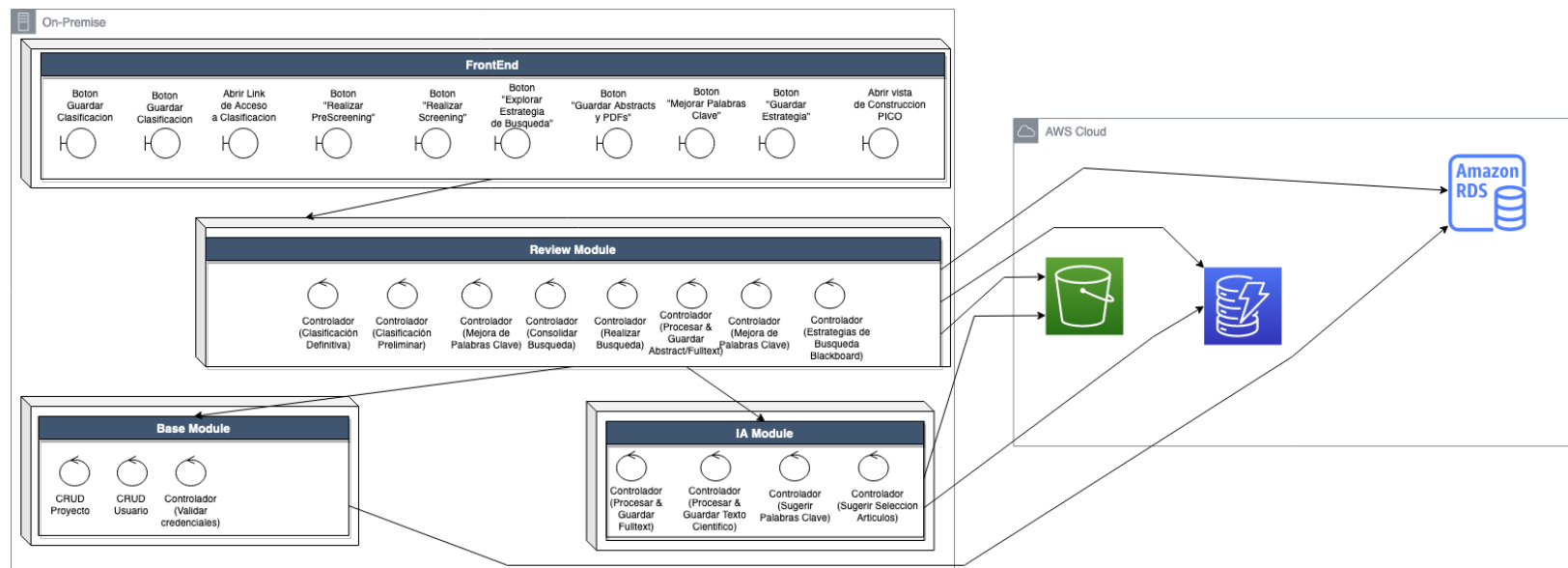


Figura 2. Vista de casos de uso.



**Figura 4.** Vista de despliegue**Figura 5.** Vista física

### 5.1.5. ATAM

Con el fin de identificar si los patrones de diseño propuestos son los correctos para responder a las necesidades de cada driver arquitectónico y del sistema en general, se utilizó la metodología ATAM. El valor agregado que trae utilizar esta metodología es entender qué situaciones potencialmente problemáticas se pueden generar al implementar cada uno de los patrones arquitectónicos, para reducir su impacto.

La metodología ATAM tiene como punto de entrada los requerimientos funcionales y no funcionales, los drivers y patrones arquitectónicos escogidos para cada driver arquitectónico, y las vistas 4+1. En esta sección se abordará el único elemento restante de las 4 fases de ATAM: Trade-offs, donde se evalúan las ventajas y desventajas de cada patrón arquitectural escogido en cada caso específico, lo cual se presenta a continuación [7].

A continuación, se presentan las consideraciones para tener en cuenta con cada uno de los patrones arquitecturales escogidos.

- *Primera iteración*

En la tabla 13 se presentan los patrones arquitectónicos escogidos, y las consideraciones que deberían tenerse en cuenta para mitigar las desventajas de cada uno de ellos.

**Tabla 13.** Tradeoffs de los patrones arquitecturales propuestos, primera iteración

ID	Patrón Arquitectónico	RNF-1	RNF-2	RNF-3	RNF-4	RNF-5	RNF-6	RNF-7	Desventajas y consideraciones
1	Cliente servidor	X	X	X					Para que realmente aumente la seguridad, el acceso al backend se debe restringir a que solo se permita desde la ubicación del FrontEnd. Además, se debe balancear la carga del procesamiento entre los dos módulos.
2	Arquitectura MVVM	X	X						La reusabilidad del código es limitada [80]. Aunque es difícil extender las clases, en angular se pueden importar los templates de los componentes requeridos. La mayor utilización de memoria que representan estos templates se compensa utilizando lazy loading [81].
3	Suite de pruebas		X						Mayor tiempo de desarrollo al inicio del proyecto. A mediano plazo se observa una ganancia de tiempo.
4	Arquitectura en capas			X	X		X		Se hace necesario desplegar cada componente por separado, pero las ventajas son gigantescas
5	Parameter store			X					Se requiere configurar VPC, permisos, accesos, lo cual representa una inversión de tiempo adicional, pero la ventaja en seguridad es gigante.
6	ECS				X				Desplegar en AWS ECS implica un menor acceso al control. Utilizando AWS EventBridge

			y AWS Cloudwatch Logs se puede lograr un monitoreo apropiado a futuro [82, 83].
7	Single Container	X	Los contenedores de Docker son una excelente solución en términos de portabilidad, pero es necesario tener ciertos cuidados con los puntos de vulnerabilidad de la imagen construida [84].
8	SoA	X	La arquitectura por servicios se puede mejorar si el servidor no mantiene estados, es decir, es RESTful [85].
9	Code Repository	X	Github permite un control de versionamiento, manejo de repositorios y colaboración muy buenos. A futuro se puede complementar con un CI/CD pipeline.
10	S3 buckets	X	Usar S3 Buckets permite guardar objetos, archivos e información fácilmente, pero es necesario hacer un adecuado manejo de roles y permisos para evitar asuntos de seguridad, ya que este api queda expuesto [86].
11	NoSQL Storage	X	DynamoDB permite guardar registros y objetos noSQL. es necesario hacer un adecuado manejo de roles y permisos para evitar asuntos de seguridad, también se debe tener en cuenta que difiere ampliamente de SQL [87].
12	RESTful APIs	X	Para que un servidor RESTful sea capaz de persistir información para usarla en otras consultas requiere el uso de bases de datos y repositorios de objetos, esto ya está incluido en esta arquitectura.
13	SQL Storage	X	-

- *Segunda iteración*

En la tabla 14 se presentan los patrones arquitectónicos escogidos, y las consideraciones que deberían tenerse en cuenta para mitigar las desventajas de cada uno de ellos.

**Tabla 14.** Tradeoffs de los patrones arquitecturales propuestos, segunda iteración

#	Driver Arquitectónico	Solución/Patrón arquitectural	Consideraciones
<b>Búsqueda automatizada</b>			
14	RNF-A-1 RNF-A-2	Behaviour Driven Development	Ver #3
15	RNF-A-3	Adaptador	Se debe construir el módulo adaptador en una capa diferente a la lógica para lograr un bajo acoplamiento y alta cohesión
16	RNF-A-4	BlackBoard	Este patrón implica un mayor gasto de tiempo en configuración de recursos, se debe también validar que el flujo de información sea correcto. Sin embargo, si funciona correctamente los resultados son excelentes.



17	RNF-A-5	Manejo de excepciones	Se debe procurar enviarlas a un registro a través de logs.
<b>Selección de estudios</b>			
18	RNF-B-1 RNF-B-2	Behaviour Driven Development	Ver #3
19	RNF-B-3	Manejo de excepciones	Ver #17
20	RNF-B-4	Adaptador	Ver #15
21	RNF-B-5	BlackBoard	Ver #16
<b>Extracción de datos</b>			
23	RNF-C-1	Manejo de excepciones	Ver #17
24	RNF-C-2	Adaptador	Ver #15
25	RNF-C-3	ETL Behaviour Driven Development	Es muy importante limitar el tiempo de estos componentes. Un procesamiento excesivo, sin importar que tan bueno sea el resultado es detrimental en la fase actual del proyecto Ver #3
<b>Análisis de texto</b>			
26	RNF-D-1	BlackBoard	Ver #16
27	RNF-D-2	Adaptador	Ver #15
28	RNF-D-3	Manejo de excepciones Behaviour Driven Development	Ver #17 Ver # 3
29	RNF-D4	ETL	Ver #25
<b>Usuario</b>			
31	RNF-E-1	ECS - Escalamiento elástico del componente	Tener precaución con el estimado de costos
32	RNF-E-1	Behaviour Driven Development	Ver # 3
<b>Manejo de documentación</b>			
33	RNF-F-1	BlackBoard	Ver #17

## 5.2. Módulo de Inteligencia Artificial

En la sección anterior se diseñó el sistema donde va a estar alojado el algoritmo y se diseñó la forma en la que se integrará los componentes del módulo de inteligencia artificial con la aplicación, de modo que fuera posible realizar el proceso de inicio a fin sin entrar en detalles. En esta sección se mejorará el flujo de información, pasos, y modelos del módulo de inteligencia artificial en el marco de la metodología CRISP-DM.

### 5.2.1 Marco metodológico: CRISP-DM

El marco metodológico estándar para realizar un proyecto de ciencia de datos es CRISP-DM. Esta metodología consta de 6 fases: 1. Entendimiento del negocio: La mayoría de esta sección fue abordada en marco teórico y trabajos relacionados. Aquí se busca representar el flujo de trabajo.

2. Entendimiento de los datos: Entradas y salidas, metadata. 3. Procesamiento de los datos: Descripción del flujo de trabajo que permite procesar los datos. 4. Modelos de inteligencia artificial: Proceso de entrenamiento y selección de modelos. 5. Evaluación: Entrenamiento y evaluación de los modelos. Relativo al proceso de escoger el mejor y verificar que responda el problema de negocio. 6. Despliegue: Despliegue en entorno de producción, en parte abordado en las vistas 4+1 en la sección previa. [3,88,89].

### 5.2.2 Entendimiento de negocio

Los módulos de soporte de toma de decisiones buscan mejorar el tiempo de desempeño y la calidad de la revisión sistemática, a través del apoyo brindado al usuario en dos fases críticas de la elaboración de una revisión sistemática:

- **Soporte para la construcción de estrategias de búsqueda:** El usuario podría tener un mejor desempeño si, por un lado, pudiera conocer qué palabras clave reducen la calidad de los resultados para excluirlas y qué palabras clave se podrían incluir para mejorarlos, y, por otro lado, qué campo de búsqueda podría utilizarse para cada fragmento de la estrategia de búsqueda para mejorar los resultados.
- **Soporte para la selección (inclusión y exclusión) de estudios:** El usuario podría tener un mejor desempeño si pudiera conocer si cada manuscrito tiene más probabilidad de pertenecer al grupo de artículos de interés o de pertenecer al grupo de artículos excluidos.

Para poder realizar esto se requiere tener:

- Una estrategia de búsqueda funcional, con la que se puedan encontrar resultados, para ser clasificados por el usuario
- Clasificación de los resultados encontrados. El usuario deberá clasificar un subconjunto de los manuscritos encontrados en resultados en las categorías “incluir”, “excluir” y “neutro”.

### 5.2.3 Entendimiento de los datos

En esta sección, luego de identificar las entradas y salidas de cada módulo de inteligencia artificial, se describirán las fuentes de información, variables de interés, metadata y características de los datasets.

#### 5.2.3.1 Datos de entradas a los módulos de inteligencia artificial.

##### ***Módulo de Soporte en la construcción de estrategias de búsqueda:***

- Referencias clasificadas manualmente por el usuario dentro de las categorías mencionadas, palabras clave de la estrategia de búsqueda y abstracts pre-procesados con técnicas de PLN

##### ***Módulo de soporte en la selección (inclusión y exclusión) de estudios:***

- Referencias clasificadas manualmente por el usuario dentro de las categorías mencionadas, palabras clave de la estrategia de búsqueda y abstracts pre-procesados con técnicas de PLN.

El usuario clasifica las referencias utilizando una vista de la página web específicamente diseñada para facilitar el proceso. Las referencias que se le presentan al usuario son cargadas y compiladas por el servidor backend, desde múltiples fuentes, como se discutirá posteriormente en esta sección. Es importante tener en cuenta que esta clasificación no corresponde a la clasificación definitiva, sino a clasificar un pequeño subconjunto de artículos, con el único propósito de alimentar los módulos de inteligencia artificial. La lista de referencias es obtenida de las fuentes una única vez por cada clasificación. Una vez obtenida se almacena dentro de la base de datos no-SQL DynamoDB.

Ya que la mayoría de las referencias no contienen el abstract debido a las limitaciones de los APIs, estos se deben llenar manualmente por el asesor de Digital Medtools. Para llenarlos existe una vista en la página web que facilita y agiliza el proceso de modo que el asesor puede visitar la página web (Aquella especificada por el DOI). El asesor puede completar la referencia tanto con el abstract como texto plano, como con el texto completo en formato .PDF, el cual se procesa con el módulo de extracción de datos para obtener el texto plano que contiene.

#### *5.2.3.2 Datos de salida de los módulos de inteligencia artificial*

Los datos de salida de los módulos de inteligencia artificial buscan actuar como soporte para la toma de decisiones. A continuación, se presentan los mismos para cada uno de los módulos.

***Módulo de Soporte en la construcción de estrategias de búsqueda:***

- Puntaje que indica si una palabra clave debería excluirse o no.

***Módulo de soporte en la selección (inclusión y exclusión) de estudios:***

- Sugerencia que indica si un artículo debiera incluirse o excluirse

#### *5.2.3.3 Fuentes de información.*

La principal fuente de información para la fase de pre-screening es el API de SCOPUS de búsquedas (Search API). Este es un API gratuito, expuesto al público, donde se pueden realizar consultas. También se utiliza Crossref API, para traer otros artículos faltantes. No se utilizó SCOPUS Abstracts API, el API que retorna los abstracts de los manuscritos solicitados, ya que este API requiere una membresía estimada en 140.000 dólares y no tiene una capa gratuita para desarrolladores o emprendedores. Por otro lado, dado que Digital Medtools no está afiliado a la PUJ, existe un conflicto legal de intereses que impide utilizar el proxy de la PUJ como punto de acceso a SCOPUS.

Para la fase de screening se pueden importar referencias de múltiples fuentes subiendo archivos .RIS, que son leídos y procesados por el servidor de Digital Medtools. De esta forma se garantiza que el usuario pueda realizar la búsqueda en múltiples bases de datos, según sea su elección. Los abstracts obtenidos por este medio igualmente son procesados y posteriormente almacenados, por lo que este punto de entrada consiste en una forma alternativa para realizar el mismo proceso y usando tantas bases de datos como el usuario guste.

Vale la pena destacar que para la fase de pre-screening solo se realizó la búsqueda en SCOPUS, a pesar de que existen otras fuentes de igual o mayor valor para la obtención de metadata de investigación biomédica, tales como Pubmed, Web of Science y Google Scholar [101]. Debido a que el único propósito de esta fase era obtener un pequeño set de manuscritos que el usuario pudiera utilizar para alimentar los módulos de inteligencia artificial. De todos modos, en la aplicación se presentará el número de resultados obtenidos al realizar la búsqueda en estas tres bases de datos. En cuanto al proceso de la clasificación real, o screening, esta búsqueda en SCOPUS no afecta, dado que el usuario tiene la capacidad de importar referencias de todas las bases de datos que requiera, a través de exportar dichas referencias en formato .RIS y luego importarlas en Digital Medtools.

Por otro lado, de forma alternativa a buscar nuevas palabras clave dentro de las referencias utilizando el módulo de inteligencia artificial, existe la posibilidad de usar los términos de lenguaje controlados DeCS y MeSH, los cuales se pueden obtener desde Lilacs y Pubmed. Sin embargo, aunque esto es una excelente alternativa para complementar los resultados, trae consigo dos consideraciones: la primera es que, dado que los términos DeCS y MeSH tienen una estructura de grafos, es necesario que sea el usuario quien decida cuáles elementos del árbol de grafos podrían ser incluidos, y esto implica diseñar un set de múltiples vistas

interactivas que le permitan tomar las decisiones de una forma sencilla. Por otro lado, se hace necesario explorar de forma iterativa y computacionalmente eficiente la estructura de grafos para obtener los resultados adecuados y precisos, por lo que, además, el algoritmo debe ser resistente a errores. Como se puede observar en BIREME (2019), el autor de este repositorio de github diseñó e implementó una solución tecnológica capaz de desarrollar exactamente esta misma tarea, en un plazo de más de 5 meses, por lo que podría considerarse un proyecto aparte, ya que tiene sus propios objetivos y KPIs, una carga laboral y criterios de aceptación considerables y a que este componente no hace parte de los objetivos principales del proyecto. Por esto, y dado que de todas formas se puede realizar esta tarea de forma manual, se decidió posponer su implementación para la próxima iteración de la arquitectura de Digital Medtools [31,102].

#### 5.2.3.4 Variables de interés.

Las variables de interés son: 1. La estrategia de búsqueda. 2. La forma en la que el usuario clasifica los manuscritos encontrados al ejecutar la estrategia de búsqueda. 3. La lista de palabras claves que compone un abstract, tras ser procesado con PLN. 4. Las n características obtenidas tras realizar Information Retrieval sobre el corpus o set de documentos.

#### 5.2.4 Procesamiento de los datos

Para lograr alimentar los módulos de inteligencia artificial que realizan las sugerencias se requiere un flujo de trabajo específico, dividido en 3 fases. A continuación, en la figura 6 se encuentra representado.

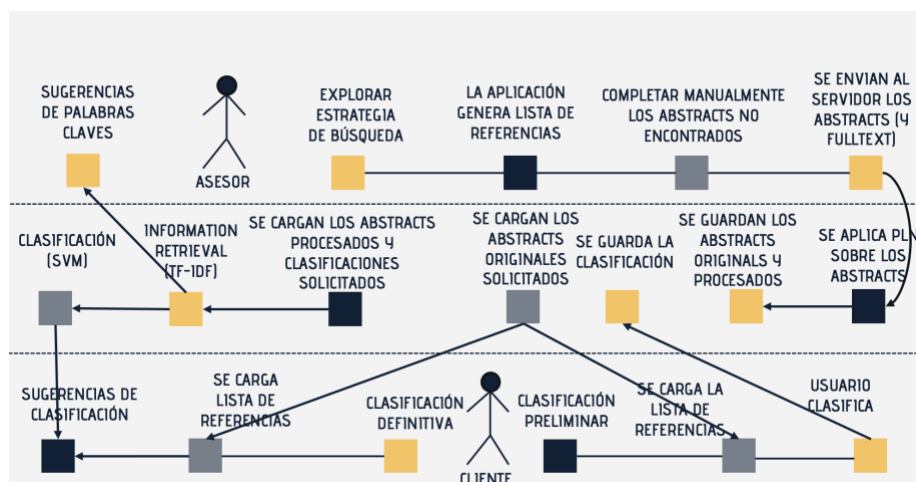


Figura 6. Procesamiento de los datos

Las 4 fases que contiene este proceso son:

1. **Explorar estrategia de búsqueda:** El asesor construye una estrategia de búsqueda PICO, la aplicación realiza la búsqueda en SCOPUS, genera la lista preliminar de referencias, la mayoría sin abstracts. El asesor completa manualmente los abstracts restantes utilizando la plataforma de Digital Medtools. Todos los abstracts se envían al servidor, donde el módulo de PLN se encarga de convertir el abstract a un vector de palabras procesadas. Finalmente se guardan los abstracts originales y procesados en AWS DynamoDB.

Una vez terminado este paso, Digital Medtools genera un link de acceso para que el cliente pueda clasificar los manuscritos

2. **Clasificación Preliminar:** El cliente accede al link que el asesor le brinda, la aplicación carga la lista de referencias, donde se traen desde el servidor los abstracts originales no procesados con PLN, para

que el usuario pueda ver títulos y abstracts. El cliente clasifica a los artículos en las categorías “Incluir”, “Neutro”, “Excluir”, y guarda la clasificación. Esta es enviada a AWS RDS Postgres.

3. **Módulo de sugerencia de Palabras Clave:** Cuando el usuario ya ha realizado clasificaciones, el asesor podrá presionar el botón “Mejorar Palabras Clave”. Esto hará que en el servidor se recolecten los abstracts previamente procesados con PLN, desde DynamoDB, y la clasificación de artículos según el cliente, desde AWS RDS Postgres. Con estos insumos, se puede ejecutar la técnica TF-IDF para puntuar la importancia de cada palabra clave y cada palabra presente en los documentos, con el fin de identificar qué palabras incluir y cuáles excluir.
4. **Módulo de sugerencia de selección de manuscritos:** Adicional al elemento anterior, utiliza SVM para realizar la clasificación.

### 5.2.5 CRISP-DM: Modelado

Ver sección 5.4 Implementación del modulo de inteligencia artificial

### 5.2.6 CRISP-DM: Evaluación

Ver sección 6.2, Resultados de la optimización

### 5.2.7 CRISP-DM: Despliegue

Ver sección 5.1.5.3.4, Vista física.

## 5.3. Implementación de la solución

En la sección anterior se pudieron identificar los casos de uso más importantes, los subsistemas de la arquitectura, la forma en que se tiene planeado desplegarla y las aplicaciones y servicios de la nube que se deberían incluir, junto con las decisiones arquitectónicas más importantes para lograr que el sistema funcione adecuadamente. En esta sección, se presentarán y detallarán las capas presentes dentro de la arquitectura en capas del ecosistema Digital Medtools, y las aplicaciones y estructuras construidas durante este proyecto en cada una de ellas.

Capa de presentación: Es la capa más exterior del ecosistema de Digital Medtools, es el único punto de acceso para las demás capas y aplicaciones de las otras capas del ecosistema Digital Medtools. Todas las aplicaciones construidas en esta capa fueron desarrolladas en Angular, el cual utiliza la arquitectura MVVM. Por el momento, solo existe una aplicación:

- DigitalMedtools FrontEnd: Módulo que contiene la capa de presentación de la arquitectura por capas, que será desplegada y ejecutada en el navegador del usuario. Esta aplicación es el único punto de acceso al ecosistema de Digital Medtools para el usuario. Para los procesos relacionados a las revisiones sistemáticas, esta aplicación consume los servicios de la aplicación Backend Reviews. Para los procesos relacionados a CRUD de usuarios y proyectos, esta aplicación consume los servicios de la aplicación Backend Base.

Capa Lógica: Es la capa intermedia del ecosistema Digital Medtools. Solo se puede acceder a ella a través de la capa de presentación. Para implementar las aplicaciones de esta capa, se utilizó Python, con Django. Se utilizó el paquete Pytest para construir la suite de pruebas requerido por la metodología Behaviour Driven Development. Esta capa contiene los controladores vinculados a las acciones realizadas por el usuario en la capa de presentación y provee los servicios necesarios (Es una arquitectura orientada a

servicios) para que esta funcione correctamente. A continuación, se presentan las aplicaciones construidas en esta capa:

- **Backend Reviews:** Aplicación desplegada como servidor backend que provee los servicios relacionados a la gestión de información en el marco de la elaboración de estrategias de búsqueda y clasificación de artículos. Tiene la responsabilidad de proveer al módulo DigitalMedtools FrontEnd de todos los servicios que sean necesarios para su correcto funcionamiento. Este módulo consume a Backend Base para las validaciones de seguridad y de nivel de acceso y roles de usuarios y proyectos, y a la aplicación Backend AI, para las operaciones relacionadas a análisis de textos, extracción de datos, inteligencia artificial y procesamiento de lenguaje natural.
- **Backend base:** Aplicación desplegada como servidor backend que se encarga de realizar todas las validaciones de seguridad, de gestionar los procesos CRUD de usuarios y proyectos, de obtener y guardar vínculos entre usuarios, proyectos y roles. Como se mencionó a lo largo de las metodologías ADD y ATAM, es un componente que requiere alta disponibilidad y por esta razón fue separado. Idealmente debería estar desplegado en ECS para facilitar el escalamiento y reducir los tiempos de respuesta ya que las bases de datos están desplegadas en AWS.
- **Backend AI:** Aplicación desplegada como servidor backend responsable de las operaciones relacionadas a análisis de textos, extracción de datos, inteligencia artificial y procesamiento de lenguaje natural. Contrario a la mayoría de los proyectos de Inteligencia Artificial, este no es un script 1-file sino una aplicación desplegada como servidor backend RESTful siguiendo las mejores prácticas de programación orientada a objetos en Python.

**Capa de Cloud Services:** Esta es la capa más interna del ecosistema de Digital Medtools, solo se puede acceder a ella a través de la capa lógica. En esta capa se utilizaron componentes de la capa gratuita de AWS. A continuación, se presentan las infraestructuras que se crearon:

- **Parameter Stores:** Responsables de guardar información sensitiva como llaves de acceso y credenciales de conexión.
- **DynamoDB:** Base de datos No SQL responsable de almacenar, por un lado, los estados de las búsquedas incluyendo las palabras clave, y por otro lado los abstracts originales y procesados con PLN.
- **RDS PostgreSQL:** base de datos relacional SQL.
- **S3 Buckets:** Almacenar objetos JSON siguiendo el patrón blackboard.

## 5.4 Implementación del modulo de Inteligencia Artificial

Como se mencionó previamente este módulo se divide en 2 componentes que requieren 3 etapas: Procesamiento de Lenguaje Natural, Information Retrieval y Clasificación.

### 5.4.1 Procesamiento de Lenguaje Natural

Para el procesamiento de lenguaje natural se construyó una clase *NaturalLanguageProcessor* capaz de 1. Eliminar puntuación y signos, 2. tokenizar por palabras, 3. convertir a minúsculas, 4. eliminar puntuación, 5. eliminar todos los caracteres no-alfanuméricos, 6. remover las stopwords en inglés (Ya que todos los manuscritos de SCOPUS se encuentran en inglés), 7. remover los tokens vacíos, 8. Stemmizar, para preservar únicamente las raíces de las palabras, 9. Remover los tokens de longitud cero.

Se utilizó stemmización en vez de lematización porque la lematización tiene en cuenta el contexto para realizar la tokenización, y aunque esto es de utilidad para la selección de manuscritos, es contraproducente

Por otro lado, por tema de consumo de recursos de AWS, se evitó realizar ambas operaciones sobre cada abstract. El uso de stemming en vez de lematización en la fase de sugerencia en la selección de artículos se compensó a través de la inclusión de bigramas dentro del vectorizador TF-IDF. Los abstracts y fulltexts a los que se les aplicó esta técnica fueron guardados tanto en su formato original como procesado en AWS DynamoDB.

**Figura 7.** Clase NaturalLanguageProcessor para procesamiento de lenguaje natural

Para lograr implementar el modulo de inteligencia artificial capaz de sugerir qué palabras incluir y cuales excluir, se debieron seguir los siguientes pasos:

- Vale la pena recordar que este proceso se hace en tiempo real, ya que esta embebido en uno de los módulos de la aplicación desplegada como servidor backend, que también se conecta con la infraestructura de la nube. Debido a la eficiencia con que fue diseñado el código, no genero ningún impacto en la carga del sistema, tardando solo un par de segundos en responde.

```

N = len(corpus_dict.keys())
DF = {}
for doi in corpus_dict:
    tokens = corpus_dict[doi].split(" ")
    for w in tokens:
        try:
            DF[w].add(doi)
        except:
            DF[w] = {doi}

```

**Figura 8.** Componente de frecuencia de términos. Basado en los conceptos expuestos en [97,98]

```

def doc_freq(word):
    c = 0
    try:
        c = len(DF[word])
    except:
        pass
    return c

```

**Figura 9.** Componente contador de palabras. Basado en los conceptos expuestos en [97,98]

```

weights = classification
tf_idf = {}
for doi in corpus_dict:
    tokens = corpus_dict[doi].split(" ")
    weight = weights[doi]*3
    counter = Counter(tokens)
    words_count = len(tokens)
    for token in np.unique(tokens):
        tf = weight * counter[token]/words_count
        df = doc_freq(token)
        idf = np.log((N+1)/(df+1))
        tf_idf[token] = tf*idf

```

**Figura 10.** Componente TF-IDF. Basado en los conceptos expuestos en [97,98]

```

keywords_TF_IDF = []

keywords_vectorized = NaturalLanguageProcessor.ProcessAndClean(" ".join(keywords)).split(" ")

for index in range(len(keywords)):
    vectorized_keyword = keywords_vectorized[index]
    original_keyword = keywords[index]
    if vectorized_keyword in tf_idf:

```



```
keywords_TF_IDF.append({'key':original_keyword,'value':tf_idf[vectorized_keyword]})

else:

keywords_TF_IDF.append({'key':original_keyword,'value':None})

result['TFIDF']= keywords_TF_IDF
```

**Figura 11.** Componente adaptador de formato de palabras clave

#### 5.4.3 Clasificación utilizando Machine Learning

Para lograr implementar el modulo de inteligencia artificial capaz de sugerir qué artículos científicos se podrían incluir y cuáles excluir, se siguieron los siguientes pasos:

1. Traer el diccionario de clasificaciones para cada referencia, lo cual fue realizado por el cliente en una fase anterior, también traer los abstracts pre-procesados utilizando PLN, lo cual fue realizado luego de que el asesor subió los abstracts en un paso pasado. Estos datos son traídos desde AWS (Ver figura 12).
2. Establecer los parámetros del vectorizador TF-IDF, y el tipo y parámetros del modelo clasificador, en este caso SVM (Ver figura 13).
3. Dividir la lista de identificadores DOI de los abstracts en 5 grupos para el 5-fold cross-validation (Ver figura 14).
4. En cada iteración del 5-fold crossvalidation, se tomó uno de los fragmentos como dataset de testing y los otros cuatro como dataset de training. Se entrenó el modelo utilizando la función train\_n\_test() (Ver figuras 15 y 16).

```
#El API requiere que se envíe el ID de proyecto
projectid = requestData['projectid']
suggestions_classificador_id = 0
# Cargo las clasificaciones del usuario desde DynamoDB. Cada articulo fue
# Clasificado manualmente por cada usuario
classifications = load_multiple_classifications(projectid, suggestions_classificador_id) #Labels

for key in classifications:

    if classifications[key] != -1:
        classifications[key]=1

# Obtengo la lista de articulos que fueron clasificados en ese muestreo
DOI_LIST = list(classifications.keys())
# Con esta lista, puedo traer del repo blackboard, el manuscrito procesado
# Con PLN, en una fase previa, justo al momento de guardar el original.
# Es decir, se guardar simultaneamente en la base de datos el abstract
#Original y el procesado
corpus_dict = ArticleRepo.getAvailableProcessedAbstracts(DOI_LIST)
```

**Figura 12.** Traer información requerida para la clasificación

```
Tfidf_vect = TfidfVectorizer(max_features=400)
model = svm.SVC(C=2.0, kernel="rbf", degree=2, gamma='scale')
result['accuracy']="NaN"
result['crossvalidation'] = {}
```

**Figura 13.** Establecer parámetros del modelo, del vectorizador

```
@singleton
class FoldSampler:

    def __splitter(self,doiList:List):

        shuffledDoiList = sklearn.utils.shuffle(doiList)
        partition_list = []
        N_PARTITIONS = 5

        ITEMS_PER_PARTITION = len(shuffledDoiList)//N_PARTITIONS
        counter = 0
        for PARTITION in range(N_PARTITIONS):
            doi_in_partition = []
            for index in range(ITEMS_PER_PARTITION):
                doi_in_partition.append(shuffledDoiList[counter])
                counter =counter+1

            partition_list.append(doi_in_partition)
        return partition_list

    def Split(self,list1):
        list2=self.__splitter(list1)
        return list2
```

**Figura 14.** Dividir el dataset en 5 partes para 5-fold cross-validation

```
for index in range(len(folds)):
    test_dois = folds[index]
    train_dois = []
    for index2 in range(len(folds)):
        if index2 == index:
            continue
        train_dois += folds[index2]
    corpus_dict_train = []
    corpus_dict_test = [ ]
    trainY = []
    testY = []

    for doi in train_dois:
        corpus_dict_train.append(corpus_dict[doi])
        trainY.append(classifications[doi])
    for doi in test_dois:
        corpus_dict_test.append(corpus_dict[doi])
        testY.append(classifications[doi])
    Tfidf_vect.fit(corpus_dict)
    trainX = Tfidf_vect.transform(corpus_dict_train)
    testX = Tfidf_vect.transform(corpus_dict_test)

    accuracy = train_n_test(model,trainX,trainY,testX,testY)
    result['crossvalidation'][str(index)] = accuracy
    accuracies.append(accuracy)
```

**Figura 15.** Explorar cada iteración del 5-fold cross validation [99]

```
def train_n_test(model,trainX,trainY,testX,testY):

    model.fit(trainX,trainY)

    predictions_SVM = model.predict(testX)

    accuracy = accuracy_score(predictions_SVM, testY)*100

    return accuracy
```

**Figura 16.** Entrenamiento y testeo de los modelos. Se calcula la precisión del modelo para cada score [99]

## 6. PRUEBAS Y RESULTADOS

En esta sección se demuestran los resultados obtenidos tras implementar el proyecto. Se abordarán los resultados desde la perspectiva del cumplimiento de los requerimientos funcionales. Se logro cubrir 18 de los 20 requerimientos funcionales especificados en el diseño.

Como se mencionó previamente, la aplicación permite que el flujo de trabajo sea muy flexible, ya que los usuarios pueden realizar nuevas iteraciones dentro de una misma fase o volver a fases pasadas, pudiendo incluso redefinir el proyecto, los criterios de inclusión y exclusión y las palabras clave.

### 6.1. Pruebas de funcionalidad

- **RF-SCR-1 Screening: Permite la selección de títulos y resúmenes [CUMPLIDO]**

La aplicación permite seleccionar títulos y resúmenes como se muestra en la figura 17.

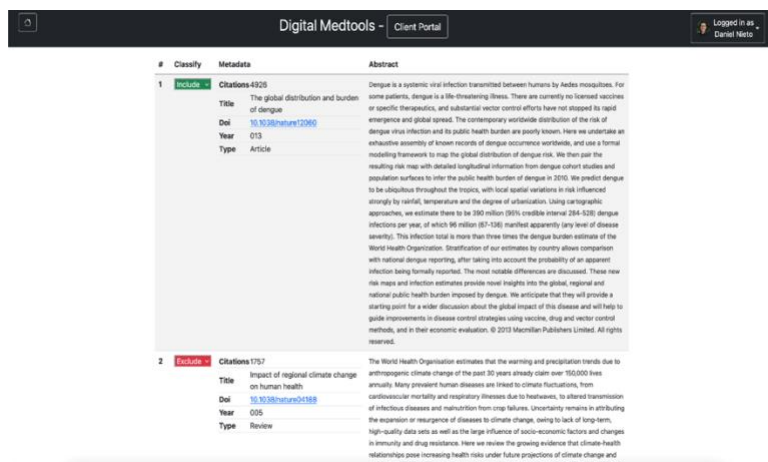


Figura 17. Screening de selección (Digital Medtools).

- **RF-SCR-2 Screening: Permite la utilización de texto completo [CUMPLIDO]**

La aplicación cuenta con la capacidad de leer PDF, extraer texto, y guardarlo para ser analizado por el módulo de PLN.

- **RF-SCR-3 Screening: Permite categorizar más referencias [CUMPLIDO]**

Luego de realizar una clasificación y guardar los datos, es posible generar una nueva orden para categorizar o etiquetar más referencias.

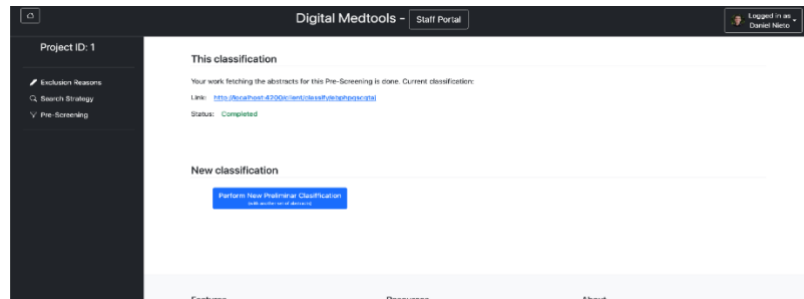


Figura 18. Screening de categorización (Digital Medtools).

- **RF-SCR-4 Screening: Análisis de texto [CUMPLIDO]**

La aplicación es capaz de transformar texto en lenguaje científico en texto que los algoritmos pueden procesar, utilizando vectorizaciones con Procesamiento de Lenguaje Natural.

<input type="checkbox"/>	doi	sectionIndex	content
<input type="checkbox"/>	10.110991...	abstract	CHIKV has been responsible for significant human morbidity for (probably) several hundred years; yet in spite of its prevalence, CHIKV epidemi...
<input type="checkbox"/>	10.110991...	abstract_processed	chikv respons significant hum morbidity probably several hundr years yet spit prevalenc chikv epidemiology mechanisms virulenc pathogenesis...
<input type="checkbox"/>	10.110991...	abstract	The interferon (IFN) system is an extremely powerful antiviral response that is capable of controlling most, if not all, virus infections in the abs...
<input type="checkbox"/>	10.110991...	abstract_processed	interferon ifn system extremely powerful antiviral respons capabl controlling virus infections absenc adaptiv immunity howevr virus still replicat...
<input type="checkbox"/>	10.111111...	abstract	nsect guts present distinctive environments for microbial colonization, and bacteria in the gut potentially provide many beneficial services to t...
<input type="checkbox"/>	10.111111...	abstract_processed	nsect guts present distinctiv environments microbial colonization bacteri gut potentially provid many beneficial servic hosts insects display wid...
<input type="checkbox"/>	10.111111...	abstract	The mosquito Aedes (Stegomyia) albopictus (Skuse) (Diptera: Culicidae), originally indigenous to South-east Asia, islands of the Western Pacific ...

Figura 19. Screening de análisis (Digital Medtools).

- **RF-SCR-5 Screening: Vista ciega a las decisiones de otros. [CUMPLIDO]**

Las clasificaciones se pueden realizar por separado. El asesor comparte un link a los clientes, donde se carga la vista donde darán su calificación a las referencias, y de este modo están ciegos a las decisiones de otros.

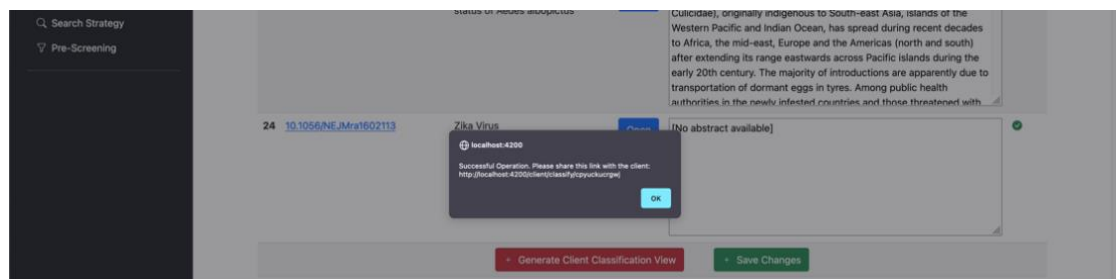
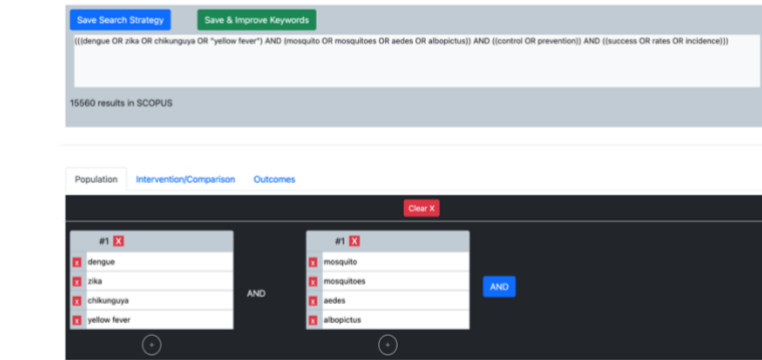


Figura 20. Screening de vistas (Digital Medtools).

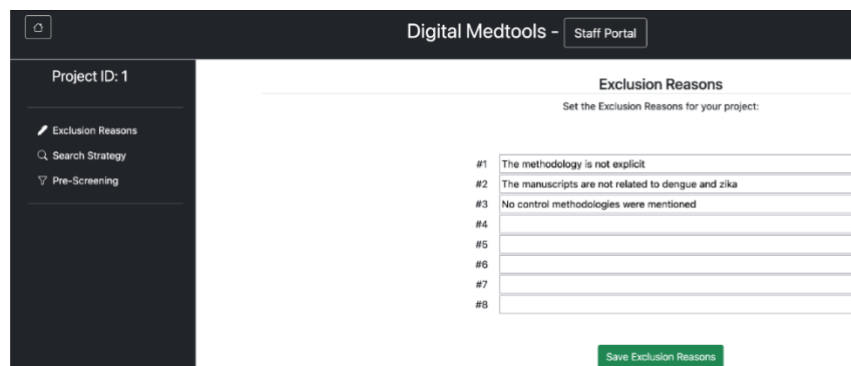
- **RF-SCR-7 Screening: Enfoque en Sensibilidad y especificidad de la búsqueda [CUMPLIDO]**

Se hizo un trabajo minucioso en el diseño del constructor de estrategia de búsquedas, se diseñó de tal forma que pudiera seguir el formato de la metodología PICO, quizá la más conocida y efectiva, para realizar búsquedas medicas de alta calidad. Se incorporó también un contador de resultados para que para el usuario fuera más fácil identificar el número de resultados totales y tener una mejor idea del estado la estrategia de búsqueda.



**Figura 21.** Screening de constructor de búsquedas (Digital Medtools).

- **RF-SCR-8 Screening: Criterios de inclusión/exclusión [CUMPLIDO]**  
Se diseñó una vista dedicada exclusivamente a satisfacer este requerimiento:



**Figura 22.** Screening de criterios (Digital Medtools).

- **RF-SR-1 Soporte de revisión sistemática: Permite de-duplicación [POSTERGADO].**  
Este paso tiene como pre-requisito contar con múltiples fuentes de manuscritos, por lo que se posterga
- **RF-SR-2 Soporte de revisión sistemática: Facilitación de la interacción entre las partes interesadas [CUMPLIDO].**  
La modalidad de trabajo en la que el staff orquesta el proceso y permite la participación del usuario en cada decisión cumple este objetivo.
- **RF-SR-3 Soporte de revisión sistemática: Desarrollo de protocolo, soporte para incluir elementos PICO [CUMPLIDO].**  
Ver RF-SCR-7. Se realizó un gran esfuerzo por incluir un módulo PICO interactivo, también se incluyeron elementos de protocolo como criterios de exclusión.
- **RF-SR-4 Soporte de revisión sistemática: Permite la extracción de datos [CUMPLIDO].**  
Ver RF-REF-5. Se puede extraer y procesar el texto científico que se encuentra dentro de los PDF.
- **RF-REF-2 Referencias: Exportación de referencias [CUMPLIDO].**
- **RF-REF-3 Referencias: Aleatorización de referencias [POSTERGADO].**

Este paso tiene como pre-requisito contar con múltiples fuentes de manuscritos, por lo que se posterga

- **RF-REF-4 Referencias: Exportación de decisiones [CUMPLIDO].**

El staff cuenta con un botón donde puede decidir exportar las decisiones de los revisores a .csv

- **RF-REF-5 Referencias: Importación de PDF [CUMPLIDO].**

El staff cuenta con la posibilidad de importar PDF e incluirlos al procesamiento de texto.

#	DOI	Title	Preview	Abstract	PDF	Saved
1	10.1038/nature12060	The global distribution and burden of dengue	Open	Dengue is a systemic viral infection transmitted between humans by Aedes mosquitoes. For some patients, dengue is a life-threatening illness. There are currently no licensed vaccines or specific therapeutics, and substantial vector control efforts have not stopped its rapid emergence and global spread. The contemporary worldwide distribution of the risk of dengue virus infection and its public health burden are poorly known. Here we undertake an exhaustive assembly of known records of dengue occurrence worldwide, and use a formal	Upload	✓
2	10.1038/nature04198	Impact of regional climate change on human health	Open	The World Health Organisation estimates that the warming and precipitation trends due to anthropogenic climate change of the past 30 years already claim over 150,000 lives annually. Many prevalent human diseases are linked to climate fluctuations, from cardiovascular mortality and respiratory illnesses due to heatwaves, to altered transmission of infectious diseases and malnutrition from crop failures. Uncertainty remains in attributing the expansion or resurgence of diseases to climate change, owing to lack of long-term, high-quality	Upload	✓
3	10.1056/NEJMoa1600651	Zika virus associated with microcephaly	Open	A widespread epidemic of Zika virus (ZIKV) infection was reported in 2015 in South and Central America and the Caribbean. A major concern associated with this infection is the apparent increased incidence of microcephaly in fetuses born to mothers infected with ZIKV. In this report, we describe the case of an expectant mother who had a febrile illness with rash at the end of the first trimester of pregnancy while she was living in Brazil. Ultrasonography performed at 29 weeks of gestation revealed microcephaly with calcifications in	Upload	✓

Figura 23. Screening de importación de PDF (Digital Medtools).

- **RF-BUS-1 Búsqueda: Permite buscar referencias [CUMPLIDO]**

La aplicación se conecta con SCOPUS para traer la metadata, utilizando el Search API. Complementa algunos abstracts con CrossRef, y los demás son introducidos manualmente. No es posible consumir los servicios de SCOPUS a través del proxy Javeriana por asuntos legales, y una suscripción cuesta alrededor de 140.000 dólares anuales.

```

Array(24) (Object, Object, Object, Object, Object, Object, Object, Object, Object, Object, Object, Object, Object, Object, Object, Object, Object, Object, Object, Object, Object, Object, Object, Object)
  0: Object { SCOPUS_ID: "SCOPUS_ID:14076804730", date: "2013-04-25", title: "The global distribution and burden of dengue", ... }
  1: Object { SCOPUS_ID: "SCOPUS_ID:27844536963", date: "2009-11-17", title: "Impact of regional climate change on human health", ... }
  2: Object { SCOPUS_ID: "SCOPUS_ID:184955126409", date: "2016-03-10", title: "Zika virus associated with microcephaly", ... }
  3: Object { SCOPUS_ID: "SCOPUS_ID:33644847197", date: "2006-03-11", title: "Climate change and human health: Present and future risks", ... }
  4: Object { SCOPUS_ID: "SCOPUS_ID:79955622230", date: "2018-01-01", title: "Dengue: A continuing global threat", ... }
  5: Object { SCOPUS_ID: "SCOPUS_ID:184904761947", date: "2014-01-01", title: "Global antibiotic consumption 2000 to 2010: An analysis of national pharmaceutical sales data", ... }
  6: Object { SCOPUS_ID: "SCOPUS_ID:35748929370", date: "2007-11-16", title: "Dengue", ... }
  7: Object { SCOPUS_ID: "SCOPUS_ID:7684089343", date: "2018-01-01", title: "Insect fat body: Energy, metabolism, and regulation", ... }
  8: Object { SCOPUS_ID: "SCOPUS_ID:8036468861", date: "2002-02-01", title: "Epidemic dengue/dengue hemorrhagic fever as a public health, social and economic problem in the 21st century", ... }
  9: Object { SCOPUS_ID: "SCOPUS_ID:18495538083", date: "2012-04-12", title: "Current concepts: Dengue", ... }
  10: Object { SCOPUS_ID: "SCOPUS_ID:72249193145", date: "2009-12-24", title: "A Wolbachia symbiont in Aedes aegypti limits infection with Dengue, Chikungunya, and Plasmodium", ... }
  11: Object { SCOPUS_ID: "SCOPUS_ID:10944252070", date: "2004-01-01", title: "Emerging flaviviruses: The spread and resurgence of Japanese encephalitis, West Nile and dengue viruses", ... }
  12: Object { SCOPUS_ID: "SCOPUS_ID:1803862395", date: "2000-12-01", title: "Serpins: structure, mechanism, and function", ... }
  13: Object { SCOPUS_ID: "SCOPUS_ID:184937064884", date: "2015-06-30", title: "The global distribution of the arbovirus vectors Aedes aegypti and Ae. albopictus", ... }
  14: Object { SCOPUS_ID: "SCOPUS_ID:8036461364", date: "2002-01-01", title: "Dengue: An update", ... }
  15: Object { SCOPUS_ID: "SCOPUS_ID:3364532211", date: "2006-04-01", title: "Seasonality and the dynamics of infectious diseases", ... }
  16: Object { SCOPUS_ID: "SCOPUS_ID:100951519905", date: "2011-08-29", title: "Successful establishment of Wolbachia in Aedes populations to suppress dengue transmission", ... }
  17: Object { SCOPUS_ID: "SCOPUS_ID:74449092224", date: "2018-02-01", title: "Present and future arboviral threats", ... }
  18: Object { SCOPUS_ID: "SCOPUS_ID:33744411844", date: "2006-01-01", title: "Genome microevolution of chikungunya viruses causing the Indian Ocean outbreak", ... }
  19: Object { SCOPUS_ID: "SCOPUS_ID:184947491870", date: "2015-11-10", title: "Safeguarding human health in the Anthropocene epoch: Report of the Rockefeller Foundation-Lancet Commission on planetary health", ... }
  20: Object { SCOPUS_ID: "SCOPUS_ID:184962082454", date: "2016-07-01", title: "Zika virus", ... }
  21: Object { SCOPUS_ID: "SCOPUS_ID:1803741254", date: "1999-01-01", title: "West Nile fever - A reemerging mosquito-borne viral disease in Europe", ... }
  22: Object { SCOPUS_ID: "SCOPUS_ID:184944217408", date: "2004-09-01", title: "Critical review of the vector status of Aedes albopictus", ... }
  23: Object { SCOPUS_ID: "SCOPUS_ID:184964452166", date: "2016-04-21", title: "Zika Virus", ... }
length: 24

```

Figura 24. Screening de motor de búsqueda (Digital Medtools).

- **RF-GES-2 Gestión: Soporte para múltiples proyectos: Usuario puede crear proyectos, editarlos [CUMPLIDO]**

El staff puede crear múltiples proyectos desde la plataforma de staff. El usuario no tiene permitido hacerlo ya que Digital Medtools no ofrece la aplicación, sino el servicio de asesoría utilizando la aplicación.

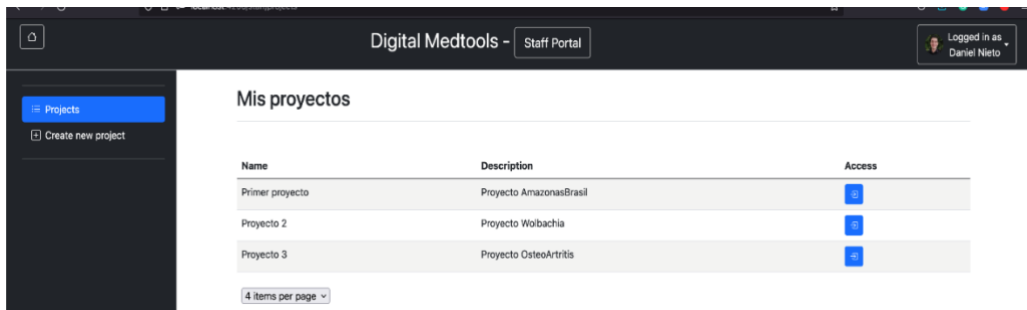


Figura 25. Screening de soporte (Digital Medtools).

- **RF-GES-4      Gestión: Restaurabilidad de trabajo [CUMPLIDO]**

Toda la metadata de los proyectos y los modelos de los diferentes componentes son persistidos a AWS DyanamODB y AWS RDS PostgreSQL donde se encuentran protegidos, por lo que los datos del trabajo tienen un espacio virtual asignado.

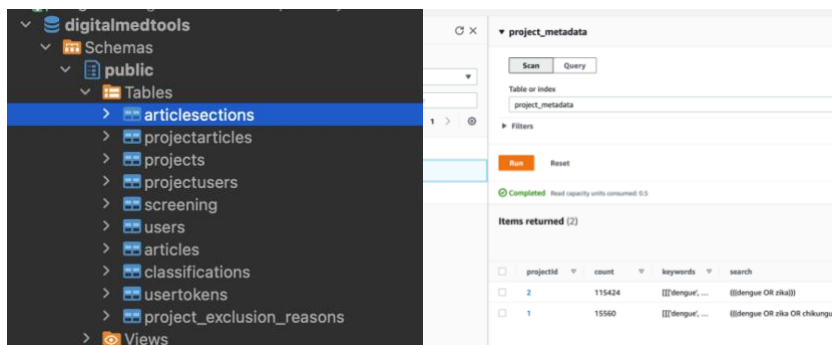


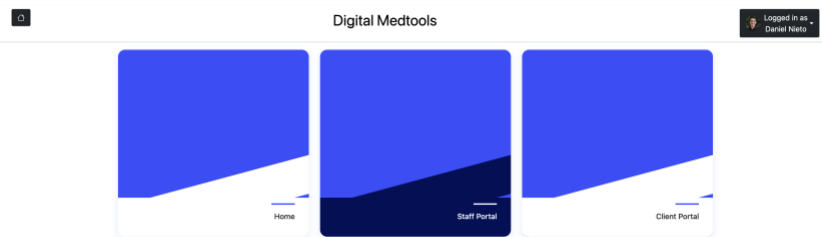
Figura 26. Screening de localización.

- **RF-BUS-5      Búsqueda: Resultado de palabras clave (o similar) [CUMPLIDO]**

En RF-SCR-7 se puede observar que de hecho la herramienta no permite introducir estrategias de búsqueda manuales sino solamente palabras clave. Además, permite sugerir que palabras incluir y cuales excluir.

- **RF-GES-5      Gestión: Gestión de roles [CUMPLIDO]**

Se crearon dos plataformas en la página web, una para clientes y otra para staff. Ambos perfiles tienen roles muy distintos. El del staff como orquestador del flujo de trabajo, con acceso a todos los servicios de Digital Medtools, y el del cliente como el responsable del criterio de aceptación y de gestionar las decisiones del contenido, con acceso limitado.



**Figura 27.** Screening de gestión de roles (Digital Medtools).

## 6.2. Garantía de alta calidad de los resultados

El uso de esta aplicación garantiza alta calidad de los resultados obtenidos, siempre y cuando exista un buen manejo y un compromiso del investigador por leer y revisar en detalle cada una de las referencias, clasificarlas adecuadamente, y construir una estrategia de búsqueda de alta calidad. Esta garantía se da dado que la aplicación fue diseñada para seguir los criterios PRISMA, AMSTAR y ROBIS especificados para estas fases [21, 22, 23, 24, 27, 28, 29, 36, 43].

## 6.3 Resultados de la optimización

En esta sección se presentan los resultados de la aplicación en términos de optimización del proceso

### 6.3.1 Optimización de la elaboración de estrategias de búsqueda

Se observó que el algoritmo fue capaz de identificar qué palabras clave se podían incluir o excluir. En el ejemplo que se probó se utilizó la estrategia de búsqueda:

*TITLE-ABS-KEY(((dengue OR zika OR chikunguya OR "yellow fever") AND (mosquito OR mosquitoes OR aedes OR albopictus)) AND ((control OR prevention)) AND ((success OR rates OR incidence) AND (invasion OR invasive OR foreign)))*

El algoritmo retornó que se debía excluir la palabra clave “invasion” para obtener mejores resultados, dado que el algoritmo TF-IDF logró identificar que esta palabra era relevante en el set de referencias clasificadas por el usuario como “Excluidas”. Esto se pudo obtener al identificar tras valorar que este término era más frecuente en los manuscritos valorados como “Por excluir”

Keyword	Score
dengue	1
zika	2
chikunguya	0
yellow fever	-6
mosquito	-3
mosquitoes	0
aedes	0
albopictus	1
control	3
prevention	2
success	2
rates	5
incidence	1
invasion	-20
invasive	1
foreign	1

**Figura 28.** Resultados del módulo de sugerencias de exclusión de palabras clave

### 6.3.2 Optimización en calidad

Para la fase de extracción de características se evaluó el desempeño de los algoritmos TF-IDF y Bag of Words (BoW), con diferentes combinaciones de hiper-parámetros. Por otro lado, para la fase de aprendizaje



de máquina, se evaluó el desempeño de los algoritmos SVM y Naive Bayes, igualmente con diferentes combinaciones de hiper-parámetros. Dado que ambos eran requeridos para entrenar y evaluar los modelos. Se tuvieron en cuenta las combinaciones de ambos parámetros para medir el desempeño.

Se evaluaron 33 combinaciones de hiper-parametros, cada una con 20 iteraciones de evaluación, donde en cada una de ellas se realizaba 5 fold-cross validation para evaluar los resultados de la clasificación desde 5 perspectivas distintas del dataset. En resumen, para cada combinación de hiper-parámetros se realizó 5-fold cross-validation 20 veces, y se ejecutaron 100 modelos cuyos resultados de precisión, recall y auc fueron promediados. Para obtener estos resultados, en total se ejecutaron 3300 modelos. Esto fue posible dado el gran esfuerzo que se hizo para construir una solución con una arquitectura de software muy sólida.

Se seleccionó el modelo Multinomial Naive-Bayes Alpha=1,5 porque obtuvo el mejor puntaje al realizar 3300 iteraciones entre todas las combinaciones de hiper-parámetros. De la misma manera, para la fase de extracción de características se seleccionó TF-IDF normalizado con L2, con n-gramas entre 1 y 2, un máximo de 500 features y smooth idf (Modelo 33). Los resultados detallados de este proceso se pueden observar en la tabla 15

**Tabla 15.** Resultados del entrenamiento y evaluación de los modelos de inteligencia artificial

Modelo	Configuración TF-IDF	Iteraciones	Cross-Validations	Model runs	Accuracy	auc	$\sigma$ Accuracy	$\sigma$ auc
<b>SVM-SVC C=2,0 poly 2-Degree</b>	TF-IDF smooth_idf-0 ngram_range-1;2 min_df-1 binary-0 use_idf-1 sublinear_tf-0 max_features-500	20	5	100	75,2222	74,2371	11,6967	35,1674
<b>SVM-SVC C=2,0 poly 2-Degree</b>	TF-IDF smooth_idf-0 ngram_range-1;2 min_df-1 binary-0 use_idf-1 sublinear_tf-1 max_features-500	20	5	100	76,7778	76,8036	10,9523	32,9614
<b>SVM-SVC C=2,0 poly 2-Degree</b>	TF-IDF smooth_idf-0 ngram_range-1;2 min_df-1 binary-1 use_idf-1 sublinear_tf-1 max_features-500	20	5	100	74,8889	74,2708	10,3101	33,9491
<b>SVM-SVC C=2,0 poly 2-Degree</b>	TF-IDF smooth_idf-1 ngram_range-1;2 min_df-1 binary-1 use_idf-1 sublinear_tf-1 max_features-500	20	5	100	77,2222	74,6925	10,87	35,6856
<b>SVM-SVC C=2,0 poly 2-Degree</b>	TF-IDF smooth_idf-1 ngram_range-1;2 min_df-1 binary-0 use_idf-1 sublinear_tf-1 max_features-500	20	5	100	77,8889	81,8899	8,4302	27,2044
<b>SVM-SVC C=2,0 poly 2-Degree</b>	TF-IDF smooth_idf-1 ngram_range-1;3 min_df-1 binary-0 use_idf-1 sublinear_tf-1 max_features-500	20	5	100	77,1111	79,381	9,4519	29,6802
<b>SVM-SVC C=2,0 poly 2-Degree</b>	TF-IDF smooth_idf-1 ngram_range-1;2 min_df-1 binary-0 use_idf-1 sublinear_tf-1 max_features-300	20	5	100	78,5556	82,4018	10,9705	28,9814
<b>SVM-SVC C=2,0 poly 2-Degree</b>	TF-IDF smooth_idf-1 ngram_range-1;2 min_df-1 binary-0 use_idf-1 sublinear_tf-1 max_features-200	20	5	100	77,7778	80,9226	8,5046	28,3944
<b>SVM-SVC C=2,0 poly 2-Degree</b>	TF-IDF smooth_idf-1 ngram_range-1;2 min_df-1 binary-0 use_idf-1 sublinear_tf-1 max_features-250	20	5	100	77,4445	77,2718	10,0448	33,5573
<b>SVM-SVC C=2,0 poly 2-Degree</b>	TF-IDF smooth_idf-1 ngram_range-1;2 min_df-1 binary-0 use_idf-1 sublinear_tf-1 max_features-300	20	5	100	77,4445	77,002	9,4036	32,675
<b>SVM-SVC C=2,0 poly 2-Degree</b>	TF-IDF smooth_idf-1 ngram_range-1;2 min_df-1 binary-0 use_idf-1 sublinear_tf-1 max_features-300	20	5	100	78,1111	77,8671	8,7153	32,6213
<b>SVM-SVC C=2,0 poly 2-Degree</b>	TF-IDF smooth_idf-1 ngram_range-1;2 min_df-1 binary-0 use_idf-1 sublinear_tf-1 max_features-500	20	5	100	77,2222	79,5179	10,401	30,8103
<b>SVM-SVC C=2,0 poly 2-Degree</b>	TF-IDF smooth_idf-1 ngram_range-1;3 min_df-1 binary-0 use_idf-1 sublinear_tf-1 max_features-500	20	5	100	76,7778	74,7282	10,2464	35,2121
<b>SVM-SVC C=2,0 poly 2-Degree</b>	TF-IDF smooth_idf-1 ngram_range-1;2 min_df-1 binary-0 use_idf-1 sublinear_tf-1 max_features-300 norm-l2	20	5	100	77,3334	81,3135	9,725	28,324

<b>SVM-SVC C=2,0 poly 2-Degree</b>	TF-IDF smooth_idf-1 ngram_range-1;2 min_df-1 binary-0 use_idf-1 sublinear_tf-1 max_features-300 norm-l1	20	5	100	77,1111	76,5724	9,319	33,2885
<b>SVM-SVC C=2,0 poly 2-Degree</b>	BoW ngram_range-1;2 min_df-1 binary-0 max_features-300 norm-l2	20	5	100	62,5556	74,3343	12,0951	21,9081
<b>SVM-SVC C=2,0 poly 2-Degree</b>	BoW ngram_range-1;3 min_df-1 binary-0 max_features-300 norm-l2	20	5	100	62,4445	73,3413	11,9002	23,9616
<b>SVM-SVC C=2,0 poly 2-Degree</b>	BoW ngram_range-2;3 min_df-1 binary-0 max_features-300 norm-l2	20	5	100	46,1111	15,2808	6,7697	30,0106
<b>SVM-SVC C=2,0 poly 2-Degree</b>	BoW ngram_range-1;3 min_df-1 binary-1 max_features-300 norm-l2	20	5	100	67,3334	68,6935	12,7148	33,9548
<b>SVM-SVC C=2,0 poly 2-Degree</b>	BoW ngram_range-1;3 min_df-3 binary-1 max_features-300 norm-l2	20	5	100	67,5556	70,9256	13,4636	30,6888
<b>SVM-SVC C=2,0 poly 2-Degree</b>	TF-IDF smooth_idf-1 ngram_range-1;2 min_df-1 binary-0 use_idf-1 sublinear_tf-1 max_features-500 norm-l2	20	5	100	76,3334	79,2302	10,3143	29,569
<b>SVM-SVC C=2,0 poly 2-Degree</b>	TF-IDF smooth_idf-1 ngram_range-1;2 min_df-1 binary-1 use_idf-1 sublinear_tf-1 max_features-500 norm-l2	20	5	100	76,1111	79,0556	10,87	30,5809
<b>SVM-SVC C=2,0 poly 2-Degree</b>	TF-IDF smooth_idf-1 ngram_range-1;2 min_df-1 binary-0 use_idf-1 sublinear_tf-1 max_features-300 norm-l2	20	5	100	77,5556	78,8889	9,0694	31,6108
<b>SVM-SVC C=2,0 linear 2-Degree</b>	TF-IDF smooth_idf-1 ngram_range-1;2 min_df-1 binary-0 use_idf-1 sublinear_tf-1 max_features-500 norm-l2	20	5	100	77,1111	80,3264	9,5829	29,5806
<b>SVM-SVC C=1,0 linear 2-Degree</b>	TF-IDF smooth_idf-1 ngram_range-1;2 min_df-1 binary-0 use_idf-1 sublinear_tf-1 max_features-500 norm-l2	20	5	100	78,7778	90,3968	8,6637	9,1132
<b>SVM-SVC C=1,0 linear 1-Degree</b>	TF-IDF smooth_idf-1 ngram_range-1;2 min_df-1 binary-0 use_idf-1 sublinear_tf-1 max_features-500 norm-l2	20	5	100	76,8889	77,004	10,0727	32,6419
<b>SVM-SVC C=1,0 rbf 1-Degree</b>	TF-IDF smooth_idf-1 ngram_range-1;2 min_df-1 binary-0 use_idf-1 sublinear_tf-1 max_features-500 norm-l2	20	5	100	77,6667	83,8532	10,1731	23,8945
<b>SVM-SVC C=0,8 rbf 1-Degree</b>	TF-IDF smooth_idf-1 ngram_range-1;2 min_df-1 binary-0 use_idf-1 sublinear_tf-1 max_features-500 norm-l2	20	5	100	79,6667	91,1508	9,3504	9,8977
<b>SVM-SVC C=0,8 rbf 2-Degree</b>	TF-IDF smooth_idf-1 ngram_range-1;2 min_df-1 binary-0 use_idf-1 sublinear_tf-1 max_features-500 norm-l2	20	5	100	78	89,8661	11,1648	10,8477
<b>Multinomial Naive-Bayes</b>	TF-IDF smooth_idf-1 ngram_range-1;2 min_df-1 binary-0 use_idf-1 sublinear_tf-1 max_features-500 norm-l2	20	5	100	78,8889	90,253	8,7218	9,2118

<b>Multinomial Naive-Bayes Alpha=0,8</b>	TF-IDF smooth_idf-1 ngram_range-1;2 min_df-1 binary-0 use_idf-1 sublinear_tf-1 max_features-500 norm-l2	20	5	100	79,6667	91,1994	10,119	10,5252
<b>Multinomial Naive-Bayes Alpha=0,5</b>	TF-IDF smooth_idf-1 ngram_range-1;2 min_df-1 binary-0 use_idf-1 sublinear_tf-1 max_features-500 norm-l2	20	5	100	79,5556	91,1032	9,3056	9,5751
<b>Multinomial Naive-Bayes Alpha=1,5</b>	TF-IDF smooth_idf-1 ngram_range-1;2 min_df-1 binary-0 use_idf-1 sublinear_tf-1 max_features-500 norm-l2	20	5	100	79,6667	91,129	10,119	10,6639

#### 6.4 Comparación con los resultados y el ahorro de tiempo de otras herramientas

En la tabla 16, a continuación, se compara Digital Medtools con otras herramientas. Se utilizó como base para construir la tabla Jin *et al* (2021), que identificó las metodologías de inteligencia artificial usadas en algunas de las herramientas más populares [75]. Se muestran el accuracy y AUC como medidas de precisión del modelo, y el WSS @95 (Work Saved over random Sampling at 0,95 recall), que representa la reducción en esfuerzo de muestreo para obtener el 95% de la literatura relevante en el área. A partir de este valor se puede estimar el tiempo en segundos ahorrado por referencia en la fase de selección de estudios [69]. Digital Medtools obtuvo un valor WSS de 0,1183 a un recall de 0,80. Es probable que el alto porcentaje de referencias incluidas, tal como en el id 4 de la tabla 16, produjera un bajo valor de WSS. Adicionalmente, el bajo número de abstracts analizados en este estudio, dadas las limitaciones de recursos, de investigadores y de tiempo, debió tener un gran impacto en esta medida. Queda pendiente fortalecer este análisis y enfocar el entrenamiento de los modelos a esta métrica para próximos estudios.

**Tabla 16.** Resultados del entrenamiento y evaluación de los modelos de inteligencia artificial

Referencia	Aplicación	Modelo	Cross-Validations	Corpus	% Incluidos	Abstracts por Revision	Accuracy	AUC	WSS @0,95
-	Digital Medtools	Naïve Bayes + TF-IDF	5-Fold CV X10 iteraciones	1	53,84%	52	0,7997	0,9113	WSS@80 = 0,1183
[69]	Rayyan	SVM	2-Fold CV	2	0,5-21,7%	273-1030	-	0,87	0,49
[103]	-	Naïve Bayes/SVM + TF-IDF	5-Fold CV	1	27%	2000	0,9	-	0,7
[104]	-	Random Forest + TF-IDF	5-Fold CV	1	50%	2000	0,8	0,9	WSS@80 = 0,68
[105]	AbstrackR	SVM + TF-IDF	No Data	4	1,5-59%	5243-47141	0,266-0,647	-	0,672-0,884
[106]	ASReview	Naïve Bayes + TF-IDF	No Data	2	0,66-1,66%	2544-5782	-	-	0,67-0,92
[107]	Robot Analyst	Latent Dirichlet + TF-IDF	No	22		86-2000	-	0,4729 – 0,9196	0,07-0,71

Modificado, procesado y sintetizado a partir de [69, 75, 103, 104, 105, 106, 107]

## 7. CONCLUSIONES

En este capítulo se presentan las conclusiones obtenidas tras culminar las distintas fases del proyecto

### 7.1 Conclusiones del proyecto

A partir del trabajo realizado y de la información contenida a lo largo de este documento, se puede concluir que:

- Se cumplió el objetivo principal del proyecto al diseñar e implementar la aplicación funcional, se cumplieron el 100% de los objetivos específicos, se logró cubrir 18 de 20 casos de uso identificados, y se logró elaborar el 100% de la documentación y entregables planteados.
- Se diseñó y construyó un prototipo de software funcional, capaz de optimizar el proceso y los resultados de las fases de búsqueda y selección de artículos de la elaboración de revisiones sistemáticas médicas, utilizando técnicas de inteligencia artificial
- Se logró reducir el tiempo de trabajo del proceso de identificar la literatura relevante de de 34 a 12 horas laborales. La simplicidad del flujo de trabajo observada en Digital Medtools es una gran prueba de la potencia de la herramienta construida.
- El módulo de sugerencia de inclusión/exclusión de palabras clave cumplió su objetivo de clasificar las palabras clave utilizando procesamiento de lenguaje natural e information retrieval (TF-IDF). Estas sugerencias tienen un alto valor para facilitar la construcción de la estrategia de búsqueda
- El módulo de sugerencia de inclusión/exclusión de manuscritos durante el screening logró clasificar los artículos utilizando machine learning (SVM, Bayes Naive) además de PLN y TF-IDF. Se obtuvo una precisión de 79%, y un AUC. De 91%, los cuales son resultados muy buenos para un módulo de procesamiento natural con capacidades de inteligencia artificial. Adicionalmente vale la pena aclarar que se utilizó la regularización L2, validaciones cruzadas 5-fold, y 20 ejecuciones por cada combinación de hiperparámetros por modelo, para garantizar la máxima calidad y veracidad de estos resultados.
- Se logró construir una plataforma de gran utilidad para el crecimiento exponencial e internacional que Digital Medtools tiene planeado, a través del uso de tecnologías AWS y aplicaciones RESTful.
- El flujo de trabajo permite redefinir la pregunta problema, características del proyecto y estrategia de búsqueda a utilizar, lo cual es de gran utilidad para los investigadores.
- Esta aplicación garantiza resultados de alta calidad para las fases de búsqueda de literatura y selección de estudios ya que sigue los criterios PRISMA, AMSTAR y ROBIS especificados para esta fase, pero siempre y cuando el investigador realice un trabajo detallado y juicioso.
- La arquitectura en capas aumentó la seguridad dado que la capa de presentación es la única que está expuesta al usuario y que la capa AWS Cloud solo está expuesta a la capa lógica. Además, se utilizaron tokens JWT y encriptación BlowFish en las claves y se hizo validación de roles.
- La forma modular en que fue construida la arquitectura de la aplicación permite que sus componentes puedan ser reutilizados fácilmente, no solo en otros módulos sino también en otros proyectos
- Se desplegó un servidor RESTful capaz de realizar tareas de inteligencia artificial en tiempo real, y que además cuenta con la capacidad de utilizar infraestructura de AWS Cloud. Esta combinación es poco común, y es altamente valorada en el entorno laboral, ya que la mayoría de proyectos de inteligencia artificial suelen desplegarse como módulos 1-file en jupyter notebooks.

## 8. TRABAJO FUTURO

El prototipo construido durante este proyecto permitió cubrir los requerimientos básicos y el flujo de trabajo básico para optimizar el esfuerzo en términos de tiempo y calidad, sin embargo, se pueden realizar mejoras que permitan expandir las capacidades de la aplicación, su usabilidad y el interés por parte de los usuarios. En esta sección del documento se mencionan los objetivos e ideas que pueden ser de interés para mejorar las capacidades de esta aplicación.

### *8.1 Optimización en calidad*

Aunque los resultados de los algoritmos de inteligencia artificial fueron bastante buenos respecto a lo que se espera de un módulo de procesamiento de lenguaje natural con capacidades de inteligencia artificial, es necesario evaluar durante la ejecución de proyectos con datos reales como se comportan los algoritmos y el procesamiento de lenguaje natural, para hacer los ajustes respectivos.

Un punto importante es verificar de qué forma se puede apoyar al asesor y al cliente para que los puntos donde pueda existir más riesgo de reducir la calidad se puedan resolver a través del desarrollo de nuevas características.

### *8.2 Flujo de trabajo*

Aunque en esta versión se incluyeron los requerimientos funcionales más importantes, lo cual permitió garantizar el flujo de trabajo básico y llevar a cabo a completitud el proceso de selección de referencias, queda pendiente incorporar a las características de la aplicación aquellos otros elementos de los anexos 3, 4 y 5 que se considere que aportan un alto valor desde la perspectiva de negocio.

## 9. REFERENCIAS

- [1] D. Putler and R. Krider, "A Process Model for Data Mining—CRISP-DM", in *Customer and Business Analytics*, 1st ed., D. Putler and R. Krider, Ed. New York: Chapman and Hall/CRC, 2012. doi: 10.1201/b12040
- [2] W. Ayele, "Adapting CRISP-DM for Idea Mining", *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 6, 2020. doi: 10.14569/ijacsa.2020.0110603
- [3] F. Schafer, C. Zeiselmaier, J. Becker and H. Otten, "Synthesizing CRISP-DM and Quality Management: A Data Mining Approach for Production Processes", 2018 IEEE International Conference on Technology Management, Operations and Decisions (ICTMOD), 2018, pp. 190–195. doi: 10.1109/itmc.2018.8691266
- [4] D. Abrams et al., "Area Under the Curve (AUC)", *Encyclopedia of Behavioral Medicine*, pp. 125-126, 2013. Available: 10.1007/978-1-4419-1005-9\_986
- [5] W. Stuetzle, "Cross-Validation", in *Encyclopedia of statistics in behavioral science*, B. Everitt, Ed. Chichester, England: John Wiley & Sons, 2005. doi: 10.1002/0470013192.bsa158
- [6] R. Wojcik et al., "Attribute-Driven Design (ADD), Version 2.0", Carnegie Mellon University, 2006. Available: 10.1184/R1/6572066.v1 [Accessed 6 December 2021].
- [7] R. Kazman, M. Klein and P. Clements, "ATAM: Method for Architecture Evaluation", *Resources.sei.cmu.edu*, 2000. [Online]. Available: [https://resources.sei.cmu.edu/asset\\_files/TechnicalReport/2000\\_005\\_001\\_13706.pdf](https://resources.sei.cmu.edu/asset_files/TechnicalReport/2000_005_001_13706.pdf). [Accessed: 26-Nov- 2021].
- [8] [K. de Graaf, A. Tang, P. Liang and H. van Vliet, "Ontology-based Software Architecture Documentation", 2012 Joint Working IEEE/IFIP Conference on Software Architecture and European Conference on Software Architecture, 2012, pp. 121–130. doi: 10.1109/wicsa-ecsa.212.20
- [9] K. de Graaf, P. Liang, A. Tang and H. van Vliet, "How organisation of architecture documentation affects architectural knowledge retrieval", *Science of Computer Programming*, vol. 121, pp. 75-99, 2016. doi: 10.1016/j.scico.2015.10.014
- [10] P. Kruchten, "The 4+1 View Model of architecture", *IEEE Software*, vol. 12, no. 6, pp. 42-50, 1995. Available: 10.1109/52.469759.
- [11] M. Babar and I. Gorton, "Comparison of Scenario-Based Software Architecture Evaluation Methods", 11th Asia-Pacific Software Engineering Conference, pp. 600-607, 2004. doi: 10.1109/apsec.2004.38
- [12] M. Poppendieck, "Lean Software Development", 29th International Conference on Software Engineering (ICSE 07 Companion), 2007. doi: 10.1109/icsecompanion.2007.46 [Accessed 26 November 2021].
- [13] A. Janes and G. Succi, "Lean Software Development in Action", 2014. Available: 10.1007/978-3-642-00503-9 [Accessed 6 December 2021].
- [14] B. Djulbegovic and G. Guyatt, "Progress in evidence-based medicine: a quarter century on", *The Lancet*, vol. 390, no. 10092, pp. 415-423, 2017. Available: 10.1016/s0140-6736(16)31592-6
- [15] J. McKenzie and S. Brennan, "Overviews of systematic reviews: great promise, greater challenge", *Systematic Reviews*, vol. 6, no. 1, 2017. doi: 10.1186/s13643-017-0582-8
- [16] P. Shekelle, M. Eccles, J. Grimshaw and S. Woolf, "When should clinical guidelines be updated?", *BMJ*, vol. 323, no. 7305, pp. 155-157, 2001. doi: 10.1136/bmj.323.7305.155

- [17] P. Ganeshkumar and S. Gopalakrishnan, "Systematic reviews and meta-analysis: Understanding the best evidence in primary healthcare", *Journal of Family Medicine and Primary Care*, vol. 2, no. 1, p. 9, 2013. doi: 10.4103/2249-4863.109934
- [18] J. Grimshaw, M. Eccles and J. Tetroe, "Implementing clinical guidelines: Current evidence and future implications", *Journal of Continuing Education in the Health Professions*, vol. 24, no. 1, pp. S31-S37, 2004. doi: 10.1002/chp.1340240506
- [19] A. Selva et al., "Development and use of a content search strategy for retrieving studies on patients' views and preferences", *Health and Quality of Life Outcomes*, vol. 15, no. 1, 2017. doi: 10.1186/s12955-017-0698-5
- [20] M. Pollock, R. Fernandes and L. Hartling, "Evaluation of AMSTAR to assess the methodological quality of systematic reviews in overviews of reviews of healthcare interventions", *BMC Medical Research Methodology*, vol. 17, no. 1, 2017. doi: 10.1186/s12874-017-0325-5
- [21] B. Shea et al., "AMSTAR is a reliable and valid measurement tool to assess the methodological quality of systematic reviews", *Journal of Clinical Epidemiology*, vol. 62, no. 10, pp. 1013-1020, 2009. doi: 10.1016/j.jclinepi.2008.10.009
- [22] B. Shea et al., "AMSTAR 2: a critical appraisal tool for systematic reviews that include randomised or non-randomised studies of healthcare interventions, or both", *BMJ*, p. j4008, 2017. doi: 10.1136/bmj.j4008
- [23] M. Page et al., "The PRISMA 2020 statement: an updated guideline for reporting systematic reviews", *BMJ*, p. n71, 2021. doi: 10.1136/bmj.n71
- [24] M. Rethlefsen et al., "PRISMA-S: an extension to the PRISMA Statement for Reporting Literature Searches in Systematic Reviews", *Systematic Reviews*, vol. 10, no. 1, 2021. doi: 10.1186/s13643-020-01542-z
- [25] A. Methley, S. Campbell, C. Chew-Graham, R. McNally and S. Cheraghi-Sohi, "PICO, PICOS and SPIDER: a comparison study of specificity and sensitivity in three search tools for qualitative systematic reviews", *BMC Health Services Research*, vol. 14, no. 1, 2014. Available: 10.1186/s12913-014-0579-0
- [26] M. Eriksen and T. Frandsen, "The impact of patient, intervention, comparison, outcome (PICO) as a search strategy tool on literature search quality: a systematic review", *Journal of the Medical Library Association*, vol. 106, no. 4, 2018. doi: 10.5195/jmla.2018.345
- [27] B. Shea et al., "Amstar 2 Guidance document", Amstar.ca, 2021. [Online]. Available: <https://amstar.ca/docs/AMSTAR%202-Guidance-document.pdf>. [Accessed: 11- Nov- 2021].
- [28] P. Whiting, P. Davies, J. Savović, D. Caldwell, and R. Churchill, "Evidence to inform the development of ROBIS, a new tool to assess the risk of bias in systematic reviews", *researchgate.net*, 2013. [Online]. doi: 10.13140/RG.2.1.5068.6327
- [29] P. Whiting et al., "ROBIS: A new tool to assess risk of bias in systematic reviews was developed", *Journal of Clinical Epidemiology*, vol. 69, pp. 225-234, 2016. doi: 10.1016/j.jclinepi.2015.06.005
- [30] F. Thielen et al., "How to prepare a systematic review of economic evaluations for clinical practice guidelines: database selection and search strategy development (part 2/3)", *Expert Review of Pharmacoeconomics & Outcomes Research*, vol. 16, no. 6, pp. 705-721, 2016. doi: 10.1080/14737167.2016.1246962
- [31] G. van Mastrigt et al., "How to prepare a systematic review of economic evaluations for informing evidence-based healthcare decisions: a five-step approach (part 1/3)", *Expert Review of*



- Pharmacoeconomics & Outcomes Research, vol. 16, no. 6, pp. 689-704, 2016. doi: 10.1080/14737167.2016.1246960
- [32] M. Alarcón Gil, S. Osorio Toro and G. Baena Caldas, "THE EVIDENCE-BASED MEDICINE PICO STRATEGY APPLIED TO DENTISTRY USING MESH, EMTREE AND DECS", Revista Facultad de Odontología, vol. 31, no. 1-2, 2019. doi: 10.17533/udea.rfo.v31n1-2a8
- [33] N. Ali and M. Usman, "Reliability of search in systematic reviews: Towards a quality assessment framework for the automated-search strategy", Information and Software Technology, vol. 99, pp. 133-147, 2018. doi: 10.1016/j.infsof.2018.02.002
- [34] J. Nicholson, A. McCrillis and J. Williams, "Collaboration challenges in systematic reviews: a survey of health sciences librarians", Journal of the Medical Library Association, vol. 105, no. 4, 2017. doi: 10.5195/jmla.2017.176
- [35] D. Richards, "The quality of systematic reviews in dentistry", Evidence-Based Dentistry, vol. 5, no. 1, pp. 17-17, 2004. doi: 10.1038/sj.ebd.6400242
- [36] M. Storman, D. Storman, K. Jasinska, M. Swierz and M. Bala, "The quality of systematic reviews/meta-analyses published in the field of bariatrics: A cross-sectional systematic survey using AMSTAR 2 and ROBIS", Obesity Reviews, vol. 21, no. 5, 2020. doi: 10.1111/obr.12994
- [37] B. Wijnen, G. Van Mastrigt, W. Redekop, H. Majoie, R. De Kinderen and S. Evers, "How to prepare a systematic review of economic evaluations for informing evidence-based healthcare decisions: data extraction, risk of bias, and transferability (part 3/3)", Expert Review of Pharmacoeconomics & Outcomes Research, vol. 16, no. 6, pp. 723-732, 2016. doi: 10.1080/14737167.2016.1246961
- [38] Y. Yu et al., "Assessment of the quality of systematic reviews on COVID-19: A comparative study of previous coronavirus outbreaks", Journal of Medical Virology, vol. 92, no. 7, pp. 883-890, 2020. doi: 10.1002/jmv.25901
- [39] A. Elshafay et al., "Reporting quality in systematic reviews of in vitro studies: a systematic review", Current Medical Research and Opinion, vol. 35, no. 9, pp. 1631-1641, 2019. doi: 10.1080/03007995.2019.1607270
- [40] E. Opheim, P. Andersen, M. Jakobsen, B. Aasen and K. Kvaal, "Poor Quality in Systematic Reviews on PTSD and EMDR – An Examination of Search Methodology and Reporting", Frontiers in Psychology, vol. 10, 2019. doi: 10.3389/fpsyg.2019.01558
- [41] D. Layton, "A Critical Review of Search Strategies Used in Recent Systematic Reviews Published in Selected Prosthodontic and Implant-Related Journals: Are Systematic Reviews Actually Systematic?", The International Journal of Prosthodontics, pp. 13-21, 2017. doi: 10.11607/ijp.5193
- [42] V. Montori, M. Swiontkowski and D. Cook, "Methodologic Issues in Systematic Reviews and Meta-Analyses", Clinical Orthopaedics & Related Research, vol. 413, pp. 43-54, 2003. doi: 10.1097/01.blo.0000079322.41006.5b
- [43] P. Fleming, J. Seehra, A. Polychronopoulou, Z. Fedorowicz and N. Pandis, "A PRISMA assessment of the reporting quality of systematic reviews in orthodontics", The Angle Orthodontist, vol. 83, no. 1, pp. 158-163, 2012. doi: 10.2319/032612-251.1
- [44] T. Cox, "Sensitivity and specificity", Ophthalmology, vol. 110, no. 9, p. 1858, 2003. doi: 10.1016/s0161-6420(03)00841-8
- [45] F. Shokraneh, "Automation of Systematic Reviews", Researchgate.net, 2018. [Online]. doi: 10.13140/RG.2.2.36649.03681. [Accessed: 26- Nov- 2021].

- [46] C. Marshall and A. Sutton, "Systematic Review Toolbox", Systematicreviewtools.com, 2021. [Online]. Available: <http://www.systematicreviewtools.com/>
- [47] C. Battoui, "A Text Miner analysis to compare internet and medline information about allergy medications", University of Louisville, 2004. Available: <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.86.4021&rep=rep1&type=pdf> [Accessed: 11- Nov- 2021].
- [48] T. Strzalkowski, "Natural language information retrieval", Information Processing & Management, vol. 31, no. 3, pp. 397-417, 1995. doi: 10.1016/0306-4573(94)00055-8
- [49] T. Iwakura and H. Yoshikawa, "Natural Language Processing Application to the Chemical Domain", Journal of Natural Language Processing, vol. 27, no. 4, pp. 969-973, 2020. doi: 10.5715/jnlp.27.969
- [50] H. Christian, M. Agus and D. Suhartono, "Single Document Automatic Text Summarization using Term Frequency-Inverse Document Frequency (TF-IDF)", ComTech: Computer, Mathematics and Engineering Applications, vol. 7, no. 4, p. 285, 2016. doi: 10.21512/comtech.v7i4.3746
- [51] J. Whissell and C. Clarke, "Improving document clustering using Okapi BM25 feature weighting", Information Retrieval, vol. 14, no. 5, pp. 466-487, 2011. Available: 10.1007/s10791-011-9163-y
- [52] T. Strzalkowski, F. Lin, J. Wang and J. Perez-Carballo, "Evaluating Natural Language Processing Techniques in Information Retrieval", Text, Speech and Language Technology, pp. 113-145, 1999. doi: 10.1007/978-94-017-2388-6\_5
- [53] scikit-learn developers, "Support Vector Machines", Scikit Learn, 2021. [Online]. Available: <https://scikit-learn.org/stable/modules/svm.html>. [Accessed: 05-Nov-2021].
- [54] scikit-learn developers, "1.11. Ensemble methods", Scikit Learn, 2021. [Online]. Available: <https://scikit-learn.org/stable/modules/ensemble.html>. [Accessed: 05- Nov- 2021].
- [55] C. Bustos Navarrete, M. Morales Malverde, P. Salcedo Lagos and A. Díaz Mujica, "Buhos: A web-based systematic literature review management software", SoftwareX, vol. 7, pp. 360-372, 2018. doi: 10.1016/j.softx.2018.10.004
- [56] VTS Software, "SRDB.PRO - Enterprise Systematic Review Software", SRDP.PRO, 2021. [Online]. Available: <https://www.srdb.pro/>. [Accessed: 05- Nov- 2021].
- [57] D. Bowes, T. Hall and S. Beecham, "SLuRp", Proceedings of the 2nd international workshop on Evidential assessment of software technologies - EAST 12, 2012, pp. 33-36. doi: 10.1145/2372233.2372243
- [58] C. Hamel, S. Kelly, K. Thavorn, D. Rice, G. Wells and B. Hutton, "An evaluation of DistillerSR's machine learning-based prioritization tool for title/abstract screening – impact on reviewer-relevant outcomes", BMC Medical Research Methodology, vol. 20, no. 1, p.256 2020. doi: 10.1186/s12874-020-01129-1
- [59] Z. Munn, "Software to support the systematic review process", JBI Database of Systematic Reviews and Implementation Reports, vol. 14, no. 10, p. 1, 2016. doi: 10.11124/jbisrir-2016-002421
- [60] Nested Knowledge, "Bringing Systematic Review to Life", Nested Knowledge, 2021. [Online]. Available: <https://nested-knowledge.com>. [Accessed: 05-Nov-2021].
- [61] S. Fabbri, C. Silva, E. Hernandez, F. Octaviano, A. Di Thommazo and A. Belgamo, "Improvements in the StArt tool to better support the systematic review process", Proceedings of the 20th International Conference on Evaluation and Assessment in Software Engineering, 2016. doi: 10.1145/2915970.2916013

- [62] T. Turner, E. Steele, C. Mavergames and J. Elliott, "Facilitating Web-Based Collaboration in Evidence Synthesis (TaskExchange): Development and Analysis", *JMIR Research Protocols*, vol. 7, no. 12, p. e188, 2018. doi: 10.2196/resprot.9285
- [63] C. Kohl et al., "Online tools supporting the conduct and reporting of systematic reviews and systematic maps: a case study on CADIMA and review of existing tools", *Environmental Evidence*, vol. 7, no. 1, 2018. doi: 10.1186/s13750-018-0115-5
- [64] J. Thomas and J. Brunton, "EPPI Reviewer: Software for research synthesis", *EPPI Reviewer*, 2007. [Online]. Available: <https://discovery.ucl.ac.uk/id/eprint/10003909/>. [Accessed: 05- Nov- 2021].
- [65] B. Bigendako and E. Syriani, "Modeling a Tool for Conducting Systematic Reviews Iteratively", *Proceedings of the 6th International Conference on Model-Driven Engineering and Software Development*, 2018. doi: 10.5220/0006664405520559
- [66] B. Kitchenham and S. Charters, "Guidelines for performing Systematic Literature Reviews in Software Engineering", Elsevier.com, 2007. [Online]. Available: [https://www.elsevier.com/\\_\\_data/promis\\_misc/525444systematicreviewsguide.pdf](https://www.elsevier.com/__data/promis_misc/525444systematicreviewsguide.pdf). [Accessed: 11- Nov- 2021].
- [67] J. Molléri and F. Benitti, "SESRA: a web-based automated tool to support the systematic literature review process", *Proceedings of the 19th International Conference on Evaluation and Assessment in Software Engineering*, 2015. doi: 10.1145/2745802.2745825
- [68] H. Harrison, S. Griffin, I. Kuhn and J. Usher-Smith, "Software tools to support title and abstract screening for systematic reviews in healthcare: an evaluation", *BMC Medical Research Methodology*, vol. 20, no. 1, 2020. doi: 10.1186/s12874-020-0897-3
- [69] M. Ouzzani, H. Hammady, Z. Fedorowicz and A. Elmagarmid, "Rayyan—a web and mobile app for systematic reviews", *Systematic Reviews*, vol. 5, no. 1, 2016. doi: 10.1186/s13643-016-0384-4.
- [70] L. Kellermeyer, B. Harnke and S. Knight, "Covidence and Rayyan", *Journal of the Medical Library Association*, vol. 106, no. 4, 2018. doi: 10.5195/jmla.2018.513
- [71] J. Babineau, "Product Review: Covidence (Systematic Review Software)", *Journal of the Canadian Health Libraries Association / Journal de l'Association des bibliothèques de la santé du Canada*, vol. 35, no. 2, p. 68, 2014. doi: 10.5596/c14-016
- [72] "DRAGON: An online tool for systematic review", *Health & Environmental Research Online (HERO)*, 2021. [Online]. Available: [https://hero.epa.gov/hero/index.cfm/reference/details/reference\\_id/4593151](https://hero.epa.gov/hero/index.cfm/reference/details/reference_id/4593151). [Accessed: 05- Nov- 2021].
- [73] B. Wallace, K. Small, C. Brodley, J. Lau and T. Trikalinos, "Deploying an interactive machine learning system in an evidence-based practice center", *Proceedings of the 2nd ACM SIGHIT symposium on International health informatics - IHI '12*, pp. 819-824, 2012. doi: 10.1145/2110363.2110464
- [74] M. Kahili-Heede and K. Hillgren, "Colandr", *Journal of the Medical Library Association*, vol. 109, no. 3, 2021. doi: 10.5195/jmla.2021.1263.
- [75] D. Jin et al., "Artificial intelligence in radiology", *Artificial Intelligence in Medicine*, pp. 265-289, 2021. doi: 10.1016/b978-0-12-821259-2.00014-4
- [76] V. Jakkula, "Tutorial on support vector machine (SVM)", *School of EECS, Washington State University*, 2011. Available: <https://course.ccs.neu.edu/cs5100f11/resources/jakkula.pdf> [Accessed 5 November 2021].

- [77] B. Liu, Z. Jiang and K. Fang, "Stable predictive control based on Toeplitz equation and LS-SVM", 2008 Chinese Control and Decision Conference, 2008. doi: 10.1109/ccdc.2008.4597588
- [78] H. Zhang, "The Optimality of Naive Bayes", University of New Brunswick, 2004. [Online]. Available: <https://www.cs.unb.ca/~hzhang/publications/FLAIRS04ZhangH.pdf>.
- [79] D. Berrar, "Bayes' Theorem and Naive Bayes Classifier", in Encyclopedia of Bioinformatics and Computational Biology: ABC of Bioinformatics, S. Ranganathan, M. Gribskov, K. Nakai and C. Schönbach, Ed. US: Elsevier, 2018.
- [80] Abhisek, G., "MVVM-1: A General Discussion", Medium, 2021. [online]. Available: <https://medium.com/swift-india/mvvm-1-a-general-discussion-764581a2d5d9> [Accessed 4 November 2021].
- [81] F. Yan, Z. Xu, Y. S. Zhong, Z. HaiBei, and C. Y. Ge, "Research on Performance Optimization Scheme for Web Front-End and Its Practice", Communications, Signal Processing, and Systems, pp. 883–890, 2021. doi: 10.1007/978-981-15-8411-4\_118
- [82] Amazon Web Services, "Amazon EventBridge | Event Bus", Amazon Web Services, 2021. [online] Available: <https://aws.amazon.com/eventbridge/> [Accessed 4 November 2021].
- [83] Amazon Web Services, "What is Amazon CloudWatch Logs?", Amazon Web Services, 2021. [online] Available: <https://docs.aws.amazon.com/AmazonCloudWatch/latest/logs/WhatIsCloudWatchLogs.html> [Accessed 4 November 2021].
- [84] K. Brady, S. Moon, T. Nguyen, and J. Coffman, "Docker Container Security in Cloud Computing", 10th Annual Computing and Communication Workshop and Conference (CCWC), 2020.
- [85] U. Isikdag, "Design patterns for BIM-based service-oriented architectures", Automation in Construction, vol. 25, pp. 59-71, 2012. doi: 10.1016/j.autcon.2012.04.01
- [86] Amazon Web Services, "What is Amazon S3? - Amazon Simple Storage Service", Amazon Web Services, 2021. [online] Available: <https://docs.aws.amazon.com/AmazonS3/latest/userguide/Welcome.html> [Accessed 5 November 2021].
- [87] Amazon Web Services, "DynamoDB", Amazon Web Services, 2021. [online] Available at: <https://docs.aws.amazon.com/dynamodb/> [Accessed 5 November 2021].
- [88] R .E. Krider, "A Process Model for Data Mining—CRISP-DM", Customer and Business Analytics, pp. 43–56, 2012. ISBN: 9780429185687
- [89] W. Ayele, "Adapting CRISP-DM for Idea Mining", International Journal of Advanced Computer Science and Applications, vol. 11, no. 6, 2020. doi: 10.14569/ijacsa.2020.0110603
- [90] E. Beller et al., "Making progress with the automation of systematic reviews: principles of the International Collaboration for the Automation of Systematic Reviews (ICASR)", Systematic Reviews, vol. 7, no. 1, 2018. doi: 10.1186/s13643-018-0740-7
- [91] S. van der Mierden, "Software tools for literature screening in systematic reviews in biomedical research", ALTEX, vol. 36, no. 3, pp. 508-517, 2019. doi: 10.14573/altex.1902131
- [92] W. Wu, K. Akers, E. Hu, A. Sarkozy, P. Vinson, "Digital Tools for Managing Different Steps of the Systematic Review Process", Library Scholarly Publications. 2018, pp. 136. Available: <https://digitalcommons.wayne.edu/libsp/136/> [Accessed 4 November 2021].
- [93] J. Eden, L. Levit, A. Berg and S. Morton, Finding what works in health care. Washington D.C.: National Academies Press (US), 2011. ISBN: 978-0-309-16425-2

- [94] A. Tsou, J. Treadwell, E. Erinoff and K. Schoelles, "Machine learning for screening prioritization in systematic reviews: comparative performance of Abstrackr and EPPI-Reviewer", *Systematic Reviews*, vol. 9, no. 1, 2020. doi: 10.1186/s13643-020-01324-7
- [95] Y. Jin, X. Ren, L. Yu, Y. Wang, Q. Huang, X. Li, X. Ren, J. Lou, K. Gao, M. Chen, W. Hu, X. Zeng, H. Shang, "Artificial intelligence for the development and implementation guidelines for traditional Chinese medicine and integrated traditional Chinese and western medicine", *TMR Modern Herbal Medicine*, vol. 4, no. 2, 2021. Available: <https://www.tmrjournals.cn/public/articlePDF/20210501/9f420c661daf2a3484a0aef0f501aa2c.pdf> [Accessed 5 August 2021].
- [96] A. Scott, C. Forbes, J. Clark, M. Carter, P. Glasziou and Z. Munn, "Systematic review automation tool use by systematic reviewers, health technology assessors and clinical guideline developers: tools used, abandoned, and desired", 2021. doi: 10.1101/2021.04.26.21255833
- [97] W. Scott, "TF-IDF for Document Ranking from scratch in python on real world dataset.", Medium, 2021. [Online]. Available: <https://towardsdatascience.com/tf-idf-for-document-ranking-from-scratch-in-python-on-real-world-dataset-796d339a4089> [Accessed: 17- November- 2021].
- [98] W. Scott, "Document Retrieval using TF-IDF Weighted Rank and TF-IDF Cosine Similarity", GitHub, 2021. [online]. Available: <https://github.com/williamscott701/Information-Retrieval/blob/master/2.%20TF-IDF%20Ranking%20-%20Cosine%20Similarity%2C%20Matching%20Score/TF-IDF.ipynb> [Accessed 17 November 2021].
- [99] G. Bedi, "Simple guide to Text Classification(NLP) using SVM and Naive Bayes with Python", Medium, 2021. [online]. Available: <https://medium.com/@bedigunjit/simple-guide-to-text-classification-nlp-using-svm-and-naive-bayes-with-python-421db3a72d34> [Accessed 17 November 2021].
- [100] E. Hudlicka, "Requirements elicitation with indirect knowledge elicitation techniques: comparison of three methods", *Proceedings of the Second International Conference on Requirements Engineering*, 1996, pp 4-11. doi: 10.1109/icre.1996.491424
- [101] M. Falagas, E. Pitsouni, G. Malietzis and G. Pappas, "Comparison of PubMed, Scopus, Web of Science, and Google Scholar: strengths and weaknesses", *The FASEB Journal*, vol. 22, no. 2, pp. 338-342, 2007. doi: 10.1096/fj.07-9492lsf
- [102] BIREME, 2019, "GitHub - bireme/PICO-BackEnd: Search BVS via PICO (Patient, Intervention, Comparison, Outcome)", GitHub, 2021 [online]. Available at: <https://github.com/bireme/PICO-BackEnd> [Accessed 30 November 2021].
- [103] M. Maaz, "Viability of machine learning to reduce workload in systematic review screenings in the health sciences: a working paper", arXiv.org, 2021. [Online]. Available: <https://arxiv.org/abs/1908.08610>. [Accessed: 03- Dec- 2021].
- [104] X. Li, R. Al-Zaidy, A. Zhang, S. Baral, L. Bao and C. Giles, "Automating Document Classification with Distant Supervision to Increase the Efficiency of Systematic Reviews", arXiv.org, 2021. [Online]. Available: <https://arxiv.org/abs/2012.07565>. [Accessed: 03- Dec- 2021].
- [105] A. Gates, C. Johnson and L. Hartling, "Technology-assisted title and abstract screening for systematic reviews: a retrospective evaluation of the Abstrackr machine learning tool", *Systematic Reviews*, vol. 7, no. 1, 2018. doi: 10.1186/s13643-018-0707-8
- [106] R. van de Schoot et al., "An open source machine learning framework for efficient and transparent systematic reviews", *Nature Machine Intelligence*, vol. 3, no. 2, pp. 125-133, 2021. doi: 10.1038/s42256-020-00287-7

- [107] P. Przybyła et al., "Prioritising references for systematic reviews with RobotAnalyst: A user study", *Research Synthesis Methods*, vol. 9, no. 3, pp. 470-488, 2018. doi: 10.1002/jrsm.1311

## 10. ANEXOS

### Anexo 1

#### Herramientas que abordan menos de 3 fases

207 herramientas de las 245 abordan menos de 4 fases de la elaboración de una revisión sistemática. La mayoría de ellas son herramientas multidisciplinarias que permiten realizar la automatización de tareas genéricas, que no hacen parte exclusivamente del proceso de elaboración de una revisión sistemática. Estas herramientas son presentadas a continuación:

**Tabla A1.** Herramientas de automatización que abarcan menos de 3 fases del ciclo de elaboración de revisiones sistemáticas

Fases abordadas	Herramientas
3	Citavi, Colandr, COVID-NMA, Docear, Doctor Evidence, EROS: Early Review Organizing System, EvidenceSET, Graph Commons, JSTOR Text Analyser, Medline (PubMed) trend, MontyLingua, NodeXL, ORBIT (Outcome Reporting Bias in Trials) Matrix Generator, PaperBot, PEx, Publish or Perish, Researchr, Syras, Sysrev, Table Builder, Textpresso, The albatross plot, Trip Rapid Reviews, Wikindx
2	A2A (Apples to Apples), Abtrackr, AlvisIR Food semantic search engine, AntConc, Automatic VArant evidence DAtabase (AVADA), BioReader (Biomedical Research Article Distiller), Carrot2, CIneMA, colwiz, Comparative Toxicogenomics Database (CTD), Concept Encoder, Coremine Medical, CrowdCARE, EndNote, Epistemonikos, EviAtlas, Forest Plot Generator, Forester, G-Bean, GAPScreeener, GATE, Gephi, GoPubMed, GRADEpro, GSscraper, Import.io, Litmaps, Mapping MEDLINE, MarkerDB, Medline Transpose, Mendeley, Menoci, Meta-Essentials, metaDigitise, NAILS (Network Analysis Interface for Literature Studies), NetMetaXL, Network explorer, NLM Medical Text Indexer (MTI), OpenMEE, Papers, ParsCit, PEx-Graph, Pimiento, PRISMA2020 flow diagrams, PubMed PICO Tool, PubMed2XL, PubReMiner, PubVenn, Quetzal, RCT Tagger, REviewER, RevManHAL, revtools, RobotReviewer, robvis, ROSES flowchart, SENT, Site Content Analyzer, SLRONT, Spá, Spotfire-DXP, SWIFT-Active Screener, SWIFT-Review, Systematic Review Assistant - a prototype, topictagger, VOSViewer, Web-based AMSTAR Checklist, Wizfolio, Yale MeSH Analyzer, yEd Graph Editor, Zotero

- 
- 1** 3D Covariate Visualiser, Aigaion, AND - Author Name Disambiguation, Article-based PubMed Search Engine (APSE), Bebop, BeCAS, BERT, BEST (Biomedical Entity Search Tool), BibDesk, Bibus, BioBERT, CERC, Cientopolis Scolr, Citation and Text-Based Framework for Retrieving Publications for Literature Reviews, citationchaser, CiteULike, CitNetExplorer, CLAMP (Clinical Language Annotation, Modeling, and Processing), Comprehensive Meta-Analysis (CMA), Confidence Interval Calculator, cTAKES (clinical Text Analysis and Knowledge Extraction System), Cytoscape, Data Abstraction Assistant (DAA), DBPedia, Diagrams.net, Disputatron, Do-Search, DoCTER, Drug Herb Interaction Query Website, Drug-Gene Interaction Database (DGIdb), DTA MA (Diagnostic Test Accuracy Meta-Analysis), Engauge Digitizer, Eppi-Mapper, Evidence Base, EXACT: EXtracting Accurate efficacy and safety information from ClinicalTrials.gov, FastText, Fiddle: File Drawer Liberation Effort, FLC 3.0 critical appraisal tools, Graph2Data, GRIM (granularity-related inconsistency of means) test calculator, HCE, HelioBLAST, Inciteful.xyz, JabRef, Journal/Author Name Estimator (Jane), Kaggle, KBibTeX, Leaf, litsearchr, Manuscript Matcher, MAVIS: Meta-Analysis via Shint, MedTerm Search Assist, MeSH on Demand, Meta-Analysis Toolbox, Meta-CART, MetaEasy, metafor, MetaGenyo, MetaInsight, MetaLight, MetaMap, metamisc, MetaWin, MetaXL, Metta, MIX 2.0, NetConfer, NimbleMiner, nineteen, OmixLitMiner, OpenMetaAnalyst, OutcomesApp, Paperpile, Plot Digitizer, Polyglot Search, Practical Meta-Analysis Effect Size Calculator, PRISMA Flow Diagram Generator, Pybliographer, Pymeta, PythonMeta, R.ROSETTA: an interpretable machine learning framework, rebase, RefDB, Reference Manager, Referencer, RefWorks, ReVis, RIS Export, RobotAnalyst, RobotSearch, Sample Size Search (SSS) Tool for PubMed, Science of Science (Sci2) Tool, SciRef, Scolr (Support for Collaborative, Open, Literature Reviews), Screenatron, Search Builder 1.0, Search Whiteboard, Sebzer, SEED, SensPrecOptimizer, Sente, Systematic Review Data Repository (SRDR), TextBlob: Simplified Text Processing, UNITEX, Voyant Tools, WebPlotDigitizer, Weka, word2vec, WordStat 7, xmeta
- 

- 0** PDQ-Evidence, Twister
- 

Sintetizado y procesado a partir de la información encontrada en [46]



A continuación, se presentan los requerimientos funcionales identificados desde la perspectiva de las características más deseadas por los usuarios. Para cada uno de los estudios comparativos se identificó que puntaje de valor se le asignaba al requerimiento. Aquellas características o funcionalidades que se mencionan de forma distinta pero similar por los distintos autores se agruparon dentro de una misma categoría.

Componente	Requerimiento funcional	Relevancia según autores									Total
		[68]	[90]	[91]	[68]	[92]	[93]	[94]	[95]	[96]	
Screening	Permite la selección de títulos y resúmenes.	0	1		1			1		3	
	Permite importar texto completo.	1			1				1	3	
	Categorizar referencias adicionales.	1.5		1						2.5	
	Análisis de texto.	0							1	1	
	Revisión ciega a las decisiones de otros.	1.5					1			2.5	
	Resolución de conflictos.	1.5				1				2.5	
	Herramienta de clasificación (Machine Learning)	0.5								0.5	
	Enfoque en Sensibilidad y especificidad de la búsqueda.	0						1		1	
	Criterios de inclusión/exclusión.	0						1		1	
Soporte de revisión sistemática	Permite la de-duplicación.	1	1		1	1			1	5	
	Configuración de la revisión: Facilita interacción de las partes interesadas.	0	1							1	
	Desarrollo de protocolo. Permite estructurar búsqueda PICO	0	1							1	
	Permite la extracción de datos.	0.5			1	1		1	1	4.5	
	Permite etiquetado de datos para metaanálisis	0		0.5						0.5	
	Apoya otras etapas de la revisión.	0.5								0.5	
	Validación de calidad.	0			1	1		1		3	
Gestión de referencias	Importación de referencias.	0					1			1	
	Exportación de referencias.	2								2	
	Aleatorización de referencias.	1		1						2	
	Exportación de decisiones.	2								2	

	Importación de PDFs.	1		1		2
	Puede filtrar citas por categoría.	1				1
Resultados	Exportar resultados.	0		2		2
	Diagrama de flujo PRISMA.	0		0.5		0.5
	Genera figuras y/o tablas para la redacción del documento.	0	1	1	1	3
	Generación o síntesis del reporte o de resultados.	0	1	1	1	5
Búsqueda	Generación de la bibliografía.	0		1		1
	Permite búsqueda de referencias (Ideal: Integrado con otras DB).	1	1		1	3
	Traducción de búsqueda entre bases de datos.	0			1	1
	Literatura Gris.	0			1	1
	Búsqueda de sinónimos y tesauros.	0			1	1
	Resaltado de palabras clave (o similar).	1		1		2
	Buscador tipo Snowball.	0		1		1
Meta-análisis	Meta-análisis: Cálculos bioestadísticos.	0		1		1
	Meta-análisis: Valoración crítica: Evaluaciones de riesgo de sesgo.	0	1	1	1	3
Gestión de proceso	Soporte para múltiples usuarios.	2		1	1	6
	Soporte para múltiples proyectos.	1				1
	Elección de pantalla simple o doble antes del screening.	1.5				1.5
	Restaurabilidad de trabajo.	1.5				1.5
	Gestión de roles.	1		1		2
Otros	No requiere pago para su uso	1.5		1	1	4.5
	Soporta distintos tipos de caracteres	0		1		1
Facilidades	Hay una aplicación para móvil	1				1
	Integración con otras herramientas.	0		1		2
	Flujo en una sola herramienta. No se requirieren otras.	0			1	1
	La herramienta tiene requisitos de sistema sencillos.	1.5				1.5

	No se requiere que el usuario escriba código	1.5		1.5
	Experiencia de usuario.	0	1	1
	Corto tiempo de uso y aprendizaje.	0	1	1
Seguridad	Sitio web es seguro.	1.5		1.5
Soporte técnico	Ayuda y soporte técnico.	0	1	1
	Soporta trabajo offline.	0		0
	Existe una guía de instalación (Si corresponde).	1		1
	Existe una sección de ayuda.	1		1
Flujo de trabajo	La herramienta es flexible para realizar diferentes flujos de trabajo.	1.5	1	2.5
	Configuración de usuario rápida	1		1
	El progreso se retroalimenta al usuario.	1.5		1.5
	Monitoreo del estado del proyecto.	0	1	1

Sintetizado y procesado a partir de la información encontrada en [63, 68, 90, 91, 92, 93, 94, 95, 96].

### Anexo 3: Todos los requerimientos funcionales

A continuación, se presentan los requerimientos funcionales desde la perspectiva de las características requeridas por el usuario. Esta tabla incluye todos los requerimientos, incluso aquellos que no aplican para la versión 1.0

**Tabla A3.** Requerimientos funcionales desde la perspectiva de las características deseadas por los usuarios.

Requerimiento funcional	Descripción	Relevancia
RF-F-SCR-1	Screening: Permite la selección de títulos y resúmenes	3
RF-F-SCR-2	Screening. Permite la detección de texto completo	3
RF-F-SCR-3	Screening: Permite categorizar más referencias	2.5
RF-F-SCR-4	Screening: Análisis de texto	1
RF-F-SCR-5	Screening: Vista ciega a las decisiones de otros.	2.5
RF-F-SCR-6	Screening: Resolución de conflictos	2.5
RF-F-SCR-7	Screening: Enfoque en Sensibilidad y especificidad de la búsqueda	1
RF-F-SCR-8	Screening: Criterios de inclusión/exclusión	1
RF-F-SR-1	Soporte de revisión sistemática: Permite la de duplicación	5
RF-F-SR-2	Soporte de revisión sistemática: Configuración de la revisión, permite participación e interacción entre las partes interesadas	1
RF-F-SR-3	Soporte de revisión sistemática: Permite construir estrategia de búsqueda PICO	1
RF-F-SR-4	Soporte de revisión sistemática: Permite la extracción de datos	4.5
RF-F-SR-5	Soporte de revisión sistemática: Validación calidad	3
RF-F-REF-1	Referencias: Importación de referencias	1
RF-F-REF-2	Referencias: Exportación de referencias	2
RF-F-REF-3	Referencias: Aleatorización de referencias	2
RF-F-REF-4	Referencias: Exportación de decisiones	2
RF-F-REF-5	Referencias: Importación de Pdfs.	2
RF-F-REF-6	Referencias: Filtrar citas según categorías	1
RF-F-RES-1	Resultados: Exportar resultados	2

RF-F-RES-2	Resultados: Generación de figuras o tablas para facilitar la redacción del informe.	3
RF-F-RES-3	Resultados: Generación o síntesis del reporte o de resultados	5
RF-F-RES-4	Resultados: Generación de la bibliografía	1
RF-F-BUS-1	Búsqueda: Puede buscar referencias (Ideal: Integrado con otras DB)	3
RF-F-BUS-2	Búsqueda: Traducción de estrategia de búsqueda entre bases de datos	1
RF-F-BUS-3	Búsqueda: Literatura Gris	1
RF-F-BUS-4	Búsqueda: Búsqueda de sinónimos y tesauros	1
RF-F-BUS-5	Búsqueda: Resaltado de palabras clave	2
RF-F-BUS-6	Búsqueda: Buscador tipo Snowball	1
RF-F-MET-1	Metaanálisis: Cálculos bioestadísticos	1
RF-F-MET-2	Metaanálisis: Evaluaciones de riesgo de sesgo	3
RF-F-GES-1	Gestión: Soporte para múltiples usuarios: CRUD Usuarios y equipos	6
RF-F-GES-2	Gestión: Soporte para múltiples proyectos: Usuario puede crear proyectos, editarlos	1
RF-F-GES-3	Gestión: Elección de pantalla simple o doble antes de la progresión	1.5
RF-F-GES-4	Gestión: Restaurar estado de trabajo	1.5
RF-F-GES-5	Gestión: Gestión de roles	2

Sintetizado y procesado a partir de la información encontrada en [63, 68, 90, 91, 92, 93, 94, 95, 96].

A continuación, se presentan la tabla completa de requerimientos funcionales desde la perspectiva de los marcos metodológicos. Esta tabla incluye todos los requerimientos, incluso aquellos que no aplican para la versión 1.0.

**Tabla A4.** Requerimientos funcionales desde la perspectiva de las características deseadas para satisfacer los marcos metodológicos

Requerimiento funcional	Descripción
RF-PR-TI	Título: Mostrar en el manuscrito final si es revisión sistemática y/o metaanálisis
RF-PR-AB	Abstract: El abstract debe estar adecuadamente estructurado
RF-PR-IN-1	Introducción: Se debe incluir la justificación de la revisión
RF-PR-IN-2	Introducción: Mostrar Estrategia PICOS
RF-PR-ME-1	Metodología: Mostrar URL y meta-data de protocolo de revisión
RF-PR-ME-2	Metodología: Se deben mostrar características del estudio, criterios de elegibilidad y validar la calidad de los criterios de elegibilidad

RF-PR-ME-3	Metodología: Se deben describir de bases de datos usadas, fuentes de información y fecha de búsquedas. Se debe explicar la estrategia de búsqueda utilizada. Se deben utilizar estrategias de búsqueda optimizadas y realizar una búsqueda exhaustiva
RF-PR-ME-4	Metodología: Se debe describir el proceso de selección de estudios. Debe incluir más de un revisor. Se deben incluir restricciones de fecha, idioma y formato. Se deben recopilar todos los estudios relevantes para la temática. Deben existir esfuerzos para minimizar el sesgo en selección
RF-PR-ME-5	Metodología: Se debe describir la extracción de datos, la cual debe ser realizada por más de 1 revisor. Métodos para confirmar la información
RF-PR-ME-6	Metodología: Describir variables de interés utilizadas para comparar los estudios
RF-PR-ME-7	Metodología: Describir métodos de evaluación de riesgo de sesgo en cada estudio y en la síntesis
RF-PR-ME-8	Metodología: Describir estadísticos de resumen utilizados
RF-PR-ME-9	Metodología: Describir métodos utilizados para la combinación estadística y comparar manuscritos
RF-PR-ME-10	Metodología: Describir metodologías de evaluación de riesgo de sesgo
RF-PR-ME-11	Metodología: Describir métodos adicionales para verificar calidad de la evidencia
RF-PR-RE-1	Resultados: Incluir diagrama de exclusión PRISMA, razones de exclusión e inclusión y número de estudios excluidos en cada etapa
RF-PR-RE-2	Resultados: Características para las que se extrajeron los datos en cada estudio
RF-PR-RE-3	Resultados: Presentar riesgo de sesgo en cada estudio, evaluación del punto 12
RF-PR-RE-4	Resultados: Datos estadísticos, estimación de efecto y diagrama de bosque para riesgos y beneficios encontrados
RF-PR-RE-5	Resultados: Presentar resultados principales. Si es metaanálisis intervalos de confianza y medidas de consistencia
RF-PR-RE-6	Resultados: Presentar resultados de evaluación de riesgo. Ítem 15
RF-PR-RE-7	Resultados: Presentar otros resultados incluyendo información de la solidez de la evidencia
RF-PR-DI	Discusión Resumir principales hallazgos, solidez de la evidencia. Limitaciones a nivel de estudio/resultado y a nivel de revisión. Interpretación general de la evidencia e implicaciones para futuras investigaciones
RF-PR-OT-1	Otros: Incluir información de registro y protocolo
RF-PR-OT-2	Otros: Describir fuentes de financiación
RF-PR-OT-3	Otros: Describir conflictos de intereses
RF-PR-OT-4	Otros: Describir códigos y otro material disponible

Sintetizado y procesado a partir de la información encontrada en [21, 22, 23, 24, 27, 28, 29, 36, 43].

#### Anexo 4: Requerimientos funcionales y no funcionales para la versión 2.0 de la aplicación

En esta sección se presentan los requerimientos funcionales y no funcionales que se podrían incluir en una próxima versión, para garantizar cumplir todos los requerimientos por parte de los marcos metodológicos y por parte de las necesidades del usuario.

##### Requerimientos funcionales para la versión 2.0 de la aplicación, de las características deseadas por el usuario

A continuación, se presentan los requerimientos funcionales desde la perspectiva de usuario que serán incluidos en una próxima versión. Al implementarlos, se puede garantizar que la aplicación va a cubrir todas las necesidades más importantes de los investigadores biomédicos.

Algunas de ellas, como la resolución de conflictos y la búsqueda de Tesauros, no se pueden incluir en esta versión ya que requieren una mayor madurez de los componentes y de la arquitectura en general. Otras, como la validación de calidad y la generación de bibliografía, son características secundarias que no son indispensables para el funcionamiento de la aplicación, y aunque son deseadas, pueden ser implementadas en la siguiente iteración.

**Tabla A5.** Requerimientos funcionales deseados por los usuarios para una futura versión.

Id	Requerimiento funcional	Relevancia	Comentarios
RF-F-SCR-6	Screening: Resolución de conflictos.	2.5	Este módulo facilita la clasificación de los manuscritos en incluidos y excluidos. Dos usuarios y un tercero para desempatar podrán clasificar los artículos.
RF-F-SR-5	Soporte de revisión sistemática: Validación de calidad.	3	Eso es básicamente comprobar que se cumplieron los criterios PRISMA.
RF-F-REF-1	Referencias: Importación de referencias.	1	Es uno de los elementos que más puede facilitar el flujo de información, pero requiere una mayor madurez de la arquitectura para entender en qué punto del flujo de trabajo se pueden importar y en qué formatos, siguiendo la legislación de propiedad intelectual de los datos.
RF-F-REF-6	Referencias: Puede filtrar citas por categoría.	1	Esto fácilmente se puede hacer desde Excel, pero vendría bien incluirlo.
RF-F-RES-1	Resultados: Exportar resultados.	2	Estos elementos hacen referencia al módulo de generación de resultados, el cual es el paso posterior a la clasificación de manuscritos, y que todavía no está cubierto en los objetivos de negocio de la arquitectura actual
RF-F-RES-2	Resultados: Generación de figuras o tablas para ayudar con la redacción del documento.	3	

RF-F-RES-3	Resultados: Generación o síntesis del reporte	5	
RF-F-RES-4	Resultados: Generación de la bibliografía.	1	
RF-F-BUS-2	Búsqueda: Traducción de búsqueda entre bases de datos.	1	Uno de los elementos de mayor utilidad, pero requiere explorar en más detalle en qué momento del flujo de información se puede realizar esto.
RF-F-BUS-3	Búsqueda: Literatura Gris.	1	Está fuera del alcance por el momento ya que es una tarea muy manual y operativa, no es sencillo integrar esta información.
RF-F-BUS-4	Búsqueda: Búsqueda de sinónimos y tesauros.	1	Es otro de los elementos de mayor utilidad, pero requiere integraciones complejas, y construir clases siguiendo el patrón adaptador con un alto nivel de dificultad, ya que al ser una tarea con tanta carga debe estar diseñada hacia el performance.
RF-F-BUS-6	Búsqueda: Buscador tipo snowball.	1	Es una característica muy útil, pero requiere un poco de trabajo manual.
RF-F-MET-1	Metaanálisis: Cálculos bioestadísticos.	1	Estos elementos están por el momento fuera del objetivo de negocio de la aplicación, ya que corresponden a un caso de uso muy específico.
RF-F-MET-2	Metaanálisis: Valoración crítica de riesgo de sesgo.	3	
RF-F-GES-1	Gestión: Soporte para múltiples usuarios: CRUD Proyectos y equipos de trabajo	6	Requiere una mayor madurez de los módulos de usuario y proyectos, la cual excede los objetivos de negocio de la iteración actual de la aplicación.

Sintetizado y procesado a partir de la información encontrada en [21, 22, 23, 24, 27, 28, 29, 36, 43]



## Anexo 5. Requerimientos funcionales para la versión 2.0 de la aplicación, desde la perspectiva de los marcos metodológicos

A continuación, se presentan los requerimientos funcionales desde la perspectiva de los marcos metodológicos, que serán incluidos en una próxima versión. Al implementarlos, se puede garantizar que la aplicación va a generar revisiones sistemáticas de muy alta calidad, al contar con los elementos que son solicitados por los checklist de alta calidad. La mayoría de estos requerimientos hacen referencia a la construcción del documento al finalizar la elaboración de la revisión sistemática.

**Tabla A6.** Requerimientos funcionales requeridos por los marcos metodológicos, que serán abordados en una próxima versión.

Requerimiento funcional	Descripción
RF-PR-TI	Título: Mostrar en el manuscrito final si es revisión sistemática y/o metaanálisis.
RF-PR-AB	Abstract: El abstract debe tener una estructura adecuada.
RF-PR-IN-1	Introducción: Se debe incluir la justificación de la revisión.
RF-PR-ME-1	Metodología: Mostrar URL y meta-data de protocolo de revisión.
RF-PR-ME-5	Metodología: Se debe describir la extracción de datos, la cual debe ser realizada por más de 1 revisor.
RF-PR-ME-6	Metodología: Describir variables de interés utilizadas para comparar los estudios.
RF-PR-ME-7	Metodología: Describir métodos de evaluación de riesgo de sesgo en cada estudio y en la síntesis.
RF-PR-ME-8	Metodología: Describir estadísticos de resumen utilizados.
RF-PR-ME-9	Metodología: Describir métodos utilizados para la combinación estadística y comparar manuscritos.
RF-PR-ME-10	Metodología: Describir metodologías de evaluación de riesgo de sesgo.
RF-PR-ME-11	Metodología: Describir métodos adicionales para verificar la calidad de la evidencia.
RF-PR-RE-2	Resultados: Características para las que se extrajeron los datos en cada estudio.
RF-PR-RE-3	Resultados: Presentar riesgo de sesgo en cada estudio, evaluación del punto 12.
RF-PR-RE-4	Resultados: Datos estadísticos, estimación de efecto y diagrama de bosque para riesgos y beneficios encontrados.
RF-PR-RE-5	Resultados: Presentar resultados principales. Si es metaanálisis intervalos de confianza y medidas de consistencia.
RF-PR-RE-6	Resultados: Presentar resultados de evaluación de riesgo. Ítem 15
RF-PR-RE-7	Resultados: Presentar otros resultados incluyendo información de la solidez de la evidencia.
RF-PR-DI	Discusión Resumir principales hallazgos, incluyendo solidez de la evidencia. Limitaciones a nivel de estudio/resultado y a nivel de revisión. Interpretación general de la evidencia e implicaciones para futuras investigaciones.
RF-PR-OT-1	Otros: Incluir información de registro y protocolo.
RF-PR-OT-2	Otros: Describir fuentes de financiación.
RF-PR-OT-3	Otros: Describir conflictos de intereses.
RF-PR-OT-4	Otros: Describir códigos y otro material disponible.

Sintetizado y procesado a partir de la información encontrada en [21, 22, 23, 24, 27, 28, 29, 36, 43]