



UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

Desarrollo de herramienta de Revisión sistemática

PROPUESTA DE TEMA DE MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL EN COMPUTACIÓN

Andrés Basáez Valdivieso

MODALIDAD:
Memoria

PROFESOR GUÍA:
Ivana Bachmann

CO-GUÍA:
Jocelyn Simmonds

SANTIAGO DE CHILE
2024

1. Introducción

El presente documento de propuesta de Trabajo de Título se basa en un proyecto de la Profesora Ivana Bachmann del Departamento de Ciencias de la Computación de la Universidad de Chile [1]. Este consiste en una aplicación web orientada a investigadores, que sirva de apoyo para organizar y llevar a cabo revisiones sistemáticas de literatura científica.

Una revisión sistemática es un proceso riguroso y reproducible de recopilación, evaluación y síntesis del conocimiento y los trabajos de investigación respecto a un tema específico y bien definido. Esto permite generar un entendimiento comprensivo e integral sobre el estado actual de la investigación y la literatura respecto del área particular [16]. Además, permite identificar brechas en el estado del arte, para que puedan sugerir nuevos focos donde realizar investigación. A grandes rasgos, el proceso de revisión sistemática cuenta con tres fases principales [12]:

- Planificación de la revisión
 1. Identificación de la necesidad de una revisión.
 2. Desarrollo de un protocolo de revisión.
- Realización de la revisión
 1. Identificación y selección de estudios.
 2. Evaluación de la calidad de los estudios.
 3. Extracción y monitoreo de datos
 4. Síntesis de datos.
- Reporte de la revisión.

El flujo de trabajo convencional, sin herramientas especializadas, generalmente se hace utilizando planillas de cálculo. Esta herramienta no es ideal para el trabajo en cuestión. La estructura de las hojas de cálculo no facilita la organización, búsqueda y evaluación de los elementos de la revisión, ni cuenta con un sistema que permita clasificarlos de manera eficiente para extraer datos y tendencias o generar gráficos relevantes. La naturaleza manual de esta forma de realizar la revisión la hace lenta, propensa a errores, y queda sujeta a la capacidad individual de los investigadores de organizar la planilla. Además, la unificación de las planillas de múltiples investigadores en el equipo de revisión puede ser inconsistente sin el uso de un sistema de etiquetado más estandarizado, lo que dificulta la colaboración.

El presente proyecto nace de la necesidad de una herramienta de código abierto con un enfoque en el análisis de la biblioteca de artículos revisados durante el proceso, y la extracción y síntesis de estos datos, tal que investigadores puedan clasificar los artículos con etiquetas personalizadas, agruparlos en categorías que les parezcan relevantes durante su lectura y a partir de esto extraer datos relevantes sobre su proceso de revisión sistemática de la literatura.

Las soluciones existentes actualmente requieren de pagos de suscripciones, o no están enfocadas en este ámbito específico de la investigación, como por ejemplo JabRef [17] o Mendeley [10], por lo cual son útiles como herramientas complementarias, pero no cuentan con todo lo necesario para llevar a cabo una revisión sistemática. Esta es la motivación principal de este proyecto, y la razón por la que se busca resolver los problemas presentes y expandir sobre esta plataforma, para ofrecer una solución accesible e integral para investigadores.

Actualmente, la aplicación desarrollada para el proyecto de la profesora Bachmann no cumple con todas las funcionalidades requeridas para satisfacer las necesidades de sus usuarios, ya sea porque no se encuentran implementadas, o se encuentran implementadas de tal manera que su uso es poco intuitivo o presentan errores no resueltos que impiden su buen funcionamiento. Además, las versiones existentes son inestables e incompletas, y son difíciles de instalar y correr, por esto, para efectos prácticos, no hay una versión funcional de la herramienta.

2. Situación Actual

El proyecto comenzó en el año 2016 por la profesora Ivana Bachmann, motivado por la necesidad de una herramienta especializada en apoyar la realización de revisiones sistemáticas, surgida durante el curso de sus estudios de postgrado. Desde su inicio, el proyecto ha pasado por varias iteraciones y ha sido desarrollado con el apoyo de dos estudiantes del DCC que han realizado su práctica en *NIC Chile Research Labs*. A pesar de este trabajo, la herramienta no se encuentra en un estado funcional. Hay funcionalidades críticas que no están implementadas, hace uso de versiones de librerías y lenguajes que se encuentran deprecadas, y su uso en general es confuso para usuarios que no estén familiarizados con el proyecto. La versión actual está desarrollada utilizando Flask y JavaScript, usando MySQL para la base de datos. A continuación se presentará a grandes rasgos el estado de la última iteración de este proyecto, y los problemas que presenta.

2.1. Funcionalidades actuales

En primer lugar, se comentarán las funcionalidades actualmente presentes en el proyecto. Existe un formulario para agregar artículos que permite registrar metadatos de artículos por parte de los usuarios, además, está la opción de añadir categorías y subcategorías dentro de estas al momento de agregar un artículo. Existe una vista donde pueden crearse categorías y metacategorías. Por último, se puede encontrar un buscador de artículos por nombre, este buscador solo reconoce coincidencias exactas con el título.

2.2. Problemas de experiencia de usuario

La experiencia de usuario presenta varios problemas que se detallan a continuación: El buscador principal solo funciona si el input del usuario coincide exactamente con los títulos disponibles, lo que dificulta la búsqueda cuando el usuario no recuerda el nombre preciso. Además, hay múltiples vistas con un propósito poco claro, y ciertas funciones, como la creación de categorías, requieren navegar entre varias ventanas y realizar backtracking para agregarlas a los artículos. El diseño de varios elementos resulta confuso al no incluir tooltips ni instrucciones claras, obligando a los usuarios a interactuar con los botones para entender su función. Por otra parte, algunos botones no realizan ninguna acción cuando no se cumplen ciertas precondiciones, sin informar al usuario sobre ello, un ejemplo de esto puede verse en la Figura 1.

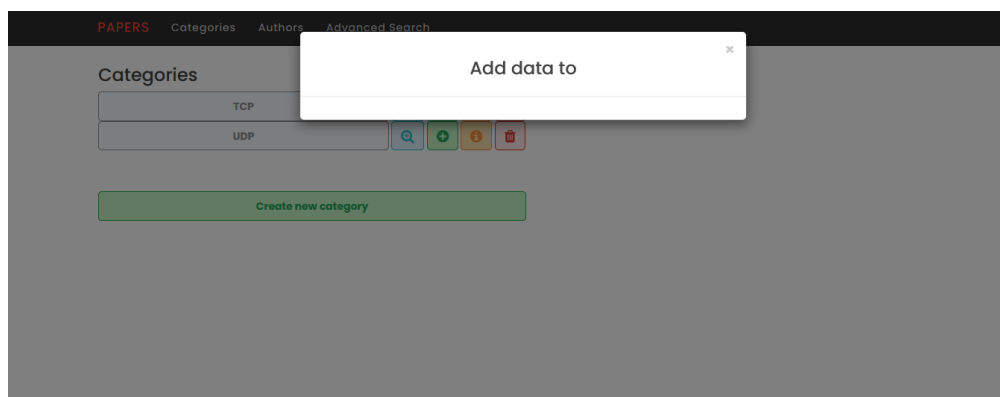


Figura 1: Ejemplo de un modal que se mantiene vacío si no se presiona otro botón no relacionado previamente.

2.3. Problemas de funcionalidad

El uso de categorías, clases y metacategorías es complicado y actualmente no ofrece ningún beneficio práctico, ya que deberían servir como criterios de búsqueda en la búsqueda avanzada, la cual no funciona correctamente dado que presenta errores y hace caer la plataforma. Si se asigna una clase a un artículo dentro de una categoría, esta no puede ser removida, solo reemplazada por otra clase dentro de la misma categoría.

Si bien se ha avanzado considerablemente desde sus orígenes, esta herramienta está incompleta, no se encuentran implementados todos los features necesarios para su funcionamiento, presenta bugs y errores que impiden su uso normal, y en general el flujo del usuario al interactuar con la plataforma es confuso y poco intuitivo, generando una mala experiencia y afectando negativamente su usabilidad. Por lo tanto, esta versión del proyecto requiere mejoras substanciales para que pueda cumplir con los objetivos del proyecto original.

3. Objetivos

Objetivo General

Crear una versión funcional de la herramienta, basándose en el proyecto actual, implementando nuevas funcionalidades y mejorando significativamente la usabilidad y experiencia de usuario, de tal manera que se facilite el proceso de revisión de literatura científica y la recopilación de datos para investigadores.

Objetivos Específicos

- Analizar la plataforma actual e identificar oportunidades de mejora.
- Diseñar una nueva versión de la herramienta con todas las funcionalidades mínimas mencionadas en la sección anterior.
- Implementar la plataforma diseñada, con un enfoque en mejorar la experiencia de usuario. Hacer que el uso de la herramienta sea más claro e intuitivo.
- Validar la usabilidad y funcionalidad de la plataforma.

Evaluación

Se determinará si la herramienta cumple con las funcionalidades necesarias verificando el flujo de trabajo utilizando artículos de prueba. Esto incluye subida de un artículo, el llenado de formulario de metadatos, el etiquetado de clases y categorías, una búsqueda exitosa en la plataforma, y la recolección de datos.

Se evaluará la usabilidad y utilidad percibida de la plataforma mediante el uso de encuestas, como la System Usability Scale [5]. Se realizarán pruebas de la solución con investigadores, el usuario objetivo de esta herramienta. En particular se espera trabajar con miembros de NIC labs, y con al menos cinco usuarios [14].

4. Solución Propuesta

4.1. Reimplementación de funcionalidades existentes

Como se ha discutido en las secciones anteriores, la solución propuesta consiste en mejorar la plataforma centrándose en su usabilidad, estabilidad, y las funcionalidades adicionales. Para este propósito, se va a recrear el proyecto existente utilizando Django como base para el backend, reemplazando a Flask. Esta decisión tiene múltiples propósitos. En primer lugar, permitirá programar desde el comienzo siguiendo buenas prácticas de programación. Django impone una estructura más estricta al proyecto, y promueve la separación de sus componentes en modelos, vistas, y templates, lo que facilitará la organización, legibilidad y modularización del código.

Además, Django permite un mejor manejo e interacción con la base de datos, tiene la capacidad de generar formularios a partir de modelos de datos, además de su sistema de ORM que permite interactuar con objetos con una sintaxis más intuitiva y concisa. También proporciona medidas de seguridad integradas, como protección ante inyecciones SQL, además de proporcionar paneles de administración para la base de datos.

Se estima que esta migración a un nuevo framework permitirá establecer una base sólida para cualquier desarrollo que se quiera hacer a futuro, con una estructura más ordenada y documentada. Adherirse a estas buenas prácticas desde un comienzo facilitará el mantenimiento y escalabilidad de la plataforma y permitirá intervenciones futuras sobre esta. Durante esta sección del trabajo también se arreglará el problema de la agrupación de categorías, y creación de súper/subcategorías, que tiene interfaz gráfica y modelo de datos, pero que actualmente no funciona.

En el frontend se mantendrá el stack tecnológico actual de HTML, CSS y JS. Se realizarán correcciones de errores, y se añadirán elementos como tooltips que sirvan de guía para los usuarios en su uso de la herramienta. Además se implementarán las vistas necesarias para la visualización de PDFs y acceder a los datos recolectados en la revisión sistemática.

4.2. Nuevas Funcionalidades

Actualmente se está considerando la librería pdf.js [13], desarrollada por *Mozilla Corporation*, para la visualización de los artículos. Se buscará implementar una manera de llenar el formulario de ingreso de artículos simultáneo al renderizado de este en una única vista, para mejorar el flujo de trabajo de los usuarios. Se deberá modificar el modelo de datos para asociar el PDF de cada artículo a sus metadatos. Se implementará una manera de referenciar información relevante de los PDFs que justifiquen su clasificación mediante el uso de librerías como pdfquery [6].

Una de las funcionalidades objetivo más importantes es la recopilación de datos de todos los artículos añadidos. Se evaluarán las métricas útiles para investigadores, y las librerías para la extracción de estas, ya sea una descarga directa en formato CSV, o la visualización de gráficos en un dashboard.

5. Plan de Trabajo

Se va a dividir el trabajo a realizar en cinco sprints de tres semanas cada uno. Dado que la revisión del stack tecnológico, creación del modelo de datos y el diseño general de la plataforma fue completado previamente, se estima que el Sprint 1, donde estaba contemplado este trabajo, requerirá menos tiempo. Esto podría generar holgura para implementar componentes opcionales adicionales o pulir la solución final implementada.

- **Sprint 1: Sistema de tags.** Desarrollo del sistema de gestión de tags para los artículos.
- **Sprint 2: Buscador unificado de Artículos.** Rediseño y streamlining de la búsqueda de artículos, con consolidación de la búsqueda regular y avanzada en una sola vista.
- **Sprint 3: Visualización del artículo, e ingreso de metadatos y tags.** Importación y visualización de archivos PDF locales, y formulario de ingreso de metadatos.
- **Sprint 4: Generación de CSV y estadísticas.** Determinación de métricas relevantes. Recolección de datos para su descarga.
- **Sprint 5: Validación y escritura de informe final.**

6. Trabajo Adelantado

A continuación se reportan los avances realizados para dar inicio al proyecto, los cuales servirán como fundación para su desarrollo de este trabajo de título.

6.1. Recopilación de requerimientos

La etapa inicial del trabajo consistió en una serie de reuniones con las profesoras Bachmann y Simmonds, donde se discutieron y definieron los requerimientos y objetivos principales de la plataforma. Durante estas reuniones, se identificaron los elementos y necesidades centrales del proyecto y se tomaron algunas decisiones de diseño de los componentes más importantes, como por ejemplo, el uso de un árbol de tags para representar más fácilmente el sistema de clasificaciones de artículos. Todas las decisiones de planeamiento y de diseño fueron posteriormente validadas para asegurar su alineación con los objetivos planteados.

6.1.1. Elaboración de historias de usuario

Como investigador realizando una revisión sistemática quiero:

6.1.2. Historias de usuario esenciales

- Quiero poder almacenar datos sobre artículos de mi revisión de tal manera que pueda extraerlos para su posterior análisis.
- Quiero poder revisar el artículo mientras lleno información sobre sus metadatos para ingresarlos a la Plataforma.
- Quiero poder clasificar los artículos con tags personalizados.
- Quiero poder agrupar tags en ciertas categorías más amplias para organizar mi trabajo.
- Quiero poder buscar papers por sus metadatos (incluyendo apariciones en el abstract) y tener la opción de revisarlo más a fondo o entrar editar dichos metadatos.
- Quiero poder ver todos los papers que tengo guardados.
- Quiero poder extraer un CSV con métricas relevantes a mi revisión.

6.1.3. Historias de usuario opcionales

- Quiero poder revisar a qué parte de un paper corresponde a un tag para justificar la elección.
- Quiero poder hacer análisis de los datos de la revisión en la plataforma.

6.1.4. Refinamiento y priorización

También a lo largo de estas reuniones se discutió la importancia de cada elemento, permitiendo evaluar su criticidad relativa en el contexto del proyecto. Esto es útil para determinar y priorizar objetivos, orientar el desarrollo, y estimar el nivel de atención y tiempo que se le dedicará a cada funcionalidad. Dado esto, se clasifican la mayoría de las funcionalidades descritas anteriormente como esenciales, ya que son críticas para cumplir los objetivos dispuestos para este proyecto. Por otro lado, las menos centrales son clasificadas como opcionales, tales como el dashboard de visualizaciones, o la asociación de una sección de un artículo con sus

tags. Estas funcionalidades pueden desarrollarse si es que se dispone de tiempo para ello terminando el resto del trabajo requerido, con la opción de dejarlas como trabajo propuesto. Esto permite centrarse en entregar un producto final viable, dejando posibles mejoras para iteraciones siguientes.

6.2. Preparación de ambiente de trabajo

Se crea un repositorio de GitHub [2] para hacer versionamiento del código, y se hace un commit inicial, que incluye la configuración base para el proyecto, utilizando Django como backend y React en el frontend. El uso de Django se discutió previamente en este informe, y estaba contemplado desde el inicio. Por otro lado, la elección de React para el frontend de la plataforma se decide por factores como la familiaridad del estudiante, permitiendo aprovechar conocimientos previos para disminuir el tiempo utilizado en aprender el stack tecnológico. Además, la amplia disponibilidad de recursos y librerías ayudará con el desarrollo, evitando gastar tiempo en crear componentes desde cero. Se prueba el funcionamiento del backend y el frontend, y de una API que permitirá transmitir información entre ellos.

6.3. Pruebas de librerías

Para el desarrollo del proyecto, fue necesario hacer pruebas sobre distintas librerías para poder cumplir apropiadamente con los requerimientos del sistema, y teniendo en cuenta características importantes, como que sean actualmente mantenidas, su facilidad de uso, y que tengan licencias de código abierto.

Para la gestión de los tags se necesita de una estructura de árbol con capacidad de drag-and-drop. Esto es para facilitar el trabajo de los investigadores al manejar los tags, de tal manera que puedan gestionar las clasificaciones relevantes y características de los artículos de una manera visual e intuitiva. Se probaron varias librerías, entre ellas se encuentran:

6.3.1. Librerías basadas en jQuery

Durante las pruebas, se evaluaron varias librerías como jqTree [4], jsTree [3] y otros plugins basados en jQuery [15], una librería de JavaScript para desarrollo web. Todas estas librerías tienen licencias permisivas, como MIT y Apache, lo que es positivo. Por otro lado, estas herramientas presentaron incompatibilidades con el framework seleccionado, requiriendo soluciones complejas para integrarlas correctamente. Tras investigar el problema, se concluyó que, aunque el uso de plugins y librerías basados en jQuery es técnicamente posible, no es recomendable en el entorno de trabajo elegido. Esto se debe a los conflictos entre la manera en que jQuery y React interactúan con el DOM (Document Object Model). Mientras React utiliza un DOM virtual, minimizando lo más posible la interacción con este, mientras que, jQuery manipula directamente el DOM real, lo que genera inconsistencias y problemas de integración y de comunicación entre ambas librerías. Además, jQuery no es una librería moderna, y proyectos relacionados como jsTree y jqTree tienen actualizaciones infrecuentes y poco substanciales, por lo cual no se consideraran como actualizados y activamente mantenidos. Por estas razones, estas alternativas fueron descartadas.

6.3.2. DevExtreme - Tree View [9]

Por otro lado, existían librerías que cumplían muy bien con las funcionalidades necesarias, pero cuya licencia no permitía su libre uso. Este es el caso de React Tree View de DevExtreme, que requiere de una licencia de 900 USD para su utilización, y la necesidad de renovarla en caso de querer hacer cualquier modificación usando estos componentes en años posteriores. Se busca que todos los componentes utilizados en el desarrollo de esta plataforma sean open source, no solo por los altos precios de las licencias, sino que para que en el futuro esta solución pueda distribuirse libremente y que sea accesible para desarrolladores que quieran construir sobre ella, modificarla, y adaptarla sin restricciones.

6.3.3. Ant Design Tree [7]

Se prueba exitosamente la librería, creando un árbol drag-and-drop utilizable. Se presentan algunos problemas inicialmente, en especial con la funcionalidad de drag-and-drop, que se logran arreglar con algo de dificultad. Cumple con tener una licencia MIT, y por tanto, es de libre uso, además es frecuentemente actualizada y mantenida. Por último se prueban otras librerías de Ant Design, como TreeSelect, que permite hacer un dropdown con los datos del árbol en el mismo formato. Considerando todos estos puntos, esta librería califica como un buen candidato según los parámetros considerados para esta elección.

6.3.4. Brimdata react-arborist [11]

La librería react-arborist fue la más simple de implementar. Tiene nodos customizables, el drag-and-drop viene listo para usar y es controlado internamente por el árbol, a diferencia del Tree de Ant Design, que requiere de más programación por parte del usuario para hacer lo mismo. Además, es una de las librerías de árboles más populares para react, tiene una amplia documentación, demos, y guías útiles. Finalmente, se decidió utilizar esta librería, ya que de todas las consideradas, era la que implementaba un sistema de drag-and-drop de manera más simple. Además, esta librería cumple con todos los requisitos mencionados anteriormente. Usando esta librería, pudo programarse un componente de árbol de tags, que permite a los usuarios arrastrar tags fácilmente y distribuirlos a lo largo del árbol a su gusto. El estado del árbol es trackeado con cada cambio, siendo retornado en un formato idéntico al input, por lo que el manejo de la información de los tags desde y hacia la base de datos se simplifica. Este componente servirá como base para la vista final de gestión de etiquetas, y que se haya logrado implementar una versión básica en esta primera iteración sin problemas disminuye el riesgo del proyecto.

6.4. Diseño de vistas y componentes

Para planear el desarrollo de una plataforma que responda a las necesidades y objetivos planteados con anterioridad, se crean una serie de mockups de las vistas necesarias. Estos servirán de guía para el desarrollo de esta herramienta, y permitieron validar tempranamente el diseño de esta.

6.4.1. Selección de revisión sistemática

Esta vista permitirá gestionar y seleccionar distintas revisiones, como se ve en la Figura 2, se tienen tarjetas conteniendo datos como título, fechas de inicio y término, y estado de completitud. La idea para esta vista surge de la necesidad de ver información de revisiones anteriores, o de participar en más de una revisión de manera simultánea, sin la necesidad de crear un sistema de sesiones.

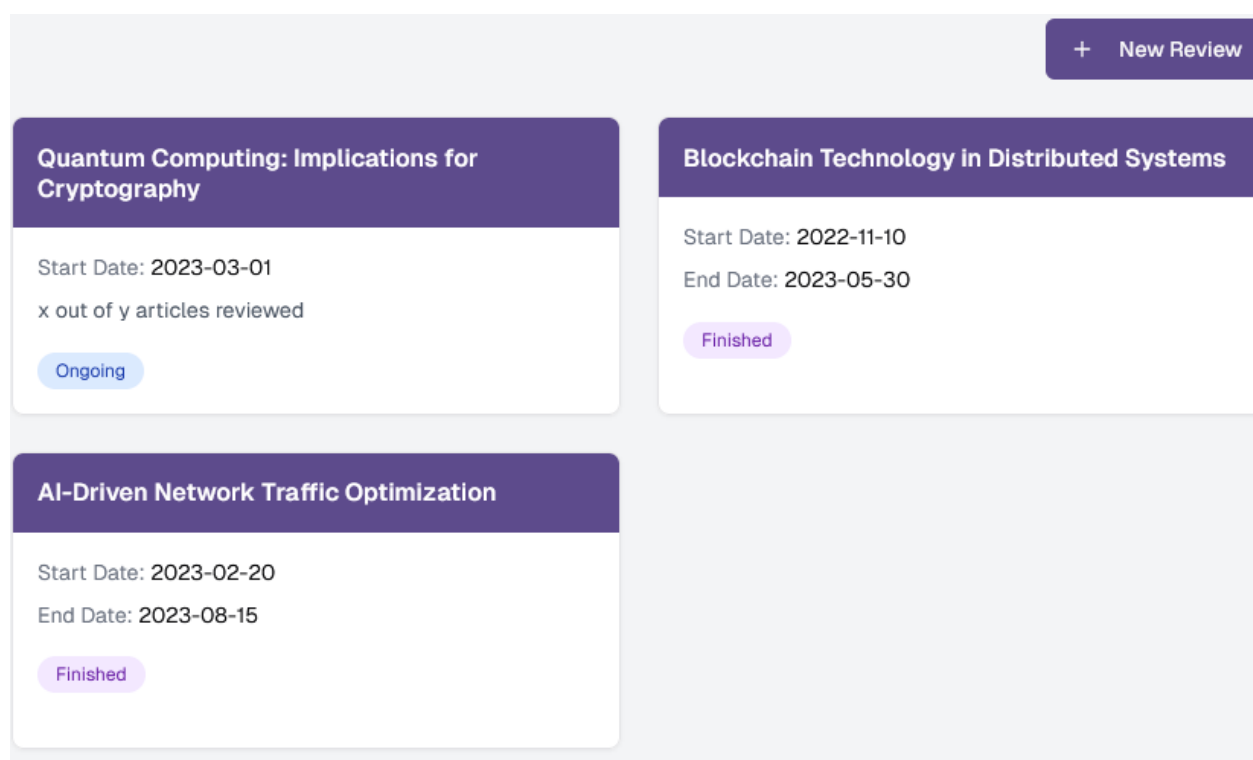


Figura 2: Componente de selección de Revisión Sistemática.

6.4.2. Búsqueda de artículos

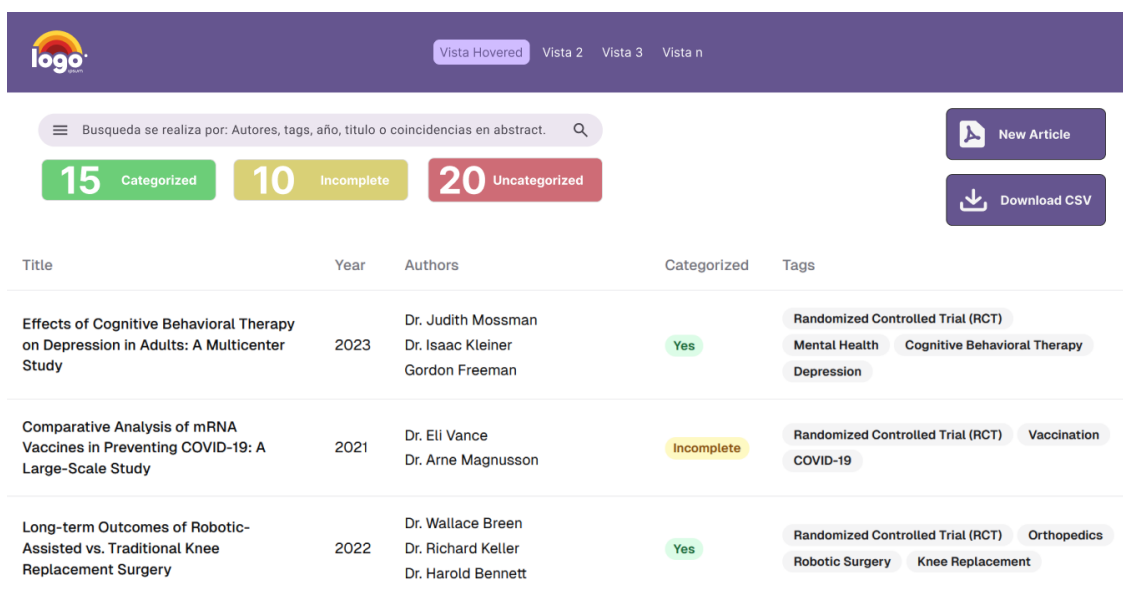
La vista presente en la figura 3 tiene varios propósitos. Como función principal, permitirá a usuarios buscar artículos de su librería por alguno de sus metadatos. Estos pueden incluir:

- Autores.
- Tags.
- Año de publicación.
- Coincidencias de texto en el título o el abstract.

Además de permitir recepcionar nuevos archivos pdf, y la exportación de toda la información en formato CSV. Por defecto, muestra una tabla scrolleable con todos los artículos pertenecientes a la revisión, presentando los datos más importantes. Por último, tiene contadores con información sobre el estado actual de la categorización de los artículos, marcándolos como:

- **Categorized:** Tiene todos los metadatos mínimos llenados.
- **Incomplete:** Tiene algunos metadatos mínimos llenados.
- **Uncategorized:** Solo el archivo PDF del artículo, sin ningún metadato.

Al diseñar esta vista, se priorizó la simplificación del proceso de búsqueda, y se procura mostrar solo los datos más importantes para una vista rápida. Como se mencionó anteriormente en este informe, existían dos buscadores distintos, uno por títulos, y otro avanzado con más opciones, en dos vistas separadas, que eran muy distintas entre sí en su uso, siendo la búsqueda avanzada difícil de navegar y poco clara en cuanto a su funcionamiento. Se espera que con esta vista unificada, la experiencia de usuario al buscar y agregar artículos mejore considerablemente, ofreciendo una alternativa moderna, intuitiva y simple de usar.



logo				
Vista Hovered Vista 2 Vista 3 Vista n				
Busqueda se realiza por: Autores, tags, año, título o coincidencias en abstract.				
New Article				
Download CSV				
15 Categorized	10 Incomplete	20 Uncategorized		
Title	Year	Authors	Categorized	Tags
Effects of Cognitive Behavioral Therapy on Depression in Adults: A Multicenter Study	2023	Dr. Judith Mossman Dr. Isaac Kleiner Gordon Freeman	Yes	Randomized Controlled Trial (RCT) Mental Health Cognitive Behavioral Therapy Depression
Comparative Analysis of mRNA Vaccines in Preventing COVID-19: A Large-Scale Study	2021	Dr. Eli Vance Dr. Arne Magnusson	Incomplete	Randomized Controlled Trial (RCT) COVID-19 Vaccination
Long-term Outcomes of Robotic-Assisted vs. Traditional Knee Replacement Surgery	2022	Dr. Wallace Breen Dr. Richard Keller Dr. Harold Bennett	Yes	Randomized Controlled Trial (RCT) Robotic Surgery Knee Replacement Orthopedics

Figura 3: Vista de búsqueda de artículos.

Cuando se selecciona un elemento de la tabla, se despliega un modal que muestra la información completa del artículo, como puede verse en la Figura 4. Esto se hace para evitar sobrecargar la vista principal, ya que algunos campos, como el abstract, el resumen y los comentarios, pueden contener texto más extenso que ocuparía demasiado espacio con información que podría no ser inmediatamente relevante para la tabla, que debe ser una vista rápida y concisa de los datos más importantes.

Además, desde este modal, se ofrece la opción de navegar a la vista de edición de metadatos, donde se pueden realizar modificaciones detalladas a la información del artículo.

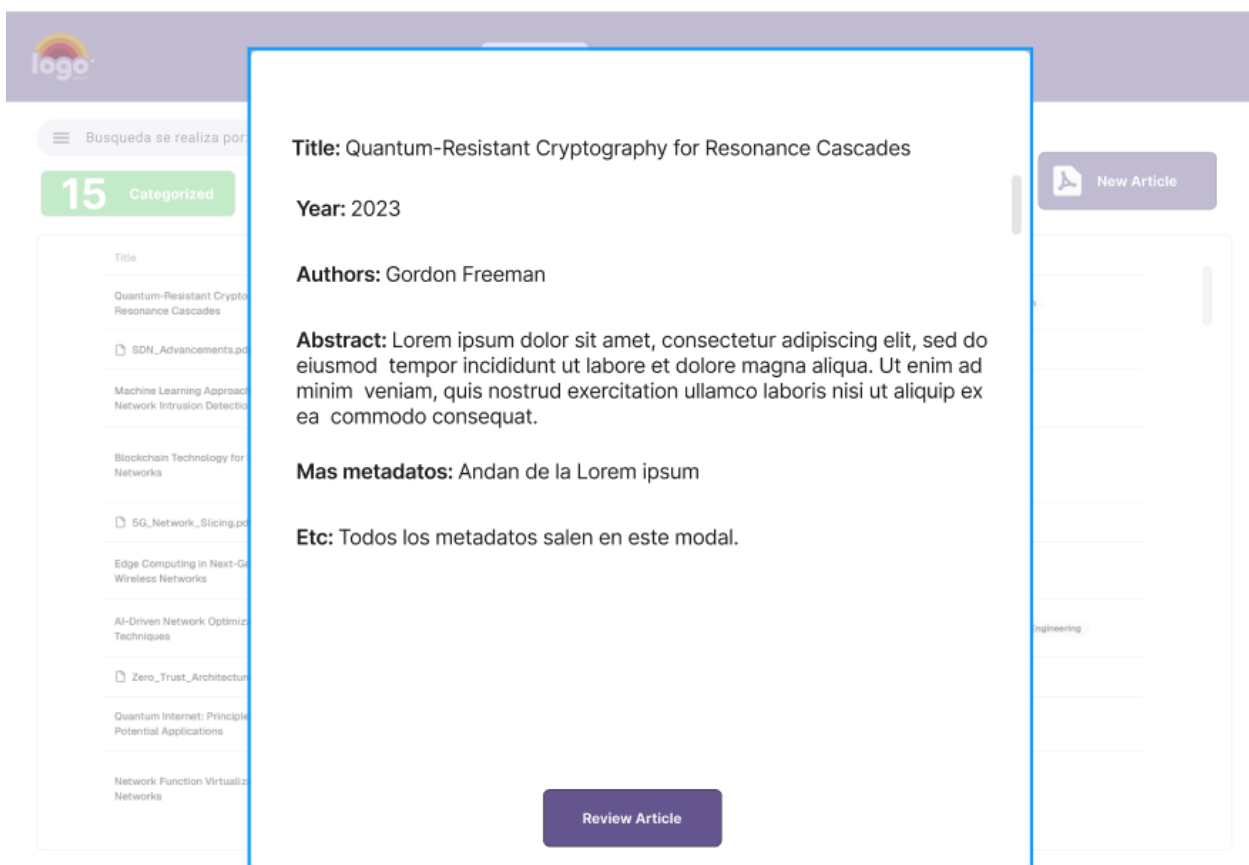


Figura 4: Modal con información detallada.

6.5. Lectura y edición de metadatos

Esta vista tiene dos componentes principales, un lector de archivos PDF y un formulario para información de los metadatos del artículo. Esto permite al usuario la revisión y la clasificación simultánea de los artículos, como puede visualizarse en la Figura 5. La inclusión de un lector de PDFs en la plataforma no es esencial para el funcionamiento de esta, pero se estimó que la presencia de este permitiría que los investigadores tengan acceso a toda la información necesaria para realizar la tarea de la clasificación de los artículos en un solo lugar, lo cual es conveniente, y le ahorrará tener que abrir el artículo en alguna otra pestaña, o revisar una impresión de este.

El componente de selección de tags que se visualiza actualmente es TreeSelect [8] de Ant Design. No se han probado actualmente más librerías de este tipo, pero se evaluarán distintas opciones, con el requisito de que el selector muestre un árbol indentado correctamente, con nodos colapsables, y que sea de selección múltiple, para permitir al usuario navegarlo con facilidad, y seleccionar los tags que correspondan para clasificar adecuadamente el artículo.

En su iteración más básica, se tendrá una sección de comentarios que permita al investigador justificar la elección de ciertos tags para el artículo, o añadir cualquier anotación que considere pertinente. Esto es útil para recordar al usuario sus decisiones de clasificación. Si hay tiempo de desarrollo para hacerlo, en esta sección se piensa implementar una manera de asociar segmentos específicos del artículo a tags, para hacer este proceso aún más simple y ágil para el usuario.

The screenshot shows a web interface for viewing and editing article metadata. The top navigation bar includes the 'logo' and tabs for 'Vista Hovered', 'Vista 2', 'Vista 3', and 'Vista n'. The main content area displays the article 'Domestic cats (*Felis catus*) discriminate their names from other words' by Atsuko Saito et al. The article text is visible, including the abstract and introduction. On the right side, there is a metadata form with fields for Title, Year, Authors, Abstract, and Tags. The Tags field is expanded, showing a tree structure with 'Node1' selected, which has sub-nodes 'Child Node1', 'Child Node2', 'Child Node3', 'Child Node4', and 'Child Node5'. A 'Submit' button is located at the bottom right of the form.

Figura 5: Vista de lectura y formulario de metadatos.

6.6. Gestión de Tags

La vista de gestión de tags, que puede verse en la Figura 6, tiene como propósito principal el manejo del árbol de tags de cada revisión sistemática. Se tiene un árbol jerárquico, presentado en la Figura 7, donde los tags pueden ser agrupados en categorías más generales, generando un sistema de clasificación en categorías y subcategorías, cada tag representando alguna clasificación definida por el usuario según la necesidades de su revisión sistemática, ya sean estas preguntas acerca de aspectos técnicos de los artículos, sobre las metodologías que usan en su investigación, o sobre el tipo de contenido.

El árbol tiene una funcionalidad de drag-and-drop, que permite reorganizar los tags de manera simple e intuitiva, permitiendo mover un nodo y reubicarlo a él y a todos sus hijos, manteniendo la estructura. Este sistema permite clasificar artículos de manera estructurada y dinámica, organizada jerárquicamente por categorías que van de generales a específicas con cada nivel del árbol. Permite asignar múltiples clasificaciones y responder distintas preguntas sobre un artículo al seleccionar hojas de las distintas ramas. Además, es escalable, ya que mientras el volumen tags puede crecer, la facilidad de navegación se mantiene, permitiendo al usuario expandir y colapsar ramas para centrarse en sus áreas de interés. También mejora la capacidad de búsqueda y filtrado, por ejemplo, permitiendo filtrar por ramas y sus subcategorías.

Además del árbol, se planea agregar una sección con información del tag seleccionado, incluyendo su título, descripción, y el número total de artículos utilizándolo actualmente, con un listado de dichos artículos, utilizando el mismo componente de tabla que se ve en la vista de búsqueda.

Tag Name: Randomized Controlled Trial (RCT)

Tag Description: A study that randomly assigns participants to groups to compare the effectiveness of different treatments or interventions.

There are currently **7** articles using this tag.

Title	Year	Authors	Categorized	Tags
Effects of Cognitive Behavioral Therapy on Depression in Adults: A Multicenter Study	2023	Dr. Judith Mossman Dr. Isaac Kleiner Gordon Freeman	Yes	Randomized Controlled Trial (RCT) Mental Health Cognitive Behavioral Therapy Depression
Comparative Analysis of mRNA Vaccines in Preventing COVID-19: A Large-Scale Study	2021	Dr. Eli Vance Dr. Arne Magnusson	Incomplete	Randomized Controlled Trial (RCT) Vaccination COVID-19
Long-term Outcomes of Robotic-Assisted vs. Traditional Knee Replacement Surgery	2022	Dr. Wallace Breen Dr. Richard Keller Dr. Harold Bennett	Yes	Randomized Controlled Trial (RCT) Orthopedics Robotic Surgery Knee Replacement
Impact of Mindfulness-Based Stress Reduction on Chronic Pain Management	2023	Cave Johnson Caroline Green	Incomplete	Randomized Controlled Trial (RCT) Chronic Pain Mindfulness
Effectiveness of a New Targeted Therapy in Advanced Stage Lung Cancer Patients	2022	Dr. Arian Vance Doug Rattmann	Yes	Randomized Controlled Trial (RCT) Oncology Targeted Therapy Lung Cancer
Evaluation of a Novel Gene Therapy Approach for Cystic Fibrosis Treatment	2023	Dr. Gregory Simmons	Incomplete	Randomized Controlled Trial (RCT) Gene Therapy Cystic Fibrosis
The Role of Gut Microbiome in Obesity: A Comprehensive Analysis	2022	Dr. Wallace Breen Dr. Judith Mossman Dr. Isaac Kleiner	Yes	Randomized Controlled Trial (RCT) Microbiome Obesity Nutrition

Figura 6: Vista de gestión de tags.

Ejemplo de arbol para investigación Medica

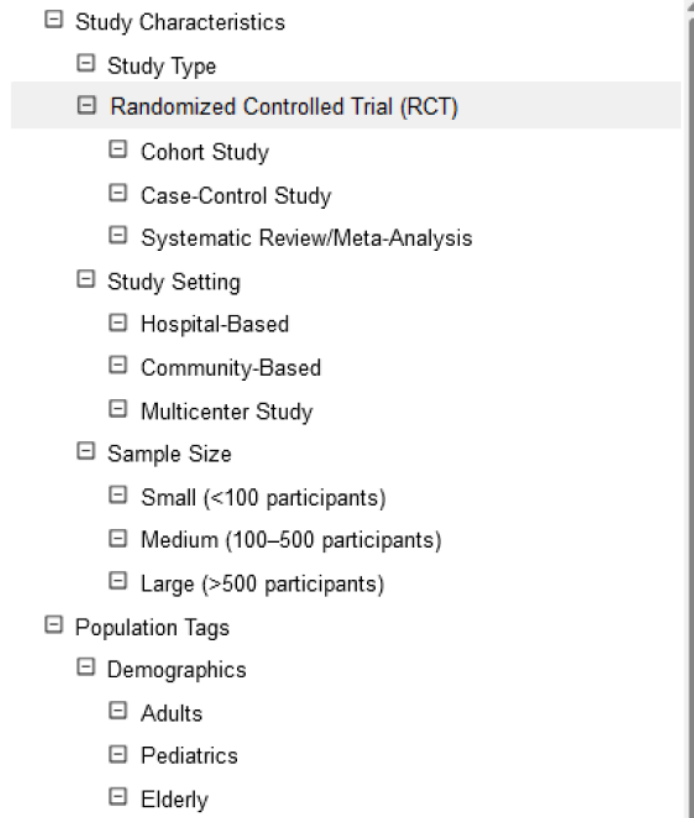


Figura 7: Árbol de tags de prueba en mas detalle

6.7. Desarrollo de modelo de datos

El modelo de datos diseñado para esta herramienta, que puede verse en la Figura 8, busca ofrecer una manera de manejar las relaciones entre los diferentes componentes del sistema, facilitando la interacción entre el frontend y el backend, de tal manera que puedan satisfacerse las funcionalidades principales de la herramienta, que consisten en el almacenamiento, organización y clasificación de artículos de las revisiones sistemáticas en las que participen sus usuarios.

- **Systematic Review:** Contiene información general de cada revisión sistemática
- **Article:** Contiene metadatos de artículos.
- **Author:** Lista de autores
- **Tags:** Información para cada tag y referencia a su padre para establecer relación de árbol.
- **ArticleTags y ArticleAuthors:** Junction tables para relacionar múltiples tags y autores a artículos sin generar redundancia.

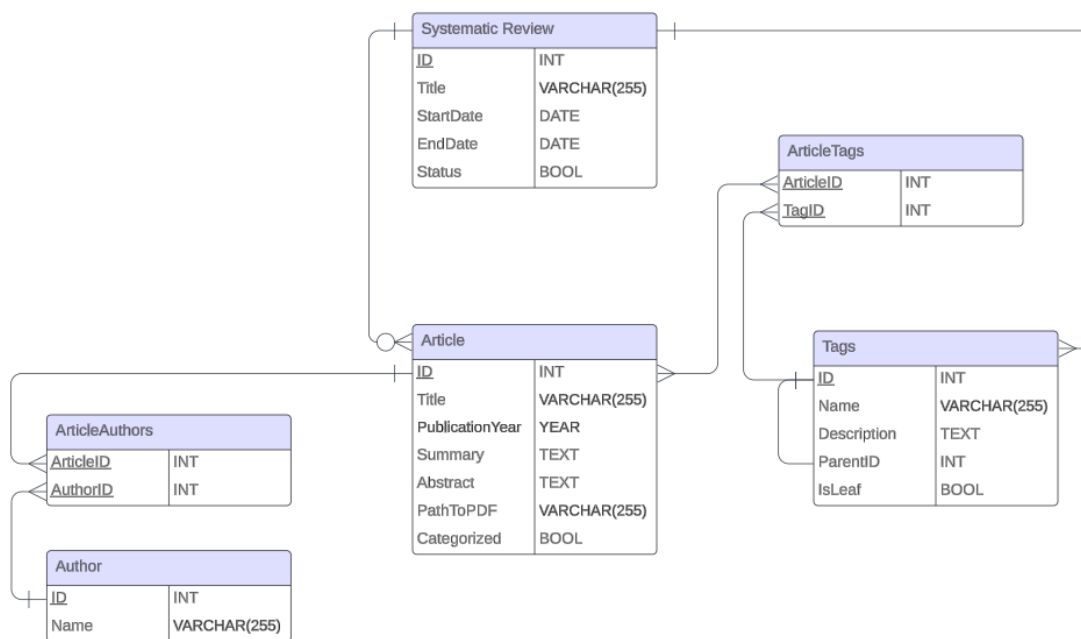


Figura 8: Modelo de datos

6.8. Síntesis

El trabajo realizado en este adelanto establece bases sólidas para el desarrollo de la herramienta. Las funcionalidades principales se encuentran planificadas, representadas en las vistas mockup y el modelo de datos. Además, se ha comenzado el trabajo de desarrollo en la gestión de tags, que fue uno de los puntos débiles más importantes de la solución anterior. Se tienen buenos resultados creando un componente árbol con datos dummy.

Todos los avances se han planeado y validado a lo largo del semestre en reuniones con las profesoras guía satisfactoriamente.

Referencias

- [1] Bachmann, Ivana: *Systematic Review Tool*, Accedido 25/09/2024. <https://github.com/tami-di/systematic-review-tool/tree/tami-updates>.
- [2] Basáez, Andrés: *SysRev*, Accedido 05/12/2024. <https://github.com/5goblins/SysRev>.
- [3] Bozhanov, Ivan: *jsTree - A jquery plugin, that provides interactive trees*, Accedido 05/12/2024. <https://www.jstree.com/>.
- [4] Braak, Marco: *jqTree - A jQuery widget for displaying a tree*, Accedido 05/12/2024. <https://mbraak.github.io/jqTree/>.
- [5] Brooke, John: *SUS: A quick and dirty usability scale*. Usability Eval. Ind., 189, Noviembre 1995.
- [6] Cushman, Jack: *pdfquery*, Accedido 25/09/2024. <https://pypi.org/project/pdfquery/>.
- [7] Design, Ant: *Tree by Ant Design - A Multiple-level structure list.*, Accedido 05/12/2024. <https://ant.design/components/tree>.
- [8] Design, Ant: *TreeSelect by Ant Design - Tree selection control.*, Accedido 05/12/2024. <https://ant.design/components/tree-select>.
- [9] DevExtreme: *React TreeView - Getting Started.*, Accedido 05/12/2024. https://js.devexpress.com/React/Documentation/ApiReference/UI_Components/dxTreeView/.
- [10] Elsevier, Mendeley: *Mendeley reference management software*, Accedido 25/09/2024. <https://www.mendeley.com/>.
- [11] Kerr, James: *React Arborist - The complete tree view component for React*, Accedido 05/12/2024. <https://github.com/brimdata/react-arborist>.
- [12] Kitchenham, Barbara: *Procedures for Performing Systematic Reviews*. Keele, UK, Keele Univ., 33, Agosto 2004.
- [13] Mozilla: *pdf.js Github Repository*, Accedido 22/09/2024. <https://github.com/mozilla/pdf.js>.
- [14] Nielsen, Jakob y Thomas K. Landauer: *A mathematical model of the finding of usability problems*. En *Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems*, CHI '93, página 206–213, New York, NY, USA, 1993. Association for Computing Machinery, ISBN 0897915755. <https://doi.org/10.1145/169059.169166>.
- [15] Team, The jQuery: *jQuery: The Write Less, Do More, JavaScript Library*. <https://jquery.com/>.

//jquery.com/year={Accedido13/12/2024},.

- [16] Temple University Libraries: *Common Review Types - Evidence Synthesis and Systematic Reviews - Research Guides at Temple University*, Accedido 08/09/2024. <https://guides.temple.edu/c.php?g=78618&p=9548660>.
- [17] The JabRef team: *JabRef citation and reference management software*, Accedido 25/09/2024. <https://www.jabref.org/>.