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

HERRAMIENTA DE APOYO A REVISIONES SISTEMÁTICAS DE LA LITERATURA EN EL ÁREA DE LA COMPUTACIÓN

TESIS PARA OPTAR AL GRADO DE MAGÍSTER EN TECNOLOGÍAS DE INFORMACIÓN

NÉSTOR ADRIÁN LÓPEZ LUQUE

PROFESORES GUÍA:
DANIEL PEROVICH GEROSA
SERGIO OCHOA DELORENZI

MIEMBROS DE LA COMISION:
CLAUDIO GUTIERREZ GALLARDO
JOCELYN SIMMONDS WAGEMANN
PEDRO ROSSEL CID

SANTIAGO DE CHILE
2019

TESIS PARA OPTAR AL GRADO DE: Magíster en Tecnologías de Información
Por: Néstor Adrián López Luque
Fecha: marzo 2019
Profesores guía: Daniel Perovich, Sergio Ochoa

Herramienta de Apoyo a Revisiones Sistemáticas de la Literatura en el Área de la Computación

Las revisiones sistemáticas de la literatura (RSL) son actividades de gran utilidad en entornos de investigación, ya que permiten identificar los principales estudios científicos en una cierta temática, a través de preguntas de investigación específicas. Estas actividades siguen un proceso recomendado, el cual varía un poco según la disciplina sobre la que se realice dicho estudio; por ejemplo, el proceso de RSL para el área de computación es levemente diferente al que se sigue para el área de medicina o de física. Este trabajo de tesis consideró sólo el proceso de RSL para computación, y particularmente para el área de ingeniería de software, para la cual hay una recomendación de proceso definido y ampliamente validado.

A pesar de la utilidad que tiene realizar RSL, requiere un gran esfuerzo por parte de los involucrados en realizarla. Este trabajo de tesis abordó esta problemática, tratando de simplificar dicho proceso y hacerlo más abordable para el usuario final (en términos del esfuerzo requerido), sin perder precisión o flexibilidad respecto al proceso recomendado para el área de ingeniería de software.

Como resultado de este trabajo de tesis se realizaron modificaciones al proceso de RSL recomendado, se definieron nuevos indicadores para usar en dicho proceso, y también los algoritmos requeridos para calcularlos. Además, se desarrolló una aplicación de software Web para definir y gestionar la mayor parte del proceso de RSL, incluyendo las actividades colaborativas que forman parte de él. El software desarrollado implementa las modificaciones propuestas al proceso, y calcula y utiliza los indicadores definidos.

La usabilidad y utilidad de la aplicación desarrollada fue evaluada a través de dos focus groups con investigadores que han realizado RSL. Además, la precisión del software para recuperar estudios relevantes para una RSL fue evaluada en la práctica, y comparada con los resultados reportados en tres procesos de revisión diferentes publicados en la literatura. El resultado de esta comparación fue muy positivo reduciendo el tiempo necesario para desarrollar todo el proceso, así como para encontrar los artículos relevantes, mostrando que la propuesta de modificación del proceso de RSL, así como su implementación a través de la aplicación Web antes mencionada, representan una ayuda real y tangible para los investigadores que se ven enfrentados a realizar esta actividad.

Tabla de Contenido

1. Introducción

1.1 Contexto

Una de las propiedades del conocimiento científico es su carácter acumulativo. El avance en el conocimiento se produce porque el saber acumulado es la base sobre la que se desarrollan las nuevas investigaciones, típicamente a través de un proceso de refutación, confirmación o la exploración de nuevas formulaciones que contribuyan a la explicación de los fenómenos de estudio [Gui15]. Es de esta forma como el conocimiento progresa, se desarrollan teorías y se explican los fenómenos del mundo físico y social [Mart01].

El trabajo científico usualmente tiene como objetivo responder preguntas de investigación, y a través de ese mecanismo, se busca avanzar el “estado del arte” (o el conocimiento actual) en un dominio específico. Para poder avanzar dicho conocimiento, es necesario saber cuál es el conocimiento disponible en torno a una pregunta de investigación, y de esa manera estar seguro que la respuesta a la pregunta planteada efectivamente logrará ese objetivo.

La forma típica de determinar el conocimiento actual en torno a una pregunta de investigación, es revisando los trabajos previos de otros investigadores que han intentado responder la misma pregunta, o una similar. Hay muchos métodos para revisar estos trabajos previos; por ejemplo, revisión sistemática de la literatura, revisión integradora, meta-análisis, revisión panorámica, revisión paraguas, etc. [Gui15, Whi14]. Una de las más exhaustivas, pero también costosas en términos del esfuerzo requerido para realizarla, es la revisión sistemática de la literatura [Pin06, Kit07].

Las revisiones sistemáticas surgen como alternativa a las revisiones tradicionales de literatura, y usan métodos sistemáticos y explícitos para identificar, valorar y seleccionar los estudios primarios relevantes, relacionados con las palabras claves que se encuentran en la pregunta de investigación a responder. Sin embargo, el esfuerzo de llevar a cabo estas revisiones es su principal restricción [Pin06].

En términos prácticos, este proceso inicia con la extracción de las palabras claves y sus respectivos sinónimos, desde las peguntas de investigación. Luego, a partir de estas palabras claves se realiza la búsqueda de estudios primarios en diferentes bibliotecas digitales. Los resultados de estas búsquedas deben ser depurados y evaluados para seleccionar los artículos relevantes. Para finalizar esta etapa se realiza una verificación de la calidad de la RSL [Pin06].

A partir de la masificación de Internet y la implementación de bibliotecas digitales en línea, se ha producido una explosión en la cantidad de literatura que se encuentra a disposición a través de la red. Si bien esta situación a priori puede verse como muy beneficiosa, el cúmulo de información disponible puede convertirse en una dificultad a la hora de establecer el conocimiento acumulado en un dominio específico. En la actualidad podemos perdernos en una gran cantidad de información en el que encontramos, desde información irrelevante a información esencial. Este gran volumen de literatura disponible requiere, por una parte, conocer y discernir la información relevante de aquella que no lo es, y posteriormente evaluar, juzgar y localizar la documentación recuperada [Gui15].

Como se mencionó antes, para realizar una RSL es necesario ejecutar una serie de tareas de forma ordenada; sin embargo, el esfuerzo requerido para llevarlas a cabo es alto, y el proceso a seguir es muy riguroso. Por lo tanto, realizarlas de forma manual no sólo es demandante, sino propensa a muchos errores, debido a los miles de artículos que el investigador debe clasificar, extrayendo la metadata de cada uno y almacenándola en alguna aplicación que le permita organizarla para su respectivo análisis. Esto también hace que el investigador pueda equivocarse en el armado de la cadena de búsqueda a alguna biblioteca en particular, o en la extracción y almacenamiento de la metadata. Por ejemplo, el método requiere aplicar diversos criterios para determinar las bibliotecas digitales sobre las cuales buscar, desarrollar las cadenas de búsqueda por medio de las palabras claves que se encuentran en las preguntas de investigación, extraer y analizar la información de alto nivel de todos los estudios primarios encontrados (título, resumen, autores y palabras claves), realizar la depuración y clasificación de los trabajos recuperados, y finalmente realizar el análisis y la integración del conocimiento contenido en los artículos seleccionados.

En el Departamento de Ciencias de la Computación (DCC) de la Facultad de Ciencias Físicas y Matemáticas de la Universidad de Chile se realiza mucha investigación, y de buen nivel [DCC18]. Por lo tanto, sus académicos y alumnos de postgrado se ven frecuentemente enfrentados al desafío de realizar una revisión sistemática. Dado el carácter evolutivo del conocimiento, esta actividad se repite una y otra vez.

1.2 Problema a Abordar

En el caso de las revisiones sistemáticas en el área de computación, al igual que en la mayoría de las áreas, realizar este proceso llega a ser muy engorroso debido a la cantidad de fuentes de información disponibles, y a las múltiples instancias que existen para cometer errores. Si bien hay varias propuestas de procedimientos para realizar estas revisiones, prácticamente todas ellas requieren de al menos 2 o 3 personas para poder llevarse a cabo [Kit07], [Ria10]; esto con el fin de hacer que el esfuerzo requerido sea más abordable, y al proceso más robusto ante potenciales errores y sesgos.

Además de todo lo antes planteado, el investigador también debe dedicar tiempo y esfuerzo a definir una estructura para almacenar y clasificar la información de la metadata asociada a los estudios recuperados. Esta metadata típicamente se usa para realizar la depuración de la lista de trabajos a considerar, y clasificarlos según sus necesidades. Debido a la cantidad de trabajo que requiere realizar una RSL existen varias instancias para cometer errores, cuando el proceso se realiza de forma manual. Esto hace a dichos RSL potencialmente deficientes debido a limitaciones de alcance y/o de profundidad.

En la actualidad se han desarrollado algunas herramientas para apoyar algunas actividades del proceso de revisión sistemática. Estos desarrollos han sido auspiciados principalmente por universidades e instituciones independientes. Entre las herramientas más destacadas se encuentran SLuRp (Systematic Literature unified Review Program) [Bow12], StArt (State of the Art through systematic review) [Her12] y RSL-Tool [Fer10].

Al hacer un análisis de las capacidades de estas herramientas, se identificó que la principal limitante que tienen es la conexión a bibliotecas digitales, es decir, a las fuentes de información. En muchos casos estas herramientas no están desarrolladas para apoyar revisiones sistemáticas en el área de computación, y por lo tanto requieren que el investigador genere su propia biblioteca de artículos a procesar. Esto además ser una limitante para el investigador, le crea la tarea adicional de tener que exportar los resultados de las bibliotecas digitales a la herramienta seleccionada. Por otra parte, la documentación para el uso de estas herramientas es usualmente escasa, por lo que el investigador debe invertir un esfuerzo no despreciable en aprender a configurarlas y a usarlas. El realizar todo el proceso de forma manual implica que el investigador esté sujeto a las siguientes implicancias:

- Gran cantidad de tiempo y esfuerzo invertido en la búsqueda en bibliotecas digitales de calidad relacionadas con el área de computación, además del tiempo invertido en la generación de las cadenas de búsqueda, su aplicación (selección de artículos) en cada biblioteca digital, y la gestión inicial de la gran cantidad de resultados de las búsquedas para su depuración.

- La alta probabilidad de cometer errores humanos al momento de aplicar las cadenas de búsqueda a cada fuente de información (biblioteca digital), o de omitir alguna cadena al momento de aplicarlas a dichos repositorios, además de tener que buscar herramientas que le permita a los investigadores gestionar la información resultante, facilitando de alguna manera su depuración inicial para reducir el riesgo de cometer errores en la depuración de los resultados.

- El esfuerzo que el investigador debe que invertir para poder familiarizarse con cada biblioteca digital, ya que cada una tiene definidos sus procesos de búsqueda de

forma independiente, y difieren entre ellas en la forma de plantear los criterios de búsqueda.

1.3 Esbozo de la Solución

Para ayudar a paliar el desafío planteado, en este trabajo de tesis se desarrolló una herramienta de software para apoyar a la realización de revisiones sistemáticas, particularmente aquellas realizadas en el ámbito de la computación. Para ello se buscó en algunos casos automatizar, y en otros apoyar, las actividades consideradas en dicho proceso. El proceso general de RSL que apoya la herramienta adhiere al propuesto por Kitchenham et al. [Kit07], puesto que es uno de los más aceptados por investigadores de Ciencias de la Computación [Fel17], [Soo16].

Inicialmente la herramienta desarrollada utiliza como base la información de las publicaciones que están disponibles en DBLP (Digital Bibliography & Library Project) [DBL18], y luego complementa los metadatos faltantes de dichos estudios con información obtenida desde administradores de referencias como Mendeley [Men18]. Esta decisión se tomó a partir de una serie de pruebas que se realizaron para verificar la completitud de los artículos almacenados en DBLP, todos pertenecientes al área de las ciencias de la computación. A futuro se analizarán posibles extensiones.

El software desarrollado en esta tesis apoya la realización de las siguientes actividades dentro del proceso de una RSL [Kit07]: (1) definición de las cadenas de búsqueda en base a los términos y sinónimos plateados por los investigadores, (2) definición de las fuentes de información que serán utilizadas en la RSL (DBLP y Springer), (3) realización de las búsquedas y recuperación de los metadatos básicos de los estudios resultantes (por ejemplo, título, resumen, y autores de los estudios), (4) eliminación de estudios duplicados, y (5) apoyo al proceso de evaluación colaborativa de la calidad de los estudios recuperados.

La herramienta desarrollada permite al investigador ingresar las palabras claves o términos (con sus respectivos sinónimos) que se encuentran en las preguntas de investigación que pretende responder. En base a estas palabras claves, la herramienta automáticamente le propondrá las cadenas de búsqueda a utilizar para recuperar los estudios primarios pertinentes.

Una vez concluida esa etapa, el software procederá a realizar la búsqueda en la base de datos local. La búsqueda se realizará considerando el contenido del título, el resumen (abstract) y las palabras claves (si las hubiere) de los documentos almacenados en la base de datos. Adicionalmente se consulta esta información en el repositorio de capítulos de libros de la editorial Springer, a través de una API (Application Programming Interface),

pues se observó que dichas publicaciones no estaban en DBLP, y muchas de ellas aparecián en las RSL reportadas en la literatura.

La metadata de cada artículo (título, resumen, autores y palabras claves) se almacena en una estructura que permite realizar una primera depuración automática de los estudios duplicados. Después de eso, el software procede a presentar los resultados al investigador en un formato ordenado, para que éste pueda realizar el resto de las etapas de la revisión de manera asistida. Además de la precisión y automatización de las búsquedas, este trabajo pretende además generar un ahorro de tiempo y esfuerzo en la realización de este proceso, en comparación con lo que tendría que invertir la persona si lo realizara de forma manual.

1.4 Objetivos de la Tesis

El objetivo general de este trabajo de tesis es desarrollar una herramienta que brinde apoyo a los investigadores durante la realización de revisiones sistemáticas de la literatura en el área de la computación. Con esto se busca que la herramienta permita reducir el tiempo, la complejidad y el esfuerzo requerido para realizar la búsqueda, clasificación y gestión de los estudios primarios, así como la reducción de errores debido al procesamiento manual. Para lograr este objetivo se definieron los siguientes objetivos específicos:

- Desarrollar un servicio que genere de forma automática las cadenas de búsqueda a partir de las palabras claves que el investigador ingrese. Esto permite ahorrar tiempo y disminuir de forma significativa el error humano en este proceso.
- Desarrollar un servicio que automatice las búsquedas de estudios primarios, y permita una fácil clasificación y depuración inicial de los resultados, con el fin de reducir el esfuerzo de procesamiento de dichos estudios.
- Desarrollar un servicio que automatice la gestión de la metadata de los estudios primarios, y de los estudios mismos, con el fin de brindar mayor eficiencia al proceso de revisiones sistemáticas.
- Desarrollar la herramienta que integre los servicios antes mencionados, con el fin de facilitarle las revisiones sistemáticas de la literatura a los investigadores.

1.5 Metodología

En el desarrollo de este trabajo de tesis se llevaron a cabo las siguientes.

Fase Preliminar:

1. *Delimitación y alcance de la herramienta dentro del proceso de las revisiones sistemáticas*: Esto se refiere al desglose y definición de los requisitos funcionales

que la herramienta deberá cumplir para realizar el apoyo de las revisiones sistemáticas en el área de la computación. Entre las tareas de esta fase se encuentra el determinar las fuentes de información, estudiando y definiendo la forma en que se accederá a cada una. Además, se requiere definir de manera formal los procesos que se verán automatizados por la herramienta.

2. *Determinar la factibilidad de la tecnología a utilizar*: En este paso se determina la plataforma tecnológica, el lenguaje de programación y gestor de bases de datos a utilizar, así como un análisis de conexión a las diferentes bibliotecas digitales. Luego de este análisis se determinó utilizar DBLP como fuente principal para la herramienta.
3. *Diseño de las pruebas funcionales a realizar*: En esta etapa se diseña la matriz de pruebas funcionales que se realizan a la aplicación una vez terminado el desarrollo.

Fase de Desarrollo:

1. *Desarrollo de la solución*: Esta etapa consistió en el diseño de la herramienta y su implementación utilizando la plataforma elegida previamente.
2. *Desarrollo de pruebas funcionales*: Esta etapa consistió en realizar pruebas funcionales de la herramienta obtenida.
3. *Desarrollo de la documentación*: Se desarrolló un manual de usuario de forma digital en un documento de texto y documentación técnica de la herramienta.

Fase de Evaluación:

1. *Selección de participantes*: En esta actividad se seleccionó la población de investigadores en el área de la computación que participará en la validación de la herramienta, determinando por medio de encuestas, la percepción de los participantes acerca de la utilidad y la usabilidad de la herramienta.
2. *Evaluación de la confiabilidad y utilidad aparente de la herramienta*: Se replicó el proceso completo de varias revisiones sistemáticas con y sin la herramienta acotando para fines prácticos las cadenas de búsqueda, con el fin de comprobar si la herramienta minimiza de forma significativa los falsos positivos y falsos negativos, con el fin de verificar su confiabilidad.

1.7 Estructura del Documento

Este documento de tesis se estructura en 7 capítulos. El Capítulo 2 introduce los conceptos y términos más importantes relacionados con el proceso de revisión sistemática de la literatura, las herramientas que en la actualidad apoyan este proceso, un detalle las partes que éstas consideran y las que dejan fuera. El Capítulo 3 se define el proceso para una RSL que se apoya por medio del software, se analizan las fuentes o

bibliotecas digitales y se plantea una estrategia a seguir para obtener los artículos. Además, se plantea una visión de la solución final a alto nivel y se lista los requisitos que debe cumplir el software. El Capítulo 4 contiene una descripción de la arquitectura y las tecnologías empleadas para el desarrollo de la solución, y el modelo de datos con una explicación de cómo se estructura el almacenamiento de la información. El Capítulo 5 detalla la implementación de la aplicación, por medio de la interfaz de usuario de la misma, y cómo éstas apoyan las diferentes etapas del proceso. El Capítulo 6 describe la validación de la solución, basada en la replicación de revisiones sistemáticas reportadas en la literatura, y las pruebas de usabilidad y utilidad realizadas con los usuarios. El Capítulo 7 presenta las conclusiones de esta tesis y el trabajo a futuro.

2. Marco Teórico

En este capítulo se define el detalle del proceso para realizar una revisión sistemática de la literatura, luego se realiza un análisis de las principales bibliotecas digitales en el área de la computación, y al final se describen y comparan las principales características de las diferentes herramientas que existen en el mercado para apoya las RSL.

2.1 Revisión Sistemática de la Literatura

Una revisión sistemática de la literatura (o RSL) se puede definir como un proceso formal y ordenado, que tiene como objetivo obtener los estudios primarios más relevantes relacionados con una o más preguntas de investigación. Estas revisiones se hacen con el fin de crear una base sólida de conocimiento, que sirva como fundamento para la investigación a realizar [Oko10]. Kitchenham define los siguientes pasos para realizar una revisión sistemática de la literatura [Kit07] (ver Figura 1).

Tal como se muestra en la Figura 1, el proceso de RSL se inicia con la “definición de las preguntas de investigación” (etapa 1), a partir de las cuales se procede a la identificación del conjunto de términos (o palabras) claves. Estos términos están asociados al conocimiento que se desea obtener para poder responder una o más preguntas de investigación. Una vez definidos los términos con sus respectivos sinónimos, éstos son combinados para obtener el conjunto de cadenas de búsqueda. Estas cadenas serán luego usadas para realizar la búsqueda de los estudios primarios (reportados en artículos científicos), por lo que es fundamental que se generen todas las posibles combinaciones de palabras claves. Para ello se utiliza como conectores a los diferentes operadores lógicos, AND y OR principalmente.

En la etapa “proceso de búsqueda” (etapa 2), el investigador debe seleccionar las bibliotecas digitales donde va a buscar la información. Es importante que dichas fuentes de información tengan alta influencia en el área de investigación. Esta etapa también considera el hecho de que el investigador debe además familiarizarse con las características de cada una de estas fuentes, haciendo uso de la interfaz Web de cada biblioteca o haciendo uso de las especificaciones de sus API (cuando están disponibles). Esto es requerido para consultar el contenido de las bibliotecas digitales, y saber cuál es el formato en el que se reciben los resultados de las búsquedas.

Lo siguiente en la etapa 2 es la aplicación de las cadenas de búsqueda sobre las bibliotecas digitales. Para ello el investigador debe ejecutar las cadenas definidas sobre cada una de las bibliotecas seleccionadas, pudiendo requerir ajustar las cadenas para su efectiva aplicación a cada biblioteca en particular. Como resultado de esto se obtiene una gran cantidad de estudios primarios relacionados con la investigación, junto con otros

estudios que no necesariamente son relevantes para responder las preguntas de investigación planteadas.

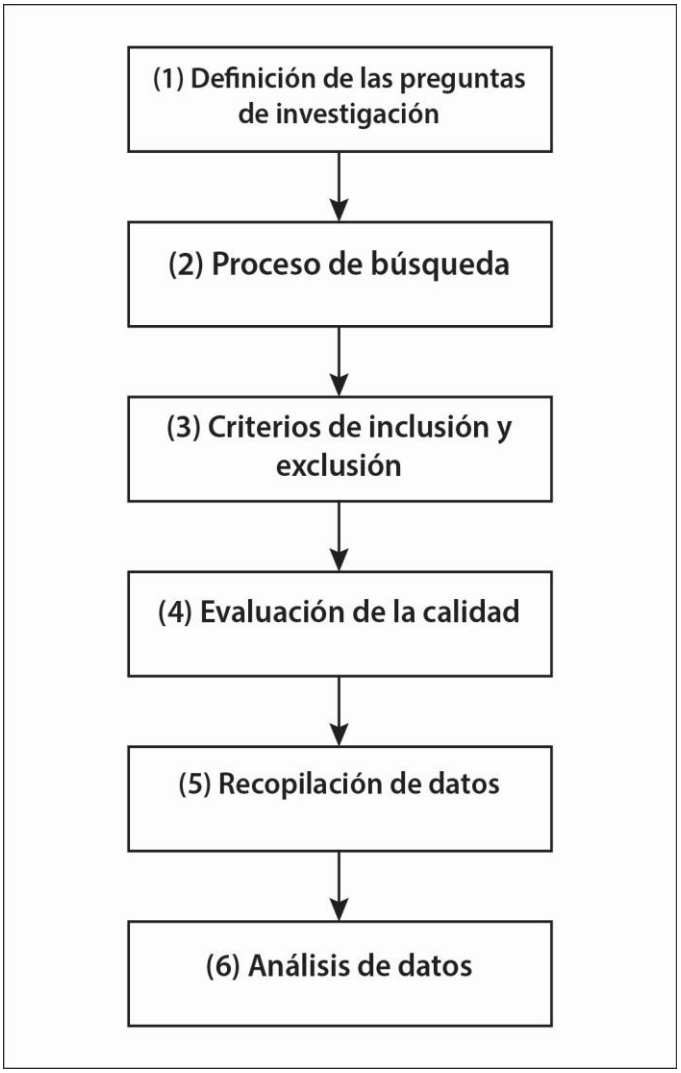

Figura 1: Etapas del proceso de una Revisión Sistemática de la Literatura propuestos por Kitchenham [Kit07].

En esta etapa el investigador debe extraer la metadata de cada artículo para su posterior análisis. Sin embargo, en el caso de que las bibliotecas no posean una forma automática de exportar la metadata, el investigador deberá realizar este proceso de forma manual lo cual es una tarea engorrosa, debido al esfuerzo y el tiempo que deberá invertir en ella. Este proceso manual de ingreso de cada cadena de búsqueda, en cada biblioteca digital definida como fuente de datos, puede superar las capacidades de una persona. Por esto puede resultar recomendable hacer esta etapa con el apoyo de una herramienta de software.

En la etapa identificada como “criterios de inclusión y exclusión” (etapa 3) se inicia con la eliminación de los resultados que son obviamente irrelevantes, o que se encuentran duplicados en los resultados. Esto se realiza entre varios investigadores quienes analizan de forma rápida el contenido de los documentos, y determinan si vale la pena incluirlos o excluirlos del análisis posterior.

En la etapa “evaluación de la calidad” (etapa 4) el objetivo es tratar de valorar si los resultados obtenidos son relevantes para el tema de investigación, o si se ha dejado por fuera algún estudio que tenga relevancia para responder las preguntas de investigación planteadas.

La etapa llamada “recopilación de datos” (etapa 5) se refiere a la extracción de la metadata de los estudios primarios, donde el investigador extrae los resultados de las búsquedas y clasifica los artículos por título, autor y otros campos que él considere pertinentes. De esa manera es más factible manejar la información extraída de las bibliotecas, que a veces puede ser muy voluminosa.

En el “análisis de datos” (etapa 6), el investigador aplica criterios cualitativos principalmente semánticos a cada estudio y realiza una depuración más rigurosa, dando como resultado un listado de estudios que tienen una fuerte relación con la investigación, para finalmente evaluar la calidad del proceso realizado.

Cada una de las etapas del proceso descrito requiere un esfuerzo no menor de parte del investigador además del tiempo y concentración en la manipulación de una gran cantidad de información.

2.2 Fuentes de Información

En cuanto a las fuentes de información (bibliotecas digitales), se analizaron la mayoría de las que tienen mayor relevancia y que son frecuentemente utilizadas para realizar SLR en el área de las ciencias de la computación [Kit06] [Bri14], ya que es el dominio de aplicación escogido para este proyecto. Las características que se analizaron de estas fuentes de información fueron las siguientes:

- *Existencia de una API*: Se refiere a que si la biblioteca digital proporciona algún mecanismo para extraer la metadata de los artículos que almacena mediante una API.

- *Protocolo de acceso*: En el caso que la biblioteca tuviese una API, se analizó la forma de acceder a la misma ya sea utilizando el protocolo SOAP, o solicitudes HTTP, entre otros.

- *Limitantes en la metadata*: En el caso que la herramienta tuviese API, se analizó si existe algún tipo de limitantes que la biblioteca tenga, por ejemplo, si limita la cantidad de resultados que se puedan extraer.

- *Formato de los resultados*: Se refiere al tipo de formato en que la API retorna la metadata de los artículos resultantes de una búsqueda. Por ejemplo, estos resultados podrían entregarse en formato JSON o XML.

- *Tipos de operadores soportados*: Se refiere a los operadores lógicos que soporta la biblioteca digital para formar las cadenas de búsqueda, por ejemplo, usando AND, OR, y NOT, entre otros.

- *Metadata*: Se refiere a la metadata que la biblioteca proporciona para apoyar los procesos de búsqueda de información, por ejemplo: título, resumen y palabras claves de los artículos almacenados.

- *Campos de búsqueda*: Se refiere al componente a través del cual la biblioteca digital permite realizar las búsquedas, por ejemplo, en título, resumen y palabras claves, entre otros.

- *Búsqueda por tipo*: Indica si se puede limitar las búsquedas por tipo de artículo, como por ejemplo, capítulos de libros, artículos de conferencia o de revista, entre otros.

Tabla 1: Características de las Bibliotecas Digitales Consideradas (información consultada en diciembre 2018).

Fuente	**API**	**Tipo de acceso**	**Limitantes metadata**	**Resultados**	**Búsqueda por tipo**	**Operadores**
IEEE	Sí	HTTP	Sí	XML	No	Sí
ScienceDirect	Sí	HTTP	No	JSON	No	Sí
Web of Science	Sí	SOAP	Sí	XML	No	Sí
ACM Digital Library	No	N/A	N/A	N/A	N/A	N/A
Springer	Sí	HTTP	No	JSON	Sí	Sí
DBLP	Sí	HTTP	No	XML	No	Sí

En la tabla 1 se muestra una tabla resumen con las principales características de las bibliotecas consideradas.

Para verificar la completitud de DBLP se realizó un experimento que consistió, en primera instancia, en buscar en DBLP todos los artículos recuperados en la RSL reportada en [Ver18], que involucró 95.715 artículos (en la sección 3.2 se da más detalles del mismo). Se observó que los artículos fueron encontrados en DBLP, exceptuando algunos capítulos de libros publicados por Springer. Por esa razón en este trabajo de tesis se tomó a DBLP como base para realizar estas búsquedas, y al resto de las bibliotecas digitales como fuente de información extra para completar el espacio de datos.

2.3 Herramientas de Apoyo a las RSL

En la actualidad existen herramientas que apoyan distintas partes del proceso de revisión sistemática de la literatura. Algunas de estas herramientas están orientadas a áreas específicas, como por ejemplo, la medicina, para la cual existen varias estrategias que tratan de apoyar este tipo de revisiones. Debido a eso, puede llegar a ser complicado para un investigador el elegir una herramienta que pueda cumplir sus expectativas.

En muchos casos el investigador también deberá dedicar tiempo a buscar una aplicación que le permita automatizar parte del proceso. Sin embargo, no siempre podrá encontrar alguna que se ajuste a sus necesidades, ya que usualmente estas herramientas sólo automatizan la extracción de la información de algunas pocas bibliotecas digitales, por lo que es posible que el investigador decida realizar el proceso de forma manual. A pesar que no existe una clasificación oficial (o reconocida) de herramientas de apoyo a las RSL, existe una aplicación llamada SRToolBox [Fer10] que las clasifica de acuerdo a una serie de características. Dichas características se describen a continuación:

- *Desarrollo de Protocolo*: Es la capacidad de las herramientas en recopilar y almacenar, de parte del investigador, detalles en cuanto a la investigación.

- *Búsqueda Automatizada*: Es la capacidad de la aplicación de realizar de forma automática búsquedas en las diferentes bibliotecas digitales, a partir de las cadenas de búsqueda definida. En algunos casos estas aplicaciones pueden requerir una entrada, como por ejemplo, un archivo en algún formato específico indicado por la biblioteca digital.

- *Selección de Estudios*: Tiene que ver con la funcionalidad que provee una herramienta para seleccionar algún estudio en particular por parte del investigador, para su posterior evaluación, ya sea buscando en el título o en algún otro atributo de la metadata.

- *Evaluación de la Calidad*: Es la capacidad que tiene una herramienta de verificar y definir criterios de calidad por parte del investigador.

- *Extracción de Datos*: Es la capacidad de una herramienta en particular de extraer en forma automática la metadata de las diferentes bibliotecas digitales.

- *Análisis de Texto*: Es la facultad de poder realizar análisis estadístico sobre el cuerpo del documento.

- *Meta Análisis*: Es la capacidad de poder realizar análisis estadístico sobre la metadata de los documentos.

- *Generación de Informes*: Indica la habilidad de la herramienta para generar informes análisis realizado por la misma.

- *Colaboración*: La capacidad de la herramienta para permitir que cada revisión sistemática pueda ser realizada por más de una persona a través de un proceso de colaboración.

- *Gestión de Documentos*: La capacidad que tiene una herramienta para que los usuarios puedan gestionar y manipular los datos extraídos o producidos durante la revisión sistemática.

Para efectos prácticos, en este documento se le dará mayor énfasis a las herramientas que cuentan con las capacidades Búsqueda Automatizada y Extracción de Datos. Esto debido a que la solución propuesta se enfoca en estas características principalmente, las cuales tienen una relevancia muy alta en la complejidad de una RSL, y en el tiempo que debe invertir un investigador en este proceso.

Para determinar la forma en que las herramientas actuales apoyan las revisiones sistemáticas, se realizó una comparación de ellas, de acuerdo a la información que se presenta en sus respectivos sitios Web. En la evaluación y calificación de herramientas, dentro de este proyecto se verificaron las siguientes características:

- *Tipo de aplicación*: Describe el tipo de aplicación de la herramienta; por ejemplo, sistema Web o aplicación cliente.

- *Metadatos*: Representan los datos principales que se extraen, como por ejemplo, título, autores, año de publicación, DOI, resumen, y palabras claves.

- *Fuentes de información*: Representan las bibliotecas digitales desde donde se extrae información.

- *Ranking*: Este es un valor que calcula la herramienta para puntuar la relevancia de los artículos según los criterios de búsqueda utilizados.

- *Definición de proyecto*: La capacidad de la herramienta de poder definir un proyecto por cada revisión sistemática.

- *Colaboración*: La capacidad de una herramienta para poder permitir que varios usuarios trabajen simultáneamente en la misma revisión sistemática.

- *Almacenamiento de preguntas*: El apoyo que brinda la herramienta para el almacenamiento de las preguntas de investigación a responder.

- *Generación de cadenas de búsqueda*: La capacidad de generar diversas cadenas de búsquedas a partir de los términos extraídos de las preguntas de investigación, así como sus sinónimos, para luego aplicar dichas cadenas a las diferentes bibliotecas digitales.

- *Uso o descarga libre de la herramienta*: Tiene que ver con la posibilidad de utilizar la herramienta de forma gratuita.

- *Soporte*: Tiene que ver con la cantidad de soporte que se le da a la herramienta.

Como se muestra en la Tabla 2, estas herramientas se especializan principalmente en extraer la información de unas pocas bibliotecas digitales, y se enfocan en un área de investigación en específica, con excepción StArt -Tool que en su documentación dice ser compatible con una gran variedad de estas bibliotecas digitales. Sin embargo, dicha herramienta le traslada el trabajo al investigador, para que él cree y adapte las cadenas de búsqueda, y así poder extraer de cada biblioteca digital un archivo BibTe y cargarlo a la aplicación para que la herramienta pueda acceder a la metadata.

En la Tabla 2 también se puede apreciar que, en el caso de las herramientas orientadas a la medicina (Abstrackr, Rayyan, Textpresso entre otras), estas tienen como principal fuente la biblioteca Pubmed, mientras que en las del área informática se evidencia una gran variedad de bibliotecas digitales a las que se orientan las herramientas.

Tabla 2. Comparación entre herramientas y las bibliotecas digitales asociadas.

#	Herramientas & Características	Tipo Aplicación	Bibliotecas							
			PubMed	IEEEXplore	ACM	Science Direct	Springer	Web of Science	Scopus	DBLP
1	StArt -Tool [Her12]	Escritorio	Sí	Sí	Sí	Sí	Sí	Sí	Sí	No
2	RSL-tool [Fer10]	Escritorio	No	Sí	Sí	No	No	No	No	No
3	Abstrackr (Beta) [ABS18]	Web	Sí	No	No	No	No	No	No	No
4	Rayyan [RAY18]	Web	Sí	No	No	No	No	No	No	No
5	SESRA [SES18]	Web	No	Sí	No	No	Sí	No	No	No
6	Textpresso [TEXT18]	app Cliente Servidor/ Web	Sí	No	No	No	No	No	No	No
7	SRDB.PRO [SRD18]	SaaS	Sí	No	No	No	No	No	No	No
8	Parsifal [PAR18]	Web	No	No	No	Sí	No	No	Sí	No
9	DoctorEvidence [DOC18]	Web	Sí	No	No	No	No	No	No	No
10	Colandr [COL18]	Web	No	No	No	No	No	No	Sí	No

En la Tabla 3 se puede apreciar que las herramientas tienen enfoques diversos y tratan de abordar el proceso de forma distinta. En su mayoría el soporte y la evolución de cada una es poca o nula, y en muchos casos intentan llevar un registro parcial del proyecto (RSL), pero no acompañan totalmente al investigador en la generación de las preguntas de investigación, y tampoco ayudan al investigador a adaptar las cadenas de búsquedas para que puedan ser utilizadas apropiadamente en las diferentes bibliotecas digitales.

Otro factor importante para la comparación de estas herramientas, es su apoyo a los investigadores en la selección y clasificación de los documentos, ya que esta tarea puede volverse engorrosa debido a la gran cantidad de estudios (generalmente en el orden de miles) que pueden estar relacionados con los términos de las cadenas de búsqueda. Por lo tanto, el investigador deberá aplicar una serie de criterios o preguntas a cada artículo, con el fin de evaluar la relevancia del mismo con respecto a las preguntas de investigación enunciadas. Al observar los resultados de las diferentes herramientas se puede percibir que la mayoría no le da mucha importancia a esta funcionalidad, y por lo tanto le pasan esta tarea al investigador.

Tabla 3. Comparación entre herramientas y características asociadas.

#	Herramientas & enlace	Ranking	Definición del proyecto	Colaboración	Alamcenamiento de preguntas	Generación de cadenas de búsqueda	Uso libre	Soporte
1	StArt -Tool	Sí	Sí	No	No	No	Sí	Bajo
2	RSL-tool	No	No	No	No	Sí	Sí	Ninguno
3	Abstrackr (Beta)	No	Sí	Sí	No	No	Sí	Medio
4	Rayyan	No	Sí	Sí	No	No	Sí	Medio
5	SESRA	No	Sí	Sí	Sí	Sí	Sí	Medio
6	Textpresso	No	Sí	No	Sí	Sí	Sí	Medio
7	SRDB.PRO	No	Sí	Sí	Sí	Sí	No	Alto
8	Parsifal	No	Sí	Sí	No	Sí	Sí	Medio
9	DoctorEvidence	No	Sí	Sí	No	Sí	No	Alto
10	Colandr	No	Sí	Sí	Sí	Sí	Sí	Medio

2.4 Resumen

Una RSL tiene como objetivo obtener los estudios primarios más relevantes relacionados con una o más preguntas de investigación, y para ello se deben cumplir las etapas de la Figura 1. En cuanto a las fuentes de información, es importante destacar a DBLP como una fuente con una buena completitud en cuanto a artículos que refieren a tecnologías de información. En el caso de las herramientas que prestan apoyo a las RSL, luego de analizarlas se puede observar que todas tienen una cantidad significativa de limitantes las cuales en ocasiones le generan trabajo adicional al investigador. En el Capítulo 5 se expone una nueva definición del proceso de RSL para adaptarlo a la herramienta el cual está basado en el propuesto por Kitchenham, y se presenta además la concepción de la solución propuesta en este trabajo de tesis.

3. Concepción de la Solución

En este capítulo se explican los componentes básicos de la solución planteada, incluyendo el proceso de RSL a apoyar, las fuentes de información consideradas, la estructura del escenario de interacción del sistema con otros componentes externos, y los principales requisitos de dicho sistema.

3.1 Definición del Proceso RSL

Como parte del proyecto se definió el proceso para realizar una revisión sistemática de la literatura, al cual la aplicación dará apoyo. La Figura 2 muestra dicho proceso.

Tal como se mencionó antes, este proceso está basado en la propuesta de Kitchenham [Kit07]. La principal diferencia entre el proceso de Kitchenham y el propuesto para la solución planteada, es que el primero está pensado para realizarlo de forma manual, por lo que refuerza más las etapas de depuración e inclusión de resultados en diferentes niveles. En cambio, el segundo se enfoca más en las actividades a automatizar, para así facilitar parte de la extracción y clasificación de los resultados con el fin de apoyar la depuración final.

El proceso propuesto para realizar una revisión sistemática de la literatura se inicia con la “definición de las preguntas de investigación” (etapa 1), a partir de las cuales se procede a la “identificación del conjunto de términos (o palabras) claves” (etapa 2). Estos términos están asociados al conocimiento que se desea obtener para poder responder las preguntas de investigación. Una vez definidos los términos, éstos son combinados automáticamente por el sistema para obtener el conjunto de cadenas de búsquedas, lo cual se lleva a cabo en la “definición de las cadenas de búsqueda” (etapa 3). Estas cadenas son luego usadas por el sistema para realizar la búsqueda de los estudios primarios, por lo que es fundamental que se generen todas las posibles combinaciones de ella, estas son generadas por el sistema de forma automática a partir de los términos definidos en la etapa 2.

Para generar las combinaciones de palabras claves, se utilizan los operadores lógicos AND y OR para conectarlas, y obtener así las diferentes cadenas de búsqueda. Estas cadenas son la base de la consulta que el sistema genera para aplicar a la base de datos de estudios sobre la que opera el sistema.

En la siguiente etapa, llamada “aplicación de las cadenas de búsqueda a las bibliotecas digitales” (etapa 4), el sistema aplica las cadenas definidas sobre las fuentes de datos; o sea, realiza las búsquedas. En nuestro caso estas fuentes de datos son dos: la base de datos local (que es una versión enriquecida de la base de datos de DBLP), y los capítulos de libros de Springer (que no se encuentran en DBLP) que se extraen directamente de la

biblioteca digital de dicha editorial. Como resultado de la búsqueda se obtiene una gran cantidad los estudios relacionados con las preguntas de investigación, junto con otros estudios que no necesariamente son relevantes para responder dichas preguntas.

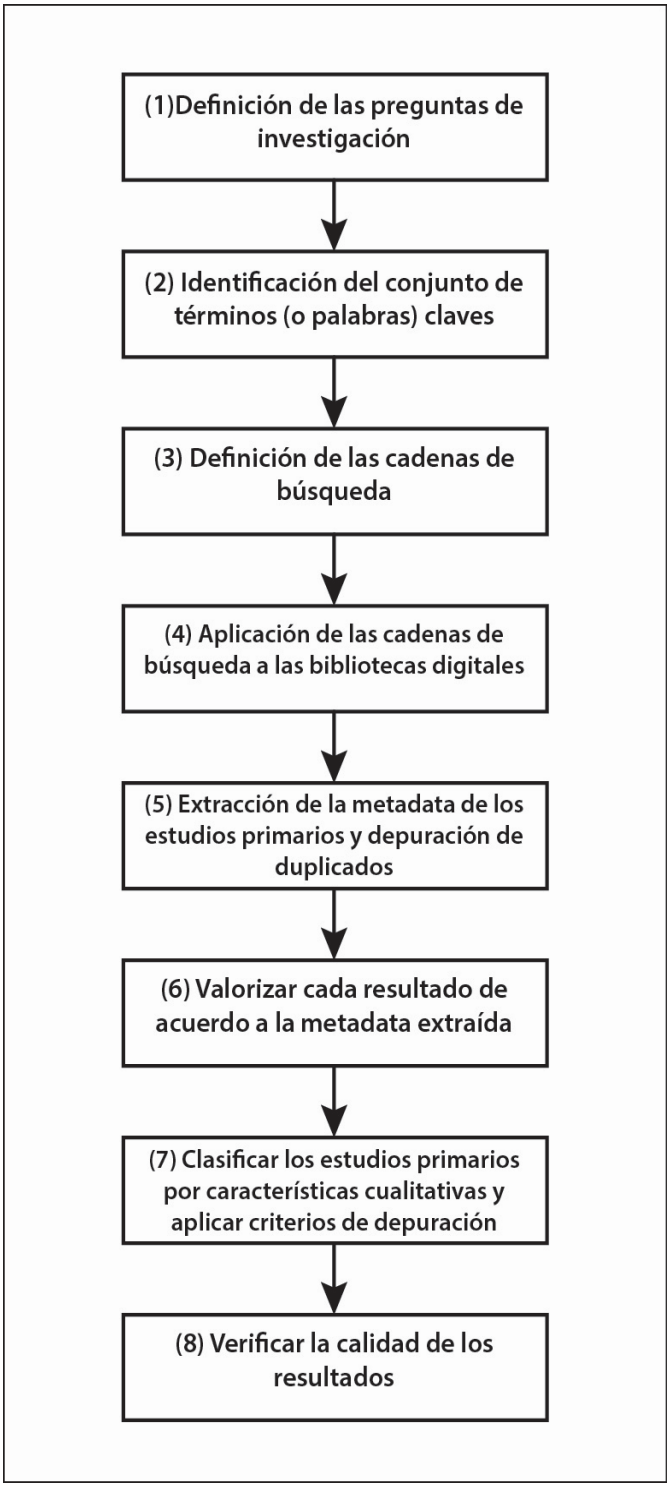

Figura 2: Etapas del proceso de una Revisión Sistemática de la Literatura propuesto.

En la siguiente etapa, denominada “extracción de la metadata de los estudios primarios y depuración de duplicados” (etapa 5), el sistema toma los resultados de las búsquedas y clasifica los artículos por título, autor y otros campos relevantes para realizar luego la selección de estudios. De esa manera es más factible manejar la información extraída de las bibliotecas. La metadata de los artículos puede llegar a ser mucha, como por ejemplo el título y el DOI, también sirven como insumo para ayudar a eliminar información duplicada, obtenida como resultados de las diferentes búsquedas.

La etapa “valorizar cada resultado de acuerdo a la metadata extraída” (etapa 6) tiene como fin categorizar los artículos extraídos, de acuerdo a su relación con los términos de búsqueda definidos previamente. De esa manera, se facilita la selección de aquellos estudios que están altamente relacionados con las preguntas de investigación que se pretenden responder. Para lo cual el sistema aplica una serie de cálculos sobre la metadata de cada artículo para generar esta valorización.

La etapa 7, llamada “clasificación de los estudios primarios por características cualitativas, y aplicación de criterios de depuración”, es realizada en forma manual por el investigador, siguiendo la recomendación de Kitchenham [Kit07]. Si bien la lógica de esta etapa no cambió respecto a la recomendación antes mencionada, la herramienta de apoyo a las RSL desarrollada en esta tesis facilita dicha actividad. Por ejemplo, permite el trabajo colaborativo de los investigadores participantes en la RSL, quienes pueden trabajar simultáneamente en determinar cuáles estudios serán usados para el análisis final, y cuáles no. La coordinación de las acciones individuales de los participantes es manejada por el sistema, el cual les entrega retroalimentación permanente (awareness) a los participantes acerca del estado actual de la RSL.

La etapa 8, llamada “verificación de la calidad de los resultados”, también se realiza en forma manual y corresponde a la misma actividad que propone Kitchenham. En esta actividad los investigadores verifican que los estudios extraídos contengan evidencia (resultados válidos) que respalden dicho estudio, la cual podrá ser luego utilizada para tratar de responder las preguntas de investigación formuladas en la etapa 1. Si bien esta actividad de verificación se realiza en forma manual, el sistema desarrollado provee el DOI de cada artículo y un link para que los investigadores puedan acceder directamente a los mismos, siempre que tengan una suscripción vigente (personal o institucional) para acceder a la biblioteca digital que los almacena; o sea, a la fuente de información.

3.2 Fuentes de Información

Como se mencionó en el marco teórico, se analizaron las fuentes de información (bibliotecas digitales) más relevantes en el área de las ciencias de la computación: ACM Digital Library, IEEEXplore Digital Library, ScienceDirect, Web of Science y Springer. Como resultado de este análisis se observaron principalmente dos cosas que están

interrelacionadas; en primer lugar, cada biblioteca digital contiene un subconjunto del espacio de datos sobre el que un investigador quisiera realizar sus RSL. Por lo tanto, para realizar una RSL el investigador debería buscar información en todas esas fuentes, y luego unificar y depurar los resultados para poder llevar a cabo la actividad.

En segundo lugar, se observó que cada fuente de datos tiene diferencias (y muchas veces limitaciones) para realizar las búsquedas sobre ellas. Estas diferencias van desde la sintaxis y capacidad de expresión del lenguaje de consulta que se puede utilizar para recuperar los estudios almacenados en una biblioteca digital, hasta los datos y el formato de las respuestas que devuelven los buscadores. En muchos casos la información resultante de una búsqueda es una parte de lo que un investigador querría obtener, por lo tanto, él o ella deberá buscar el resto de la información a mano y agregarla a su repositorio de información local para poder realizar así las etapas de selección de artículos consideradas en las RSL. Como se mencionó antes, ambas observaciones están estrechamente relacionadas, por lo que la solución a una de ellas debe considerar la solución a la otra.

Para dar solución al primer aspecto, o sea a la incompletitud de las bibliotecas digitales primarias (es decir, ACM, IEEE, etc.), se estudió la posibilidad de utilizar un repositorio secundario de publicaciones, particularmente DBLP, el cual ya integra información de las bibliotecas digitales primarias. Para poder determinar qué tan apropiado era basarse en DBLP, había que establecer la diferencia entre el contenido de DBLP, y la suma de los contenidos en las bibliotecas digitales primarias. Para ello se realizó un experimento que consistió, en primera instancia, en buscar en DBLP todos los artículos recuperados en la RSL reportada en [Ver18], que involucró 95.715 artículos. El estudio tomado como referencia utilizó sólo las bibliotecas digitales primarias.

El objetivo de este experimento era buscar patrones de inclusión o exclusión de artículos entre DBLP y las bibliotecas digitales primarias. Como resultado de este análisis se vio que todos los artículos que estaban en fuentes primarias también estaban en DBLP, excepto los capítulos de libros y los libros de la biblioteca de la editorial Springer. Este resultado permitió reducir significativamente el problema original, viendo que el espacio de búsqueda podía realizarse sobre la base de datos de DBLP, más la biblioteca digital de Springer. Otro punto importante es que los datos de DBLP son de acceso público y descargables de forma gratuita, lo que permite bajarlos a una base de datos local y manejarlos de manera apropiada para llevar a cabo las RSL. Por lo tanto, se decidió tomar como base esa solución para llevar a cabo el proceso de RSL propuesto.

Una limitante de esta solución es que la metadata de los artículos almacenados en DBLP no contienen información como el resumen o puede no estar completa, lo cual afecta los resultados de una RSL. Por lo tanto, se decidió enriquecer la información de DBLP con metadatos obtenidos desde otras fuentes de información como Mendeley. Por lo tanto, el

sistema de apoyo a este proceso debía interactuar de manera automática con dichas fuentes de datos para intentar recuperar los metadatos faltantes.

Una opción para recuperar los metadatos faltantes era realizar Web scrapping, lo cual requiere que la solución sea capaz de realizar las búsquedas en las diferentes bibliotecas digitales, y luego capturar la metadata desde la página Web de cada artículo, parseando el código HTML de cada página. El inconveniente que presenta este método es que es vulnerable a cualquier cambio en la estructura del código HTML de las páginas Web de las bibliotecas digitales. Además, el tiempo que involucran las búsquedas usando este método es relativamente alto.

Otra opción para recuperar los metadatos era proporcionarle a la aplicación la capacidad de comunicarse con las APIs disponibles de las bibliotecas digitales. Esto conlleva a que cuando el sistema realice las búsquedas, éstas puedan tener un tiempo de espera que depende de las diversas bibliotecas digitales.

Luego de realizar varias pruebas con las diferentes APIs disponibles, se escogió por un tema de velocidad de respuesta y simplicidad en la interacción, utilizar la API de Mendeley [Men18] para obtener algunos de los metadatos que no estaban en DBLP; particularmente, el resumen y las palabras claves de los artículos almacenados en la base de datos local.

3.3 Concepción Inicial de la Solución

Como un primer paso para desarrollar la solución Web que apoyaría el proceso de RSL, se decidió que la aplicación consultara la información contenida en una base de datos local, la cual sería una réplica de DBLP (fuente externa que alimenta la base de datos local mediante ETL). Esta base de datos sería extendida con la información de los capítulos de libros obtenidos de Springer (fuente externa que es consultada por la aplicación web), tal como lo muestran las flechas que indican el flujo de control en la Figura 3. Dicha información sería además enriquecida con la metadata extraída desde fuentes externas como Mendeley [Men18].

La solución propuesta plantea la utilización de un proceso tipo ETL (extracción, transformación y carga) para alimentar la base de datos local. Como se mencionó antes, esta base de datos es, en primera instancia, una réplica de DBLP. El proceso ETL debe descargar desde su fuente de datos, un archivo XML con los artículos allí reportados, y luego los carga en una base de datos documental local.

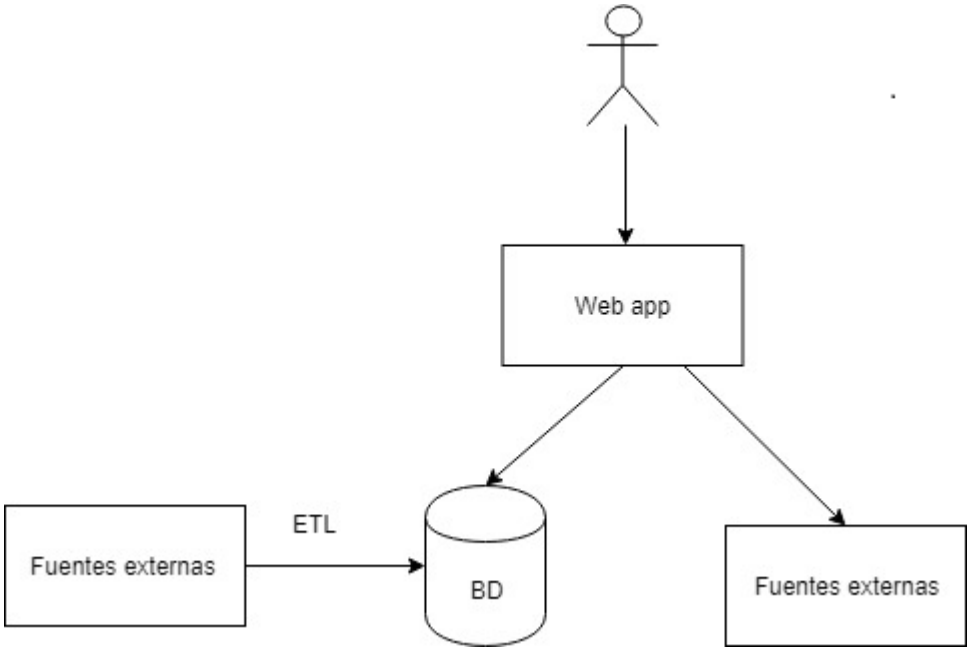

Figura 3: Diagrama estructural del escenario de funcionamiento de la solución propuesta (las flechas indican el flujo de control)

Otra alternativa para llevar a cabo este proceso es hacerlo por demanda, es decir, que en el momento que los investigadores realicen una búsqueda de artículos, la aplicación consulte las bases de datos correspondientes y actualice la copia local con dicha información. Sin embargo, la obtención de los resultados de las búsquedas utilizando dicha información podría tomar bastante tiempo, representando un potencial obstáculo para los investigadores que realizan la RSL. Por esa razón se decidió pensar la solución en los términos planteados en la primera opción.

3.4 Principales Requisitos de la Solución

A continuación se describen brevemente los principales requisitos de la solución, considerando las etapas del proceso de RSL antes indicadas, y las limitaciones actuales para realizar este proceso de forma manual, o apoyado con las herramientas indicadas en el Capítulo 2. Los requisitos también consideran la decisión de diseño estructural de la solución que se describió en la Sección 3.3.

Nombre: *R01 – Autenticación de usuarios*

Descripción: La aplicación debe permitir el acceso de los usuarios por medio de un enlace (link) que posee un token generado por el sistema (similar al que genera la aplicación Overleaf). Este enlace será enviado al

correo electrónico del investigador correspondiente al momento de registrarse.

El usuario también recibirá un enlace de acceso a su correo electrónico con un token cada vez que sea agregado a un proyecto. Dicho enlace lo conducirá directamente a la RSL en cuestión.

Nombre: *R02 - Creación y participación colaborativa en una RSL*

Descripción: La aplicación debe permitir la creación y participación colaborativa de los usuarios en diversas revisiones sistemáticas de la literatura. En cada RSL los investigadores deben poder definir lo siguiente:

- Título y descripción de la RSL.
- Participantes que colaboran en dicho estudio.
- Preguntas de investigación.
- Términos de búsqueda.

Nombre: *R03 - Búsqueda de artículos*

Descripción: El sistema debe permitir generar las cadenas de búsqueda a partir de los términos o palabras claves definidas por los investigadores, y extraer de la base de datos local todos los artículos que coincidan con dichas cadenas. Además, deberá hacer uso de la API de Springer para extraer los capítulos de libros que coincidan con las cadenas de búsqueda, y luego unificar ambos resultados.

Nombre: *R04 - Calcular la coincidencia de cada artículo*

Descripción: Para el caso de la búsqueda dentro de la base de datos local, se deberá crear un indicador que determine el grado de coincidencia de los artículos, con respecto a los términos de las cadenas de búsqueda. Los resultados serán mostrados a través de diferentes categorías, considerando primero a aquellos estudios que contienen todos los términos de las cadenas de búsqueda en su metadata. A continuación de esos resultados se presentarán al usuario, aquellos artículos que coinciden con todos los términos menos uno, y finalmente los que contienen todos los términos menos dos. Esto se hace con el fin de mitigar el riesgo que algún artículo relevante para responder la(s) pregunta(s) de investigación, quede fuera debido a diversas situaciones, como por ejemplo:

- Una pobre o incorrecta definición de los términos de búsqueda por parte de los investigadores involucrados en la RSL.
- Una definición poco representativa de la temática del artículo, en el título, palabras claves o resumen del mismo.
- Que la metadata de los artículos del repositorio local se encuentre incompleta.

Nombre: *R05 - Valorizar la relevancia de los artículos extraídos*

Descripción: La aplicación debe realizar una valorización de los artículos extraídos de acuerdo a su coincidencia con los términos (palabras claves) utilizados en la búsqueda. De esa manera se ayuda a los investigadores a depurar los resultados de las mismas. Para ello se deberá tomar en cuenta la coincidencia de los términos en el título, el resumen y en la sección de palabras claves de cada artículo almacenado en la base de datos del sistema.

Nombre: *R06 - Apoyo en la selección de artículos*

Descripción: La aplicación debe dar la posibilidad de que los investigadores puedan determinar la inclusión o exclusión de artículos de forma colaborativa, de acuerdo a características cualitativas de los estudios. Para esto se deberán seguir las siguientes reglas:

- Durante la etapa de selección, cada artículo puede tomar el estado incluido, pendiente o excluido, lo cual es indicado por los investigadores (usuarios del sistema).
- Los investigadores sólo podrán indicar su opinión acerca de la inclusión o exclusión de aquellos artículos que se encuentren en estado “pendiente”.
- Para que un artículo sea incluido en la etapa de selección, es necesario que al menos dos investigadores voten a favor de incluirlo en el estudio; o sea, en la selección de artículos potencialmente relevantes, que requieren un análisis más detallado. Esto es parte de las recomendaciones que se indican en el proceso propuesto por Kitchenham [Kit07].

- Para que un artículo sea excluido en la etapa de selección, es necesario que al menos dos investigadores voten a favor de excluirlo. Al igual que en el punto anterior, esta forma de proceder es también parte de las recomendaciones del proceso de Kitchenham.

Nombre: *R07 - Estados de la RSL*

Descripción: La aplicación deberá manejar tres posibles estados para cada RSL:

- *Definición*: Desde que se crea la RSL, ésta permanece en ese estado mientras se define su título, descripción, preguntas de investigación, y los términos o palabras claves de las cadenas de búsqueda.
- *Selección*: Una RSL estará en este estado desde que se inicia la búsqueda de artículos, e inclusive durante la inclusión o exclusión de los mismos, hasta que todos los investigadores participantes en la RSL deciden terminar esta etapa.
- *Cerrado*: Una RSL estará en este estado cuando se termina la etapa de selección de estudios; allí se muestran únicamente los artículos incluidos. Además, en esta etapa los investigadores pueden realizar comentarios sobre cada artículo.

Nombre: *R08 - Apoyo al proceso ETL usando los datos de DBLP*

Descripción: El sistema debe proporcionar un mecanismo de extracción, transformación y carga de información desde la base de datos de DBLP, hacia la base de datos local. Este proceso se deberá realizar de forma periódica y automática, y deberá evitar recargar la información ya cargada y que no fue modificada en DBLP.

Nombre: *R09 - Completar metadata con fuentes externas*

Descripción: De forma periódica, el sistema deberá completar la metadata de los artículos de la base de datos local utilizando otras fuentes de información, como por ejemplo Mendeley. Esta tarea deberá iniciar con los artículos de los años más recientes, y marcándolos con un campo (flag) que determine si el artículo fue “encontrado”, “no

encontrado” o “no es posible recuperar su metadata”, por ejemplo, por falta de un DOI. Para completar la metadata de los artículos se definieron dos estrategias complementarias:

- Cargar una lista de artículos de la base de datos local, cuya metadata nunca ha sido buscada en Mendeley por parte del sistema.
- Cargar una lista con los artículos cuya metadata ya ha sido buscada en Mendeley [Men18], pero que aún tiene metadatos incompletos. Dado que la información de Mendely se actualiza periódicamente, realizar más de una búsqueda de los metadatos de un artículo permite eventualmente completar la información de la base de datos local, con la información de dichas actualizaciones.

Nombre: *R10 - Registro de carga de artículos*

Descripción: Cuando se realice la carga de artículos desde DBLP se deberá dejar un registro de la cantidad de artículos que fueron agregados y el tiempo en el que transcurrió la carga, de la misma manera cuando se realice la búsqueda de metadata en Mendeley se deberá dejar un registro del tiempo en que transcurrido el proceso, cuantos artículos fueron encontrados en Mendeley, cuantos no se pudieron encontrar y cuantos no fue posible buscarlos.

3.5 Perfiles de Usuario del Sistema

En la Tabla 4 muestra los tipos de usuarios del sistema, y los principales servicios que les ofrece la aplicación.

Tabla 4: Tipos de usuario soportados en el sistema

Tipo de usuario	Descripción
Investigadores	Son los usuarios principales del sistema. Interactúan con éste para crear nuevas revisiones sistemáticas de la literatura, crear grupos de trabajo para cada RSL, ingresar las preguntas de investigación, definir los grupos de términos o palabras claves, seleccionar y clasificar los resultados de las búsquedas.
Usuario administrador	El administrador es un usuario técnico, el cual debe tener la capacidad de dar soporte al sistema en caso de ser necesario. Por lo tanto, este tipo de usuario debe tener conocimientos en bases de datos documentales, servicios REST y aplicaciones Web.

3.6 Resumen

La definición del proceso para el desarrollo de una RSL planteado en este capítulo está basado en la definición de Kitchenhan, pero adaptado para que pueda ser implementado en la aplicación propuesta. Luego del análisis y pruebas de las diferentes fuentes de información que existen para el área de la computación, se decidió replicar de forma local la base de datos de DBLP y enriquecer la metadata de los artículos desde fuentes como Mendeley. Además, se complementaron los resultados almacenados en la base de datos local (o sea, aquellos extraidos por la solución desarrollada), con los capítulos de libros de la editorial Springer, los cuales fueron recuperados a través de su API.

En cuanto a la concepción inicial de la solución propuesta, se determinó que fuera una aplicación Web que cuente con un proceso ETL (Extraction, Transformation and Load) que le permite actualizar los artículos y la metadata de los mismos de manera batch. Se definieron además una serie de requisitos funcionales, los cuales fueron verificados por algunos usuarios de interés en la herramienta. En el siguiente capítulo se detalla el diseño de la aplicación, así como las estrategias utilizadas para que la aplicación entregue sus servicios al usuario.

4. Diseño del Sistema

En este capítulo se presenta el diseño de la solución, partiendo por su estructura y el modelo de datos utilizado. Luego se explican los mecanismos de ranking y completitud propuestos, y las tecnologías escogidas para la implementación del sistema.

4.1 Estructura del Sistema

En el flujo de control de la Figura 4 se puede observar la arquitectura propuesta para la solución, la cual está planteada en capas. Allí se puede ver que el sistema tiene una capa de presentación (front-end), una capa lógica conformada por un back-end, un proceso ETL para actualización automática de los datos, y una capa de datos conformada por una base de datos documental. A continuación se describen los principales componentes de la arquitectura:

- *Front-end*: Este componente representa a la aplicación Web con la que interactúan los usuarios que desean desarrollar una revisión sistemática de la literatura, y mediante ella se procesan y presentan los resultados. En el caso del acceso a la misma se tomó la decisión de flexibilizar el proceso de autenticación de usuarios en el sistema, por lo que existen dos formas de acceder:

  a. Cuando un usuario ingresa su correo electrónico en la página de registro, el sistema le envía un correo con una URL que le permite acceder al listado de sus proyectos. Esta URL permanece activa hasta que el usuario solicite una nueva.

  b. Una vez que el correo electrónico de un investigador esté registrado en el sistema, y el investigador es registrado como participante en una RSL por otro investigador, la aplicación le envía un correo con una URL que lo direcciona directamente al proyecto (RSL en el que fue registrado).

- *ETL (extracción, transformación y carga)*: Este componente permite a la aplicación comunicarse con las fuentes externas de información, extrayendo de forma periódica los metadatos y actualizando la base de datos local cuando corresponda.

- *Fuentes externas*: Son las fuentes de conocimiento externas desde donde el ETL extrae información para cargarla a la base de datos local de forma periódica. Ejemplo de fuentes externas son: DBLP y Mendeley [Men18].

- *Back-end*: Tiene entre sus principales tareas la comunicación entre la aplicación Web y la base de datos, creando las RSL, las búsquedas de los artículos coincidentes con las cadenas de búsqueda, realizando la autenticación en base a

tokens, generando un ranking de los artículos extraídos, y apoyando la selección de los artículos basada en las opiniones de los participantes.

- *Base de datos*: Está compuesta por una base de datos documental, particularmente ésta es MongoDB [Mon18], y un índice desarrollado con Apache Lucene [Luc18].

- *Springer API:* Esta es consultada en el momento de crear una RSL con el fin de complementar los resultados obtenidos de la base de datos con capítulos de libros desde Springer.

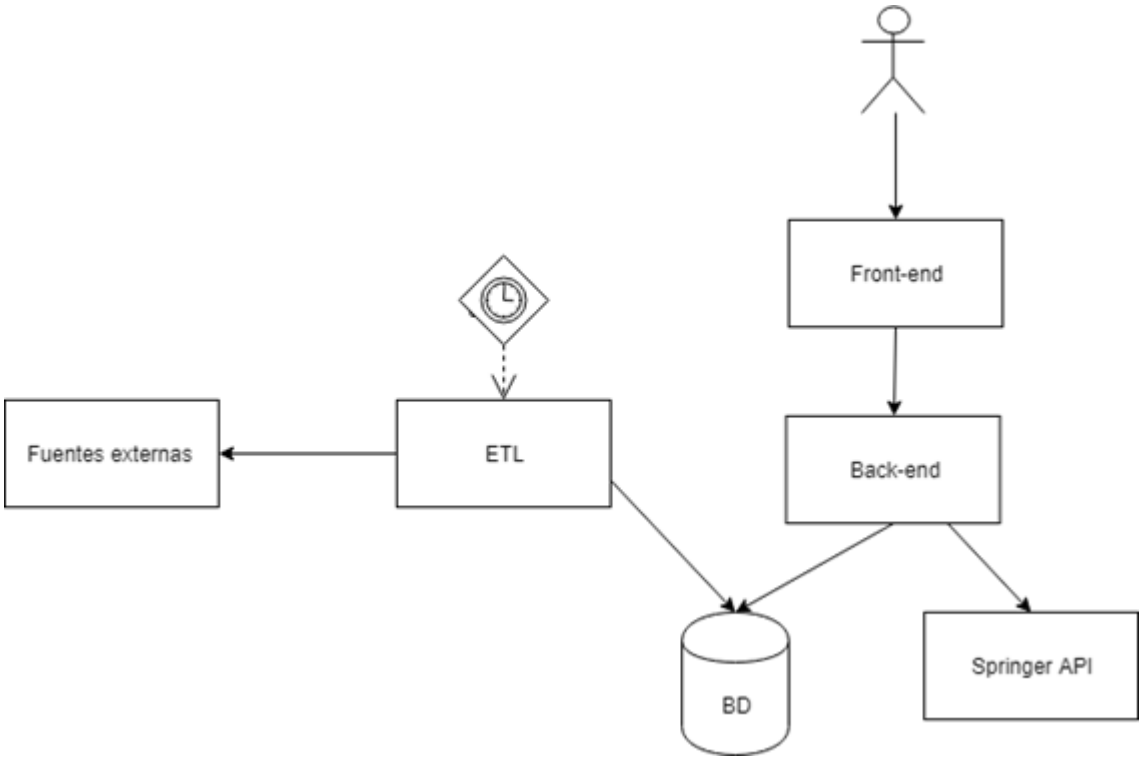

Figura 4: Arquitectura de la solución propuesta (las flechas indican el flujo de control)

Para la explicación de la arquitectura del proyecto se tomará como base la Figura 2, donde se expone el proceso propuesto para realizar una revisión sistemática de la literatura. Dicho proceso está estructurado en 8 etapas, las cuales se abordan en el diseño de la arquitectura.

Las etapas 1 y 2, es decir, la definición de las preguntas de investigación y la identificación de los términos (o palabras) claves, son gestionadas por el front-end del sistema, donde el usuario puede definir las preguntas de investigación y los términos de búsqueda. Luego el back-end gestiona el almacenamiento de la información en la base de datos.

En las etapas 3 y 4, que se refieren a la formación y aplicación de las cadenas de búsqueda sobre la base de datos local; o sea la del sistema. Para ello, el front-end envía los términos de búsqueda al back-end, el cual es el encargado de formar las cadenas de búsqueda y estructurar las consultas que enviará a la base de datos local y también a Springer a través de su API.

Para las etapas 5 y 6, que se refieren a la extracción de la data y a la valoración de la misma, el back-end aplica la consulta a la base de datos y genera una colección con todos los artículos que coincidieron con los términos de búsqueda. Luego verifica la metadata de los artículos para valorizarlos de acuerdo con su coincidencia con los términos y palabras claves. Más tarde, envía esta información al front-end para que los investigadores puedan aplicar los criterios de depuración, clasificar los estudios de acuerdo a características cualitativas, y finalmente verificar la calidad de los resultados, tal como se indica en las etapas 7 y 8 del proceso.

4.2 Estrategia ETL

En el flujo de datos de la Figura 5 se expone la arquitectura del proceso ETL (extracción, transformación y carga) para la cual se tomaron las siguientes consideraciones:

- Existen dos fuentes externas de información: DBLP y Mendeley. Éstas exponen la metadata de artículos científicos en formato XML.

- El proceso que implementa el ETL obtiene la entrada de artículos de las fuentes de datos, y debe convertir dicha información a un formato JSON para su posterior almacenamiento en la base de datos local.

- El ETL debe actualizar un índice desarrollado con Apache Lucene [Luc18], con el fin de optimizar las búsquedas en la base de datos.

- Los campos (atributos de los artículos) que debe almacenar el índice son el título, resumen y palabras claves, los cuales se toman en cuenta para los procesos de búsqueda.

- Estas actualizaciones se deben realizar de forma periódica (una vez por semana) y automática.

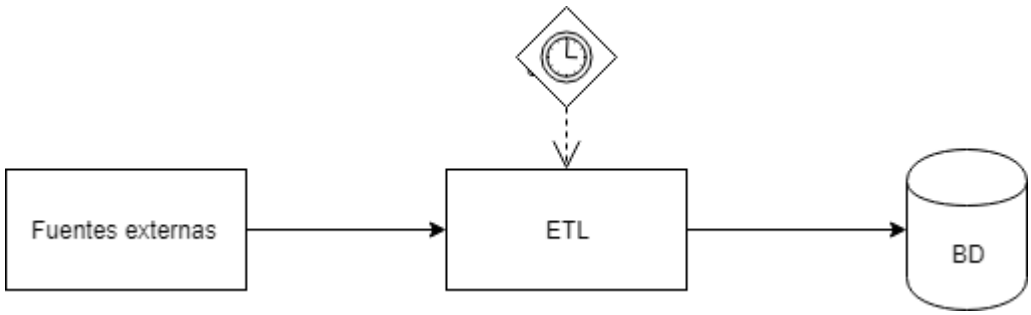

Figura 5: estrategia de extracción de datos (flujo de datos).

A continuación se explica brevemente la estrategia de actualización de la base de datos local, utilizando como fuente externa DBLP:

- *Verificar y extraer fuente*: El ETL verifica que exista un nuevo archivo de datos expuesto por DBLP (disponible para su descarga). Para ello, verifica el campo ETag de la última versión de la base de datos de DBLP disponible públicamente. Si el ETag solicitado es distinto al almacenado en la base de datos local, entonces el sistema se actualiza con la nueva versión, y se procede a descargar el archivo de DBLP. De lo contrario, no se realiza ninguna acción.

- *Verificar mdate*: El atributo mdate corresponde a un campo de control que utiliza DBLP para almacenar la fecha en que se modificó por última vez la información de cada artículo. En este paso se realiza una consulta a la base de datos local para obtener el mdate con la fecha más reciente. Luego se lo almacena en memoria para así determinar si la información que se mantiene en la base de datos local está o no actualizada.

- *Preparación de datos y carga*: Una vez descargado el archivo que expone DBLP con su base de datos en formato XML, éste se descomprime y es procesado. Este archivo XML contiene más de cuatro millones de registros por lo que el Back-end lo va cargando en memoria por partes con el fin de no sobre cargar el equipo. Este proceso puede durar varios minutos por lo que se han configurado para que se realice una vez por semana en la madrugada.

- *Actualización*: Al procesar un artículo se verifica el campo mdate que contiene la última fecha de modificación del mismo, y se compara con el mdate almacenado en memoria. En caso que el mdate del artículo analizado sea mayor que el registrado en el sistema, se procede a actualizar o insertar el artículo en la base de datos local según corresponda. De lo contrario, se prosigue con el procesamiento del siguiente artículo.

4.3 Modelo de Datos

La Figura 6 muestra un modelo entidad-relación, notación IE (Information Engineering) [Teo11] de los datos manejados en la base de datos local. Allí se pueden ver los elementos sobre los que se trabaja, y los atributos de cada uno de ellos.

Dada la naturaleza de los datos involucrados en el sistema, se tomó la decisión de utilizar la base de datos documental MongoDB, para lo cual se crearon una serie de colecciones. La colección principal es una réplica de la base de datos de DBLP.

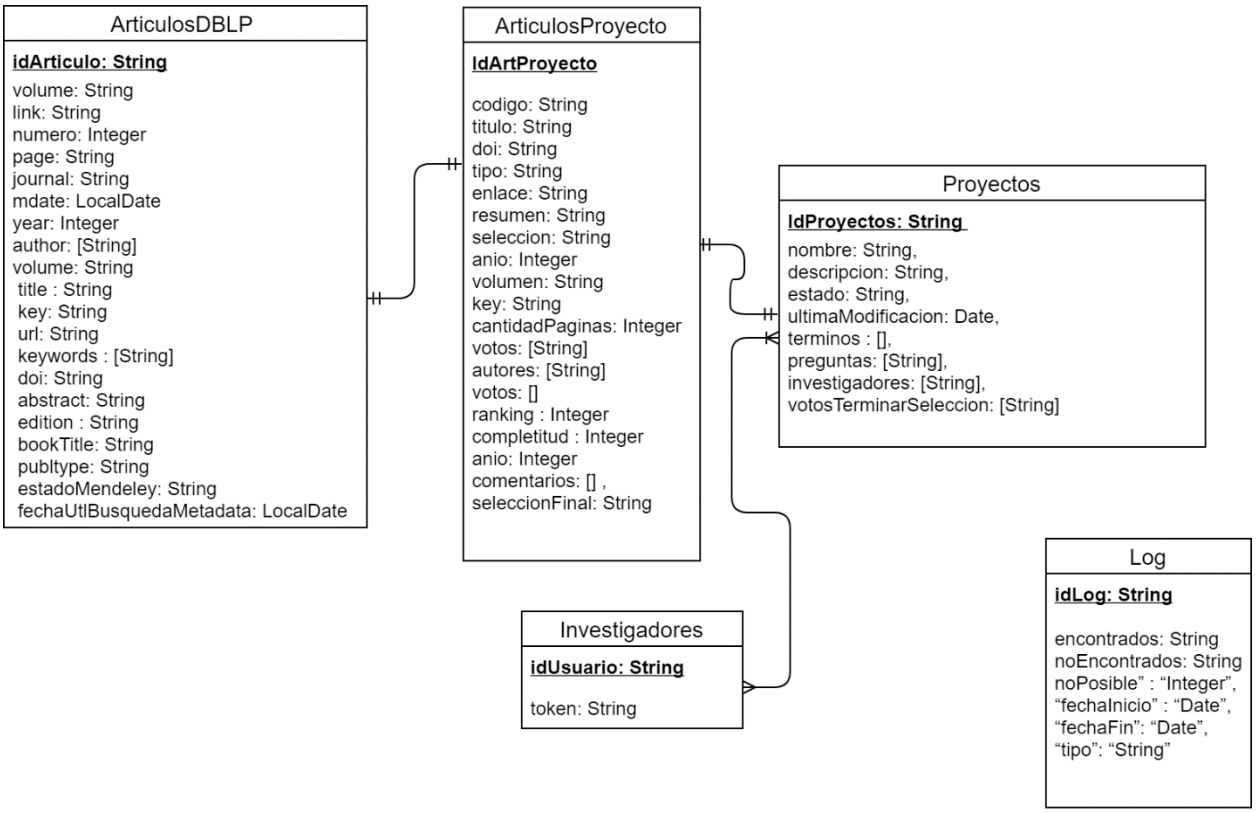

Figura 6: Diagrama de la base de datos.

A continuación se describe la estructura del archivo JSON asociado a una publicación, la cual combina los atributos de un artículo de revista y uno de conferencia:

DBLP:
{ "idArticulo": "String" ,
"volume": "String",
"link": "String",
"numero": "Integer",
"page": "String",
"journal": String,
"mdate": "LocalDate",

“year”: “Integer”,
“author”: [String],
“volume”: String,
“title” : “String”,
“key”: “String”,
“url”: “String”,
“keywords” : [String],
“doi”: “String”,
“abstract”: “String”,
“edition” : “String”,
“bookTitle” : “String”
“publtype”: “String”,
“estadoMendeley”: “String”
“fechaUtlBusquedaMetadata”: “LocalDate” }

En el archivo JSON antes mostrado se aprecian los campos que se desean almacenar como metadata de cada artículo. La base de datos de DBLP no proporciona el resumen, ni las palabras claves de cada estudio, por lo que éstas se tienen que obtener por medio de la API de Mendeley [Men18]. Para el control de este proceso se crearon dos campos adicionales “estadoMendeley” y “fechaUtlBusquedaMetadata”. El primero puede tomar los siguientes valores:

- *encontrado*: Significa que el artículo fue encontrado en Mendeley [Men18] y su metadata ya fue guardada en la base de datos local.

- *no encontrado*: Significa que el artículo fue buscado en Mendeley [Men18] pero no se lo encontró, por lo tanto, se lo deja en este estado para que se busque de nuevo posteriormente, en caso de que se actualicen las fuentes externas.

- *no es posible*: Este estado significa que el artículo no tiene un DOI para ser buscado. De esta forma el proceso de reintentos de búsqueda no lo toma en cuenta.

Es posible que el artículo no posea un valor para este campo (atributo), lo que significa que no ha sido analizado. En el caso del campo “fechaUtlBusquedaMetadata”, éste contiene la última fecha en que fue buscado el artículo.

A continuación se muestra la descripción del archivo JSON asociado a una RSL creada en el sistema, la cual se caracteriza como un proyecto.

Proyectos:

{ “idProyectos” : “String”,

```
“nombre” : “String”,
“descripcion” : “String”,
“estado” : “String”,
“ultimaModificacion” : “Date”,
“terminos” : [{ “sinonimos” : [“String”],
                “valor”: “Integer” }],
“preguntas” : [“String”],
 “investigadores” : [{“correo:String”}],
  “votosTerminarSeleccion” : [“String”]}
```

El esquema “Proyectos” almacena información de cada revisión sistemática de la literatura. Para controlar el progreso de un proyecto se definió el campo estado, el cual indica la etapa en que se encuentra el proyecto:

- *definición*: Indica que el proyecto se ha iniciado y tiene las siguientes características:
    - Se está definiendo el nombre del proyecto, descripción, preguntas de investigación, colaboradores y términos (o palabras claves).
    - El investigador puede definir un peso diferente a cada término, el cual será tomado en cuenta al momento de calcular el ranking (nivel de relevancia) para cada artículo.
    - En esta fase la aplicación provee a los investigadores un número aproximado de los artículos que coinciden con los términos o palabras claves definidas.

- *selección*: Esta etapa se inicia con la búsqueda de los artículos que coinciden con los términos o palabras claves. En ella el investigador puede realizar las siguientes acciones:
    - Verificar la metadata de los artículos y visualizar la misma en diferentes formas (lista de artículos, lista resumen, cuadrícula), con el fin de hacer más cómoda la depuración.
    - Puede votar (opinar) a favor de incluir o excluir un artículo de la etapa final.

- *cerrado*: Ésta se inicia cuando todos los investigadores han indicado que desean terminar la etapa de selección, y en esta última etapa los investigadores pueden:
    - Realizar una última depuración de los artículos seleccionados.
    - Añadir comentarios u observaciones a cada artículo con el fin de compartirlos con el resto del equipo.

Para llevar el control de los investigadores que han solicitado terminar la etapa de selección, se utiliza el campo “votosTerminarSeleccion”, el cual tiene un registro de los

investigadores que han solicitado finalizar esta fase. Otro campo fuera de los definidos por los investigadores es el de “ultimaModificacion”, el cual almacena la última fecha de modificación del proyecto.

El esquema de “Investigadores” permite llevar el control de los usuarios del sistema, además de tener un registro del token de cada usuario. El token es asignado por el sistema y permite crear el enlace que se les envía a los investigadores para acceder al mismo. El identificador de los investigadores es el correo electrónico de cada usuario. A continuación se muestra el esquema definido para almacenar los investigadores.

```
Investigadores:
        { “idUsuario” : “String”,
          “token” : “String” }
```

El esquema “ArticulosProyecto” se crea luego que se realiza la búsqueda de los artículos que pertenecen a una RSL. Este esquema almacena los datos de todos los artículos que coincidieron con los términos de búsqueda definidos por los investigadores. La razón por la cual se crea una copia de la información es que se quiere preservar los datos del RSL tal como se analizaron al momento de crear el RSL, ya que, en un futuro, los datos de los artículos podrían cambiar pues potencialmente pueden ser refinados por el proceso de ETL. A continuación se presenta dicho esquema.

```
ArticulosProyecto:
        {  “codigo” : “String”,
           “titulo” : “String”,
           “doi” : “String”,
           “tipo” : “String”,
           “enlace” : “String”,
           “resumen” : “String”,
           “seleccion” : “String”,
           “anio” : “Integer”,
           “volumen” : “String”,
           “key” : “String”,
           “cantidadPaginas” : “Integer”,
           “votos” : [“String”],
          “autores” : [“String”],
          “keywords” :[“String”]
          “votos”: [{ “correo” : “String”,
                                “voto” : “String”  }],
           ranking : “Integer”,
           completitud : “Integer”,
           “anio” : “Integer”,
```

“comentarios” : [{“idUsuario”: “String”,
“Comentario” : “String” }] ,
seleccionFinal: “String”}

La principal diferencia entre este esquema y el de DBLP son los campos (atributos) que se explican a continuación:

- *ranking*: El valor de este campo indica el nivel de relevancia potencial de cada artículo para ayudar a responder la pregunta de investigación que dio origen a la RSL. La relevancia se determina revisando el lugar donde se encuentran los términos de búsqueda dentro de un artículo.

- *seleccion*: Se refiere al estado final del artículo en la etapa de Selección. Este estado puede tomar los siguientes valores: pendiente, incluido o excluido.

- *comentarios*: Una vez que termina la etapa de selección, los investigadores pueden agregar comentarios a los artículos seleccionados. Dichos comentarios son almacenados en este campo, el cual contiene el identificador del autor que realiza el comentario, además del comentario u observación realizada.

- *seleccionFinal*: Una vez terminada la etapa de selección, los investigadores pueden realizar una evaluación final incluyendo o excluyendo los artículos antes seleccionados. Esta evaluación se almacena en el campo selecionFinal.

- *votos*: Este campo es un arreglo que contiene un campo compuesto por el correo electrónico que identifica al investigador, y un voto que se refiere a la decisión de incluir o excluir el artículo en la etapa de selección. Este arreglo permite controlar si un artículo es incluido o excluido en la etapa de cierre, para lo cual al menos dos investigadores deben de coincidir en esa decisión para que la misma se vuelva definitiva.

El esquema “Log” que se muestra a continuación contiene un registro de las cargas periódicas realizadas para completar la metadata del esquema DBLP.

Log:
{
“encontrados” : “Integer”,
“noEncontrados” : “Integer”,
“noPosible” : “Integer”,
“fechaInicio” : “Date”,
“fechaFin” : “Date”,
“tipo” : “String”

}

Este esquema almacena la fecha y hora en que inició el proceso de ETL, y la fecha y hora en que éste finalizó. Además, contiene un registro de la cantidad de artículos que fueron encontrados en Mendeley, los que no se encontraron y los que no son posibles de encontrar debido a que no poseen un DOI. El tipo de tarea realizada también se almacena en un campo el cual puede ser:

- artículos no buscados: el cual se refiere a que los artículos procesados nunca se han buscado en Mendeley.

- artículos no encontrados: el cual se refiere a que los artículos procesados ya se han buscado en Mendeley, pero no se encontraron.

4.4 Estrategia de Ranking y Completitud

En el caso de la estrategia para clasificar y valorar los artículos en la solución propuesta, se implementaron dos indicadores que apoyan a los investigadores en esta tarea, el primero de ellos es el "*ranking"* y el segundo es la *"completitud"*. Para establecer el cálculo del ranking se realizaron varias pruebas y la fórmula se fue ajustando hasta lograr el objetivo deseado. Este campo se determina computando diversos pesos, que se establecen de acuerdo al lugar donde se encuentran los términos de búsqueda dentro de la metadata asociada a los artículos. Para ello se sigue el proceso que se indica a continuación:

- Se le solicita al investigador que valorice cada término de búsqueda, el cual incluye un conjunto de sinónimos de dicho término. En caso de que el investigador no realice esta valoración, el sistema tomará por defecto el valor 1 para cada término de búsqueda. Esto significa que el peso (relevancia) de todos los términos es igual.

- Una vez que el sistema extrae la metadata de los artículos, éste verifica que el título, resumen o las palabras claves contengan alguno de los sinónimos definidos en las cadenas de búsqueda, valorizando con un 4 si los términos aparecen en las palabras claves, un 3 si éstos parecen en el título, y un 1 si están en el resumen. A estos valores luego se los multiplica por el peso que le dio el investigador (o el sistema) a cada término de búsqueda. Finalmente, este valor es elevado a una potencia que es igual a la cantidad de grupos de sinónimos (términos de búsqueda) que coincidieron con la información contenida en el título, resumen o palabras claves de cada artículo.

- Este valor luego es incluido en la metadata de cada artículo, lo que permitirá al investigador estimar de una forma rápida la relevancia que tiene cada estudio, según la pregunta de investigación que se quiere responder. Este dato es utilizado

por la herramienta para ordenar los artículos y ayudarles a los investigadores en su procesamiento.

El siguiente ejemplo muestra el proceso para generar el ranking de artículos, suponiendo que se tienen los siguientes grupos de términos o palabras claves:

- ranking
- judgments, judgements

A modo de ejemplo, consideremos los dos términos anteriores, con una valorización de parte del investigador de 1 para “ranking” y de 100 para “judments, judgements”. Consideremos además el siguiente artículo para ilustrar el cálculo del ranking.

> *Título*: Using Supervised Machine Learning to Automatically Build Relevance Judgments for a Test Collection
>
> *Palabras claves*: test collections, evaluation, qrels, relevance judgments, machine learning, doc2vec, tf-idf
>
> *Resumen*: This paper describes a new approach to building the query based relevance sets (qrels) or relevance judgments for a test collection automatically without using any human intervention. The methods we describe use supervised machine learning algorithms, namely the Naïve Bayes classifier and the Support Vector Machine (SVM). We achieve better Kendall's tau and Spearman correlation results between the TREC system ranking using the newly generated qrels and the ranking obtained from using the human-built qrels than previous baselines. We also apply a variation of these approaches by using the doc2vec representation of the documents rather than using the traditional tf-idf representation.

El proceso de cálculo se divide en cuatro partes:

- *Lo primero es valorar el título*. Como sólo el segundo término coincide con los términos de búsqueda, se coloca el valor 100 definido por el usuario, multiplicado por 3 que es el valor definido para los términos que coinciden en el título. El resultado se eleva a la potencia 1, porque sólo el segundo grupo de sinónimos coincidió con la información del título del artículo, por lo tanto el valor final obtenido es: $(3 * 100)^{1}$ = 300

- *Lo siguiente es valorar las palabras claves*. Como sólo el segundo término coincide con la información del artículo, se coloca el valor 100 definido por el usuario, multiplicado por 4 que está definido para los términos que coinciden en las palabras claves. El resultado se eleva a la potencia 1, porque sólo el segundo

término (grupo de sinónimos) coincidió con la información de las palabras claves de dicho estudio, obteniéndose así un valor final igual a: $(4 * 100)^{1} = 400$

- *El tercer paso será valorar el resumen*. Como los dos grupos de sinónimos coinciden con la información del resumen, cada uno aporta el valor definido por el usuario 1 y 100 respectivamente. Estos son multiplicados por 1, que está definido por el sistema para los términos que coinciden en el resumen. Estos valores se suman y el resultado se eleva a la potencia 2, porque los dos términos (grupos de sinónimos) estaban contenidos en el resumen del artículo. Por lo tanto el valor final obtenido es el siguiente: $(1 * 1 + 1 * 100)^{2} = 10.201$

- *Finalmente, calcular el valor final*. En el último paso se suman los valores calculados anteriormente:
  300 + 400 + 10.201 = 10.901

El segundo indicador que calcula el sistema desarrollado es la "*completitud"*, la cual determina si los todos términos o palabras claves buscadas, fueron encontrados en la metadata de los artículos; particularmente en el título, resumen y palabras claves. Este indicador (completitud) podrá tomar tres valores: 3, si todas las cadenas de sinónimos (enlazadas con el operador AND) coincidieron en al menos un campo de la metadata del artículo; 2, si todas menos una cadena de sinónimos coincidieron en la metadada del artículo; y 1, si todas menos dos cadena de sinónimos coincidieron en la metadada del artículo. Por lo tanto, en el ejemplo anterior el valor de la completitud en el artículo sería 3. Sin embargo, si agregamos otra cadena de sinónimos que incluya palabras claves como "effort" y "cost", el valor de completitud para dicho artículo sería 2, ya que ni "effort", ni "cost" se encuentran en el título, resumen o palabras claves.

4.5 Tecnologías Utilizadas

En cuanto a las tecnologías elegidas para la implementación de la solución, se consideró utilizar una base de datos documental debido al tipo de información (metadata de documentos académicos) que se pretende almacenar, y un índice desarrollado con Apache Lucene para minimizar los tiempos de búsqueda de artículos. Para el front-end se escogió desarrollarlo con Angular 4, y un back-end con Java 1.8 y Spring 4 por el soporte que dan estas tecnologías para desarrollar servicios REST (para el API ofrecido al front-end) y la simplicidad de acceso a datos (incluyendo la conexión a MongoDB). El resumen de estas tecnologías se detalla en la Tabla 5.

Tabla 5: Tecnologías utilizadas en la implementación de la solución

Componente	Herramientas utilizadas
Front-end	Angular 4
Back-end	Java 1.8, Spring 4
Bases de datos	MongoDB 3.6.2 y Apache Lucene 2.9.4
ETL	Java 1.8, Spring 4, API Mendeley

4.6 Resumen

Se determinó que la solución tuviese una arquitectura en capas, la cual cuenta con el desarrollo de un Front-end que gestiona la interacción con el investigador, un back-end que gestiona la comunicación con la base de datos y un ETL que se encarga de actualizar la base de datos de forma automática y periódica. En canto a la tecnología de la base de datos se decidió que fuese MongoDB por la naturaleza de los datos a almacenar, que son de tipo documento. Además, se determinó crear un ranking al momento de realizar la búsqueda de artículos para un RSL con el fin de facilitar al investigador la tarea de encontrar los artículos relevantes. En el Capítulo 5 se presenta la implementación de la aplicación a partir del diseño planteado.

5. Implementación de la Aplicación

Este capítulo se divide en cinco secciones, las cuales presentan las principales interfaces del sistema desarrollado. En cada sección se explican las tareas que los usuarios pueden realizar, como por ejemplo el “Ingreso al Sistema”, el cual detalla la forma en que los usuarios pueden registrarse y acceder al mismo, el “Acceso a los Proyectos” donde se describe cómo acceder a las RSL desarrolladas, la “Creación de una RSL” la cual detalla los pasos que se deber realizar para crear un nuevo proyecto, la “Configuración de la Búsqueda” donde se expone las configuraciones que se deben efectuar para realizar la búsqueda de artículos, y la “Selección de Artículos” donde se explica el proceso para incluir o excluir los artículos en una RSL.

5.1 Ingreso al Sistema

La Figura 7 muestra la interfaz de usuario donde los investigadores deben ingresar para solicitar acceso al sistema. Una vez ingresado su correo en la caja de texto indicada con el rótulo 1, y dando click al botón registrarse (rótulo 2), el sistema envía un correo electrónico al investigador con un enlace que contiene un link de acceso a la RSL para dicho usuario. Este token no caduca, y por lo tanto puede ser utilizado por el usuario las veces que lo necesite.

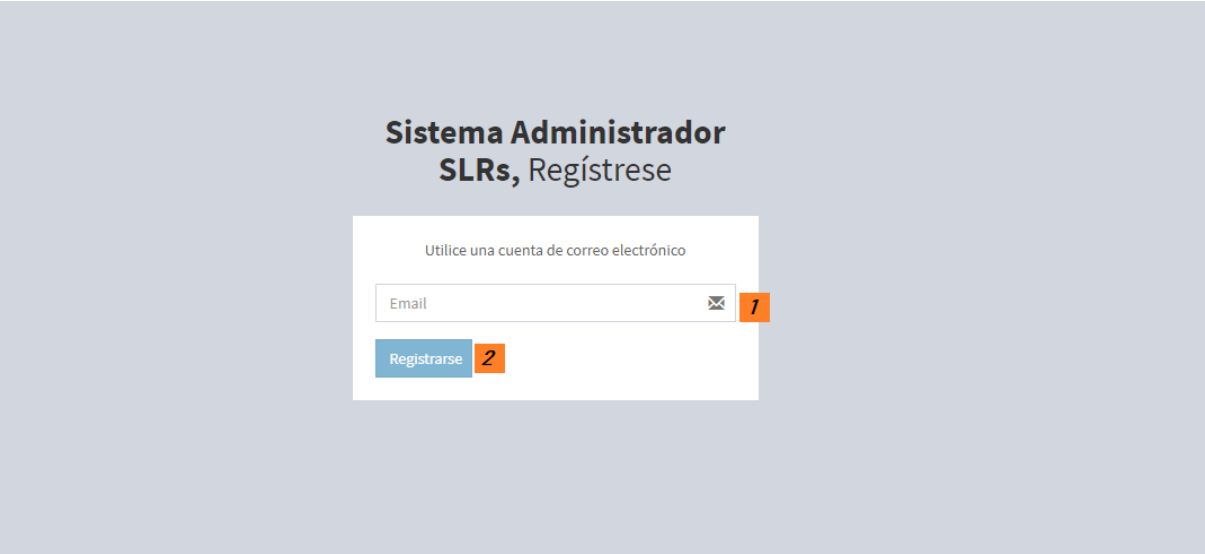

Figura 7: Registro de usuarios del sistema.

5.2 Acceso a los Proyectos

Una vez que el usuario ingresa a la aplicación por medio del enlace enviado a su correo, éste accederá a la sección “Mis Proyectos”. Esta es la interfaz que se muestra en la Figura 8, la cual contiene un listado de todas las RSLs en las cuales ha participado o está

participando dicho investigador (rótulo 1). Además, muestra un resumen de cada RSL es desplegado en el panel principal (rótulo 2), el cual a su vez le permite al usuario:

- Verificar la información básica de cada proyecto (rótulo 3): el título, estado de la RSL (definición, selección o cerrado), última fecha de modificación, y el correo electrónico de los investigadores participantes.

- Acceder a las RSLs en las cuales participa el investigador (rótulo 3) con el fin de trabajar en las mismas.

- Crear un nuevo proyecto (rótulo 4).

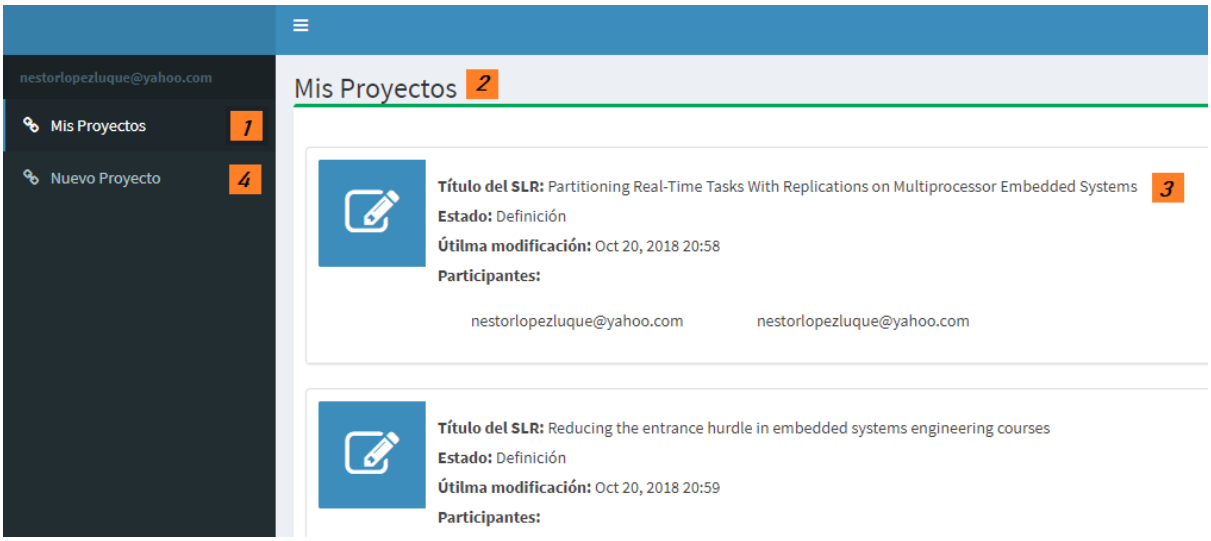

Figura 8: Listado de proyectos en los que participa el investigador.

5.3 Creación de una RSL

Una vez que el usuario ha decidido crear una nueva revisión sistemática de la literatura (rótulo 4 de la figura 8), la aplicación despliega la interfaz mostrada en la Figura 9. En este momento la RSL se encuentra en la fase de definición, dando la posibilidad a los usuarios de realizar las siguientes acciones:

- Editar la definición de la RSL (rótulo 1) agregando un título y una descripción al proyecto (rótulo 2).

- Agregar nuevos investigadores. Para esto, el investigador debe agregar el correo electrónico del resto de los participantes en el área de “Registro de Usuarios” (rótulo 3). Al incluirlos, la aplicación les envía a los investigadores un correo electrónico con un enlace directo al proyecto para que puedan colaborar en su realización.

- Eliminar usuarios. En este caso la aplicación le quita el acceso a la RSL que previamente se le proporcionó, por lo que el enlace que se le envió al incluirlo en el proyecto quedará deshabilitado (rótulo 3).

- En caso que el usuario quiera regresar a la pantalla de “Mis proyectos”, puede hacerlo mediante el enlace en el rótulo 4 que aparece en el panel izquierdo.

Además, en esta pantalla el usuario puede desplegar un menú en la etiqueta “Proyecto”, la cual se encuentra en el panel izquierdo. Allí se le presenta una serie de enlaces que dirigen de forma rápida a diferentes áreas de la definición de la RSL.

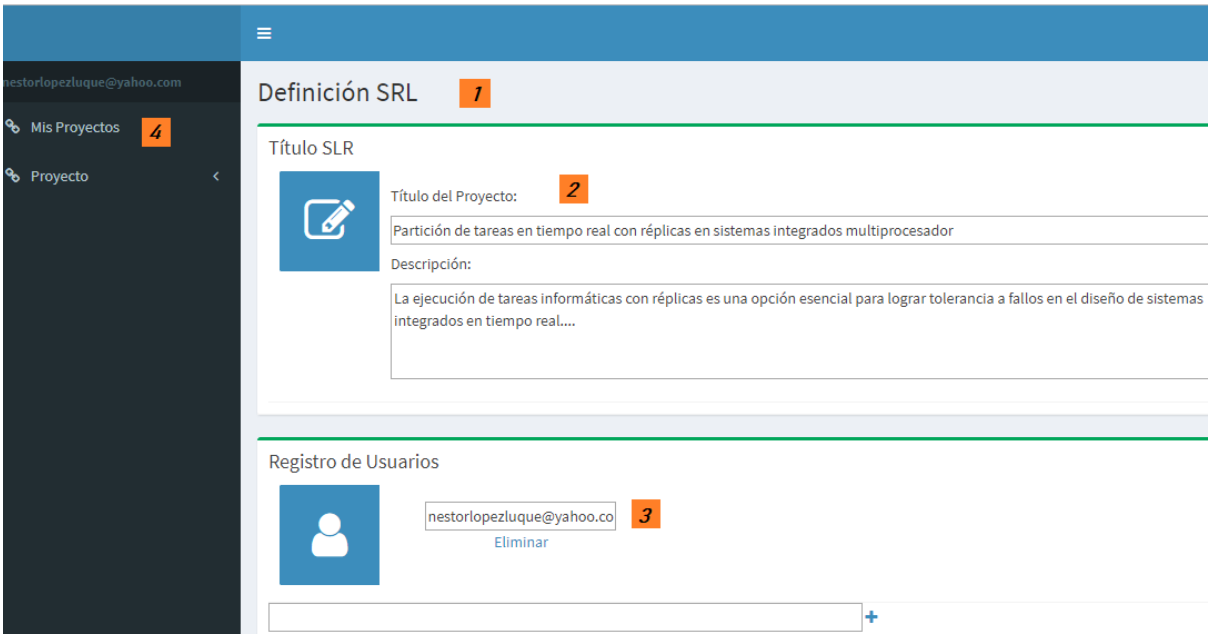

Figura 9: Funcionalidad asociada a la etapa de definición de una RSL

5.4 Configuración de la Búsqueda

En la Figura 10 se muestran otros campos que los investigadores pueden definir durante la etapa de definición de la RSL. Ejemplos de estos campos son los siguientes:

- Agregar o editar las preguntas de la investigación (rótulo 1), esto con el fin de tener un registro de las mismas. Este paso corresponde a la etapa 1 del proceso propuesto para RSL de la figura 2.

- Agregar o editar los términos de búsqueda (rótulo 2), con sus respectivos sinónimos, los cuales permitirán generar las cadenas de búsqueda correspondientes. En este campo el investigador pude utilizar el * (asterisco) en el

caso que desee seleccionar todas las posibles terminaciones de una palabra. Por ejemplo, si agrega la palabra program* estaría agregando las palabras program, programing, programs entre otras. Este paso corresponde a la etapa 2 del proceso propuesto para RSL de la figura 2.

- Agregar o editar una valorización por cada término buscado, con sus respectivos sinónimos (rótulo 3). Cada valor agregado debe ser un número entero positivo el cual será tomado en cuenta al momento de calcular el ranking. Dicho ranking define el nivel de relevancia de cada artículo con respecto a los términos de búsqueda.

- El investigador también puede verificar, en tiempo real, la cantidad de artículos que coinciden con los términos de búsqueda (rótulo 4). En esta parte se muestran tres etiquetas que contienen los valores que corresponden a los artículos que coinciden con:
    - todos los términos.
    - todos los términos menos uno.
    - todos los términos menos dos.

- El investigador también puede verificar la cadena de búsqueda utilizada en la recuperación de artículos (rótulo 5). Este paso corresponde a la etapa 3 del proceso propuesto para RSL de la figura 2.

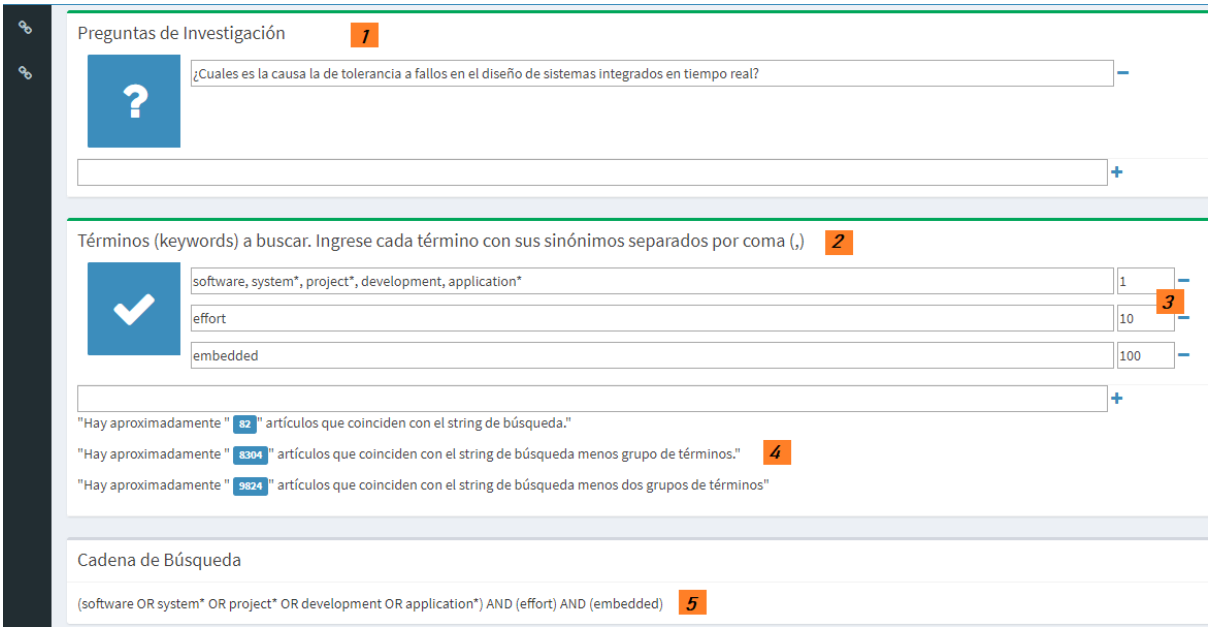

Figura 10: Interfaz de definición de preguntas de investigación y términos de búsqueda

En la Figura 11 se muestra los últimos campos a editar antes de terminar con la etapa de definición y pasar a la etapa de selección. Allí se pueden realizar las siguientes acciones:

- Definir el intervalo de años (rótulo 2) que se desea agregar para limitar la extracción de los artículos (en un formato de cuatro dígitos). En caso de dejar los campos vacíos la aplicación no aplica ningún filtro, y en caso de agregar sólo el primer valor, se mostrarán los artículos desde el año indicado. En caso de agregar sólo el segundo valor, se mostrarán los artículos hasta el año indicado.

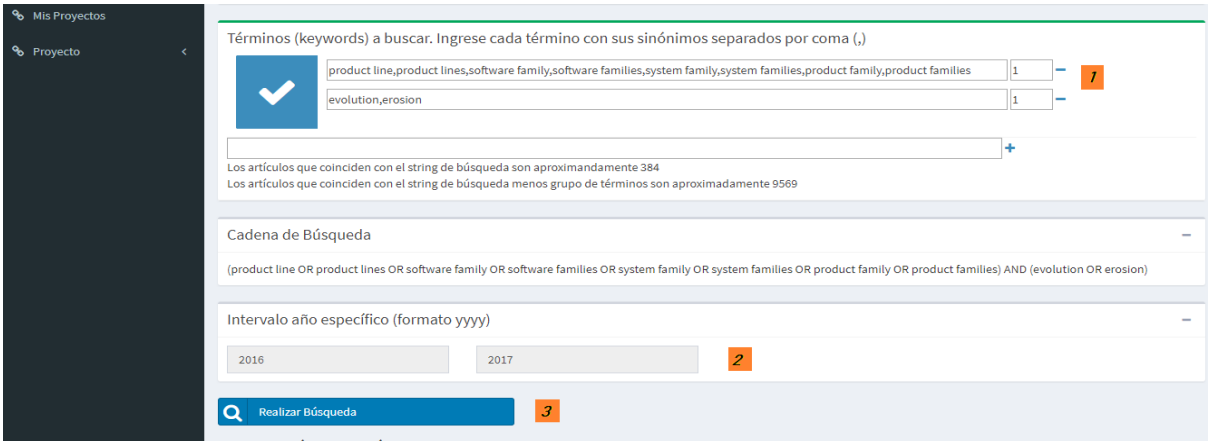

Figura 11: Definición de filtro de años y búsqueda de artículos

- Realizar la búsqueda (rótulo 3). Esta acción hará que el sistema realice la búsqueda de los artículos que coinciden con los términos definidos por los investigadores. Este paso corresponde a las etapas 4 y 5 del proceso propuesto para RSL de la figura 2. Sobre el resultado de la búsqueda el sistema define el ranking y la completitud de cada artículo. Además, los ordena de acuerdo al ranking con el fin de facilitar a los investigadores la tarea de analizar los artículos, lo que corresponde a la etapa 6 del proceso propuesto para RSL de la figura 2.

5.5 Selección de Artículos

Luego que se ha realizado la búsqueda la RSL, se pasa a la etapa de selección, donde ya no se podrá editar el título, ni la descripción de la RSL, ni agregar nuevos investigadores. Sin embargo, en esta etapa queda abierta la posibilidad de volver a redefinir los términos de búsqueda, con sus sinónimos (rótulo 1 de la Figura 11). De esta manera es posible volver a realizar la búsqueda, hasta que se considere que los términos están bien afinados. Se permite esto con la salvedad de que si el usuario selecciona algún artículo, esta selección no tendrá efecto al volver a realizar la búsqueda (rótulo 3 de la Figura 11).

En la Figura 12 se muestran los artículos que coinciden con los términos definidos por los investigadores. Esta fase corresponde a la etapa 7 del proceso propuesto para RSL de la figura 2 y En esta fase de selección los investigadores podrán:

- Acceder al listado de los artículos que coinciden con las cadenas de búsqueda (rótulo 1).

- Acceder al título, resumen, palabras claves, y autores, entre otros, con el fin de decidir si incluir o excluir los artículos para la etapa final.

- Verificar ranking de cada artículo (rótulo 2), el cual es calculado por el sistema a partir de los términos de búsqueda y la valorización de estos términos indicada por el usuario.

- Verificar la completitud (nivel de cobertura) de cada artículo (rótulo 3), la cual se representa visualmente por medio de símbolos “check”. Si el artículo tiene tres checks, significa que tiene una mayor completitud o cobertura de términos de búsqueda, que aquellos que tienen dos checks, y así sucesivamente.

- El investigador también puede acceder al artículo por medio del enlace que se presenta en esta sección, siempre y cuando el usuario tenga acceso a la biblioteca digital (fuente de información) de dicho artículo.

- El investigador también puede evaluar la relevancia del artículo con respecto a la pregunta de investigación formulada. Además, puede tomar la decisión de incluir o excluir dicho estudio, tomando en cuenta que se necesitan al menos dos votos coincidentes para definir si el artículo se incluye o excluye definitivamente de la RSL.

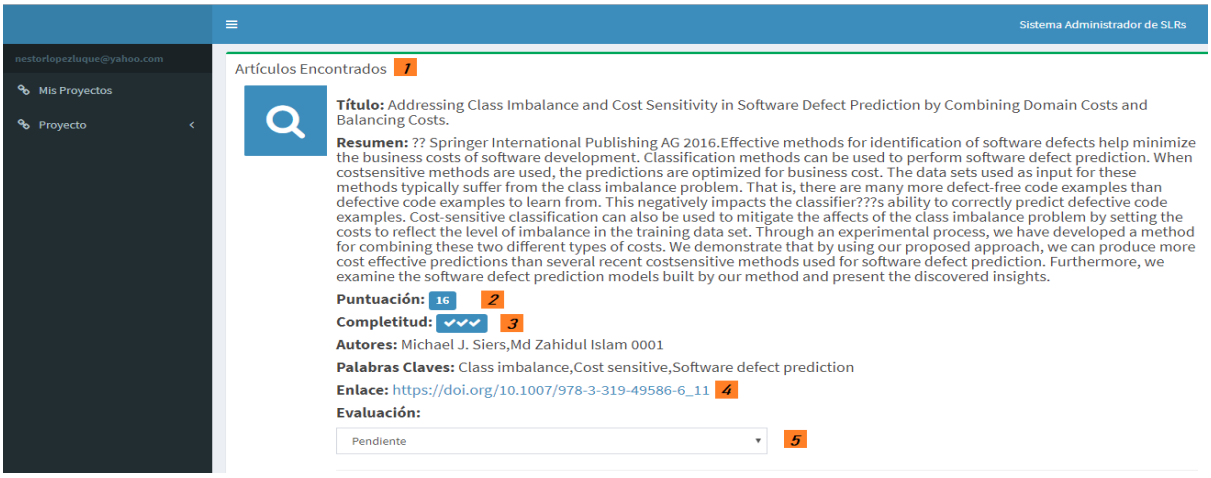

Figura 12: Resultados de búsqueda de artículos

Como se muestra en la Figura 13, durante la etapa de selección de artículos (rótulo 1) se presenta una serie de opciones. Éstas permiten visualizar los resultados de diferentes formas, estructuras de presentación de los datos (rótulo 2), con el fin de facilitar la tarea de depuración inicial de artículos.

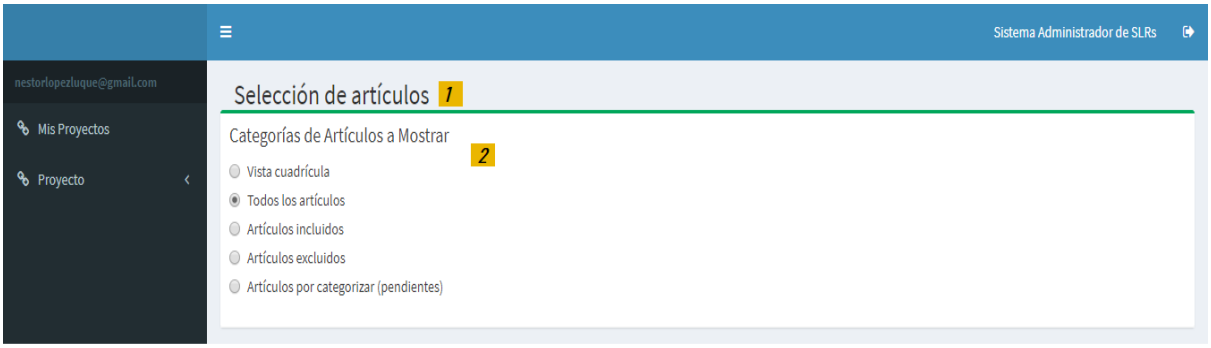

Figura 13: Opciones para mostrar la selección de artículos para una RSL

Las opciones de visualización son las siguientes (rótulo 2):

- *Vista cuadrícula*: En esta vista se muestran todos los artículos en una presentación tipo hoja de cuadrícula. Allí los investigadores pueden votar a favor de incluir o excluir cada uno de los artículos.

- *Todos los artículos*: En esta vista se visualizan los artículos en una lista donde el investigador puede revisar la metadata, y al igual que en la vista anterior, puede decidir incluir o excluir artículos.

- Artículos incluidos: En esta vista se visualizan sólo los artículos que han sido incluidos para la etapa final (para esto se necesita de dos votos a favor de incluirlos).

- Artículos excluidos: En esta vista se visualizan sólo los artículos que han sido excluidos para la etapa final (para esto se necesita de dos votos a favor de excluirlos).

- Artículos por categorizar (pendientes): En esta vista se incluyen los artículos pendientes por clasificar, en caso que los investigadores dejen artículos pendientes y pasen a la fase de cierre, estos no se reflejarán.

Una vez que todos los participantes han decidido terminar la etapa de selección de artículos, se inicia la etapa de cierre de la RSL, en la cual los investigadores sólo tienen acceso a los artículos incluidos en la etapa de selección. Como se muestra en la Figura

14, en esta etapa se puede realizar una depuración final (rótulo 1) y añadir comentarios (rótulo 2). Este paso corresponde a la etapa 8 del proceso propuesto para RSL de la figura 2.

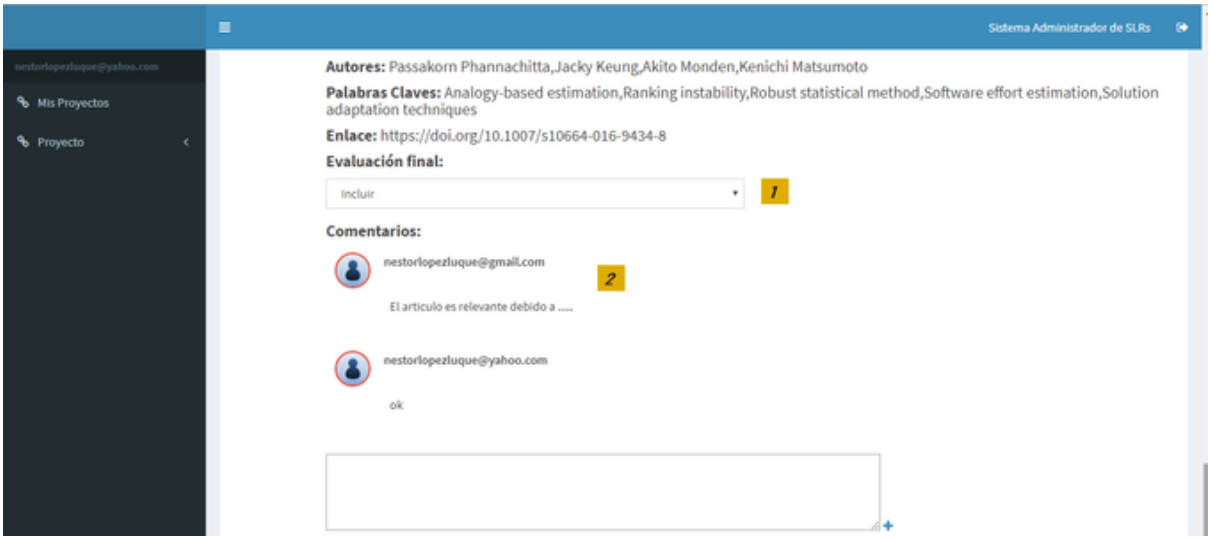

Figura 14: Acceso a los artículos en la etapa de cierre.

5.6 Resumen

Para el diseño de las interfaces de usuario se procuró que fuesen lo más intuitivas posibles para el investigador que conoce el proceso de una RSL. En cuanto al acceso al sistema se implementó a través de un mecanismo sencillo, donde el investigador recibe por email un link que le da acceso al grupo de proyectos en los que está participando o ha participado ya en el pasado.

En cuanto a la definición de una nueva RSL, en la aplicación se trató de que exista un orden lógico en los pasos a seguir, con el fin de que el investigador pueda sentirse cómodo y familiarizado con la aplicación. En la etapa de selección de artículos se desarrollaron diferentes tipos de vistas con el fin de hacer esta tarea lo menos engorrosa posible para el investigador. En el próximo capítulo se muestran los resultados de replicar algunas RSL utilizando la aplicación aquí mostrada.

6. Validación

La validación de la herramienta consistió en dos partes:

- Replicar RSL realizadas de forma manual, reportadas en la literatura del área de ingeniería de software. Se hizo esto con el fin de comparar los resultados reportados en esos artículos, contra aquellos obtenidos a partir del uso de la herramienta reportada en esta tesis. Así se buscó medir la eficacia de la herramienta.

- Realizar pruebas con usuarios expertos en RSL en el área de las ciencias de la computación. Se hizo esto con el fin de recibir retroalimentación en cuanto a la utilidad, usabilidad y eficiencia de la herramienta.

A continuación se describen ambos tipos de evaluaciones.

6.1 Replicación de Revisiones Sistemáticas de la Literatura

Para esta etapa se replicaron tres revisiones sistemáticas de la literatura, dos de las cuales habían sido realizadas por académicos del Departamento de Ciencias de la Computación de la Universidad de Chile, y la tercera se seleccionó de entre las RSL reportadas en la literatura. Estas pruebas tenían como objetivo:

- Verificar que los artículos seleccionados de forma manual también eran recuperados por la herramienta propuesta.

- Evaluar la clasificación que realiza la herramienta respecto a los artículos encontrados, verificando la relevancia que ésta asignó a los artículos seleccionados de forma manual.

A continuación se detallan cada una de las revisiones sistemáticas de la literatura seleccionadas:

6.1.1 Caso de Estudio 1

Título de la revisión sistemática de la literatura: “Software product line evolution: A systematic literature review” [Mar18].

Preguntas de investigación:
What approaches have been reported regarding the analysis and support of SPL evolution?
How do these approaches define an evolution process?
How do these approaches define SPL erosion?

How are these approaches evaluated?
How are these approaches validated?
Are these approaches general-purpose or domain-specific?

Cadena de búsqueda:
(product line OR product lines OR software family OR software families OR system family OR system families OR product family OR product families) AND (evolution OR erosion)

Bibliotecas consultadas en la revisión original:
ACM, IEEEXplore, Science Direct, Scopus, Springer, Wiley

Esta revisión sistemática de la literatura reporta que en la primera fase se extrajeron 3.387 artículos, y luego de aplicar filtros y eliminar duplicados quedó un total de 2.212. Estos restantes fueron analizados considerando su título, resumen y palabra claves. Como resultado, se seleccionaron 218 artículos que fueron leídos totalmente para obtener un resultado final de 60 artículos seleccionados de forma manual.

Por otra parte, al utilizar la aplicación propuesta, se recuperaron 9.207 artículos, los cuales fueron ordenados por la misma de acuerdo a su ranking. En el caso de los 60 artículos seleccionados de forma manual en la RSL original, se encontraron 59 entre el total de los artículos extraídos por la aplicación. Estos 59 artículos se reportaron dentro de los primeros 235 papers mejor rankeados por la aplicación.

Los artículos mejor rankeados por la herramienta están indicados en el Anexo B.1. El único artículo no recuperado por la herramienta se titula “Resource versioning scheme in evolutionary software product line” [Dko14], el cual no se encuentra en DBLP [DBL18] ya que pertenece a un journal poco conocido.

Al analizar los artículos ordenados por relevancia por la herramienta (anexo B.1), se puede inferir al analizar la metadata de los resultados, que a partir del paper ubicado en la posición 300, la relevancia de los artículos comienza a disminuir significativamente, por lo que es poco probable encontrar artículos relevantes para la RSL en lo que resta de la lista de resultados. Este tipo de filtrado no se puede realizar con el proceso manual, ya que no se cuenta con un ranking, por lo que se deben de analizar muchos más artículos para poder determinar cuándo parar de buscar artículos relevantes. La tabla 6 muestra un resumen comparativo de los resultados de este caso de estudio:

Tabla 6: Resumen Caso de Estudio 1

Filtrado tradicional de artículos (búsqueda manual)	*Filtrado de artículos utilizando la herramienta desarrollada*
Cantidad de artículos recuperados en la fase inicial: 3.387 Cantidad de artículos que es necesario analizar para encontrar los artículos seleccionados: 2.212 Cantidad de artículos finalmente seleccionados: 60	Cantidad de artículos recuperados en la fase inicial: 9.207 Cantidad de artículos que es necesarios analizar para encontrar los artículos seleccionados: 235 Cantidad de artículos finalmente seleccionados: 59

En la Figura 15 se puede apreciar cómo, a medida que avanza la posición asignada por la herramienta, van apareciendo los artículos relevantes que fueron seleccionados por los investigadores. Al primer artículo de los seleccionados aparece en la posición 1, y así sucesivamente hasta llegar al último en la posición 235. Particularmente para esta revisión sistemática de la literatura, al evaluar las primeras 100 posiciones se habrá encontrado el 60% de los artículos relevantes seleccionados.

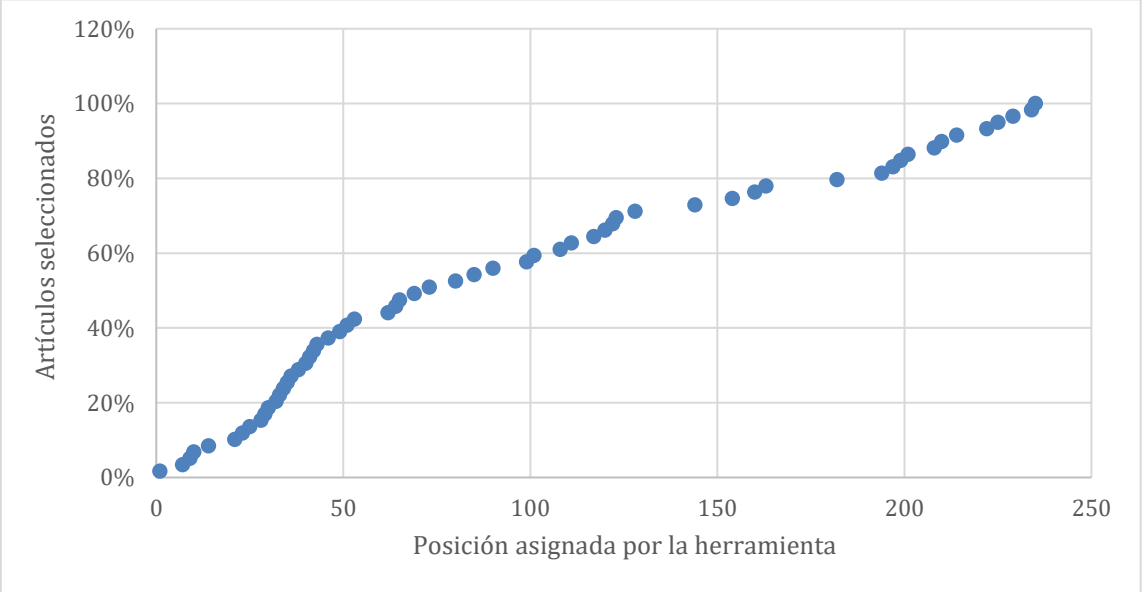

Figura 15: Artículos seleccionados versus la posición (relevancia) asignada por la herramienta.

Si consideramos la cantidad de artículos que es necesario analizar con el fin obtener la selección final de papers para realizar la RSL, la búsqueda de artículos usando la herramienta desarrollada muestra resultados muy por encima que la búsqueda manual. En este sentido, y como se muestra en la Fig. 15, para este caso el investigador sólo

necesitará revisar los primeros 250 artículos rankeados, y de esa manera encontrará los 60 artículos que fueron seleccionados en forma manual.

6.1.2 Caso de Estudio 2

Título revisión sistemática: Survey of Software Development Effort Estimation Taxonomies [Ver18].

Preguntas de investigación:
What are the main taxonomies of SDEE methods reported in the literature?
What are the most relevant classification criteria used in these taxonomies?

Cadena de búsqueda:
(Software OR system* OR project* OR development OR application*) AND (effort OR cost* OR siz*) AND (Estimat* OR Predict* OR Assess* OR Forecast* OR calcula* OR siz* OR measur* OR dimension*) AND (Taxonom* OR Categor* OR Class* OR Process* OR strateg* OR approach* OR method* OR algorithm* OR Metric* OR Unit* OR Review* OR Survey* OR Analys* OR Report* OR Ensemble OR Systematic)

Bibliotecas consultadas en la revisión original:
IEEEXplore, ACM Digital Library, Science Direct, Springer

Esta revisión sistemática de la literatura reporta que en la primera fase se extrajeron 95.715 artículos, y luego de aplicar filtros, eliminación de duplicados, criterios de inclusión y exclusión quedó un total de 1.303. Estos restantes fueron analizados por su metadata y contenido para obtener un resultado final de 17 artículos seleccionados.

En cambio, la herramienta extrajo 23.150 artículos que fueron ordenados por su ranking. De los 17 artículos seleccionados de forma manual se encontraron 16 entre el total extraído, y estos 16 se reportaron dentro de los primeros 293 artículos mejor rankeados (ver anexo B.2). El artículo no recuperado por la herramienta corresponde al estudio titulado: “Software development cost estimation: A survey” [Raj16], el cual no se encuentra en DBLP [DBL18] ya que pertenece a Indian Journal of Science, que es una revista poco conocida.

Al analizar los artículos ordenados por la herramienta se puede inferir, que al igual que en el primer caso, a partir del rango de los 300 la relevancia de los artículos comienza a disminuir significativamente. Por lo tanto, es poco probable encontrar artículos relevantes para la RSL después de esta posición. La tabla 7 muestra un resumen comparativo de los resultados de este caso de estudio:

Tabla 7: Resumen Caso de Estudio 2

Filtrado tradicional de artículos (búsqueda manual)	*Filtrado de artículos utilizando la herramienta desarrollada*
Cantidad de artículos recuperados en la fase inicial: *95.715* Cantidad de artículos que es necesarios analizar para encontrar los artículos seleccionados: *1.303* Cantidad de artículos finalmente seleccionados: *17*	Cantidad de artículos recuperados en la fase inicial: *23.150* Cantidad de artículos que es necesarios analizar para encontrar los artículos seleccionados: *293* Cantidad de artículos finalmente seleccionados: *16*

En este segundo caso (Figura 16) se puede apreciar cómo a medida que avanza la posición asignada por la herramienta van apareciendo los artículos relevantes, que fueron seleccionados por los investigadores. Donde el primer artículo de los seleccionados que aparece se le asignó la posición 7 hasta llegar al último en la posición 293. Particularmente para esta revisión sistemática de la literatura al evaluar las primeras 100 posiciones se habrá encontrado el 50% de los artículos relevantes seleccionados.

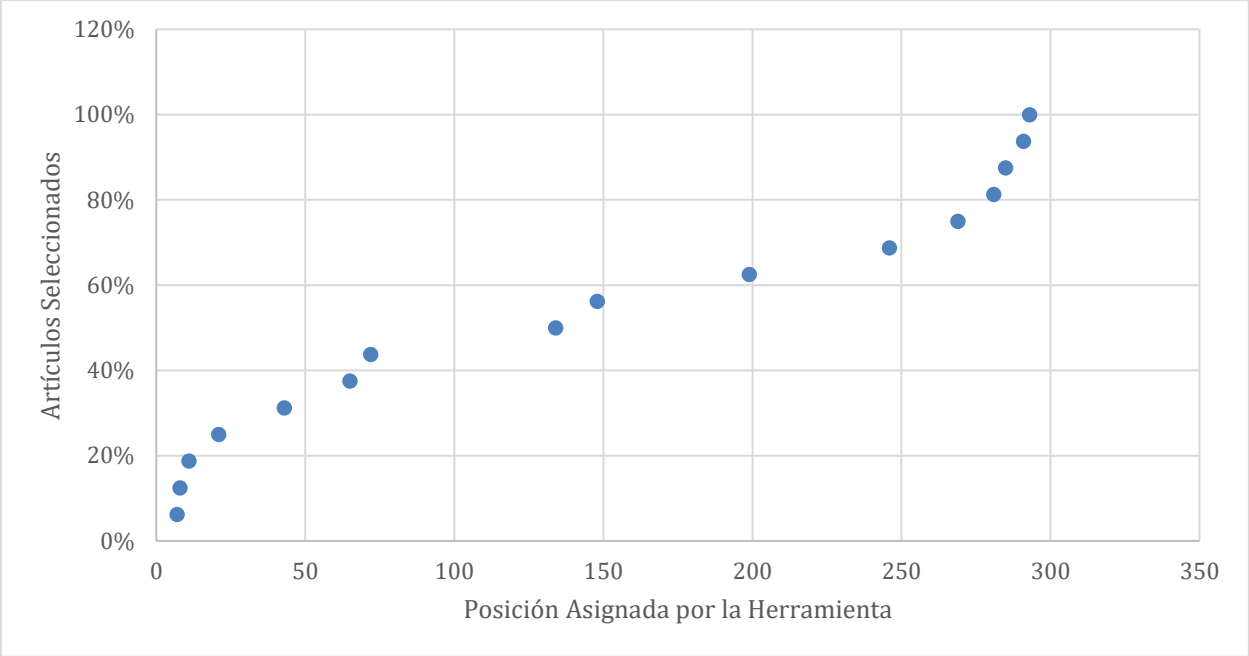

Figura 16: Artículos seleccionados frente a la posición asignada por la herramienta.

En los resultados se puede observar una considerable diferencia a favor de la herramienta, con respecto la cantidad de artículos necesarios a analizar con el fin obtener los seleccionados, comprobando su eficiencia del ranking para valorar los artículos.

6.1.3 Caso de Estudio 3

Título revisión sistemática: Systematic literature review of machine learning based software development effort estimation models [Jia12].

Preguntas de investigación:
Which ML techniques have been used for SDEE?
What is the overall estimation accuracy of ML models?
Do ML models outperform non-ML models?
Are there any ML models that distinctly outperform other ML models?
What are the favorable estimation contexts of ML models?

Cadena de búsqueda:
software AND (effort OR cost OR costs) AND (estimat* OR predict*) AND (learning OR data mining OR artificial intelligence OR pattern recognition OR analogy OR case based reasoning OR nearest neighbo* OR decision tree* OR regression tree* OR classification tree* OR neural net OR genetic programming OR genetic algorithm* OR bayesian belief network* OR bayesian net* OR ‘‘association rule OR support vector machine OR support vector regression)

Bibliotecas consultadas en la revisión original:
IEEEXplore, ACM Digital Library, Science Direct, Springer, Web of Science, IE Compendex, Google Scholar, BEST Web

Esta revisión sistemática de la literatura reporta que en la primera fase se extrajeron 3.136 artículos, y luego de aplicar filtros y eliminación duplicados quedaron 2.191. Estos restantes fueron analizados usando su metadata y contenido, para obtener un resultado final de 84 artículos seleccionados.

En cambio, la herramienta extrajo 17.836 artículos, los cuales fueron ordenados por su ranking. De los 84 artículos seleccionados de forma manual se encontraron 83 entre los recuperados por la herramienta, y estos 83 estudios se encontraban dentro de los primeros 324 artículos mejor rankeados (ver anexo B.3). Esto significa que revisando los primeros 300 - 350 artículos hubiera sido suficiente para encontrar todos los estudios relevantes, lo cual implica una reducción importante del esfuerzo de filtrado de estudios.

El artículo no encontrado por la herramienta se titula “Experiences using case-based reasoning to predict software project effort” [Kad00], el cual no se encuentra en DBLP y que pertenece a un Journal poco conocido. La tabla 8 muestra un resumen comparativo de los resultados de este caso de estudio:

Tabla 8: Resumen Caso de Estudio 3

Filtrado tradicional de artículos (búsqueda manual)	*Filtrado de artículos utilizando la herramienta desarrollada*
Cantidad de artículos recuperados en la fase inicial: 3.136 Cantidad de artículos que es necesario analizar para encontrar los artículos seleccionados: 2.191 Cantidad de artículos finalmente seleccionados: 84	Cantidad de artículos recuperados en la fase inicial: 17.836 Cantidad de artículos que es necesario analizar para encontrar los artículos seleccionados: 324 Cantidad de artículos finalmente seleccionados: 83

En la Figura 17 se puede apreciar la relación entre la posición de un paper (relevancia del mismo para el estudio) asignada por la herramienta, y el porcentaje los artículos relevantes encontrados. Al primer artículo relevante, considerando la fuente original [Jia12], la herramienta le asignó la posición 3, y al último artículo relevante le asignó la posición 324.

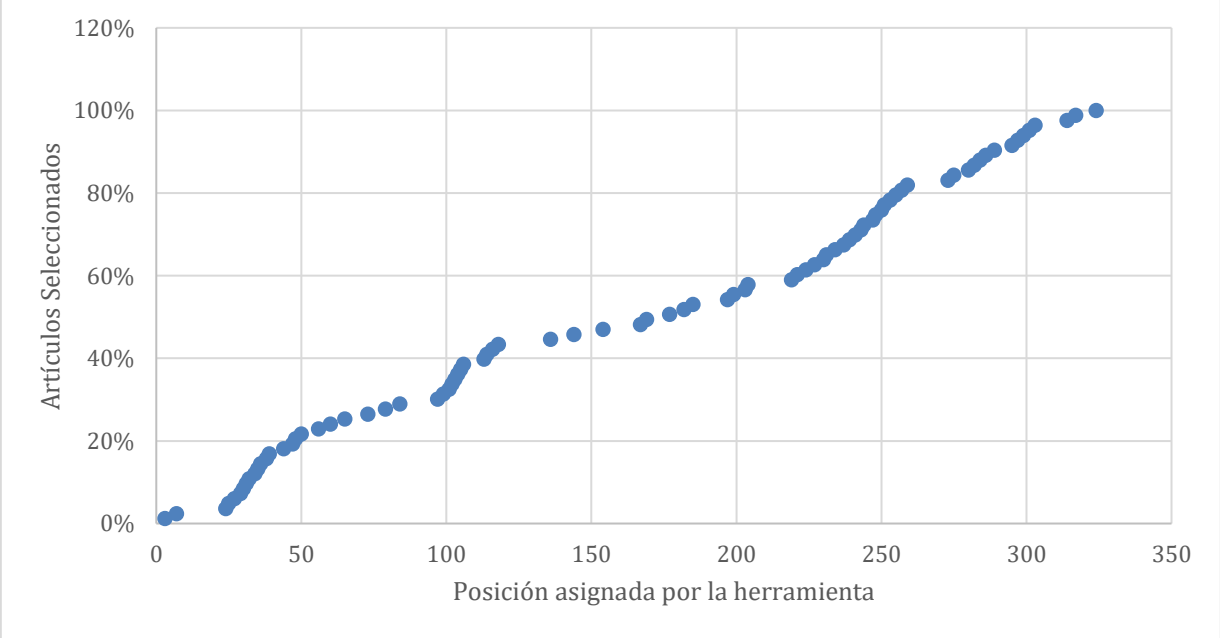

Figura 17: Artículos seleccionados frente a la posición asignada por la herramienta.

Particularmente para esta revisión sistemática de la literatura la cantidad de artículos que se seleccionó fue mayor que las dos primeras RSL, por lo que para encontrar los artículos seleccionados se tiene que llegar al menos a la posición 350 aproximadamente. En el gráfico también se puede apreciar que al evaluar las primeras 150 posiciones se habrá encontrado el 50% de los artículos relevantes seleccionados en el estudio original [Jia12].

Estos resultados, al igual que en las pruebas anteriores, muestran una considerable diferencia a favor de la herramienta, en términos de la cantidad de artículos que es necesario analizar para obtener los seleccionados para su lectura completa.

6.2 Reducción del Esfuerzo y Complejidad de Otras Actividades

Además de la búsqueda y selección de artículos, hay una serie de otras actividades que le agregan esfuerzo y complejidad a una RSL. Muchas de estas actividades se realizan de manera automática en la herramienta desarrollada, reduciendo así el esfuerzo y la complejidad de llevar a cabo una RSL. Aunque formalmente no se midió el esfuerzo o la complejidad de realizar estas tareas en los procesos de RSL que se llevan a cabo de forma manual, contrastando ambos escenarios de trabajo (es decir, con y sin la herramienta propuesta) es posible imaginarse la ganancia que representa el uso de la herramienta propuesta. La siguiente tabla contrasta ambos escenarios de trabajo, como una forma de mostrar otro aspecto de reducción del esfuerzo y complejidad de realizar una RSL debido al uso de la herramienta.

Tabla 9: Comparación de escenarios de trabajo en otras actividades de la RSL

Actividad de la RSL	Escenario Tradicional	Con la Herramienta
Generación de las cadenas de búsqueda.	Manual	Automático
Adaptación del formato de las cadenas de búsqueda, para que se ajuste a lo requerido por el buscador de cada biblioteca digital.	Manual	No requerido
Lanzamiento de cada búsqueda particular	Manual	Automático
Recuperación de resultados y formateo de resultados.	Manual	Automático
Integración de los resultados formateados, al cuerpo de estudio que apoyará a la RSL.	Manual	Automático
Eliminación de estudios duplicados.	Manual	Automático
Rankeado de artículos recuperados.	No soportado	Automático
Participación simultánea y colaborativa de los participantes en el proceso de selección de artículos.	No soportado	Soportado

Estos resultados muestran que más allá de lo que se gana en el proceso de recuperación y filtrado de artículos, que es probablemente la actividad más compleja y pesada, el uso de la herramienta facilita también la realización de varias otras actividades incluidas en una RSL.

6.3 Prueba con Usuarios

Para esta parte de la validación se realizaron dos sesiones de trabajo, en un formato similar a un grupo focal. En estas sesiones se realizó una demostración de la herramienta a académicos y estudiantes de doctorado del Departamento de Ciencias de la Computación (DCC) de la Universidad de Chile. Los participantes contaban con

experiencia en el desarrollo de revisiones sistemáticas de la literatura en el área de las Ciencias de la Computación, por lo que conocían bien el proceso y sus complejidades. La versión de la herramienta utilizada en esta demostración fue anterior a la que se mostró en el Capítulo 5, a la cual se le hicieron mejoras. La demostración de la herramienta se dividió en tres partes:

- *Introducción al software:* Se presentó el proceso de Kitchenham adaptado (propuesto en esta tesis) para realizar una revisión sistemática de la literatura (definido en la Sección 3.1). Además, se explicó qué partes del proceso son apoyadas por el software y qué partes son realizadas de forma automática por la herramienta. Luego se realizó una demostración de la herramienta para que los participantes pudieran valorar los pasos necesarios para realizar una RSL. Esta introducción a la herramienta tuvo como objetivo que los usuarios pudieran evaluar el esfuerzo y las habilidades requeridas para hacer uso de la misma durante un proceso de RSL.

- *Efectividad:* En esta parte se presentaron los resultados obtenidos al replicar una revisión sistemática de la literatura ya reportada en la literatura; mostrando cómo los artículos seleccionados de forma manual fueron clasificados por la herramienta, la cual les asignó un ranking y una posición a cada uno de ellos.

- *Usabilidad:* El objetivo de esta parte es recoger la opinión de los usuarios en cuanto a cómo la herramienta estructura una revisión sistemática de la literatura. Además, se buscó obtener observaciones de los participantes, orientadas a mejorar la usabilidad o utilidad de la misma.

Para esta parte de la validación se pudo contar con la participación de dos académicos y dos estudiantes de doctorado, quienes dieron su opinión de forma verbal. Particularmente se les consultó lo siguiente:

Tabla 10: Resultados de la evaluación con potenciales usuarios

Pregunta	# Rptas positivas	# Rptas negativas
¿El esfuerzo requerido para realizar una revisión sistemática con la herramienta es significativamente menor que el requerido para hacerla de forma tradicional?	4	0
¿Considera que las habilidades requeridas para realizar una revisión sistemática de la literatura con la herramienta son apropiadas para el tipo de usuarios potenciales?	4	0
¿Considera que el funcionamiento de la aplicación es razonable tomando en cuenta el tipo de tarea que se exige?	4	0
¿Considera que realizar una revisión sistemática de la literatura usando la herramienta es más efectivo que haciéndolo de forma tradicional?	4	0
¿Los pasos para realizar una revisión sistemática de la literatura con la herramienta son intuitivos?	4	0

Tal como se ve en la tabla anterior, los comentarios de los participantes fueron positivos, y varias de estas personas se vieron motivadas a hacer uso de la herramienta en el corto plazo. Si bien el número de participantes en esta evaluación no es significativo, los resultados son muy consistentes, mostrando que la propuesta tiene buenas chances de brindar una contribución a esta actividad.

A pesar de los buenos resultados, los participantes dieron comentarios interesantes para mejorar la herramienta. A continuación se detallan las principales observaciones que se recibieron, las cuales en su mayoría son funcionalidades adicionales que se pueden agregar en futuro:

- Agregar la opción de delimitar los años de publicación de los artículos extraídos por la aplicación. Esto permitiría replicar o extender las revisiones sistemáticas de la literatura en cualquier momento que un investigador lo desee.

- Permitir que los usuarios puedan redefinir los términos de búsqueda luego de realizar la extracción de los artículos. Esta observación fue realizada por varios usuarios, y apunta a permitir refinar los términos de búsqueda a través de un proceso iterativo.

- Permitir eliminar, del conjunto de artículos extraídos, aquellos que sean estudios secundarios o terciarios; es decir, aquellos de tipo “systematic mapping”, “surveys” u otras revisiones sistemáticas de la literatura. Esto permitiría que los usuarios no

lo tengan que hacer ese filtrado de forma manual, ya que dichos estudios no son considerados fuentes primarias y por lo tanto deberían ser eliminados.

- Agregar la capacidad de que el sistema derive de forma automática un conjunto inicial de palabras claves para ser usadas en la búsqueda, a partir de las preguntas de investigación. Como solución transitoria, se va a indicar en alguna parte de la aplicación que el registro de las preguntas de investigación tiene como propósito la documentación de la misma, y no derivar las palabras claves usadas en la búsqueda.

- Generar una notificación a los investigadores, cuando éstos estén buscando las mismas palabras clave que se buscaron ya en otra RSL. Esto permitirá reducir el esfuerzo de realizar estos procesos, y eventualmente dar a conocer trabajos (RSLs) de otros investigadores en la misma área.

- Mejorar la representación visual del indicador de completitud usado para mostrar el grado de coincidencia entre un artículo y la cadena de búsqueda. Se recomienda agregar la explicación de lo que representa este indicador en alguna parte del sistema.

- En la metadata de los artículos, resaltar los términos que coincidieron con los strings de búsqueda, con el fin de que los usuarios puedan de forma fácil identificar en qué parte de la metadata están los términos que se definieron.

- Agregar una opción para que los usuarios puedan filtrar los artículos de la revisión sistemática de la literatura, utilizando el campo *autor*.

- Agregar una barra o indicador de progreso de los artículos que faltan revisar; esto con el fin de ayudar a los investigadores a poder determinar de forma rápida cuánto trabajo aún les queda para terminar.

- Agregar la capacidad de exportar la metadata de los artículos a algún formato fácilmente manejable por los académicos.

- Agregar en la etapa de cierre del sistema una sección para registrar las respuestas a las preguntas de la evaluación de la calidad. Esto permitiría dejar de alguna forma registrados los comentarios de los participantes, con el fin de realizar un análisis más detallado de los resultados obtenidos.

Luego de recoger las opiniones de los usuarios, se decidió agregar la opción de delimitar la búsqueda de artículos por años (desde, hasta), siendo algo opcional si los usuarios quieren hacer uso de la misma. También se decidió realizar el cambio para que luego de

que el sistema realice la búsqueda de los artículos que coinciden con los términos definidos por los usuarios, el mismo pueda redefinir los términos y volver a ejecutar la búsqueda sin tener que crear una nueva revisión sistemática. Este cambio en particular se realizó entre la primera y segunda evaluación, y recibió buenos comentarios por parte de los participantes en la segunda evaluación. Dicho cambio permite ajustar (refinar) los términos de búsqueda, dependiendo de los resultados obtenidos en las pruebas iniciales. Así es posible corregir algún error o deficiencia al momento de definir los términos de búsqueda.

Una limitación que fue identificada durante la evaluación de la herramienta es la eventual no-disponibilidad de un estudio primario en la base de datos del sistema, dado que transcurre un cierto tiempo entre que un artículo de publica en la biblioteca digital de la editorial oficial de una editorial, y luego publica en DBLP. A pesar que esta demora se puede considerar marginal, es importante identificarla como una limitación de la herramienta.

6.4 Discusión

Al analizar los resultados obtenidos durante la replicación de revisiones sistemáticas, se puede inferir que la posición que la herramienta asigna a cada artículo por medio del ranking es muy útil durante la etapa de selección. Esta minimiza de forma significativa el total de artículos que son necesarios analizar para encontrar los relevantes con respecto a los términos de la investigación.
En general para revisiones sistemáticas analizadas, se encontraron que a los artículos relevantes la herramienta los ubicó dentro de las primeras 330 – 350 posiciones, ordenados por medio del ranking. Aunque éste no es un número fijo, muestra que la cantidad de artículos a revisar en el primer proceso de filtrado es significativamente menor que los reportados en la mayoría de las RSL publicadas en la literatura.

Otro aspecto importante observado durante las pruebas, es la relevancia del uso del ranking que asigna la herramienta a los artículos. Al usar este indicador en las pruebas realizadas se puede ver que dentro de los primeros 100 a 150 artículos mejor rankeados, se encontraron entre el 50 y 60% de los artículos seleccionados en los estudios originales. Por lo tanto, el uso de este ranking minimiza el esfuerzo que los investigadores deben realizar para encontrar los artículos relevantes a las preguntas de investigación.

Tal como se mostró en la sección 6.2, hay muchas otras actividades que forman parte de una RSL, que son automatizadas por la herramienta. Esto reduce aún más el esfuerzo y la complejidad de llevar a cabo una RSL.

Durante la validación con usuarios, luego que se les presentaron los pasos a seguir para crear una RSL con la herramienta, ellos concluyeron que el ahorro de tiempo debido al

uso de la aplicación era significativo durante las tareas de extracción, clasificación de la metadata y eliminación de duplicados. El tiempo total que toma realizar estas tres actividades utilizando la herramienta es usualmente menos de 10 minutos, una vez que se tienen los términos búsqueda a utilizar y los respectivos sinónimos de dichos términos. Esto contrasta significativamente con el tiempo que se tendría que invertir si los investigadores tuvieran que hacer estas actividades de forma tradicional (manual), ajustando las cadenas de búsqueda a cada biblioteca digital, extrayendo la metadata, la eliminando de duplicados y clasificándolos, lo que podría durar varias semanas de una persona trabajando a tiempo completo.

En cuanto a la usabilidad de la herramienta, a partir de los comentarios que realizaron los usuarios se puede afirmar que la herramienta es lo suficientemente coherente y entendible como para poder hacer uso de ella de forma intuitiva. Los usuarios resaltaron aspectos como el orden en que se muestra la información, o la forma colaborativa en la que ellos pueden trabajar.

De las pruebas con usuarios se puede inferir también que en general los académicos calificaron la herramienta como muy útil, y con mucho potencial a corto plazo. Teniendo en cuenta que cada vez es más común tener alumnos de doctorado realizando revisiones sistemáticas la literatura, la herramienta les podría ser de gran ayuda, permitiéndoles concentrarse más en las etapas de análisis y dejándole a la aplicación el resto de las actividades involucradas en la depuración inicial de estudios. En el caso de las observaciones que realizaron los académicos, la mayoría se orientaban a proponer funciones adicionales que se pueden automatizar con el fin de aumentar los servicios que la herramienta brinda.

7. Conclusiones y Trabajo a Futuro

Acontinuación se presenta un resumen del trabajo realizado, el impacto esperado de la solución desarrollada, las lecciones aprendidas y el trabajo a futuro.

7.1. Trabajo Realizado

Como resultado del trabajo realizado en este proyecto de tesis se logró realizar:

- Un análisis del proceso de RSL, sus implicaciones, sus principales problemas y limitantes.

- Analizar las principales soluciones que existen en el mercado, las cuales intentan apoyar este proceso. De estas herramientas también se identificaron fortalezas y debilidades.

- Se analizaron las principales bibliotecas digitales en el área de las ciencias de la computación y sus posibles formas de integración, dando como resultado la decisión de integrar esta herramienta con la base de datos de DBLP [DBL18] y con la API de Springer.

- Se definió un proceso para realizar RSL, a partir del definido por Kitchenham [Kit07], para que sirviera como base para la herramienta.

- Se definieron las tecnologías a emplear y las estrategias de diseño que se utilizarían en el desarrollo de la herramienta. Como resultado se desarrollaron diversos servicios integrados en un sistema que apoya las diferentes etapas del proceso definido.

- Se realizaron pruebas de validación de los resultados que brinda la herramienta, replicando varias RSL y comparando los resultados obtenidos en ambos escenarios. Se realizaron además sesiones de evaluación de la aplicación con usuarios expertos, con el fin de recibir retroalimentación y observaciones con respecto al sistema desarrollado.

7.2. Impacto de la Solución

El objetivo de este proyecto de tesis fue desarrollar una herramienta que apoyara a los investigadores durante la realización de revisiones sistemáticas de la literatura en el área de la computación. Este objetivo fue alcanzado al desarrollar e integrar los siguientes componentes que formaron parte de la solución:

- Un servicio que genera, a partir de los términos y sinónimos definidos por los investigadores, las cadenas de búsqueda para recuperar los estudios primarios considerados en una RSL.

- Un servicio que automatiza la búsqueda de estudios primarios a partir de las cadenas de búsqueda, permitiendo una depuración y clasificación inicial rápida y eficiente, reduciendo de forma significativa el esfuerzo de realizar esta tarea.

- Un servicio que gestiona la metadata de los estudios primarios, con el fin de presentar a los investigadores los artículos extraídos de forma ordenada de acuerdo al grado de coincidencia con los términos definidos. Esto aumenta la eficiencia en el proceso, minimizando la cantidad de artículos que se deben analizar para encontrar los más relevantes con respecto a las preguntas de investigación.

Al analizar el funcionamiento de la herramienta desarrollada se puede ver que esta reduce de forma significativa el tiempo requerido por los investigadores en la búsqueda, extracción y clasificación de artículos. Además, el uso de la herramienta minimiza el riesgo de cometer errores humanos en este proceso, y logrando reducir la complejidad y el esfuerzo que representa realizar este proceso de forma manual.

7.3. Lecciones Aprendidas

Tras dos años de trabajar en este proyecto me es posible extraer una serie de aprendizajes, de los cuales considero importante mencionar algunos de ellos:

- Fue fundamental para el éxito de este proyecto el seguimiento y el análisis exhaustivo que se le dio por parte de los involucrados, teniendo reuniones semanales con el fin de evaluar los riesgos y oportunidades en el diseño y desarrollo de la herramienta.

- La integración de los contenidos, técnicas y tecnologías vistos durante los diplomados de Ingeniería de Software y Ciencia de Datos fue muy importante para lograr que la herramienta pudiese ser eficiente y eficaz. Entre los tópicos más útiles para el desarrollo de esta tesis se puede mencionar el uso de bases de datos no relacionales, uso de índices externos con el fin dar eficiencia a las búsquedas, y el desarrollo de microservicios, entre otros.

- Considero que el enfoque temprano que se le dio al proyecto de tesis por parte del programa de magíster, así como de sus académicos, jugó un papel muy importante para poder concluir el proyecto en el tiempo establecido.

7.4. Trabajo a Futuro

De la validación con usuarios se puede inferir que existe mucho potencial para la evolución de la herramienta a medida que los usuarios puedan entregar más retroalimentación. Esto ayudará a determinar los aspectos a mejorar o funcionalidades que se le puedan seguir agregando con el fin de poder ampliar los servicios que hasta ahora brinda la herramienta.

En cuanto a los aspectos internos de la herramienta, se tiene que en un futuro se le pueda agregar más funcionalidades a parte de las que se mencionaron en la Sección 6.3, entre las que se pueden mencionar:

- Integrar más fuentes de metadata aparte de Mendeley, con el fin de extraer la metadata de los artículos que no sea posible encontrar en Mendeley.

- Crear un servicio que pueda extraer y sugerir los términos con sus sinónimos a partir de las preguntas de investigación. En una primera instancia, esta generación de sinónimos deberá estar asociada al área de las ciencias de la computación.

8. Bibliografía

[ABS18] Abstrackr (Beta). <http://abstrackr.cebm.brown.edu>. [Último acceso en diciembre de 2018].

[ACM18] ACM Digital Library. <http://dl.acm.org>. [Último acceso en Diciembre de 2018].

[Bri14] Britto, R., Freitas, V., Mendes, E., Usman, M., Effort Estimation in Global Software Development: A Systematic Literature Review. 10.1109/ICGSE.2014.11. 2014

[Bow12] Bowes, D., Hall, T., and Beecham, S. SLuRp: a tool to help large complex systematic literature reviews deliver valid and rigorous results. In Proceedings of the 2nd international wokshop on Evidential assessment of software technologies. 2012.

[COL18] Colandr. <https://www.colandrapp.com/signin>. [Último acceso en diciembre de 2018].

[DBL18] dblp computer science bibliography. <https://dblp.uni-trier.de>. [Último acceso en Diciembre de 2018]

[DCC18] Departamento de Ciencias la Computación (DCC). Investigación en el DCC. Departamento de Ciencias la Computación, FCFM, Universidad de Chile. <https://www.dcc.uchile.cl/investigacion>. [Último acceso en diciembre de 2018].

[Dko14] Ko, D., Kim, S.T., Park, S. Resource versioning scheme in evolutionary software product line, Int. J. Softw. Eng. Appl. 8 (2) 113–126, 2014

[DOC18] DoctorEvidence. <https://drevidence.com/>. [Último acceso en diciembre de 2018].

[Fel17] Felizardo, K., Souza, E., Falbo, Ricardo., Lankalapalli V., Nandamudi, Mendes, E., Nakagawa, E., (2017). Defining Protocols of Systematic Literature Reviews in Software Engineering: A Survey. 202-209. 10.1109/SEAA.2017.17.

[Fer10] Fernández-Sáez, M. G. Bocco, F.P. Romero. RSL-tool - a tool for performing systematic literature reviews. In Proceedings of the 2010 International Conference on Software and Data Technologies. 2010.

[Fer11] Ferreira-González, I., Urrutia, G. y Coello, P. Revisiones sistemáticas y meta análisis: bases conceptuales e interpretación. Madrid: Revista Española de Cardiología. 2011.

[Gar17] García-Peñalvo, F. Revisión Sistemática de Literatura en los Trabajos de Final de Máster y en las Tesis Doctorales, España, Salamanca: Universidad de Salamanca. 2017.

[Gre11] Gree, S. and Higgins, J. Cochrane Handbook for Systematic Reviews of Interventions, S.l. School of Social and Community Medicine, University of Bristol. 2011.

[Gui15] Guirao, G., Silamani, J.A. Utilidad y tipos de revisión de literatura. Ene 9(2), ISSN: 1998-348X. 2015.

[Her12] Hernandes, E., Zamboni, A., Fabbri, S., and Di Thommazo, A. Using GQM and TAM to evaluate StArt – a tool that supports Systematic Review. CLEI Electronic Journal. 2012.

[IEE18] IEEE Xplore Digital Library. < https://www.ieee.org/index.html>. [Último acceso en diciembre de 2018].

[Jia12] Jianfeng W., Shixian L., Zhiyong L., Yong H., Changqin H. Systematic literature review of machine learning based software development effort estimation models. Department of Computer Science, Sun Yat-sen University, Guangzhou, China. 2012.

[Kad00] Kadoda, G., Cartwright, M., Chen, L., Shepperd, M. Experiences using case-based
reasoning to predict software project effort, in: Proceedings of the Conference on Evaluation and Assessment in Software Engineering, Keele University, UK, 2000.

[Kit04] Kitchenham, B. Procedures for Performing Systematic Reviews. Australian: National ICT Australia Ltd, ISSN: 1353-7776. 2004.

[Kit07] Kitchenham, B., Charters, S., Budgen, D., Brereton, P., Turne, M.S, Magne, J., Mendes, E., Visaggio, G. Guidelines for performing Systematic Literature Reviews in Software Engineering, Version 2.3. EBSE Technical Report EBSE-2007-01. Keele University and University of Durham. July, 2007.

[Kit06] Kitchenham, B., Mendes, B., Travassos G., A systematic review of cross- vs. within- company cost estimation studies, EASE'06 Proceedings of the 10th international conference on Evaluation and Assessment in Software, 2006

[Luc18] Apache Lucene Core. <https://lucene.apache.org/core/>. [Último acceso en Diciembre de 2018].

[Mar18] Marques, M., Simmonds, J., Rossel, P.O, Bastarrica, M.C. Software product line evolution: A systematic literature review. Information and Software Technology 105 (2019) 190–208. 2018.

[Mart01] Martínez-Richart, M., Cabrero-García, J., Tosal-Herrero, B. Romá-Ferri, M. y Morneo-Vizcaya, M. Búsqueda bibliográfica en enfermería y otras ciencias de la salud. Alacant: Publicaciones Universidad de Alicante. 2001.

[Men18] Mendeley Reference. <https://www.mendeley.com/>. [Último acceso en diciembre de 2018].

[Mit13] Mitchell Ryan, Instant Web Scraping with Java, ISBN 978-1-84969-688-3, Birmingham B3 2PB, UK. Packt Publishing. 2013.

[Mon18] MongoDB. <https://www.mongodb.com/es>. [Último acceso en diciembre de 2018].

[Oko10] Okoli, C., Schabram, K., (2010). A Guide to Conducting a Systematic Literature Review of Information Systems Research. SSRN Electronic Journal. 10. 10.2139/ssrn.1954824.

[PAR18] Parsifal. <https://parsif.al/>. [Último acceso en diciembre de 2018].

[Pin06] Pino, F., García, F. y Piattini, M. Revisión Sistemática de Mejora de Procesos de Software en Micro, Pequeñas y Medianas Empresas. ISSN (Versión en línea): 1885-4486, Madrid: REICIS Revista Española de Innovación, Calidad e Ingeniería del Software, Vol. 2. 2006.

[Ria10] Riaz, M., Sulayman, M., Salleh, N., Mendes, Emilia. (2010). Experiences conducting systematic reviews from novices' perspective.

[RAY18] Rayyan. <https://rayyan.qcri.org/>. [Último acceso en diciembre de 2018].

[SES18] SESRA. <http://sesra.net/>. [Último acceso en diciembre de 2018].

[Sha96] Shaw, M., David, G. Software architecture: Perspectives on an Emerging Discipline. Prentice Hall. ISBN:0131829572. 1996.

[Sci18] Science Direct, Explore scientific, technical, and medical research on ScienceDirect. <http://www.sciencedirect.com/>. [Último acceso en diciembre de 2018].

[Soo16] Soomro, A., Salleh, N., Mendes, E., Grundy, J., Burch, G., Nordin, A., (2016). The Effect of Software Engineers' Personality traits on Team Climate and Performance: a Systematic Literature Review. Information and Software Technology. 73. 10.1016/j.infsof.2016.01.006.

[Spr18] Springer Link. <https://link.springer.com/>. [Último acceso en diciembre de 2018].

[SRD18] SRDB.PRO. <https://www.srdb.pro/default>. [Último acceso en diciembre de 2018].

[Srt18] SRToolBox. <http://systematicreviewtools.com/>. [Último acceso en diciembre de 2018].

[Teo11] Teorey, T., Lightstone, S., Jagadish T., Database Modeling and Design, 5th Edition, ISBN: 9780123820211, Morgan Kaufmann. 2011

[TEX18] Textpresso. <http://www.textpresso.org/index.html>. [Último acceso en diciembre de 2018].

[Ver18] Vera, T., Ochoa, S.F., Perovic, D. Survey of Software Development Effort Estimation Taxonomies, Technical Report TR/DCC-2018-2, Computer Science Department, University of Chile. March 2018.

[Web18] Web Of Science. <https://login.webofknowledge.com>. [Último acceso en diciembre 2018].

[Whi14] Whittemore, R; Chao, A; Jang, M; Minges, K.E. y Park, C. Methods for knowledge synthesis: an overview. Heart Lung. 43,5, pp. 453-61. 2014.

Anexo A: Ejemplo de construcción de las cadenas de búsqueda

El siguiente ejemplo muestra de la generación de las cadenas de búsqueda, la cual está acotada para fines prácticos con sólo algunos términos y sinónimos. Los términos a buscar son los siguientes: software, effort, development

Los sinónimos de cada término, a incluir en las cadenas de búsqueda, son los siguientes:

- software: program*, application, package
- effort: measurement, cost, resource
- development: elaboration, working out, improvement

Estos términos son enlazados utilizando operadores AND, y los sinónimos se enlazan con operadores OR. Por lo tanto, la cadena resultante sería la siguiente: (software OR program* OR application OR package) AND (effort OR measurement OR cost OR resource) AND (development OR elaboration OR “working out” OR improvement).

La cadena de búsqueda podría ser útil si las bibliotecas digitales donde se las va aplicar soportan el uso de wildcards, búsqueda de frases y operadores lógicos, de lo contrario la cadena se deberá descomponer en varias equivalentes. Por ejemplo, si una biblioteca digital soporta wildcards, el uso de frases y el operador AND, pero no soporta el operador OR, para realizar la búsqueda de los términos y sinónimos seleccionados se necesitarían 64 (4 × 4 × 4) búsquedas distintas: (software AND effort AND development), (software AND effort AND elaboration), etc.

Anexo B: Artículos seleccionados en las revisiones sistemáticas de la literatura utilizadas en la etapa de validación.

B.1 Caso de Estudio 1

A continuación se detalla el título cada artículo seleccionado en la RSL “Software product line evolution: A systematic literature review” [Mar18], con su respectivo ranking y posición dadas por la herramienta:

Título del estudio	*Ranking*	*Posición*
Requirements Prioritization Decision Rule Improvement for Software Product Line Evolution.	104	1
Requirements Evolution in Software Product Lines: An Empirical Study.	104	7
Partially safe evolution of software product lines.	104	9
A Case Study on the Evolution of a Component-based Product Line.	104	10
Structuring the modeling space and supporting evolution in software product line engineering.	104	14
Linear Evolution of Domain Architecture in Service-Oriented Software Product Lines.	101	21
Towards understanding requirement evolution in a software product line an industrial case study.	101	23
Software Product Line Evolution with Cardinality-Based Feature Models.	101	25
Restructuring Variability in Software Product Lines using Concept Analysis of Product Configurations.	71	28
EvoFM: feature-driven planning of product-line evolution.	71	29
Evolving software requirements and architectures using software product line concepts.	71	30

Flexible support for managing evolving software product lines.	71	32
An Environment for Managing Evolving Product Line Architectures.	71	33
Automatic and Incremental Product Optimization for Software Product Lines.	71	34
Towards feature-driven planning of product-line evolution.	71	35
Evolving a Product Family in a Changing Context.	71	36
Integrated Management of Variability in Space and Time in Software Families.	71	38
Guaranteeing Configuration Validity in Evolving Software Product Lines.	71	40
Understanding Feature Evolution in a Family of Product Variants.	71	41
Supporting Model Maintenance in Component-based Product Lines.	71	42
Towards a Solution for Change Impact Analysis of Software Product Line Products.	71	43
Leveraging variability modeling to address metamodel revisions in Model-based Software Product Lines.	71	46
Architectural evolution of FamiWare using cardinality-based feature models.	71	49
A theory of software product line refinement.	71	51
DarwinSPL: an integrated tool suite for modeling evolving context-aware software product lines.	71	39
Coevolution of variability models and related software artifacts - A fresh look at evolution patterns in the Linux kernel.	68	62
Extracting and Evolving Mobile Games Product Lines.	68	64
Building reliable and maintainable Dynamic Software Product Lines: An investigation in the Body Sensor Network domain.	68	65

On the Use of Variability Operations in the V-Modell XT Software Process Line.	66	69
Model-driven support for product line evolution on feature level.	44	73
State-Based Modeling to Support the Evolution and Maintenance of Safety-Critical Software Product Lines.	41	80
Investigating the safe evolution of software product lines.	41	85
Codifying architecture knowledge to support online evolution of software product lines.	41	90
A Software Modeling Odyssey: Designing Evolutionary Architecture-Centric Real-Time Systems and Product Lines.	41	99
Higher-order delta modeling for software product line evolution.	41	101
SPLEMMA: a generic framework for controlled-evolution of software product lines.	41	108
Model-based product line evolution: an incremental growing by extension.	41	111
Co-evolution of models and feature mapping in software product lines.	41	117
Variability evolution and erosion in industrial product lines: a case study.	41	120
Model-driven planning and monitoring of long-term software product line evolution.	41	122
Achieving Knowledge Evolution in Dynamic Software Product Lines.	41	123
Evolution support mechanisms for software product line process.	41	128
Issues in software product line evolution: complex changes in variability models.	38	144
Making Software Product Line Evolution Safer.	38	154
Formal Definition of Feature Models to Support Software Product Line Evolution.	38	160

Supporting Evolution in Model-Based Product Line Engineering.	38	163
Evolution in software product lines: two cases.	38	182
Supporting Online Updates of Software Product Lines: A Controlled Experiment.	11	194
Maintaining software product lines - an industrial practice.	11	197
Refactoring the Documentation of Software Product Lines.	11	199
From product architectures to a managed automotive software product line architecture.	11	201
Aligning Coevolving Artifacts Between Software Product Lines and Products.	11	208
An Architectural Approach to Support Online Updates of Software Product Lines.	11	210
Components meet aspects: Assessing design stability of a software product line.	11	214
A prototype-based approach for managing clones in clone-and-own product lines.	11	222
A Feature Model Based Framework for Refactoring Software Product Line Architecture.	11	225
Capturing variability in space and time with hyper feature models.	9	229
CompAS: A new approach to commonality and variability analysis with applications in computer assisted orthopaedic surgery.	9	234
Variability Change Management Using the Orthogonal Variability Model-Based Traceability.	9	235
Resource versioning scheme in evolutionary software product line.	N/A	N/A

B.2 Caso de Estudio 2

A continuación se detalla el título cada artículo seleccionado en la RSL “Survey of Software Development Effort Estimation Taxonomies” [Ver18], con su respectivo ranking y posición signados por la herramienta:

Título del estudio	*Ranking*	*Posición*
Analogy-based software development effort estimation: A systematic mapping and review.	86528	7
Systematic literature review of machine learning based software development effort estimation models.	86528	8
A Systematic Review of Software Development Cost Estimation Studies.	86528	11
Systematic literature review of ensemble effort estimation.	66521	21
A review of studies on expert estimation of software development effort.	22720	43
A Review of Surveys on Software Effort Estimation.	20993	65
Software development cost estimation approaches - A survey.	20993	72
Effort estimation in agile software development: a systematic literature review.	20738	134
Effort estimation in global software development - a systematic review.	20738	148
Software cost estimation meets software diversity.	2713	199
Data Mining Techniques for Software Effort Estimation: A Comparative Study.	2713	246
The Use of Simulation Techniques for Hybrid Software Cost Estimation and Risk Analysis.	2713	269
Software resources estimation	2713	281

Integrating non-parametric models with linear components for producing software cost estimations.	2713	285
Software economics: status and prospects.	2713	291
Reliability and Validity in Comparative Studies of Software Prediction Models.	2713	293
Software development cost estimation: A survey.	N/A	N/A

B.3 Caso de Estudio 3

A continuación se detalla el título cada artículo seleccionado en la RSL Systematic literature review of machine learning based software development effort estimation models [Jia12], con su respectivo ranking y posición asignados por la herramienta:

Título del estudio	*Ranking*	*posición*
Improve Analogy-Based Software Effort Estimation Using Principal Components Analysis and Correlation Weighting.	86528	3
Filtering of Inconsistent Software Project Data for Analogy-Based Effort Estimation.	86528	7
Integration of the grey relational analysis with genetic algorithm for software effort estimation.	86528	24
A study of the non-linear adjustment for analogy based software cost estimation.	86528	25
LSEbA: least squares regression and estimation by analogy in a semi-parametric model for software cost estimation.	86528	27
Improved estimation of software project effort using multiple additive regression trees.	86528	29
Quasi-optimal case-selective neural network model for software effort estimation.	86528	30
A study of mutual information based feature selection for case based reasoning in software cost estimation.	86528	31
An empirical validation of a neural network model for software effort estimation.	86528	32

Estimation of software project effort with support vector regression.	86528	34
Can genetic programming improve software effort estimation? A comparative evaluation.	86528	35
Optimization of analogy weights by genetic algorithm for software effort estimation.	86528	36
Improving analogy-based software cost estimation by a resampling method.	86528	38
GA-based method for feature selection and parameters optimization for machine learning regression applied to software effort estimation.	86528	39
The adjusted analogy-based software effort estimation based on similarity distances.	86528	44
Software development cost estimation using wavelet neural networks.	86528	47
A study of project selection and feature weighting for analogy based software cost estimation.	86528	48
Ensemble of neural networks with associative memory (ENNA) for estimating software development costs.	86528	50
Analogy-X: Providing Statistical Inference to Analogy-Based Software Cost Estimation.	86528	56
A flexible method for software effort estimation by analogy.	86299	60
Bagging Predictors for Estimation of Software Project Effort.	66521	65
A Probabilistic Model for Predicting Software Development Effort.	66521	73
A Simulation Tool for Efficient Analogy Based Cost Estimation.	66521	79
An investigation of artificial neural networks based prediction systems in software project management.	66521	84
Comparison of estimation methods of cost and duration in IT projects.	65828	97
A Pattern Recognition Approach for Software Engineering Data Analysis.	65828	99

Predicting project delivery rates using the Naive-Bayes classifier.	65795	101
Modeling Development Effort in Object-Oriented Systems Using Design Properties.	65795	102
Improving the COCOMO model using a neuro-fuzzy approach.	65793	103
Reliability and Validity in Comparative Studies of Software Prediction Models.	65599	104
Further Comparison of Cross-Company and Within-Company Effort Estimation Models for Web Applications.	65576	105
Cross-company vs. single-company web effort models using the Tukutuku database: An extended study.	65566	106
Software development cost estimation: Integrating neural network with cluster analysis.	22720	113
Comparison of artificial neural network and regression models for estimating software development effort.	22720	114
Software effort estimation by analogy and "regression toward the mean".	22720	116
Machine Learning Approaches to Estimating Software Development Effort.	22720	118
Software Cost Estimation Models Using Radial Basis Function Neural Networks.	20993	136
Applying fuzzy neural network to estimate software development effort.	20993	144
A comparison of software effort estimation techniques: Using function points with neural networks, case-based reasoning and regression models.	20993	154
Impact Analysis of Missing Values on the Prediction Accuracy of Analogy-based Software Effort Estimation Method AQUA.	20738	167
Software Effort Estimation using Machine Learning Techniques with Robust Confidence Intervals.	20738	169
Software Effort Prediction Using Regression Rule Extraction from Neural Networks.	20738	177

On the problem of the software cost function.	20738	182
A Controlled Experiment to Assess the Benefits of Estimating with Analogy and Regression Models.	20738	185
Estimating Software Project Effort Using Analogies.	20738	197
A comparison of techniques for developing predictive models of software metrics.	20738	199
Fuzzy case-based reasoning models for software cost estimation	20738	203
Using Artificial Neural Networks and Function Points to Estimate 4GL Software Development Effort.	20738	204
An Empirical Study of Analogy-based Software Effort Estimation.	20738	219
Human Performance Estimating with Analogy and Regression Models: An Empirical Validation.	20738	221
An empirical analysis of software effort estimation with outlier elimination.	20738	224
Fuzzy Decision Tree Approach for Embedding Risk Assessment Information into Software Cost Estimation Model.	20738	227
Genetic programming for effort estimation: an analysis of the impact of different fitness functions	20738	230
Analysis of attribute weighting heuristics for analogy-based software effort estimation method AQUA+.	20738	231
Empirical Data Modeling in Software Engineering Using Radical Basis Functions.	20738	234
An investigation of machine learning based prediction systems.	2713	237
Hybrid Intelligent Design of Morphological-Rank-Linear Perceptrons for Software Development Cost Estimation.	2713	239
Combining probabilistic models for explanatory productivity estimation.	2713	241
Using Public Domain Metrics To Estimate Software Development Effort.	2713	243

How Valuable is company-specific Data Compared to multi-company Data for Software Cost Estimation?	2713	244
Comparing Software Prediction Techniques Using Simulation.	2713	247
Software Project Similarity Measurement Based on Fuzzy C-Means.	2713	248
Software cost estimation using an Albus perceptron (CMAC).	2713	250
On the use of Bayesian belief networks for the prediction of software productivity.	2713	251
Estimating software development effort with connectionist models.	2713	253
Improving analogy software effort estimation using fuzzy feature subset selection algorithm.	2713	255
BBN based approach for improving the software development process of an SME - a case study.	2713	257
How effective is Tabu search to configure support vector regression for effort estimation?	2713	259
An Assessment and Comparison of Common Software Cost Estimation Modeling Techniques.	2484	273
Bayesian Network Models for Web Effort Prediction: A Comparative Study.	2484	275
Investigating the use of Support Vector Regression for web effort estimation.	2484	280
Neuro-genetic prediction of software development effort.	2484	282
Combining techniques to optimize effort predictions in software project management.	2484	284
Examining the Feasibility of a Case-Based Reasoning Model for Software Effort Estimation.	2458	286
Software effort estimation based on weighted fuzzy grey relational analysis.	2458	289
Optimal Project Feature Weights in Analogy-Based Cost Estimation: Improvement and Limitations.	2458	295

A replicated assessment and comparison of common software cost modeling techniques.	2458	297
Effort estimation modeling techniques: a case study for web applications.	2458	299
A Comparison of Techniques for Web Effort Estimation.	2458	301
Further Investigation into the Use of CBR and Stepwise Regression to Predict Development Effort for Web Hypermedia Applications.	2458	303
A comparative study of two software development cost modeling techniques using multi-organizational and company-specific data.	2458	314
A Comparative Study of Cost Estimation Models for Web Hypermedia Applications.	2458	317
Software Metrics Data Analysis - Exploring the Relative Performance of Some Commonly Used Modeling Techniques.	2458	324
Experiences using case-based reasoning to predict software project effort.	N/A	N/A