Universidad Central “Marta Abreu” de Las Villas.
Facultad Matemática, Física y Computación
Licenciatura en Ciencia de la Computación

Trabajo de Diploma

Darkaiv

SISTEMA PARA LA EXTRACCIÓN AUTOMÁTICA Y PUBLICACIÓN DE METADATOS DE PUBLICACIONES CIENTÍFICAS

Autores:
Felipe Antonio Enriquez Rodríguez
Luis Daniel Hernández Morales

Tutor:
Dr. Didiosky Benítez Erice

Santa Clara, 2016

Hacemos constar que el presente trabajo fue realizado en la Universidad Central "Marta Abreu" de Las Villas como parte de la culminación de los estudios de la especialidad de Ciencia de la Computación, autorizando a que el mismo sea utilizado por la institución, para los fines que estime conveniente, tanto de forma parcial como total y que además no podrá ser presentado en eventos ni publicado sin la autorización de la Universidad.

_____ _____

Firma de los Autores

Los abajo firmantes, certificamos que el presente trabajo ha sido realizado según acuerdos de la dirección de nuestro centro y el mismo cumple con los requisitos que debe tener un trabajo de esta envergadura referido a la temática señalada.

_____ _____

Firma del Tutor

Firma del Jefe de Departamento

No necesito saberlo todo, tan sólo necesito saber dónde encontrar aquello que me hace falta, cuando lo necesite.
-Albert Einstein-

La suerte de la que mucho se habla realmente no existe, solo se confunde con la combinación de preparación y oportunidad.
-Anónimo-

Dedicatoria

A mis padres, por amarme desde el día en que nací y por darme todo el apoyo para hacer realidad mis sueños.

A mi hermana Yumislaine: hablar del amor de hermanos es redundar.

A mis primos Douglas y Sandor, sin ellos no hubiese sido posible llegar hasta aquí.

A Amaya, lo más hermoso que me ha dado la vida. Te amo.

Felipe Antonio Enriquez Rodríguez

Dedicatoria

A mi hermano Javie, por hacerme ver el mundo de una forma diferente y ser la causa de muchas de mis decisiones, sin que me arrepienta de ninguna.

Luis Daniel Hernández Morales

Agradecimientos

A mis abuelas Isabel y Noelia, que son mis otras madres.

A mis tías todas: Isabel, Adela, Carmen y Sonia. A mis dos tíos Carlos. A mi prima Sandra. A mi tía Teresa, que le hubiese encantado verme graduado.

A mi tío Israel, que no está todos los días pero que su gesto siempre llega en el momento más preciado.

A mi primo Carlos Alberto, médico de la familia las 24 horas del día.

Al resto de mi familia, por el apoyo.

A mi madrina Mireya, que a pesar de la distancia jamás se ha olvidado de mí.

A mis amigos de antes: Pedro, Ariam, Tony y Armando, más que amigos, hermanos varones que la vida me dio.

A Rafa, Arturo y Ernesto, mi infinito agradecimiento por aparecer de uno en uno y convertirse en parte importante de mi vida. Gracias por ayudarme a que la universidad no fuese solo estudios.

A mis suegros Carmen y Eduardo, que me quieren como un hijo.

A mi tutor Dr. Didiosky Benítez Erice, por su asesoría y orientación.

A todos los profesores que alguna vez me impartieron clases, por haberme ayudado a convertirme en lo que soy.

Felipe Antonio Enriquez Rodríguez

Agradecimientos

A mi padre, por ser un ejemplo a seguir, aunque muchas cosas no las veamos igual.

A mi madre, por todos los "te amo" de las conversaciones por teléfono, aunque ella diga que los míos son para seguirle el juego.

A mi hermano, por ir a despertarme cada fin de semana con una sonrisa, incluso cuando necesitaba dormir un poco más: ¡te quiero Javi!!!

A mis abuelos, por estar pendiente a todo lo que necesitaba.

A mis tíos, primos y resto de la familia.

A mis amigos, por no faltar cuando había que hablar de algo importante (o a veces sin importancia), tomar, joder, jugar futbol, salir, cogerla con alguien… recuerden que faltó jugar futbol al frente del rectorado y disparar el cañón del club.

A mi tutor, por darme la oportunidad de hacer un trabajo que fue un reto y que realmente quería hacer.

A los profesores, que año tras año forman hombres y mujeres además de profesionales.

A todos los que un día tuve y tengo como paradigmas.

A todos los que ayudaron a forjar mi carácter y forma de pensar: con los que me llevé bien, pero muy especialmente con los que no, que me ayudaron a comprender la necesidad de crecer, ya que esto no se logra siempre por el camino más fácil.

¡A los que se fueron, a los que quedaron… y a los que vendrán!!!

A Andy, por creer en la idea del vínculo entre nuestra carrera y la de Ciencias de la Información, esfuerzo al que todavía se le debe bastante, aunque el último día te dejé embarcada.

En fin, ¡gracias a todos!!!

Luis Daniel Hernández Morales

Resumen

En la Universidad Central "Marta Abreu" de Las Villas (UCLV) se han llevado a cabo varias iniciativas orientadas a la creación de una biblioteca digital universitaria. El problema principal con estas iniciativas ha radicado en que el proceso de creación de estas bibliotecas se ha apoyado fundamentalmente en el trabajo manual, tedioso y sujeto a errores. Hasta la fecha solo se han logrado publicar cantidades poco representativas del volumen de información digital que existe en la institución. Por esto es que se necesita desarrollar una herramienta informática que permita la extracción de metadatos de documentos científicos y su publicación en una biblioteca digital.

Darkaiv es un producto informático dirigido a la extracción automática de metadatos y publicación de documentos de carácter científico. El presente trabajo está orientado a explicar las características de esta herramienta, así como sus principales funcionalidades e interacción con el usuario. Para ello, se describen las etapas de análisis de requerimientos, diseño e implementación, además de los patrones de diseño utilizados para obtener una solución eficiente y escalable.

Como resultado de la investigación se obtiene una herramienta que posibilita el manejo de grandes colecciones de documentos para la conformación de bibliotecas digitales, combina la extracción automática (utilizando varias tecnologías) con la revisión manual de los metadatos y brinda una evaluación de los metadatos atendiendo a su completitud.

Abstract

Many information management specialists at the Central University "Marta Abreu" of Las Villas (UCLV) have undertaken several initiatives aimed at creating a University digital library. The main issue with these initiatives has been that the process of creating these libraries is manual, tedious and error-prone work. So far, they have been able to publish an unrepresentative volume of all the digital information available in the institution. This is why we need to develop a software tool that allows the metadata extraction and publishing scientific papers in a digital library.

Darkaiv is a software product aimed at the automatic metadata extraction and the publishing of scientific papers. This work is oriented to explain the features of this tool and its interactions with the users. In order to do this, the stages of requirements analysis, design and implementation have been described, also, the design patterns used for an efficient and scalable solution.

As a result of the investigation, a tool is obtained that enables handling of large document collections for the creation of digital libraries, combines automatic extraction (using different technologies) with manual review of metadata and provides an evaluation of metadata according to its completeness.

Tabla de contenido

Lista de Figuras

Lista de Tablas

Introducción

Las Tecnologías de la Información y la Comunicación (TIC por sus siglas en español) han potenciado el acceso sin precedentes a la información y el conocimiento. La información se ha convertido en el eje promotor de cambios sociales, económicos y culturales. El auge de las telecomunicaciones ha producido una transformación de las tecnologías de la información y de la comunicación, cuyo impacto ha afectado a todos los sectores de la economía y de la sociedad (Cruz Regalado 2009).

Ante el panorama de implementación de las TIC, la sociedad ha variado sus procesos de información y comunicación; las barreras espaciales y de tiempo desaparecieron y la globalidad es una extensión posible. A raíz de esto, la información ha crecido exponencialmente, su circulación se da en grandes cantidades y su calidad se pone en entredicho. Son múltiples los cambios que se pueden observar a partir de la implementación de las TIC, pues ha sido tan profundo su impacto que el tema de brecha digital se retoma hoy en día como fundamental para lograr índices de desarrollo de un país. (Ochoa Gutiérrez 2012).

Las bibliotecas no quedaron exentas de los cambios y transformaciones ocurridos con el desarrollo de las TIC y el uso generalizado de Internet. Las TIC cambiaron el entorno de trabajo de las bibliotecas respecto al modo de hacer los procesos y prestar los servicios. Ello condujo a la aparición de las llamadas bibliotecas digitales (Huespe 2012).

Una biblioteca digital, en su definición más simple, es una colección de documentos digitales, almacenados en diferentes formatos electrónicos (Universidad Autónoma Metropolitana 2016). Para facilitar la búsqueda y recuperación, las bibliotecas digitales asignan información descriptiva (metadatos) sobre el contexto, calidad, condición o características de estos documentos (Senso & Rosa Piñero 2003). No en pocos casos, esta información en las bibliotecas digitales, es de baja calidad o incompleta. Esto se debe, entre otras cuestiones, a que el proceso de carga de los metadatos suele ser una tarea tediosa, que consume tiempo y muchas veces propensa a errores (Casali & Deco 2013).

En la Universidad Central "Marta Abreu" de Las Villas (UCLV) se han llevado a cabo varias iniciativas orientadas a la creación de una biblioteca digital universitaria. Entre los sistemas utilizados para alcanzar este fin se puede mencionar *GreenStone Digital Library Software*. El sitio web Libros UCLV ha formado parte también de las iniciativas realizadas en la institución. El problema principal con estas iniciativas ha radicado en que el proceso de creación de estas bibliotecas se ha apoyado fundamentalmente en el trabajo manual, sujeto a errores, y hasta la fecha solo se han logrado publicar 2052 y 1094 documentos respectivamente. Estas cantidades se consideran poco representativas del volumen de información digital que existe en la UCLV y que necesita ser incluido en la biblioteca digital de la universidad.

Para dar una idea del volumen de información digital que existe en la UCLV, solamente en el Centro de Investigación en Informática (CII) existen más de 300 000 documentos digitales que actualmente no están incluidos en ninguna de las iniciativas presentes en la UCLV. Asumir la catalogación (creación de metadatos) y publicación de estos documentos de forma manual es una tarea tediosa, que consume tiempo y propensa a errores.

De este análisis surge el problema científico de esta investigación formulado en la siguiente interrogante:

¿Cómo facilitar, mediante una herramienta informática, la catalogación y publicación de documentos de carácter científico en el proceso de construcción de una biblioteca digital?

Como **objetivo general** de esta tesis se ha concebido *desarrollar una herramienta informática que permita la extracción de metadatos de documentos científicos y su publicación en una biblioteca digital.*

Para dar respuesta al problema de científico se formulan como **preguntas científicas:**

1. ¿Cuáles son los fundamentos básicos en los que podría sustentarse un proceso de extracción automática de metadatos?
2. ¿Cómo determinar la calidad de los metadatos extraídos?
3. ¿Qué aspectos deben tenerse en cuenta para la publicación de documentos digitales en una biblioteca digital?

4. ¿Cómo diseñar una herramienta informática que permita la extracción automática de metadatos de documentos científicos en formato PDF y su publicación en una biblioteca digital?

Estas interrogantes orientaron la elaboración de los siguientes **objetivos específicos**:

1. Identificar los fundamentos básicos en los que podría sustentarse un proceso de extracción automática de metadatos.
2. Identificar las métricas a emplear para medir la calidad de los metadatos.
3. Identificar los aspectos a tener en cuenta para la publicación de documentos digitales en una biblioteca digital.
4. Diseñar una herramienta informática que permita la extracción automática de metadatos de documentos científicos en formato PDF y su publicación en una biblioteca digital.
5. Implementar la herramienta diseñada.

Justificación de la investigación:

En la actualidad con el uso de las TIC la información ha crecido exponencialmente, su circulación se da en grandes cantidades y su calidad se pone en entredicho. Esto ha llevado a que exista la necesidad de que los documentos sean catalogados adecuadamente, tarea que hasta la fecha se realiza manualmente, que consume tiempo y muchas veces está propensa a errores. Un software que permita la catalogación y publicación de documentos de carácter científico en el proceso de construcción de una biblioteca digital constituiría un paso de avance en aras de resolver estos problemas.

Estructura de la tesis:

El propósito de esta tesis es desarrollar una herramienta informática que permita la extracción de metadatos de documentos científicos y su publicación en una biblioteca digital. La misma se organiza en un primer capítulo teórico, un segundo capítulo que aborda el diseño del sistema y finalmente un tercer capítulo dedicado a la descripción de la solución.

Específicamente, en el capítulo 1 se muestran los conceptos básicos y tecnologías alrededor de las bibliotecas digitales, enfatizándose en la extracción automática de metadatos y en la evaluación de su calidad. Además, se describen las principales tecnologías utilizadas en la implementación de la solución.

En el capítulo 2 se abordan los aspectos esenciales en el análisis y diseño del sistema. Se especifican los requerimientos, tanto funcionales, como no funcionales del mismo. Se definen los casos de uso a partir de dichos requerimientos, se muestran y explican los diagramas de actividades, de clases y de componentes del sistema; así como el diagrama entidad-relación de la base de datos embebida utilizada. Finalmente, se analizan los patrones de diseño que se utilizan en el sistema.

Finalmente, en el capítulo 3 se brinda una descripción general de la herramienta implementada (Darkaiv) y sus principales funcionalidades.

CAPÍTULO 1.

MARCO TEÓRICO

CAPÍTULO 1. MARCO TEÓRICO

El presente capítulo está dedicado a exponer los resultados de una revisión bibliográfica especializada y actualizada tanto nacional como internacional. La misma se ha realizado con el objetivo de resumir aspectos fundamentales del marco teórico-conceptual elaborado con vistas a darle solución al problema de investigación planteado. Primeramente, se hace una introducción abordando los conceptos relacionados con las bibliotecas digitales, así como los principales softwares libres que se utilizan para crearlas. Luego se aborda el tema referente a los metadatos bibliográficos y a los sistemas para la extracción automática de estos. Por último, se analizan las características de las tecnologías a utilizar en el desarrollo de la aplicación como son: ActiveJDBC, Java, Maven NetBeans y H2.

1.1 Bibliotecas Digitales

Según Tramullas (2002:8), *"una **biblioteca digital** es un sistema de tratamiento técnico, acceso y transferencia de información digital, estructurado alrededor del ciclo de vida de una colección de documentos digitales, sobre los cuales se ofrecen servicios interactivos de valor añadido para el usuario final"*. Esta definición de Tramullas se refiere a recursos de información en formato digital a los que se accede mediante diferentes dispositivos electrónicos y pone en evidencia que el concepto de biblioteca digital está estrechamente ligado a las TIC.

Precisamente el uso de las TIC establece un grupo de diferencias entre las bibliotecas digitales y las bibliotecas clásicas o tradicionales. Según lo planteado por varios autores (BuenasTareas 2013; Miranda 2010), algunas de estas son:

- El modo de acceso a las bibliotecas digitales es a través de un portal web, o sea, que se puede llegar a la información deseada a distancia y por lo tanto es necesario mínimamente tener accesibilidad a internet. De esto se puede deducir que la biblioteca digital cierra solo si el servidor cae.
- En la biblioteca digital los recursos son de tipo electrónicos y se puede acceder a la información contenida en ellos con los medios electrónicos compatibles. En este

sentido, no hay que retirar, trasladar o devolver los libros, porque otros consultantes los requieren.

- En las bibliotecas digitales las informaciones no se estropean ni se desgastan.

Ahora bien, a pesar de que las bibliotecas digitales permiten un mayor acceso y organización al contenido existente en la red de redes, no están al alcance de todos. A entender de Marcum (2005) si se pudiera hacer que todos los estudiantes de América (y del resto del mundo, de países menos favorecidos, de países reprimidos, de hecho, de cualquier parte del mundo con acceso a la informática) usen los ordenadores como puertas de acceso a las bibliotecas del mundo, eso sería un logro verdaderamente digno de las nuevas tecnologías. Pero para lograr esto las bibliotecas digitales deben desarrollar tres características generales:

1. Ser una colección global de recursos importantes para la investigación, la enseñanza y el aprendizaje.
2. Ser de fácil acceso para todo tipo de usuarios, tanto principiantes como expertos.
3. Estar gestionada y mantenida por profesionales que se consideren administradores del patrimonio intelectual y cultural.

De acuerdo con Brito Neves y Alburquenque (2007) las bibliotecas digitales se encuentran entre los sistemas de información más complejos, no por ser proyectos digitales, sino por su multidisciplinariedad. La necesidad de trabajar de forma conjunta profesionales de diferentes ámbitos, desde bibliotecarios a informáticos, ingenieros electrónicos y científicos, es la regla y no la excepción en proyectos de este tipo.

Hay que señalar varios aspectos importantes para el desarrollo de una biblioteca digital, algunos autores relacionados con esta temática (Hípola et al. 2000; Tramullas 2004) destacan que:

- Los derechos de autor y la legislación sobre la propiedad intelectual son aspectos fundamentales tanto para la creación como para la protección de una biblioteca digital y suele ser el mayor escollo para el desarrollo.

- Los metadatos (datos sobre los datos) son de gran importancia para facilitar la búsqueda y recuperación de la información, ya que permiten una búsqueda efectiva y precisa.
- Los formatos utilizados para el desarrollo de bibliotecas digitales pueden ser de tipo abierto o de tipo cerrado. Los formatos abiertos se pueden manipular con mayor facilidad (ej.: SGML, HTML y XML) mientras que los cerrados presentan más dificultades para ser modificados, como el Acrobat y el PDF.

1.1.1 Software libre para Bibliotecas Digitales

Se considera software libre a todo aquel software que respeta la libertad de los usuarios y la comunidad. A grandes rasgos, significa que ***los usuarios tienen la libertad de ejecutar, copiar, distribuir, estudiar, modificar y mejorar el software***. Es decir, el software libre es una cuestión de libertad, no de precio. En inglés en ocasiones se habla de *libre software*, en lugar de *free software*, para mostrar que no significa que sea gratuito (GNU.ORG 2016).

Precisamente en GNU.ORG (2016) se expone que un programa es software libre si los usuarios tienen cuatro libertades esenciales:

- **Libertad 1:** La libertad de ejecutar el programa como se desea, con cualquier propósito.
- **Libertad 2:** La libertad de estudiar cómo funciona el programa, y cambiarlo para que haga lo que se desee. El acceso al código fuente es una condición necesaria para ello.
- **Libertad 3:** La libertad de redistribuir copias para ayudar a su prójimo.
- **Libertad 4:** La libertad de distribuir copias de las versiones modificadas a terceros.

En la ponencia presentada por Sanjo Jose (2007) se resumen los paquetes de software libre más utilizados en el ámbito académico de varios países. En base a 44 respuestas recibidas de 500 encuestas enviadas en India, Estados Unidos, Gran Bretaña y otros países, el autor concluye que los paquetes de software libre más utilizados para bibliotecas digitales son: DSpace y EPrints en primer y segundo lugar como preferidos quizás porque tienen una arquitectura orientada a preservar la producción en el ámbito académico. En tercer lugar, el

software libre Greenstone y luego Fedora con pocas instalaciones. En este epígrafe se explican las principales características de los tres primeros.

1.1.1.1 DSpace

DSpace es un software de código abierto que provee herramientas para la administración de colecciones digitales y que comúnmente es utilizado como solución de repositorio institucional. Liberado en el 2002, como producto de una alianza de HP y el MIT, DSpace es un software completamente personalizable que soporta una gran variedad de datos, incluyendo libros, tesis, fotografías, filmes, video, datos de investigación y otras formas de contenido. En este sentido, es un software fácil de adaptar a las necesidades de cualquier organización (DURASPACE 2016).

Como herramienta informática para repositorios, DSpace organiza sus datos en comunidades, colecciones y artículos (ítems), asignándoles metadatos y permitiendo su recolección por otros sistemas mediante estándares. Escrito en Java, el software utiliza una base de datos relacional en PostgreSQL o Oracle y tiene dos interfaces, una clásica (JSPUI) que usa JSP y Java Servlet API (*Application Programming Interface*), y una nueva (XMLUI) basada en Apache Cocoon que usa XML y XSLT. En relación a la interoperabilidad, DSpace implementa el protocolo de la iniciativa de archivos abiertos para la recolección de metadatos OAI-PMH (*Open Archives Initiative-Protocol Metadata Harvesting*), y es capaz de exportar paquetes de software METS (*Metadata Encoding and Transmission Standard*). El módulo API REST de DSpace proporciona una interfaz de programación de comunidades, colecciones e ítems. DSpace 4 introdujo la API REST inicial, que no permitía la autenticación, y proporciona únicamente acceso de sólo lectura a comunidades, colecciones e ítems. En cambio, DSpace 5 permite la autenticación para acceder al contenido restringido, además de permitir crear, editar y eliminar los ítems en DSpace.

De acuerdo con DSpace (2015), este sistema permite a las organizaciones:

- Capturar y describir el material digital utilizando un módulo de flujo de trabajo de presentación, o una variedad de opciones de ingesta programáticas.

- Distribuir los activos digitales de una organización a través de Internet a través de un sistema de búsqueda y recuperación.
- Preservar los activos digitales a largo plazo.

1.1.1.2 EPrints

EPrints es un software gratuito y de código abierto para la creación de repositorios digitales de acceso abierto y uno de los principales competidores de DSpace. Es desarrollado por la *School of Electronics and Computer Science* de la University of Southampton (Reino Unido). Disponible para GNU Linux (RedHat/Fedora, Debian/Ubuntu) y MS Windows (XP/Vista/7), EPrints se ha probado con éxito en Solaris y Mac OS-X, pero se recomienda utilizarlo bajo plataforma GNU Linux (Eprints 2016).

EPrints es una herramienta muy flexible que brinda gran libertad para ampliar su funcionalidad a través de un potente sistema de plugins (extensiones). Puede manipular una gran variedad de objetos digitales, desde objetos textuales a objetos multimedia y no está limitado a comunidades y colecciones estructuradas, sino que también permite la creación de colecciones virtuales flexibles a partir de metadatos. Además, EPrints dispone de una API para programar rutinas propias: programar un plugin, personalizar la manera en que renderiza una página determinada, etc. Permite importar y crear estructuras organizacionales y clasificaciones temáticas jerárquicas (por defecto, el paquete EPrints incluye la clasificación temática de la Biblioteca del Congreso de Estados Unidos). Finalmente es válido mencionar que EPrints, al igual que DSpace, implementa el protocolo de la iniciativa de archivos abiertos para la recolección de metadatos OAI-PMH.

1.1.1.3 Greenstone

Greenstone es un paquete de software que permite la creación y utilización de una biblioteca digital, con sus correspondientes colecciones de documentos. Se distribuye bajo licencia GNU. Su desarrollo lo lleva a cabo un equipo de investigadores de la University of Waikato (Nueva Zelanda). Posee un nivel de fiabilidad, de desarrollo y de mantenimiento que ha llevado a la UNESCO a incluirlo en su programa de aplicaciones informáticas para servicios de información y documentación (Fox et al. 2001).

La arquitectura de Greenstone puede parecer, a primera vista, complicada. En primer lugar, posee un motor de indización y recuperación de información textual, llamado MG, que utiliza el modelo vectorial para el tratamiento de la información. Este motor es alimentado por un conjunto de scripts, encargados de preprocesar documentos XML, lo que asegura la capacidad de la aplicación para tratar cualquier idioma. En Greenstone, la información es gestionada mediante el sistema de gestión de bases de datos GDBM (*GNU DataBase Manager*), que también es software libre, e incluye una versión especial de MG, el MG+/MGPP, para cuando sean necesarios formularios con múltiples campos para las interfaces de búsqueda. Por último es válido mencionar que en la versión 2.61 se añadió el motor de indización Lucene[1], desarrollado por Apache Foundation, como opción adicional al MG.

El proceso de importación en Greestone incluye: la selección de los documentos a tratar, su preprocesamiento y conversión en XML, la extracción de metadatos (asignados automática o manualmente) y la indización con el motor elegido con su posterior almacenamiento en la base de datos GDBM. Para el usuario final, el proceso resulta transparente. Sin embargo, para el responsable de las colecciones, el proceso es más complejo y ofrece múltiples posibilidades relacionadas con el tratamiento de la información textual y el desarrollo de las clasificaciones y presentaciones de los resultados. Desde la versión 2.41, Greenstone incorpora una interfaz gráfica en Java, GLI (*Greenstone Librarian Interface*) que facilita el proceso de creación y administración de las colecciones (Fox et al. 2001).

1.1.3 Estándares para la descripción de recursos de información

Cuando Internet empezó a crecer y la cantidad de información disponible aumentó desmesuradamente, surgió el problema de clasificarla e identificarla de manera eficiente. Partiendo de ese problema, se comenzaron a usar los metadatos que, a consideración de Senso y Piñero (2003:99) son *"toda aquella información descriptiva sobre el contexto, calidad, condición o características de un recurso, dato u objeto que tiene la finalidad de facilitar su recuperación, autentificación, evaluación, preservación o interoperabilidad"* es

[1] **Apache Lucene** es una API de código abierto para recuperación de información, originalmente implementada en Java por Doug Cutting. Está apoyado por el Apache Software Foundation y se distribuye bajo la Apache Software License.

decir, información relativa a los propios datos que facilitan su catalogación y además proporcionan información semántica asociada.

Existen diversos tipos de metadatos, y según su función se pueden clasificar en metadatos descriptivos, estructurales o administrativos (Departamento Biblioteca de la Universidad de Cornell 2003). Los descriptivos, describen e identifican recursos de información permitiendo a los usuarios la búsqueda y recuperación de la información (ej.: Dublin Core y Etiquetas META de HTML). Los estructurales facilitan la navegación y la presentación de los recursos. Además, proporcionan información sobre la estructura interna de los documentos, así como la relación entre ellos (ej.: XML, RDF y SGML). Finalmente, los administrativos facilitan la gestión de conjuntos de recursos e incluyen la gestión de derechos sobre control de acceso y uso (ej.: MOA2).

Hay varias iniciativas en función de normalizar y estandarizar los metadatos sobre recursos de información. Sin duda, uno de más conocido es el modelo de datos Dublin Core, creado inicialmente para catalogar y compartir información sobre libros entre bibliotecas estadounidenses, pero que ahora se usa casi en la totalidad de las páginas web existentes en Internet.

1.1.3.1 Dublin-Core

Dublin Core es un modelo de metadatos elaborado y auspiciado por la DCMI (*Dublin Core Metadata Initiative*), una organización dedicada a fomentar la adopción extensa de los estándares interoperables de los metadatos y a promover el desarrollo de los vocabularios especializados de metadatos para describir recursos y para permitir a sistemas más inteligentes el descubrimiento de recursos (Méndez & Senso 2004).

Según Méndez & Senso (2004) el Dublin Core es el esquema de metainformación más utilizado a nivel mundial y cuenta con un conjunto amplio de fortalezas entre las que se en encuentran:

- Su simplicidad.
- La independencia sintáctica (que ha permitido que se integre en la estructuración de datos en XML/RDF).

- Alto nivel de normalización formal: ANSI/NISOZ39.85-2001, ISO 15836-2003.
- Crecimiento y evolución del estándar a través de una institución formal consorciada: la DCMI.
- El conjunto de elementos DC se ha convertido en una infraestructura operacional del desarrollo de la Web Semántica.

La norma ISO15836 define el *Conjunto de Elementos Dublin Core*, o lo que se conoce habitualmente como "DC simple". Esos 15 elementos básicos o definiciones básicas para describir cualquier objeto de información, se presentan habitualmente divididos en tres grupos que indican la clase o alcance de la información incluida en ellos. Estos grupos son:

1. Grupo de elementos relacionados principalmente con el *contenido* del recurso.
2. Grupo de elementos relacionados principalmente con el recurso cuando es visto como una *propiedad intelectual.*
3. Grupo de elementos relacionados principalmente con la *instanciación* del recurso.

Como elementos de contenido, Dublin Core incluye los siguientes elementos:

- **Título:** el nombre dado a un recurso, habitualmente por el autor. Etiqueta: *dc.title*
- **Claves:** los temas del recurso. Típicamente, *Subject* expresa las claves o frases que describen el título o el contenido del recurso. Etiqueta: *dc.subject*
- **Descripción:** una descripción textual del recurso. Puede ser un resumen en el caso de un documento o una descripción del contenido en el caso de un documento visual. Etiqueta: *dc.description*
- **Fuente:** secuencia de caracteres usados para identificar unívocamente un trabajo a partir del cual proviene el recurso actual. Etiqueta: *dc.source*
- **Tipo del Recurso:** la categoría del recurso. Etiqueta: *dc.type*
- **Relación:** es un identificador de un segundo recurso y su relación con el recurso actual. Etiqueta: *dc.relation*

- **Cobertura:** es la característica de cobertura espacial y/o temporal del contenido intelectual del recurso. La cobertura espacial se refiere a una región física, utilizando por ejemplo coordenadas. La cobertura temporal se refiere al contenido del recurso, no a cuándo fue creado. Etiqueta: *dc.coverage*

Como elementos dentro de la categoría de **propiedad intelectual** corresponden:

- **Autor o Creador:** la persona u organización responsable de la creación del contenido intelectual del recurso. Etiqueta: *dc.creator*
- **Editor:** la entidad responsable de hacer que el recurso se encuentre disponible en la red en su formato actual. Etiqueta: *dc.publisher*
- **Otros Colaboradores:** una persona u organización que haya tenido una contribución intelectual significativa, pero que esta sea secundaria en comparación con las de las personas u organizaciones especificadas en el elemento *Creator*. (por ejemplo: editor, ilustrador y traductor). Etiqueta: *dc.contributor*
- **Derechos:** son una referencia (por ejemplo, una URL) para una nota sobre derechos de autor, para un servicio de gestión de derechos o para un servicio que brinda información sobre términos y condiciones de acceso a un recurso. Etiqueta: *dc.rights*

Finalmente, en la categoría de **instanciación** se encuentran los siguientes elementos:

- **Fecha:** una fecha en la cual el recurso se puso a disposición del usuario en su forma actual. Esta fecha no se tiene que confundir con la que pertenece al elemento Coverage, que estaría asociada con el recurso en la medida que el contenido intelectual está de alguna manera relacionado con aquella fecha. Etiqueta: *dc.date*
- **Formato:** es el formato de datos de un recurso, usado para identificar el software y, posiblemente, el hardware que se necesitaría para mostrar el recurso. Etiqueta: *dc.format*
- **Identificador del Recurso:** secuencia de caracteres utilizados para identificar unívocamente un recurso. Ejemplos para recursos en línea pueden ser URLs y URNs. Para otros recursos pueden ser usados otros formatos de identificadores, como por ejemplo ISBN (*International Standard Book Number*). Etiqueta: *dc.identifier*
- **Lengua:** lengua/s del contenido intelectual del recurso. Etiqueta: *dc.language*

Nota: *Los sistemas DSpace, EPrints y Greenstone descritos los epígrafes 1.1.1.1, 1.1.1.2 y 1.1.1.3 respectivamente implementan el estándar Dublin Core.*

1.2 Metadatos

En el epígrafe *1.1.3 Estándares para la descripción de documentos digitales* se establece una definición de metadatos dada por Senso y Rosa Piñero (2003) donde queda claro que los metadatos son datos que describen el contenido y la estructura de otros datos con el fin de facilitar su recuperación, autentificación, evaluación, preservación o interoperabilidad.

Según Stamou et al. (2006) los metadatos se clasifican de acuerdo a tres criterios:

- **Contenido***:* se pueden separar los metadatos que describen el recurso mismo de los que describen el contenido del recurso. Es posible subdividir estos dos grupos más veces, por ejemplo, para separar los metadatos que describen el sentido del contenido de los que describen la estructura del contenido o los que describen el recurso mismo de los que describen el ciclo vital del recurso. Subdividir metadatos por su contenido es lo más común.

- **Variabilidad***:* se pueden distinguir metadatos mutables e inmutables. Los inmutables no cambian, no importa qué parte del recurso se vea, por ejemplo, el nombre de un fichero. Los mutables difieren de parte a parte, por ejemplo el contenido de un vídeo (Smith & Schirling 2006).

- **Función***:* los datos se pueden dividir de acuerdo a su función en datos subsimbólicos, simbólicos o lógicos. Los subsimbólicos no contienen información sobre su significado. Los simbólicos describen datos subsimbólicos, es decir añaden sentido. Los datos lógicos describen cómo los datos simbólicos pueden ser usados para deducir conclusiones lógicas, es decir añaden comprensión.

De acuerdo con Smith y Schirling (2006) el ciclo de vida de los metadatos comprende tres fases fundamentales: creación, manipulación y destrucción. Durante la creación los metadatos, estos se pueden crear manualmente, semiautomáticamente o automáticamente. El proceso manual puede ser muy laborioso por lo que el desarrollo de técnicas semiautomáticas o automáticas es más que deseable. Aunque el desarrollo de algoritmos tan avanzados está

siendo objeto de investigación actualmente, no es probable que la computadora vaya a ser capaz de extraer todos los metadatos automáticamente. En vez de ello, se considera la producción semiautomática mucho más realista. Ciertamente, durante la manipulación y posterior destrucción de los metadatos hay cuestiones que pueden ser manejadas automáticamente, pero hay otras donde la intervención de un humano es totalmente indispensable

Existen dos posibilidades para almacenar metadatos: depositarlos *internamente*, en el mismo documento que los datos, o depositarlos externamente, en su mismo recurso. Hoy, por lo general, se considera mejor opción la localización externa porque hace posible la concentración de metadatos para optimizar operaciones de búsqueda.

1.2.1 Extracción automática de metadatos

Uno de los grandes retos que poseen las bibliotecas digitales es proporcionar una eficiente búsqueda y consulta del material almacenado en ellas. Para poder solucionar esto es importante contar con una buena descripción de los objetos digitales que la conforman, a partir de la calidad de los metadatos descriptivos, ya que estos favorecen la precisión de las búsquedas y permiten la recuperación de aquellos objetos que mejor satisfagan las necesidades de información del usuario, teniendo en cuenta sus características y preferencias individuales (Pinilla et al. 2014).

En la actualidad, existen un grupo de bibliotecas para la extracción automática de metadatos en documentos digitales que se distinguen en objetivos, arquitectura y técnicas utilizadas. En este epígrafe se presentan y analizan dos de estas bibliotecas: *Apache Tika* y *Grobid* (Pinilla et al. 2014).

1.2.1.1 Apache Tika

Apache Tika es una biblioteca escrita en Java que se centra en la identificación automática de tipo de medio, la extracción de texto, y la extracción de metadatos (Mattmann & Zitting 2011).

De acuerdo con FILExt (2016), existen alrededor de 26 000 a 51 000 tipos de contenido, mientras que los datos se almacenan en varios formatos, como documentos de texto, hoja de

cálculo Excel, archivos PDF, imágenes y archivos multimedia, para nombrar unos pocos. En contexto como este, muchas aplicaciones en la actualidad necesitan de un apoyo adicional para facilitar la extracción de los datos de estos tipos de documentos. Apache Tika sirve para este propósito, proporcionando una interfaz de programación de aplicaciones (API) genérica para detectar y extraer datos de varios formatos de archivo.

Entre las aplicaciones de Apache Tika expuestas por tutorialspoint.com (2016) se encuentran:

- Motores de búsqueda
- Análisis de documentos
- Gestión de activos digitales
- Análisis de contenido

1.2.1.2 Grobid

GROBID significa medio de generación de datos bibliográfico (*GeneRation Of BIbliographic Data*) y es una biblioteca que se utiliza para extraer, analizar y restructurar documentos sin formato, como los PDF, en documentos TEI-codificados estructurados, con un enfoque particular en publicaciones técnicas y científicas.

De acuerdo con Grobid (2016) las funcionalidades que se encuentran disponibles son:

- Extracción y análisis de cabeceras de artículos en formato PDF. La extracción aquí cubre la información bibliográfica (por ejemplo, título, resumen, autores, afiliaciones, palabras clave, etc.).
- Extracción y análisis de referencias de artículos en formato PDF. Las referencias en las notas al pie son compatibles. Son raras en artículos técnicos y científicos, pero frecuentes en las publicaciones en las ciencias humanas y sociales.
- El análisis de las referencias en aislamiento.
- El análisis de los nombres, en particular los nombres de los autores de cabecera, y nombres de los autores en las referencias (dos modelos distintos).
- El análisis de los bloques de afiliación y dirección.
- El análisis de las fechas.

- Extracción de texto completo de los artículos en formato PDF, incluyendo un modelo de la segmentación general del documento y un modelo para la estructuración del cuerpo del texto.

GROBID incluye el procesamiento por lotes, una API REST integral, una API de Java, un marco de evaluación relativamente genérico (precisión, cobertura, etc.) y la generación semiautomática de datos de entrenamiento. Actualmente se considera una producción lista con despliegues de producción que incluyen ResearchGATE, el Archivo Investigación HAL, la Oficina Europea de Patentes, Mendeley y la Organización Europea para la Investigación Nuclear (CERN).

Después de analizados Apache Tika y Grobid como bibliotecas para la extracción automática de metadatos, se puede apreciar que existe una diferencia marcada entre ambas en cuanto su función. Esta diferencia radica en que mientras que Tika es una biblioteca genérica para detectar y extraer datos de varios formatos de archivo, Grobid es una biblioteca para la extracción de metadatos enfocada particularmente en publicaciones técnicas y científicas. En este sentido se puede afirmar que una estrategia que combine a ambas obtiene mejores resultados que utilizándolas de manera separadas. Tika 1.11 introduce un parser para la utilización de Grobid, lo que reafirma la idea anterior.

1.2.2 Catálogos en línea de Fuentes Bibliográficas

Un catálogo en línea u OPAC (del inglés: *Online Public Access Catalog*) es un catálogo automatizado de los materiales de una biblioteca, de acceso público y en línea. Esto significa que las personas tienen acceso a él desde cualquier lugar con acceso a Internet. En la actualidad existen muchos catálogos en líneas destacándose Mendeley, ArXiv, PubMed, CrossRef y WorldCat. En este epígrafe se abordan las características principales de dos de ellos: CrossRef y WorldCat.

1.2.2.1 CrossRef

CrossRef es la espina dorsal de referencia activa para las publicaciones científicas en Internet. Fundada en 2000 por una asociación sin ánimo de lucro de las más importantes editoriales científicas, la PILA (*Publishers International Linking Association*), este sistema

proporciona la infraestructura básica para enlazar las referencias de artículos procedentes de distintas editoriales y publicaciones, empleando para ello el sistema de identificador de objeto digital , conocido en inglés como digital object identifier (DOI) (CrossRef 2016).

Entre sus principales servicios, CrossRef permite acceder a la información bibliográfica de un artículo del que se conoce su DOI a través de una API REST. Esta API REST solamente funciona utilizando CrossRef DOIs y devuelve los resultados en un archivo JSON que contiene los metadatos del artículo consultado (CrossRef GitHub 2016).

1.2.2.2 WorldCat

WorldCat es la red más grande de contenidos y servicios de la biblioteca del mundo. Las bibliotecas de WorldCat están dedicadas a proporcionar el acceso a sus recursos en la Web, donde la mayoría de la gente comienza su búsqueda de información. Utilizando WorldCat se pueden buscar libros populares, CDs de música y videos (todos los elementos físicos que está acostumbrado a recibir de las bibliotecas). También permite descubrir muchos nuevos tipos de contenidos digitales, como libros de audio descargables. Se pueden encontrar citas de artículos con enlaces a su texto completo; materiales de investigación autorizadas, tales como documentos y fotos de importancia local o histórica; y las versiones digitales de los objetos raros que no están disponibles al público. Debido a que las bibliotecas de WorldCat sirven a diversas comunidades en decenas de países, los recursos están disponibles en muchos idiomas (WorldCat.org 2016).

WorldCat, al igual que CrossRef, posee una API REST para la búsqueda de archivos que se conoce como *WorldCat Search API*. A través de esta se pueden buscar libros, videos o música existentes en la base de datos de WorldCat. Los resultados de una búsqueda a través de esta API devuelven enlaces directos a las bibliotecas donde fue encontrado (WorldCat Search API 2016).

1.2.3 Calidad de los Metadatos

Un metadato es considerado de alta calidad cuando este respalda los requerimientos funcionales del sistema que esté diseñado a soportar, lo que indica que también en el contexto

de los metadatos, *"la calidad está relacionada con la aptitud para el propósito"* (Medrano et al. 2012).

Según Hillmann y Bruce (2004) existen siete dimensiones que describen la calidad de los metadatos:

- **Completitud** es el grado en que los metadatos incluyen toda la información necesaria para tener una representación ideal del objeto descrito.
- **Exactitud** se refiere a la medida en que los valores de los metadatos son correctos. Puede ser considerada como la distancia semántica entre la información que un usuario extrae del registro de metadatos y la información que el mismo usuario obtiene del documento mismo.
- **Procedencia** identifica la reputación que un registro de metadatos tiene en una comunidad. Por ejemplo, un usuario puede confiar más en metadatos generados por un experto, que en los metadatos generados por un software. La procedencia está relacionada con la percepción subjetiva que el usuario tiene acerca del origen de los metadatos.
- La **conformidad a las expectativas** mide el grado en el cual el registro de metadatos satisface los requisitos de una comunidad de usuarios. Existen varios parámetros que afectan esta calidad tales como el vocabulario, los campos necesarios para realizar una tarea especificada por el usuario y la cantidad de información para describir el objeto.
- La **coherencia y consistencia lógica** es el grado en que los metadatos utilizados en un dominio describen a un mismo recurso y corresponden a una definición o conceptualización.
- La **actualidad** guarda relación con el grado para el cual un registro de metadatos permanece actual entre cierta comunidad.
- La **accesibilidad** es el grado en que los metadatos son accesibles, en términos cognitivos, así como también físico / lógicos. Metadatos que no se pueden leer o

entender no tienen ningún valor. La accesibilidad cognitiva es medida por la facilidad con que el usuario entiende la información contenida en los metadatos. La accesibilidad física / lógica está determinada por la existencia o no de formatos incompatibles o enlaces rotos.

1.2.3.1 Métrica de Completitud

Como se describe en el epígrafe anterior la *completitud* es el grado en que la instancia de metadatos contiene toda la información necesaria para tener una representación integral del recurso descrito. Mientras que es fácil de entender para los registros y bibliotecas estáticas, este concepto es menos claro para los casos de metadatos dinámicos, donde se añade nueva información cada vez que se utiliza el recurso.

Una métrica básica de completitud expuesta por Ochoa (2008) es contar el número de campos en cada instancia de metadatos que contienen un valor no nulo. La Ecuación 1.1 expresa esta idea donde *P(i)* es uno si el campo i-ésimo tiene un valor no nulo, cero en caso contrario y N es el número de campos definidos en la norma de metadatos.

$$Qcomp = \frac{\sum_{i=1}^{N} P(i)}{N}$$

Ecuación 1.1 Métrica básica de completitud

Es importante destacar que la relevancia o importancia de los metadatos puede variar en función del tipo del documento (por ejemplo: el título de un artículo puede ser considerado más importante que la fecha de publicación). Para tener en cuenta este fenómeno, un factor de importancia (peso) podría multiplicar la presencia o ausencia de un campo de metadatos. Este factor puede ser fácilmente incluido en el cálculo de la métrica integridad como se muestra en la Ecuación 1.2 (Ochoa 2008)

$$Qwcomp = \frac{\sum_{i=1}^{N} a_i * P(i)}{\sum_{i=1}^{N} a_i}$$

Ecuación 1.2 Métrica de completitud ponderada

Donde α_i es la importancia relativa del campo i-ésimo, el valor máximo para *Qwcomp* será de uno (cuando todos los campos con una importancia diferente a cero están llenos) y un valor mínimo de cero (cuando todos los campos con una importancia diferente a cero están vacíos).

1.3 Herramientas utilizadas en la Solución

En este epígrafe se analizan las herramientas y tecnologías de software que se utilizan en la implementación del Darkaiv, con especial énfasis en el lenguaje de programación y el gestor de base de datos utilizado en la solución.

1.3.1 Java como Lenguaje de Programación

Java es un lenguaje de programación de propósito general, concurrente y orientado a objetos, que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible. Desde su introducción a finales de 1995, la intención de sus creadores ha sido permitir que los desarrolladores de aplicaciones escriban el código del programa una vez y lo ejecuten en cualquier dispositivo (conocido en inglés como *WORA*, o "*write once, run anywhere*"). En otras palabras, el código que es ejecutado en una plataforma no tiene que ser recompilado para correr en otra (Gosling et al. 2005).

1.3.2 Maven para la gestión y construcción de proyectos Java

Maven es una herramienta de software para la gestión y construcción de proyectos Java creada por Jason Van Zyl, de Sonatype, en 2002. Es similar en funcionalidad a Apache Ant (y en menor medida a PEAR de PHP y CPAN de Perl), pero tiene un modelo de configuración de construcción más simple, basado en un formato XML. Es un proyecto de nivel superior de la Apache Software Foundation. Maven utiliza un Project Object Model (POM) para describir el proyecto de software a construir, sus dependencias de otros módulos y componentes externos, y el orden de construcción de los elementos. Viene con objetivos predefinidos para realizar ciertas tareas claramente definidas, como la compilación del código y su empaquetado (Apache Maven 2016).

1.3.3 NetBeans

El NetBeans es una aplicación libre del tipo IDE (Interface Development Environment) que está dotada de muchas facilidades y ventajas. Es una herramienta utilizada por los programadores para escribir, compilar, depurar y ejecutar programas. La plataforma NetBeans permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados *módulos*. Un módulo es un archivo Java que contiene clases de Java escritas para interactuar con las APIs de NetBeans y un archivo especial (manifest file) que lo identifica como módulo. Las aplicaciones construidas a partir de módulos pueden ser extendidas agregándoles nuevos módulos. Debido a que los módulos pueden ser desarrollados independientemente, las aplicaciones basadas en la plataforma NetBeans pueden ser extendidas fácilmente por otros desarrolladores de software (Böck 2009).

1.3.4 ActiveJDBC

ActiveJDBC es una biblioteca en Java que implementa el patrón Active Record (Record Activo) y es utilizada para realizar el Mapeo Objeto-Relacional (en inglés Object Relational Mapping, ORM) con gestores de base de datos. En la actualidad las bases de datos compatibles son: SQLServer, MySQL, Oracle, PostgreSQL, H2 y SQLite3. A diferencia de otros ORM como Hibernate, ActiveJDBC infiere metadatos de la Base de Datos y no utiliza configuraciones, sólo convenciones. Estas convenciones son reemplazables en el código. Para trabajar con ActiveJDBC no hay necesidad de aprender otro QL, SQL es suficiente. El código que se emplea para utilizar este ORM es menudo (se lee como en inglés) (JavaLite 2016).

ActiveJDBC requiere la instrumentación de los archivos de clase después de ser compilados. Esto se logra con una herramienta instrumentación proporcionada por el propio proyecto. Hay tres maneras de usarlo: con un plugin de Maven, Ant, y como una clase Java independiente (sin Ant o Maven).

1.3.5 H2 como Sistema Gestor de Bases de Datos Embebida

H2 es un sistema administrador de bases de datos relacionales programado en Java. Puede ser incorporado en aplicaciones Java o ejecutarse de modo cliente-servidor. Una de las características más importantes de H2 es que se puede integrar completamente en aplicaciones Java y acceder a la base de datos lanzando SQL directamente, sin tener que pasar por una conexión a través de sockets. Está disponible como software de código libre bajo la Licencia Pública de Mozilla o la *Eclipse Public License* (H2 Developers 2016).

Según lo expuesto por REBELLABS (2014), H2 está entre los gestores de base de datos embebidas más utilizados por los programadores en Java, con cierta ventaja con respecto a SQLite3.

1.4 Conclusiones del capítulo

Luego de analizados todos los aspectos teóricos relacionados con las bibliotecas digitales y los metadatos bibliográficos salen a relucir varios aspectos importantes. Primeramente, se tiene que la producción semiautomática de los metadatos se considera mucho más conveniente que la producción automática. Además, la utilización de una estrategia que combine las bibliotecas Apache Tika y Grobid obtiene mejores resultados en la extracción de metadatos que utilizarlas de manera independiente. Finalmente, se identifican siete dimensiones para medir la calidad de los metadatos: completitud, exactitud, procedencia, conformidad a las expectativas, coherencia y consistencia lógica, actualidad y accesibilidad.

CAPÍTULO 2.

ANÁLISIS Y DISEÑO DEL SISTEMA

CAPÍTULO 2. ANÁLISIS Y DISEÑO DEL SISTEMA

El presente capítulo aborda aspectos esenciales sobre el análisis y diseño del sistema. Se especifican los requisitos, tanto funcionales, como no funcionales del mismo para tener una mejor comprensión de este. Además, se definen los casos de uso a partir de dichos requisitos, se muestran los diagramas de actividades, de clases y de componentes del sistema; así como el diagrama entidad-relación de la base de datos embebida utilizada. Por último, se analizan los patrones de diseño que se utilizan en el sistema.

2.1 Requisitos del sistema

Los requisitos del sistema establecen en detalle los servicios, restricciones y metas que debe cumplir el mismo y se definen a partir de las consultas con los usuarios. Se pueden especificar en dos grupos, los requisitos funcionales y los no funcionales.

Sommerville (2013) y Pressman (2011) precisan que los requisitos funcionales son declaraciones de los servicios que proveerá el sistema, de la manera en que este reaccionará a entradas particulares y de cómo se comportará en situaciones específicas. En algunos casos, los requisitos funcionales de los sistemas también declaran explícitamente lo que estos no deben hacer.

Estos autores afirman que los requisitos no funcionales son restricciones de los servicios o funciones ofrecidas por el sistema.

2.1.1 Requisitos funcionales

1. Crear nuevas colecciones
2. Extraer metadatos de archivos y añadirlos a una colección
3. Verificar los metadatos de un archivo con la REST API de CrossRef
4. Verificar los metadatos de un archivo con la REST API de Grobid
5. Publicar los metadatos de un archivo hacia una colección de un repositorio DSpace
6. Crear un backup (salva) de la base de datos del sistema
7. Restablecer la base de datos del sistema desde un backup existente
8. Visualizar, en todo momento, la información relativa al estado del sistema

2.1.2 Requisitos no funcionales

1. Sistema Operativo Linux, Mac Os o Windows (XP o superior)
2. Máquina Virtual de Java 1.7 o superior
3. Conexión a un servidor donde se esté ejecutando la REST API de Grobid
4. Conexión a la REST API de CrossRef
5. Consistencia visual de la herramienta en relación al diseño del gestor bibliográfico Mendeley
6. Manejo de grandes colecciones de documentos
7. Uso de bibliotecas libres para la implementación del sistema
8. Un gigabyte (GB) de memoria RAM o superior
9. 100 megabyte (MB) de espacio disponible en disco duro o superior
10. Un procesador de un gigahercio(GHz) o superior de 32 bits (x86) o 64 bits (x64)

2.2 Diagramas de casos de uso del sistema

El modelo de casos de uso se utiliza para conseguir un acuerdo con los usuarios y clientes sobre qué debe hacer el sistema para ellos. Proporcionan un medio sistemático e intuitivo de capturar requisitos funcionales dirigiendo todo el proceso de desarrollo debido a que la mayoría de las actividades como el análisis, diseño y prueba se llevan a cabo partiendo de los casos de uso.

Según Booch et al. (1998) los diagramas de casos de uso son una secuencia de acciones, incluyendo variantes, que el sistema puede llevar a cabo, y que producen un resultado observable de valor para un actor concreto.

A continuación, en la Figura 2.1, se muestra el diagrama de casos de uso y actores de la aplicación.

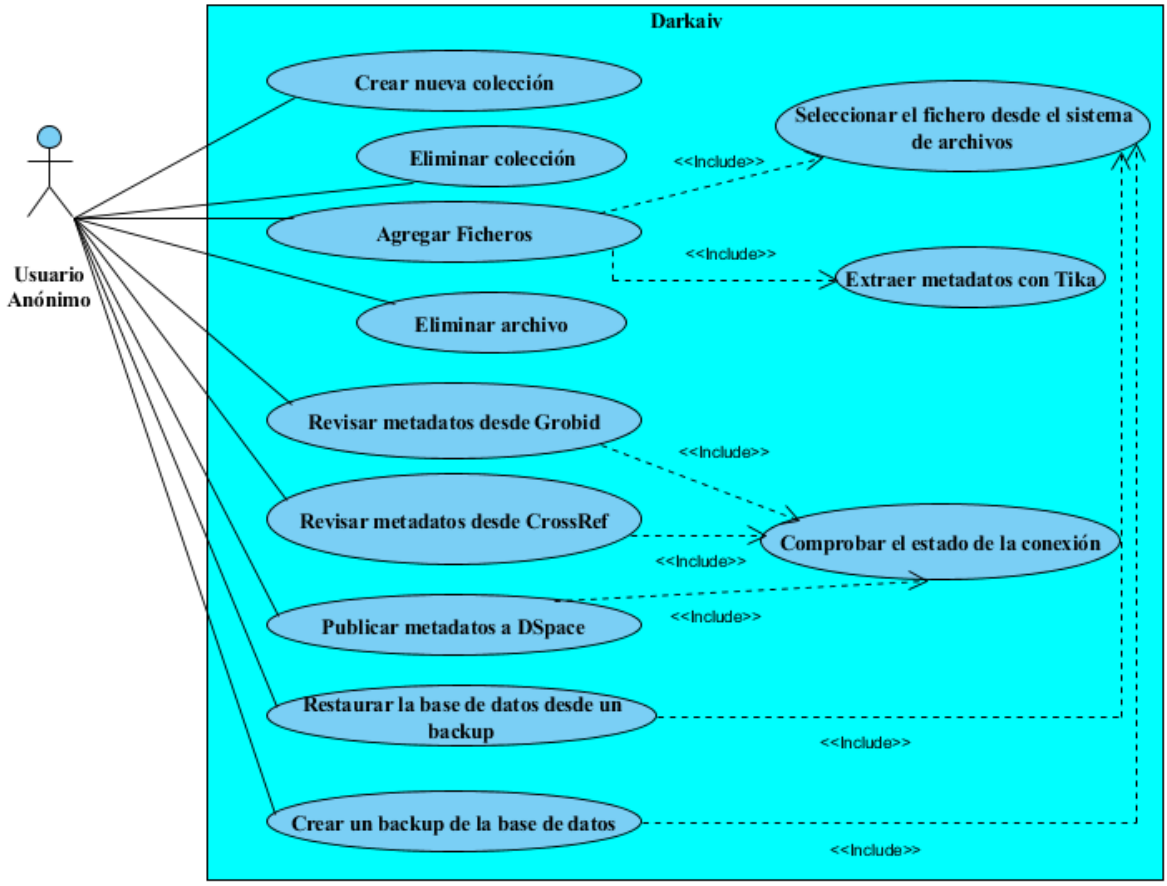

Figura 2.1 Diagrama de casos de uso y actores

2.2.1 Descripción de los casos uso del sistema

El sistema implementado está destinado a que un usuario sea capaz de llevar a cabo cada uno de los casos de uso descritos anteriormente. A continuación, se exponen características específicas de cada caso de uso.

Caso de uso	**Crear nueva colección**
Actor(es)	Usuario Anónimo
Resumen	El usuario debe seleccionar la opción "Add New Collection" e introducir el nombre y la descripción; la colección creada se introduce en la base de datos
Precondiciones	

Tabla 2.1 Descripción del caso de uso: ***Crear nueva colección***

Caso de uso	**Eliminar colección**
Actor(es)	Usuario Anónimo
Resumen	El usuario debe dar click derecho sobre la colección que desea eliminar y posteriormente confirmar esta operación
Precondiciones	Que la colección que se desea eliminar exista, y que no sea ninguna de las colecciones por defectos de la aplicación

Tabla 2.2 Descripción del caso de uso: ***Eliminar colección***

Caso de uso	**Agregar archivo**
Actor(es)	Usuario Anónimo
Resumen	El usuario debe ir al menú “File”, pulsar sobre la opción “Add Files…” y seleccionar los archivos que se desean insertar en el sistema
Precondiciones	

Tabla 2.3 Descripción del caso de uso: ***Agregar archivo***

Caso de uso	**Eliminar archivo**
Actor(es)	Usuario Anónimo
Resumen	Dar click derecho sobre el documento que se desea eliminar y seleccionar “Delete document”
Precondiciones	Que exista el archivo que se desea eliminar

Tabla 2.4 Descripción del caso de uso: ***Eliminar archivo***

Caso de uso	**Revisar metadatos desde Grobid**
Actor(es)	Usuario Anónimo

Resumen	Seleccionar los documentos a los que se les quiere revisar los metadatos; ir al menú "Review", seleccionar "From Grobid" y posteriormente confirmar esta operación
Precondiciones	Tener conexión con el servicio REST API de Grobid

Tabla 2.5 Descripción del caso de uso: ***Revisar metadatos desde Grobid***

Caso de uso	**Revisar metadatos desde CrossRef**
Actor(es)	Usuario Anónimo
Resumen	Seleccionar los documentos a los que se les quiere revisar los metadatos; ir al menú "Review", seleccionar "From CrossRef" y posteriormente confirmar esta operación
Precondiciones	Tener conexión con el servicio REST API de CrossRef

Tabla 2.6 Descripción del caso de uso: ***Revisar metadatos desde CrossRef***

Caso de uso	**Publicar metadatos a DSpace**
Actor(es)	Usuario Anónimo
Resumen	Seleccionar los documentos que se quieren publicar; ir al menú "Publish", seleccionar "DSpace" y seleccionar la colección hacia la que se quiere publicar
Precondiciones	Tener conexión con el servicio *REST* API de DSpace

Tabla 2.7 Descripción del caso de uso: ***Publicar metadatos a DSpace***

Caso de uso	**Crear un backup de la base de datos**
Actor(es)	Usuario Anónimo

Resumen	Ir al menú “Tools”, escoger la opción “Create backup” y seleccionar en la localización donde se quiere almacenar
Precondiciones	Tener espacio suficiente en disco

Tabla 2.8 Descripción del caso de uso: ***Crear un backup de la base de datos***

Caso de uso	**Restaurar la base de datos desde un backup**
Actor(es)	Usuario Anónimo
Resumen	Ir al menú “Tools”, seleccionar la opción “Restore from backup” y seleccionar el fichero desde el que se quiere hacer la recuperación
Precondiciones	Tener espacio suficiente en disco

Tabla 2.9 Descripción del caso de uso: ***Restaurar la base de datos desde un backup***

2.3 Diagrama de actividades

Los diagramas de actividades describen cómo se desarrolla un flujo de actividades entre elementos del sistema o del dominio.

De acuerdo a Booch et al. (1998) el diagrama de actividad gráficamente, es una colección de nodos y arcos que se utilizan para el modelado de los aspectos dinámicos de los sistemas y es fundamentalmente un diagrama que muestra el flujo de control entre actividades.

A continuación, en la Figura 2.2, se muestra el diagrama de actividades del flujo ideal de la aplicación

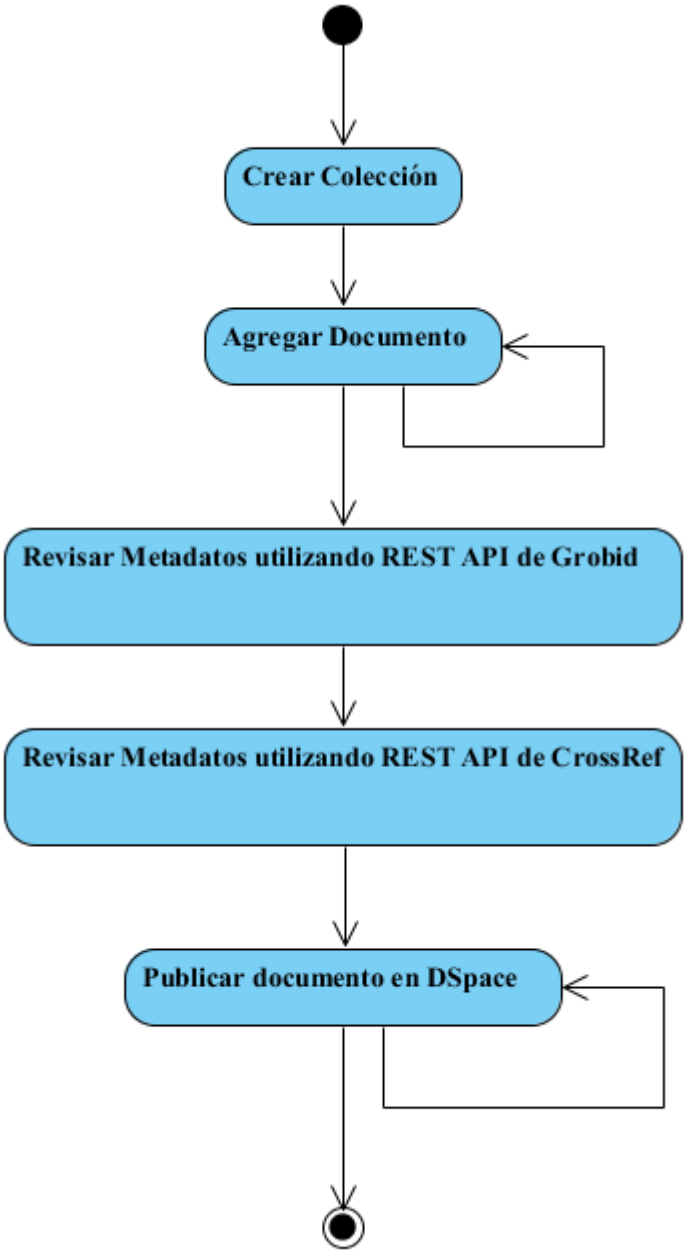

Figura 2.2 Diagrama de actividades del flujo ideal de la aplicación

2.4 Diagrama Entidad–Relación

Según define Chen (1976) el modelo Entidad/Relación puede ser usado como una base para una vista unificada de los datos, adoptando a su vez el enfoque más natural del mundo real que consiste en entidades e interrelaciones. En la Figura 2.3 se muestra el diagrama entidad relación de la base de datos del sistema.

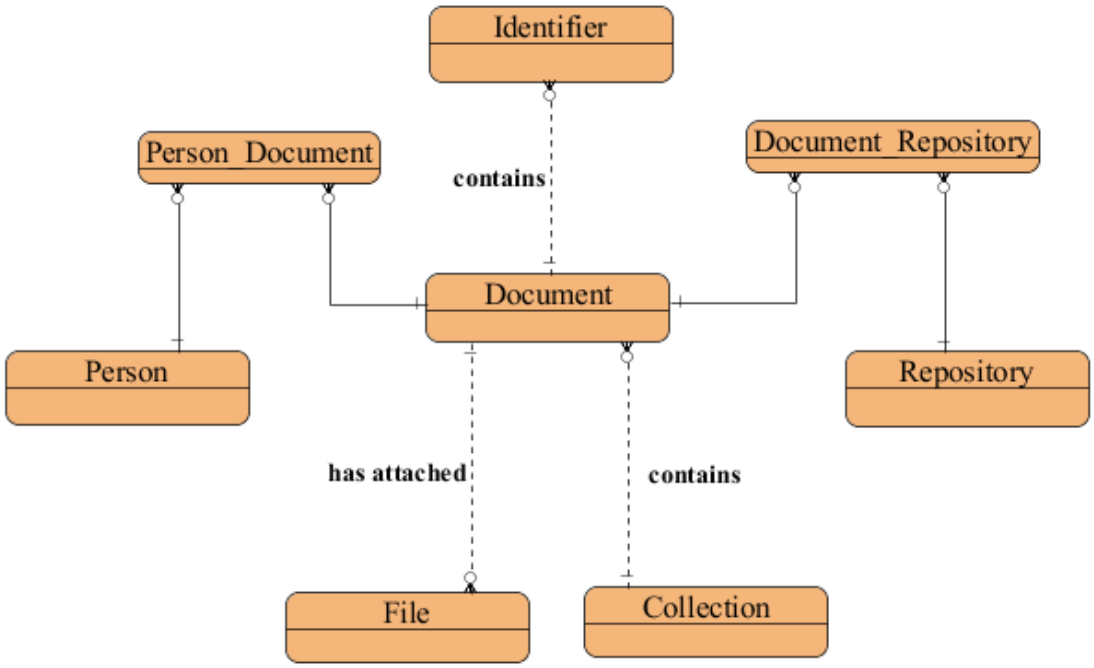

Figura 2.3 Diagrama Entidad-Relación

Document es la entidad principal del modelo, ya que en ella se almacenan la mayoría de los metadatos asociados a un archivo. En la base de datos también se almacenan otros datos relacionados con los documentos, como son las direcciones de los ficheros físicos asociados, la colección a la que pertenece, los repositorios a los cuales se han publicado, los identificadores del documento y los autores o personas relacionadas con el documento.

2.5 Patrones de diseño

De acuerdo a Alexander et al. (1977) cada patrón se asocia a un problema que ocurre una y otra vez en nuestro entorno, y entonces describe la solución de ese problema de tal manera que pueda ser utilizada un millón de veces. Aunque Alexander se refiere a patrones aplicados al diseño de edificios y ciudades, sus ideas son aplicables a los patrones de diseño orientado a objetos.

Según Gamma et al. (1993) los patrones de diseño orientado a objetos son descripciones de la comunicación de objetos y clases que pueden personalizarse para resolver un problema general de diseño en un contexto particular. En otras palabras, un patrón de diseño da

nombre, abstrae e identifica los elementos esenciales de un problema de diseño, su solución y los resultados de su aplicación.

Con la finalidad de obtener un sistema reutilizable fueron utilizados en la solución patrones de diseño como el Record Activo, Estrategia, Sistema de Mapeo, Fachada, Método Generador, Modelo Vista Controlador y N-Capas. En los siguientes subepígrafes se hace mención a cada uno de estos patrones en el contexto de la solución.

2.5.1 Record activo

Se define en Fowler et al. (2003) como Record Activo un objeto que *envuelve* una fila en una tabla de base de datos o vista, encapsula el acceso a la base de datos, y añade la lógica de dominio en esos datos. Un objeto que contiene los datos y el comportamiento. Muchos de estos datos son persistentes y tienen que almacenarse en una base de datos. Record activo utiliza el enfoque más obvio, poniendo la lógica de acceso a datos en el objeto de dominio. Este patrón se utiliza para mapear el modelo orientado a objetos que se utiliza en la capa de negocios de la aplicación al modelo relacional que se usa en la base de datos.

2.5.2 Estrategia

De acuerdo con Gamma et al. (1993), el patrón de diseño estrategia (o política) provee una vía de implementar una clase con múltiples comportamientos. Este patrón permite definir una familia de algoritmos, encapsular cada uno de ellos y hacerlos intercambiables.

En la solución este patrón se aplicó para garantizar la extensibilidad del sistema. Con su utilización, se brinda la posibilidad de implementar nuevos módulos para la validación de los metadatos diferentes a Grobid y CrossRef, y que la plataforma DSpace no sea el único repositorio hacia el cual se puedan publicar los documentos.

2.5.3 Fachada

Según Gamma et al. (1993) proporciona una interfaz unificada a un conjunto de interfaces en un subsistema. Una fachada define una interfaz de alto nivel que hace que el subsistema más fácil de usar.

La capa de comportamiento provee la clase Organizer como fachada, o punto de entrada, a los manejadores que se proveen en dicha capa para que sean utilizados por la capa de presentación.

2.5.4 Método generador

Según lo planteado en (Gamma et al. 1993), este patrón de diseño define una interfaz para crear un objeto, pero deja a las subclases decidir qué clase instanciar. Método generador permite aplazar la instanciación de una clase a las subclases.

2.5.5 Sistema de mapeo

Un sistema de mapeo es un objeto que establece una comunicación entre dos objetos independientes sin que estos conozcan los detalles de dicha comunicación (Fowler et al. 2003). Concretamente la utilización de este patrón permite desacoplar diferentes partes de un sistema manteniendo cada parte ignorante de la existencia de las otras.

En la solución se implementa una combinación de sistema de mapeo y método generador para garantizar la comunicación entre los sistemas que se utilizan. En particular, la utilización de este patrón permite crear documentos (representación propia del modelo de Darkaiv) a partir de los datos que proveen Tika, CrossRef y Grobid.

2.5.6 Modelo Vista Controlador

Atendiendo a lo expuesto por Deacon (2009) en el paradigma Modelo Vista Controlador (MVC) la entrada del usuario, el modelado del mundo externo y la retroalimentación visual para el usuario están separadas de maneras explícitas y manejadas por tres tipos de objetos, cada uno especializado para su tarea. La vista está relacionada con la salida gráfica y/o textual; el controlador interpreta las entradas (eventos) por parte del usuario, al mando del modelo y/o la vista para cambiar según sea apropiado. Por último, el modelo responde a las solicitudes de información sobre su estado (por lo general desde la vista), y responde a las instrucciones para cambiar el estado (por lo general desde el controlador).

La aplicación del patrón de diseño MVC posibilita crear sistemas flexibles y potentes. Esta arquitectura se utiliza como parte del diseño del sistema para lograr una solución desacoplada y extensible.

En el siguiente subepígrafe se hace referencia a la implementación del sistema, el cual hace uso de este patrón arquitectónico.

2.5.7 N-Capas

Según se expone en (Microsoft Corporation 2009) una arquitectura separada en capas se centra en el agrupamiento de funcionalidades dentro de capas distintas de la aplicación. Las funcionalidades dentro de cada capa están relacionadas con un papel o una responsabilidad común, mientras que la comunicación entre las capas es explícita y débilmente acoplada.

Un adecuado diseño en capas ayuda a mantener una separación de intereses, que, a su vez, apoya la flexibilidad y la facilidad de mantenimiento. En particular, en el diseño de la solución se tomaron en cuenta tres capas: en la capa de datos se mapea del modelo orientado a objetos al modelo relacional y se definen los servicios REST API que usa la aplicación, en la capa de negocios se implementan las funcionalidades del sistema y en la capa de presentación se le proporciona al usuario una interfaz para que utilice dichas funcionalidades.

2.6 Arquitectura del sistema

Tal y como se plantea en el epígrafe anterior la solución sigue un patrón de arquitectura en capas, destacándose una capa de presentación, una capa de negocio y una capa de acceso a datos. En la Figura 2.4 se muestra la arquitectura del sistema.

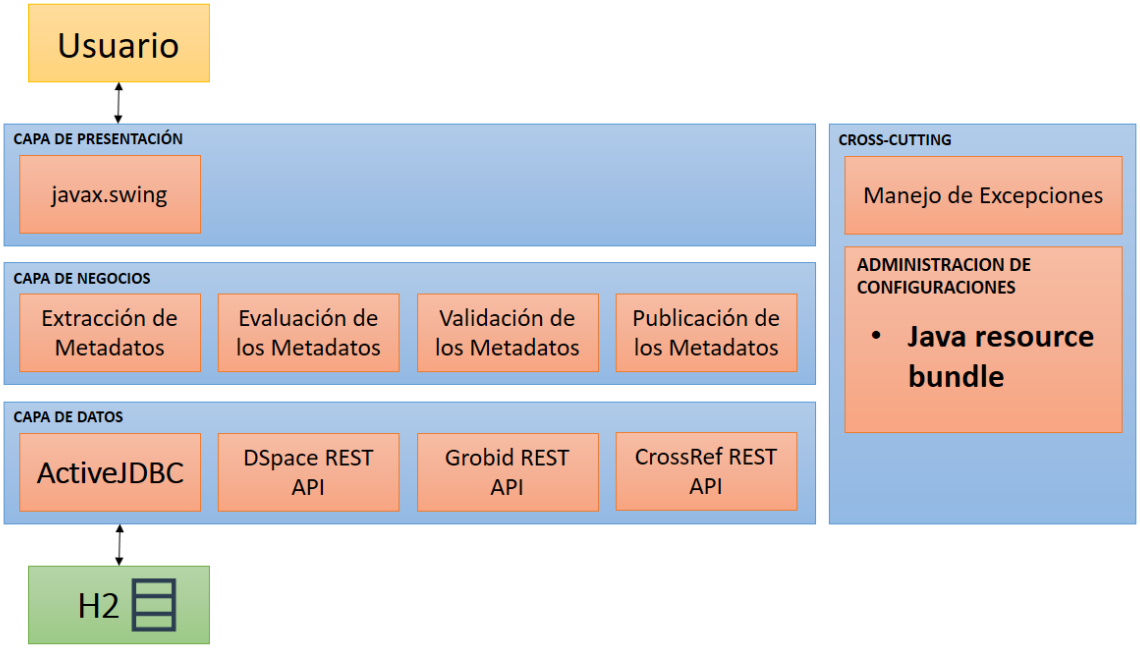

Figura 2.4 Arquitectura de la Aplicación

2.6.1 Capa de datos

En la capa de datos se utilizó una implementación para Java del patrón de diseño Record Activo (anteriormente descrito) conocida como ActiveJDBC. El mismo fue utilizado para mapeo de las clases del modelo a su representación tabular en el sistema de gestión de base de datos H2.

Como parte de esta capa también se incluyen el acceso a varios servicios web externos a los que se acceden mediante un servicio REST API; estos son: Grobid y CrossRef para la validación de los metadatos y DSpace para la publicación de los documentos.

2.6.2 Capa de negocio

En la capa de negocios se implementó un módulo por cada funcionalidad del sistema, así como paquetes auxiliares. Se utiliza el patrón de diseño estrategia en cada uno de los módulos principales de la aplicación: extracción, evaluación, validación y publicación.

En el módulo de extracción de metadatos se utiliza la biblioteca Tika para llevar a cabo dicha operación. Como parte de la evaluación de los metadatos se implementa la métrica de completitud de los metadatos. En el módulo de validación de metadatos se implementan conexiones con la REST API de CrossRef y con la de Grobid. El módulo de publicación

garantiza que esta se pueda llevar a cabo mediante una conexión con la REST API de DSpace.

2.6.3 Capa de presentación

La capa de presentación se implementó utilizando componentes de Java Swing y la interfaz se inspiró en el diseño del gestor bibliográfico Mendeley Desktop.

2.7 Diagrama de clases

Según lo planteado en Booch et al. (1998) los diagramas de clases son el esquema más común encontrado en los sistemas de modelado orientados a objetos. Un diagrama de clases muestra un conjunto de clases, interfaces, colaboraciones y relaciones.

A continuación, en la Figura 2.5, se muestra el diagrama de clases de la aplicación.

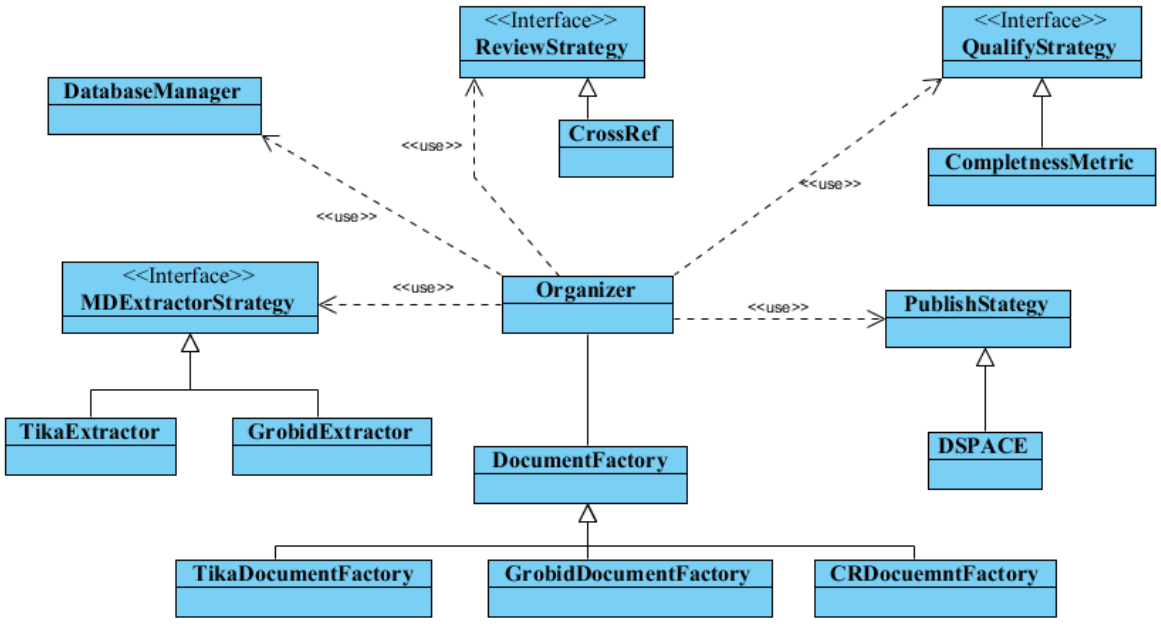

Figura 2.5 Diagrama de Clases

La implementación del módulo de extracción de metadatos utilizando la biblioteca Tika garantiza la extracción de metadatos básicos a una gran variedad de tipos de archivo.

La clase TikaExtractor implementa la interfaz MDExtractorStretegy, donde se definen los métodos necesarios para garantizar el proceso de extracción, incluyendo el método `getMetadata(File file)`.

Dentro del módulo de validación de los metadatos se realizaron dos implementaciones, una utilizando Grobid, el cual se consume como un servicio REST API y mejora considerablemente los metadatos obtenidos con Tika; y la otra utilizando CrossRef, que también se consume como un servicio REST API y tiene la característica que los datos obtenidos por esta fuente ya fueron validados por un órgano competente. Este es el orden que se recomienda en el flujo de trabajo: inicialmente Tika, luego Grobid y por último CrossRef.

Las clases CrossRef y Grobid ofrecen una implementación a la interfaz ReviewStrategy, donde se define el método necesario para hacer la revisión de un documento: `reviewMetadata(Document doc)`.

Para analizar los json devueltos por las REST API de CrossRef y Grobid se implementó un analizador sintáctico para cada uno, lo que permitió el acertado análisis de la información obtenida de estas fuentes.

Dentro del módulo de las métricas de calidad se implementó la métrica de completitud, que evalúa un conjunto de metadatos en dependencia de la existencia o no de estos. Los pesos (importancia) de cada metadato inicialmente se fijaron en uno, pero el usuario tiene la oportunidad de ajustarlos según sus necesidades.

Esta métrica se implementa en la clase CompletenessMetric, que a su vez implementa la interfaz QualifyStrategy, donde está definido el método `getMetric(Document doc)`.

Para la publicación se brinda una implementación que utiliza la REST API de DSpace 5.x, en la cual se mapean los datos de la representación interna del sistema Darkaiv a la del repositorio.

Los métodos necesarios para llevar a cabo la publicación se definen en la interfaz PublishStrategy, la cual es implementada en la clase DSpace. El método más importante para cumplir este objetivo es `publish(String collectionId, Document doc)`.

Como parte de la capa de negocios se brinda una interfaz (fachada) para que la capa de presentación se comunique con los componentes que brindan las funcionalidades descritas anteriormente. Esta fachada se implementa en la clase Organizer, donde se proveen un conjunto de método principales para responder a cada uno de los casos de uso descritos anteriormente, así como otros que ayudan a cumplimentar estas tareas.

2.8 Diagrama de componentes

Un *diagrama de componentes* muestra los elementos de un diseño de un sistema de software. Permite visualizar la estructura de alto nivel del sistema y el comportamiento del servicio que estos componentes proporcionan y usan a través de interfaces. En la Figura 2.6, se muestra el diagrama de componentes de la aplicación.

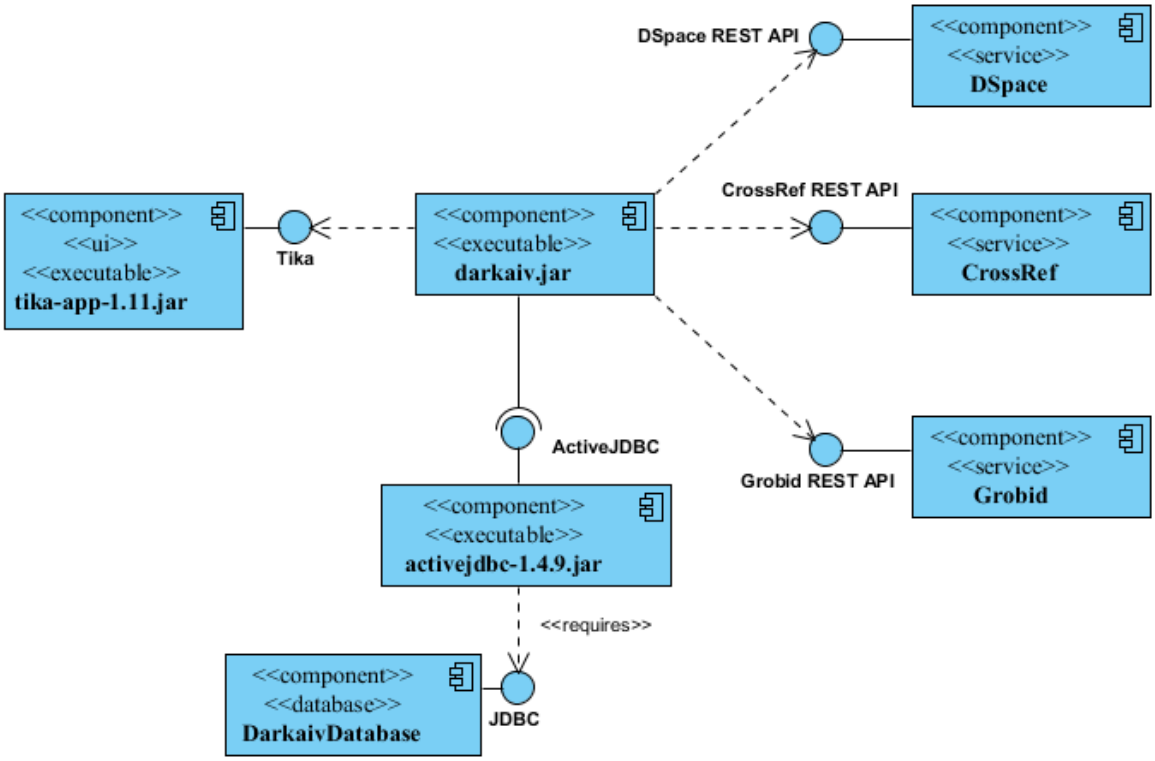

Figura 2.6 Diagrama de componentes de la aplicación

El sistema Darkaiv usa una base de datos embebida para almacenar la información acerca de los documentos a los cuales se les extrajo los metadatos. Este proceso de extracción se hace

utilizando la biblioteca Tika. La aplicación permite la validación de los metadatos extraídos usando los servicios REST API de CrossRef y de Grobid; así como la publicación de los mismos haciendo uso del servicio REST API de DSpace.

2.9 Conclusiones parciales

En este capítulo se realizó un análisis de los requerimientos funcionales y no funcionales del sistema, así como un estudio de los casos de uso que se encargan de delimitar el sistema definiendo las funciones que debe cumplir para el usuario. Se obtuvo un diseño mediante el modelado de las características principales del sistema utilizando la notación UML para una mejor comprensión de la estructura del sistema. Además, se diseñó la base de datos y la disposición de las clases para la etapa de implementación utilizando para ello los patrones de diseño con el fin de obtener una solución eficiente y escalable.

CAPÍTULO 3.

DESCRIPCIÓN DE LA SOLUCIÓN

CAPÍTULO 3. DESCRIPCIÓN DE LA SOLUCIÓN

El presente capítulo brinda una descripción general de la herramienta Darkaiv y sus principales funcionalidades. Más allá de los requerimientos técnicos, el capítulo profundiza en la interfaz de usuario de la aplicación, así como en opciones de configuración.

3.1 Breve descripción del sistema

Darkaiv persigue el objetivo de facilitar un proceso compuesto por tres fases: la extracción automática de metadatos de documentos, la revisión y validación de estos y su publicación en una biblioteca digital. Es un producto enfocado principalmente a las personas encargadas de la gestión de información en instituciones educacionales y las operaciones fundamentales que se realizan con él son: importación de documentos *(posee dos vías para su realización: seleccionar archivos con formato* ***.pdf*** *o seleccionar una carpeta para que sean añadidos todos los documentos de este formato que se encuentren en ella o en alguna de sus subcarpetas)*, revisión de documentos utilizando los servicios de Grobid y de CrossRef *(operación encaminada a mejorar la completitud y calidad de los metadatos)* y publicación de los metadatos en repositorios DSpace.

3.2 Requerimientos técnicos del sistema

Para la utilización de la herramienta Darkaiv es importante contar con un grupo de requerimientos mínimos de hardware y de software que son obligatorios para el correcto funcionamiento de la aplicación. Es por ello que se recomienda la siguiente configuración:

11. Un gigabyte (GB) de memoria RAM o superior
12. 100 megabyte (MB) de espacio disponible en disco duro o superior
13. Un procesador de un gigahercio(GHz) o superior de 32 bits (x86) o 64 bits (x64)
14. Sistema Operativo Linux, Mac Os o Windows (XP o superior)
15. Máquina Virtual de Java 1.7 o superior

3.3 Estructura de archivos y directorios de Darkaiv

La herramienta Darkaiv propone una estructura de archivos y directorios en los cuales organiza la configuración, base de datos y dependencias del sistema. A continuación, se describe el contenido del directorio raíz (ver Figura 3.1):

- **Directorio config:** En esta carpeta se encuentran los archivos de configuración de los servicios que brinda Darkaiv (extracción, revisión y publicación de los metadatos de los documentos).
- **Directorio darkaiv:** En esta carpeta se encuentra la base de datos del sistema.
- **Directorio lib:** En esta carpeta se encuentran las dependencias[2] del sistema.
- **Archivo darkaiv-1.0.jar:** Archivo ejecutable de la aplicación.

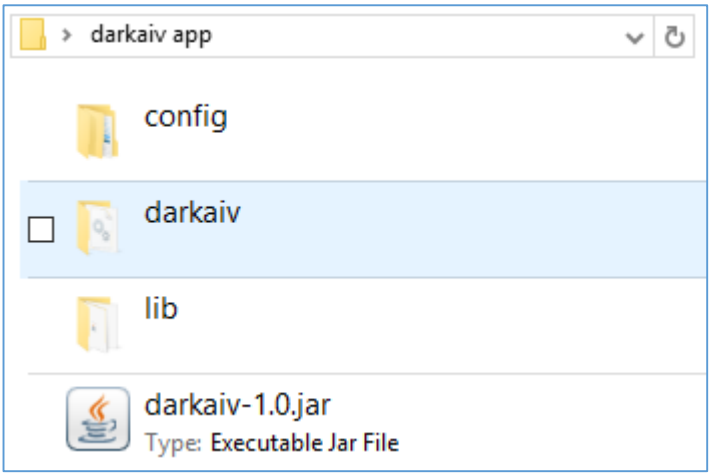

Figura 3.1 Organización de los archivos

Dentro del directorio ***config*** existen cuatro subcarpetas relacionadas con la configuración del sistema. A continuación, se describe el contenido de cada una de estas:

- **Directorio completeness_metric:** En esta carpeta se encuentran los archivos de configuración de los pesos asignados a los metadatos en dependencia del tipo de documento. De esta forma, teniendo en cuenta que Darkaiv soporta seis tipos de

[2] **Dependencias**: son aplicaciones o bibliotecas requeridas por otro programa para poder funcionar correctamente. Por ello se dice que dicho programa depende de tales aplicaciones o bibliotecas.

documentos: Book, Book Chapter, Generic, Journal Article, Conference Proceedings y Thesis, se encuentran seis archivos, estos son:

- **Archivo book.properties:** Se encuentran los pesos para los metadatos de los documentos de tipo Book.
- **Archivo book-chapter.properties:** Se encuentran los pesos para los metadatos de los documentos de tipo Book Chapter.
- **Archivo generic.properties:** Se encuentran los pesos para los metadatos de los documentos de tipo Generic.
- **Archivo journal-article.properties:** Se encuentran los pesos para los metadatos de los documentos de tipo Journal Article.
- **Archivo proceedings.properties:** Se encuentran los pesos para los metadatos de los documentos de tipo Conference Proceedings.
- **Archivo thesis.properties:** Se encuentran los pesos para los metadatos de los documentos de tipo Thesis.

- **Directorio crossref:** En esta carpeta se encuentra el archivo para la configuración de la dirección web de la plataforma CrossRef.
- **Directorio dspace:** En esta carpeta se encuentra el archivo para la configuración de las direcciones web del repositorio Dspace.
- **Directorio grobid_service:** En esta carpeta se encuentra el archivo para la configuración de la dirección web de Grobid Service.

3.4 Interfaz de Usuario

La interfaz de la aplicación está inspirada en la interfaz visual del gestor bibliográfico Mendeley, uno de los más utilizados en la actualidad en el sector académico e investigativo. Esta consta de una ventana principal y de un grupo de opciones para la gestión de los documentos de la aplicación.

3.4.1 Ventana Principal

La ventana principal de la aplicación está dividida en tres secciones acompañadas por un menú de opciones y una barra de estado del sistema (ver Figura 3.2).

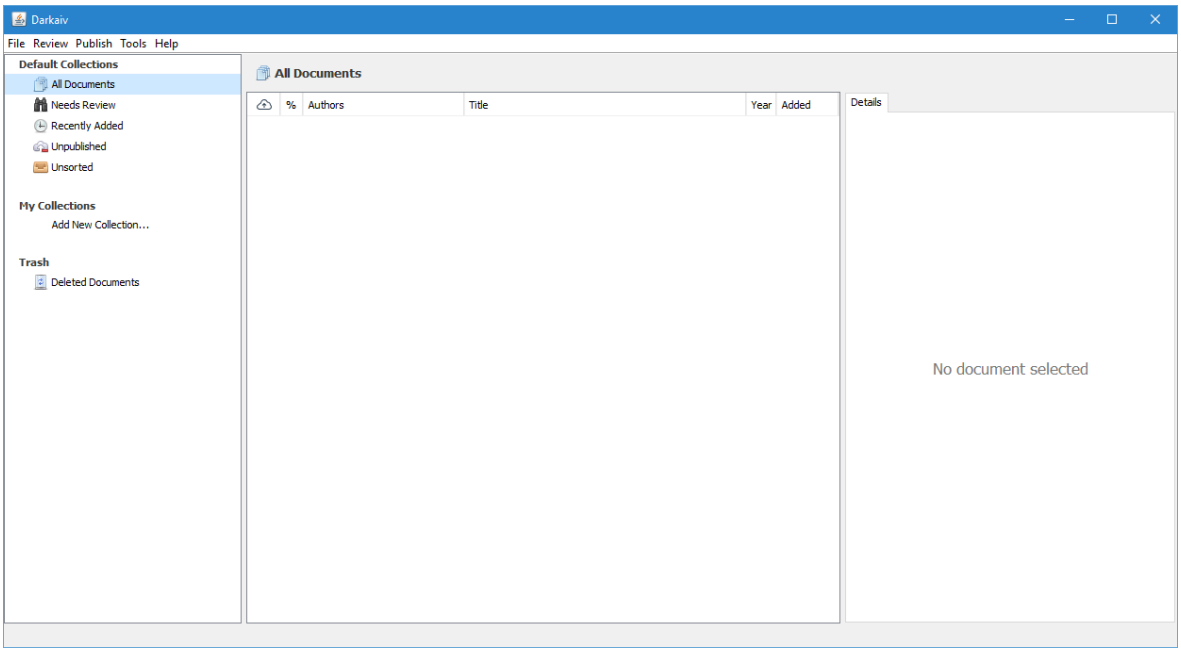

Figura 3.2 Ventana Principal del Darkaiv

El menú principal muestra un grupo de opciones agrupadas en cinco submenús:

- **Menú Files:** En este menú se ubican las opciones para la importación de archivos al sistema. La importación de archivos puede realizarse mediante dos vías: la primera, seleccionando los archivos con formato ***.pdf*** que se desean insertar; la segunda, seleccionando la carpeta que contiene los archivos deseados *(se añaden todos los documentos en formato* ***.pdf*** *que se encuentren en ella o en sus subcarpetas)*

- **Menú Review:** En este menú se ubican las opciones vinculadas a la revisión y/o completamientos de los metadatos de los documentos importados. Esta operación es posible realizarla utilizando el servicio web de Grobid Service o el de la plataforma Crossref.
- **Menú Publish:** En este menú se ubica la opción de publicar los documentos importados, así como sus metadatos, en un repositorio DSpace.
- **Menú Tools:** En este menú se ubican las opciones para el manejo de salvas de los datos de la aplicación.

- **Menú Help:** Por último, en este menú se ubican las opciones de ayuda del sistema.

Además del menú descrito anteriormente, la ventana principal del sistema incluye tres secciones dirigidas al manejo de las colecciones, los documentos que la componen, así como sus metadatos. De izquierda a derecha estas secciones son:

- **Listado de colecciones:** En esta sección se gestionan todas las colecciones del sistema, tanto las que son creadas por el usuario, como las predefinidas por la aplicación (All Documents, Needs Review, Recently Added, Unpublished, Unsorted y Deleted Documents).
- **Listado de documentos:** En esta sección se muestran los documentos pertenecientes a las colecciones e incluye un grupo de indicadores y opciones vinculadas a los procesos de revisión y publicación de documentos.
- **Detalles:** Esta sección permite el manejo de los metadatos de los documentos.

Finalmente, la parte inferior de la ventana principal incluye una barra de estado que solo es visible cuando el sistema realiza una determinada operación, como por ejemplo insertar archivos.

3.4.2 Gestión de Colecciones

Darkaiv permite organizar los documentos importados en colecciones. En este sentido, las colecciones permiten al usuario el agrupamiento de documentos de la misma clase, por ejemplo, los documentos correspondientes a una misma área de investigación. Una colección es como una carpeta y Darkaiv provee un conjunto de estas por defecto, además de permitir la creación de nuevas colecciones.

De forma predeterminada, la aplicación incluye seis colecciones de documentos. Estas colecciones son las siguientes:

- **Colección All Documents:** En esta colección se muestran todos los documentos existentes en la base de datos del sistema.
- **Colección Needs Review:** En esta colección se muestran aquellos documentos que hayan sido marcados por el usuario como que necesitan revisión.
- **Colección Recently Added:** En esta colección se muestran aquellos documentos que hayan sido insertados al sistema en los últimos tres días.

- **Colección Unpublished:** En esta colección se muestran aquellos documentos que nunca hayan sido publicados en un repositorio DSpace.
- **Colección Unsorted:** En esta colección se muestran aquellos documentos que no fueron añadidos en ninguna colección creada por el usuario.
- **Colección Deleted Documents:** En esta colección se muestran aquellos documentos que hayan sido eliminados por el usuario. Funciona como una papelera de reciclaje desde la cual el sistema permite restaurar documentos o eliminarlos permanentemente.

Como se mencionó anteriormente, el sistema permite al usuario la creación de nuevas colecciones, así como la modificación de estas. En las siguientes secciones se describen los pasos a seguir para realizar estas operaciones.

3.4.2.1 Creación de una colección

Para crear una nueva colección el usuario debe seleccionar la opción "Add New Collection" que se encuentra dentro de la lista de colecciones (ver Figura 3.3).

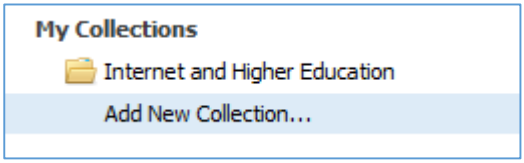

Figura 3.3 Creación de una nueva colección

Luego se muestra una ventana donde son llenados los datos referentes a la nueva colección, nombre y descripción (*name* y *description* en inglés). La colección se crea una vez que el usuario llene estos datos y presione el botón "Create" (ver Figura 3.4).

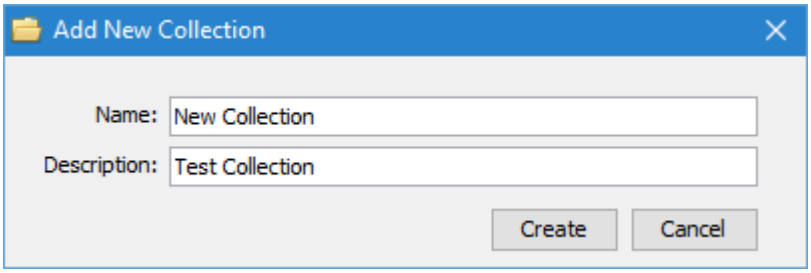

Figura 3.4 Ventana con los datos para la creación de una nueva colección

Luego de realizado este proceso la nueva colección se adiciona a la lista de colecciones que se muestran en la sección izquierda de la ventana principal.

3.4.2.2 Edición de colecciones

En relación con las colecciones, Darkaiv solamente permite editar las propiedades de aquellas creadas por los usuarios. En este sentido, no es posible modificar las colecciones predeterminadas de la aplicación. Para editar una colección se debe presionar Click Derecho sobre la que se desea modificar, mostrándose el menú contextual que se aprecia en la Figura 3.5:

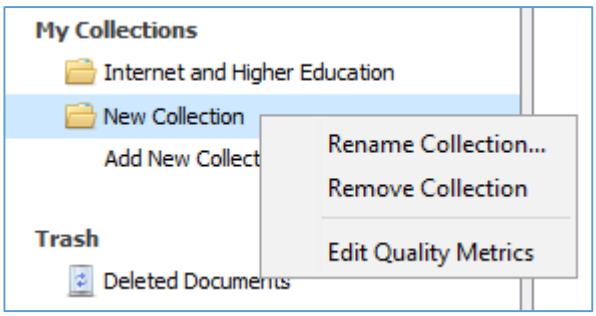

Figura 3.5 Menú de modificaciones a las colecciones

Las opciones del menú contextual de las colecciones permiten renombrar, eliminar y editar las métricas de la calidad de una colección. Estas opciones agrupadas en dos subgrupos se describen como:

- **Opción Rename Collection:** Esta opción permite renombrar la colección seleccionada. Como el nombre de cada colección es único, el sistema notifica e impide la acción en los casos de que el usuario intente poner el nombre de una colección existente.
- **Opción Remove Collection:** Esta opción permite eliminar la colección seleccionada. *(Esta operación requiere confirmación ya que la misma posee un carácter irreversible)*
- **Opción Edit Quality Metrics:** Esta opción permite editar el umbral de calidad de los metadatos de una colección tomando en cuenta una métrica de completamiento. De esta forma el sistema es capaz de mostrar en una escala de colores, el porciento

de completitud de cada documento en dependencia de la calidad estimada por el sistema. Dicha escala de colores es interpretada de la siguiente forma:

- **Verde:** Porciento de completitud de los metadatos ≥ umbral
- **Amarillo:** Porciento de completitud de los metadatos ≥ umbral / 2
- **Rojo:** Porciento de completitud de los metadatos < umbral / 2

Para un mejor entendimiento de este concepto se hace importante mostrar un ejemplo. Suponga que un usuario desea asegurar que los documentos de una colección tengan un porciento de completitud de metadatos mayor o igual a el 90%. En términos de la aplicación, esto significaría colocar como umbral 90% (ver Figura 3.6).

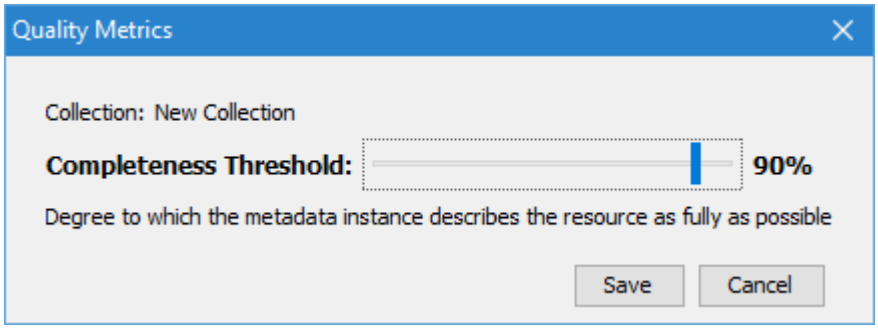

Figura 3.6 Ventana de selección del umbral. Umbral = 90%

Ciertamente, a partir de este momento los documentos de esta colección mostrarán en el campo (%) de la lista de documentos los colores distribuidos de la siguiente forma:

- **Verde:** Porciento de completitud de los metadatos ≥ 90%
- **Amarillo:** Porciento de completitud de los metadatos ≥ 45%
- **Rojo:** Porciento de completitud de los metadatos < 45%

3.4.3 Gestión de documentos

La gestión de documentos en Darkaiv se enmarca en tres procesos fundamentales: inserción, eliminación y revisión de documentos. En los siguientes epígrafes se profundiza en las particularidades de cada uno de estos procesos.

3.4.3.1 Insertar Documentos

La inserción o importación de documentos en la herramienta Darkaiv se realiza mediante las opciones Add Files y Add Folder pertenecientes al menú File de la ventana principal (ver Figura 3.7).

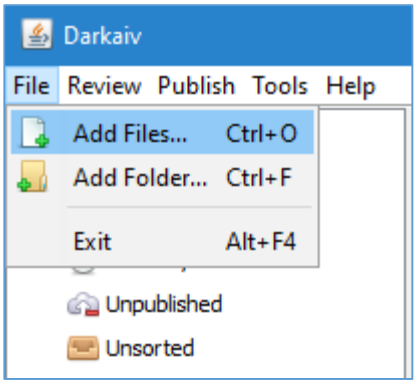

Figura 3.7 Menú File de la Ventana Principal desplegado

Al seleccionar la opción Add Files, el sistema muestra una ventana de selección de archivos mediante la cual el usuario puede seleccionar los documentos que desea importar en la aplicación. Un ejemplo de esto se muestra en la Figura 3.8 donde se seleccionan 15 archivos ubicados en una carpeta con nombre *Internet and Higher Education*.

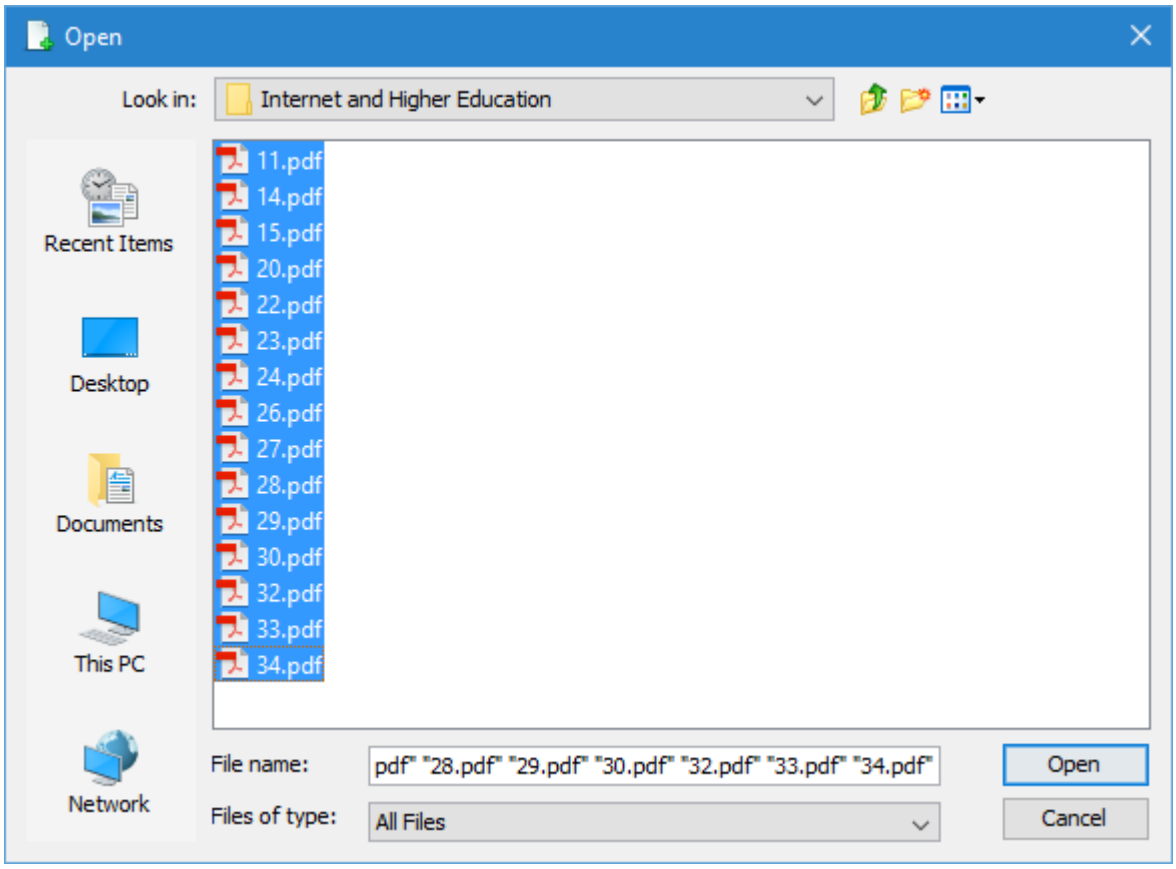

Figura 3.8 Ventana de selección de archivos

De forma muy similar el usuario también puede importar documentos mediante la opción Add Folder. Es importante destacar que solo se permite importar documentos con extensión ***.pdf***.

Una vez que se seleccionan los archivos, por cualquiera de las vías anteriormente mencionadas, el sistema extrae los metadatos de forma automática utilizando la Biblioteca Apache Tika mostrando estos en la lista de documentos de la ventana principal (ver Figura 3.9).

	%	Authors	Title	Year	Added
	64	Christine E. Nickel	A comparison of student satisfaction and value of aca...	2011	Jun 16 2016
	55		doi:10.1016/j.iheduc.2005.03.001	2005	Jun 16 2016
	55		doi:10.1016/j.iheduc.2004.09.001	2004	Jun 16 2016
	55		PII: S1096-7516(02)00102-1	2002	Jun 16 2016
	64	Debra Lee	To blog or not to blog: Student perceptions of blog eff...	2010	Jun 16 2016
	55		PII: S1096-7516(02)00130-6	2002	Jun 16 2016
	55	Carol Ritter, Barbara Polnick...	Classroom learning communities in educational leaders...	2010	Jun 16 2016
	55		PII: S1096-7516(01)00053-7	2001	Jun 16 2016
	55		doi:10.1016/j.iheduc.2007.12.003	2008	Jun 16 2016
	55	Steven R. Terrell, Martha M...	The development, validation, and application of the Do...	2009	Jun 16 2016
	55		PII: S1096-7516(01)00037-9	2001	Jun 16 2016
	55		doi:10.1016/j.iheduc.2005.06.004	2005	Jun 16 2016
	55		doi:10.1016/j.iheduc.2006.10.001	2007	Jun 16 2016
	55	Fengfeng Ke, Kui Xie	Toward deep learning for adult students in online courses	2009	Jun 16 2016
	55	Alfred P. Rovai, Mervyn J. ...	Development of an instrument to measure perceived c...	2009	Jun 16 2016

Figura 3.9 Lista de documentos de la Ventana Principal al terminar la inserción

3.4.3.2 Revisión de documentos

Una vez que son importados los documentos al sistema, Darkaiv facilita al usuario tres vías para mejorar la calidad de los metadatos extraídos de forma automática durante la importación. Dos de estas vías se basan en la utilización de servicios web, disponibles al usuario a través del menú Review de la aplicación (ver Figura 3.10). La tercera vía consiste en la modificación *manual* de los metadatos mediante el panel de detalles (Details en inglés) ubicado en la parte derecha de la ventana principal (Ver Figura 3.11).

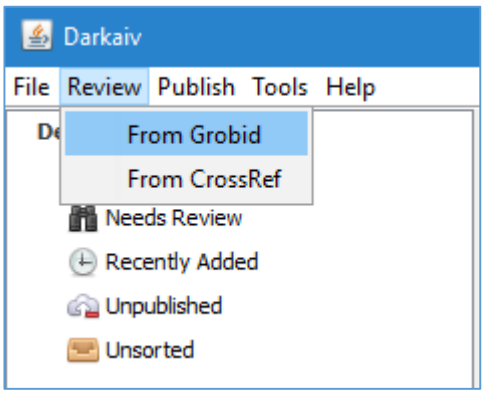

Figura 3.10 Menú Review de la Ventana Principal desplegado

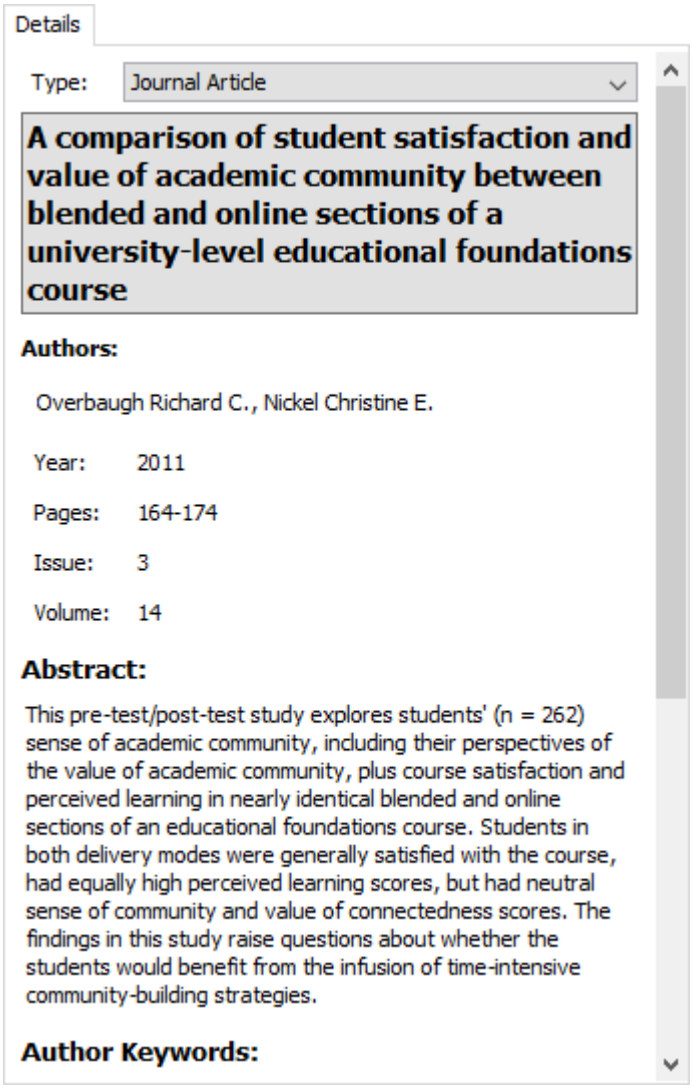

*Figura 3.11 Metadatos del documento titulado **A comparison of student satisfaction and value of academic community between blended and online sections of a university-level educational foundations course***

Para la utilización de las dos opciones del menú Review: From Grobid y From CrossRef es necesario de antemano tener configuradas las direcciones web (URL[3]) de estos servicios. Tales direcciones se especifican en los archivos *grobid.properties* y *crossref.properties* ubicados en los directorios *config/grobid_service/* y *config/crossref/* respectivamente (ver Figura 3.12 y Figura 3.13).

```
15
16      grobid.server.url=http://10.12.1.39:8080/grobid-service
```

*Figura 3.12 Archivo **config/grobid_service/grobid.properties***

```
15
16      crossref.server.url=http://api.crossref.org/works/
```

*Figura 3.13 Archivo **config/crossref/crossref.properties***

A continuación, se muestra la misma lista de documentos de la Figura 3.9 luego de aplicar la revisión de los metadatos mediante los servicios de Grobid y CrossRef. Tal y como evidencian las figuras 3.14 y 3.15 se puede apreciar un aumento considerable en la calidad de los metadatos de los documentos luego de aplicadas ambas opciones.

[3] **URL** (del inglés Uniform Resource Locator) es un identificador de recursos uniforme (Uniform Resource Identifier, **URI**) cuyos recursos referidos pueden cambiar, esto es, la dirección puede apuntar a recursos variables en el tiempo. Están formados por una secuencia de caracteres, de acuerdo a un formato modélico y estándar, que designa recursos en una red.

	%	Authors	Title	Year	Added
	82	Richard C Overbaugh, Ch...	A comparison of student satisfaction and value of ac...	2011	Jun 16 2016
	73	Alfred P Rovai, T , Mervy...	Feelings of alienation and community among higher ...	2005	Jun 16 2016
	73	Alfred P Rovai, Mervyn J ...	The Classroom and School Community Inventory: De...	2004	Jun 16 2016
	64		Development of an instrument to measure classroom...	2002	Jun 16 2016
	82	Olivia Halic, Debra Lee, Tr...	To blog or not to blog: Student perceptions of blog e...	2010	Jun 16 2016
	64		Sense of community, perceived cognitive learning, a...	2002	Jun 16 2016
	73	Carol Ritter, Barbara Poln...	Classroom learning communities in educational leade...	2010	Jun 16 2016
	64		Classroom community at a distance A comparative a...	2001	Jun 16 2016
	73	Marisa Exter, Nichole Harl...	Story of a conference: Distance education students'...	2008	Jun 16 2016
	73	Steven R Terrell, □ , Mar...	The development, validation, and application of the ...	2009	Jun 16 2016
	64		Building and sustaining community in asynchronous l...	2001	Jun 16 2016
	73	Terrie Lynn Thompson, C...	Community building, emergent design and expecting...	2005	Jun 16 2016
	73		Facilitating online discussions effectively	2007	Jun 16 2016
	73	Fengfeng Ke, Kui Xie	Toward deep learning for adult students in online co...	2009	Jun 16 2016
	73	Alfred P Rovai, Mervyn J ...	Development of an instrument to measure perceived...	2009	Jun 16 2016

Figura 3.14 Lista de documentos luego de ser revisados utilizando Grobid

	%	Authors	Title	Year	Added
	92	Richard C. Overbaugh, C...	A comparison of student satisfaction and value of ac...	2011	Jun 16 2016
	85	Alfred P. Rovai, Mervyn J...	Feelings of alienation and community among higher ...	2005	Jun 16 2016
	85	Alfred P. Rovai, Mervyn J...	The Classroom and School Community Inventory: De...	2004	Jun 16 2016
	64		Development of an instrument to measure classroom...	2002	Jun 16 2016
	92	Olivia Halic, Debra Lee, Tr...	To blog or not to blog: Student perceptions of blog e...	2010	Jun 16 2016
	64		Sense of community, perceived cognitive learning, a...	2002	Jun 16 2016
	85	Carol Ritter, Barbara Poln...	Classroom learning communities in educational leade...	2010	Jun 16 2016
	64		Classroom community at a distance A comparative a...	2001	Jun 16 2016
	85	Marisa Exter, Nichole Harl...	Story of a conference: Distance education students'...	2008	Jun 16 2016
	85	Steven R. Terrell, Martha...	The development, validation, and application of the ...	2009	Jun 16 2016
	64		Building and sustaining community in asynchronous l...	2001	Jun 16 2016
	85	Terrie Lynn Thompson, C...	Community building, emergent design and expecting...	2005	Jun 16 2016
	85	Alfred P. Rovai	Facilitating online discussions effectively	2007	Jun 16 2016
	85	Fengfeng Ke, Kui Xie	Toward deep learning for adult students in online co...	2009	Jun 16 2016
	85	Alfred P. Rovai, Mervyn J...	Development of an instrument to measure perceived...	2009	Jun 16 2016

Figura 3.15 Lista de documentos luego de ser revisados utilizando Grobid y CrossRef

3.4.3.3 Eliminación de documentos

Para eliminar un documento del sistema se debe presionar Click Derecho sobre la lista de documentos ubicada en la ventana principal de la aplicación y seleccionar la opción “Delete Documents”. Como se muestra en la Figura 3.16, esta opción es aplicada a tres documentos que poseen valores medios bajos en cuanto a la completitud de sus metadatos (color amarillo).

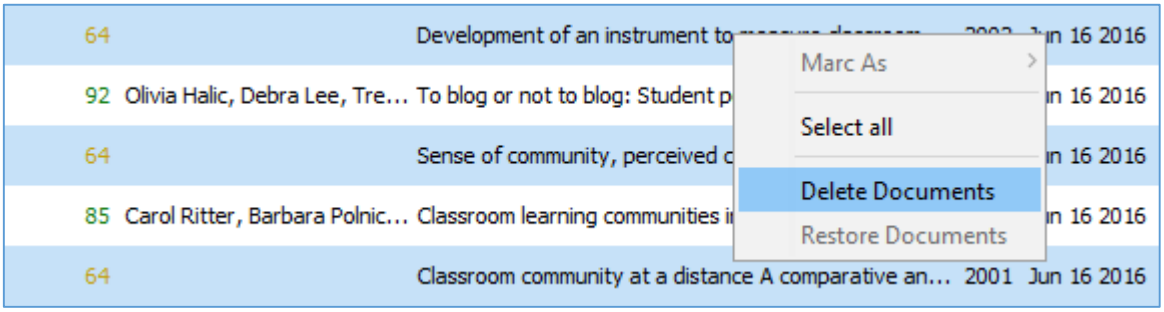

Figura 3.16 Eliminar Documentos

Una vez borrado un documento de lista de documentos, este pasa a una colección denominada "Deleted Documents", la cual funciona como una papelera de reciclaje de donde se pueden recuperar los documentos o eliminarlos completamente del sistema.

Para recuperar un documento ubicado en la colección "Deleted Documents" se utiliza la opción "Restore Documents", en cambio para eliminarlo de forma irrevocable del sistema existen dos vías fundamentales. La primera, mediante la opción "Empty Trash" de la colección Deleted Documents (ver figura debajo), mientras que la segunda, por medio de la opción "Delete Documents" en el contexto de la colección asociada a la papelera de reciclaje.

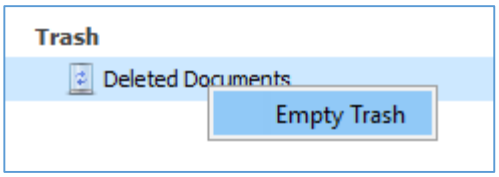

Figura 3.17 Limpiar colección de documentos eliminados (Deleted Documents)

3.4.4 Publicación de documentos

Una vez que se realicen las correcciones pertinentes a los metadatos de los documentos, con el fin de tenerlos con la calidad que se desee, Darkaiv permite la publicación de estos en un repositorio DSpace. Ciertamente, dicho repositorio debe estar debidamente configurado en el archivo *dspace.properties* que se encuentra en *config/dspace/.*

Tal y como se muestra en la Figura 3.18, son cinco los datos a introducir para configurar el repositorio Dspace que se utiliza en la publicación de documentos:

- ***dspace.server.instance:*** Nombre de la instancia de DSpace, ej. DSpaceUCLV.

- ***dspace.server.url:*** URL del Servicio Web REST[4] del DSpace.
- ***dspace.server.url.gui:*** URL de la interfaz visual del DSpace.
- ***dspace.server.user:*** Usuario a través del cual se realiza la conexión al DSpace, ej. *yoilan@uclv.edu.cu*
- ***dspace.server.password:*** Contraseña del usuario a través del cual se realiza la conexión al DSpace.

```
15
16      dspace.server.instance=LocalDSpaceServer
17      dspace.server.url=https://192.168.56.2:8443/rest
18      dspace.server.url.gui=http://192.168.56.2:8080/xmlui/
19      dspace.server.user=
20      dspace.server.password=
```

*Figura 3.18 Ejemplo de configuración de un DSpace a través del archivo **config/dspace/dspace.properties***

Una vez configurado el servidor de DSpace, el usuario podrá publicar cualquier documento seleccionado presionando la opción DSpace Collection que se encuentra en el menú Publish de la ventana principal (ver figura debajo).

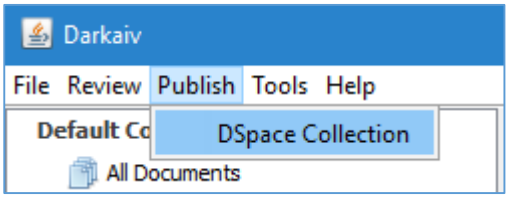

Figura 3.19 Menú Publish de la Ventana Principal desplegado

Al presionar en esta opción, al usuario se le muestra una ventana con los datos del Dspace, seguida de otra ventana que le permite navegar y seleccionar la colección de destino para los documentos que desea publicar (ver Figura 3.20).

[4] **REST** o Transferencia de Estado Representacional (**Re**presentational **S**tate **T**ransfer en inglés): arquitectura que, haciendo uso del protocolo HTTP, proporciona una API que utiliza cada uno de sus métodos (GET, POST, PUT, DELETE, etc.) para poder realizar diferentes operaciones entre la aplicación que ofrece el servicio web y el cliente.

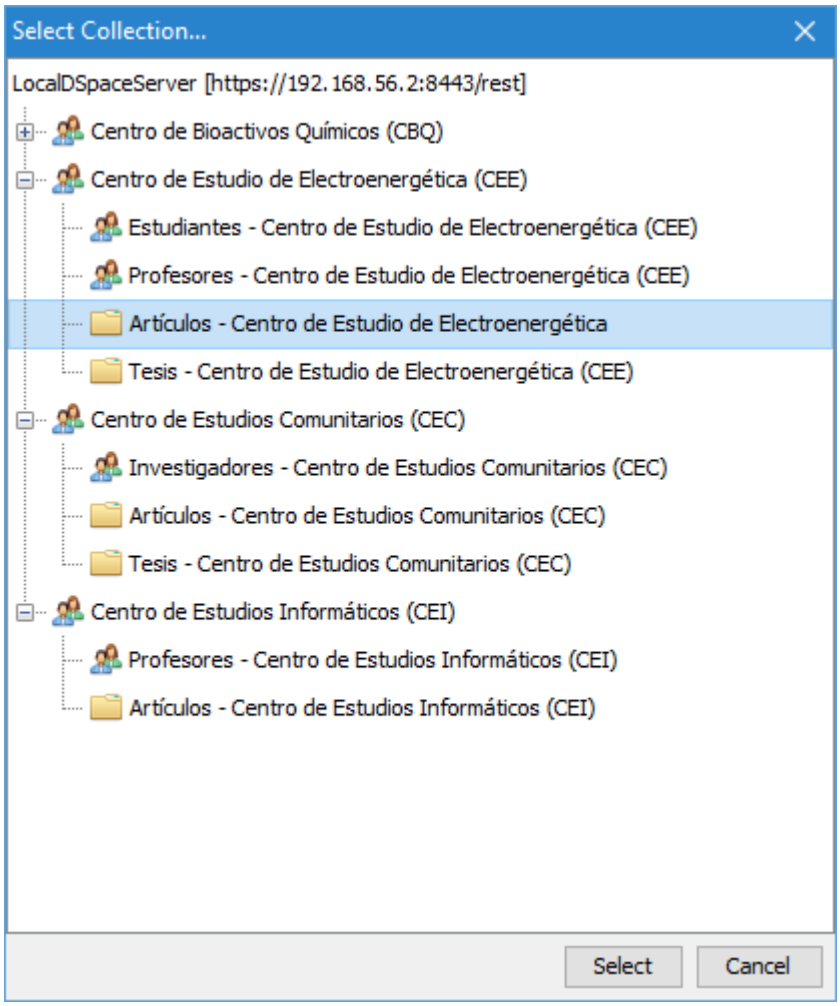

Figura 3.20 Ventana de selección de colección

Al terminar el proceso de publicación de los documentos se muestra un reporte donde se informa al usuario si el proceso de publicación concluyó satisfactoriamente, si ocurrieron errores o si fue cancelado. Dicho reporte incluye también la cantidad de documentos publicados y no publicados en el repositorio. En la Figura 3.21 se muestra un ejemplo de reporte de publicación.

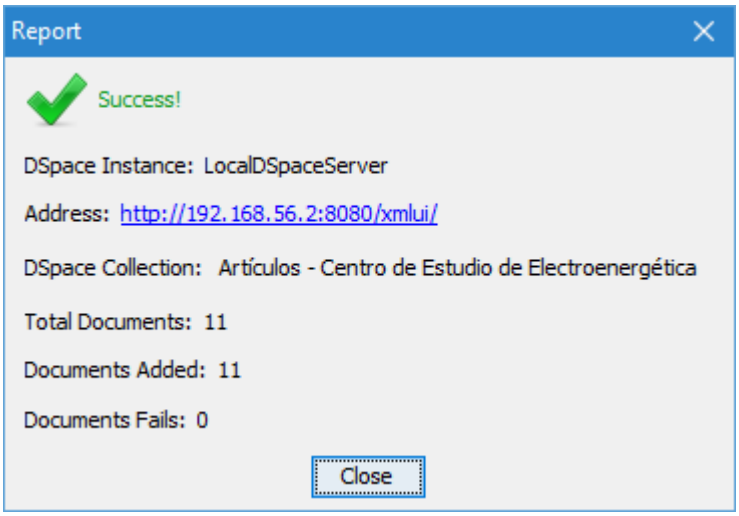

Figura 3.21 Reporte de publicación

Finalmente, en la lista de documentos de la ventana principal, también se refleja la publicación de los documentos mediante el campo representado por el icono: ☁. Todos aquellos documentos publicados por el usuario exhiben el icono ↑, mientras que los que no han sido publicados tienen el campo en blanco.

A continuación, se presenta la imagen de una lista de documentos publicados.

☁	%	Authors	Title	Year	Added
↑	92	Richard C. Overbaugh, Chri...	A comparison of student satisfaction and value of aca...	2011	Jun 16 2016
↑	85	Alfred P. Rovai, Mervyn J. ...	Feelings of alienation and community among higher ed...	2005	Jun 16 2016
↑	85	Alfred P. Rovai, Mervyn J. ...	The Classroom and School Community Inventory: Dev...	2004	Jun 16 2016
↑	92	Olivia Halic, Debra Lee, Tre...	To blog or not to blog: Student perceptions of blog eff...	2010	Jun 16 2016
↑	85	Carol Ritter, Barbara Polnic...	Classroom learning communities in educational leaders...	2010	Jun 16 2016
↑	85	Marisa Exter, Nichole Harlin...	Story of a conference: Distance education students' e...	2008	Jun 16 2016
↑	85	Steven R. Terrell, Martha M...	The development, validation, and application of the D...	2009	Jun 16 2016
↑	85	Terrie Lynn Thompson, Coll...	Community building, emergent design and expecting t...	2005	Jun 16 2016
↑	85	Alfred P. Rovai	Facilitating online discussions effectively	2007	Jun 16 2016
↑	85	Fengfeng Ke, Kui Xie	Toward deep learning for adult students in online cour...	2009	Jun 16 2016
↑	85	Alfred P. Rovai, Mervyn J. ...	Development of an instrument to measure perceived c...	2009	Jun 16 2016

Figura 3.22 Lista de documentos luego de ser publicados

4.3.5 Salva y restauración de los datos del sistema

En la herramienta Darkaiv el usuario tiene la posibilidad de realizar salvas de los datos, así como su restauración desde archivo. Para ello, el sistema ofrece en la ventana principal, las opciones de “Create Backup” y “Restore From Backup” del menú Tools. El uso de estas funcionalidades es bastante simple ya que para ambos casos solamente es necesario escoger la ubicación donde se realizar la salva o restauración de los datos de la aplicación.

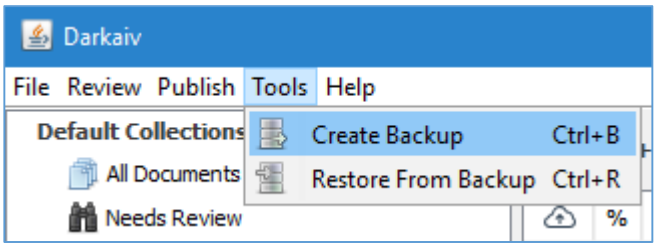

Figura 3.23 Menú Tools de la Ventana Principal desplegado

3.4 Conclusiones del capítulo

En este capítulo se describió la herramienta Darkaiv como producto informático dirigido a la extracción automática de metadatos y publicación de documentos de carácter científico. Mediante la utilización de un lenguaje orientado al usuario fue posible explicar las características de la solución tecnológica implementada, así como sus principales funcionalidades e interacción con el usuario. Finalmente, el capítulo ilustra, por medio de sus imágenes, la consistencia visual de la herramienta en relación con el diseño del gestor bibliográfico Mendeley.

Conclusiones

En este trabajo se arriban a las siguientes conclusiones:

1. La literatura académica identifica los mecanismos de producción semiautomática (automática y manual) de metadatos mucho más conveniente que aquellos donde la generación es totalmente automática. Así mismo reconoce a Dublin Core como uno de los esquemas de metadatos más utilizado en la descripción de recursos de información.

2. De las dimensiones existentes para evaluar la calidad de los metadatos (completitud, exactitud, procedencia, conformidad a las expectativas, coherencia y consistencia lógica, actualidad y accesibilidad), la completitud resulta una de las dimensiones de menor complejidad para implementar en una aplicación.

3. Fue posible constatar que la utilización combinada de bibliotecas para la extracción de metadatos (ej.: Apache Tika y Grobid) permite obtener un proceso de extracción semiautomática de metadatos de mayor calidad.

4. El uso de patrones de diseño en la implementación de la herramienta permitió obtener una solución flexible y reutilizable lo que garantiza que sea aplicable en múltiples contextos.

Recomendaciones

1. Realizar una evaluación de la herramienta tomando en cuenta el criterio de un grupo de especialistas.
2. Incorporar a la herramienta nuevas mecanismos para la extracción de metadatos (ej.: Mendeley), así como nuevas fuentes para su validación (ej.: WorldCat).
3. Implementar nuevas métricas para estimar la calidad de los metadatos producidos por la herramienta.
4. Utilizar la herramienta en la construcción de la biblioteca digital de la Universidad Central “Marta Abreu” de Las Villas.

Referencias Bibliográficas

Alexander, C., Ishikawa, S. & Silverstein, M., 1977. *A Pattern Language: Towns, Buildings, Construction*,

Apache Maven, 2016. Maven – Introduction. Available at: http://maven.apache.org/what-is-maven.html [Accessed June 17, 2016].

Böck, H., 2009. *The Definitive Guide to NetBeans Platform*, Apress.

Booch, G., Rumbaugh, J. & Jacobson, I., 1998. *Unified Modeling Language User Guide* First Edit., Addison Wesley.

Brito Neves, D.A. de & Alburquenque, E.B.C., 2007. Biblioteca digital una convergencia multidisciplinar. *Congreso ISKO-España*, pp.575–580. Available at: http://www.iskoiberico.org/wp-content/uploads/2014/09/575-580_De-Brito-Neves.pdf.

BuenasTareas, 2013. Bibliotecas virtuales y bibliotecas tradicionales, para qué sirven, ventajas, diferencias, etc. - Trabajos finales - Jocelynmagana159. Available at: http://www.buenastareas.com/ensayos/Bibliotecas-Virtuales-y-Bibliotecas-Tradicionales-Para/32156485.html [Accessed June 17, 2016].

Casali, A. & Deco, C., 2013. Asistente para el Depósito de Objetos en Repositorios con Extracción Automática de Metadatos. *... de Tecnologías de la ...*, (SEPTEMBER). Available at: https://www.researchgate.net/profile/Ana_Casali/publication/263926440_Title__An_assistant_for_loading_objects_in_repositories_using_automatic_metadata_extraction/links/0f31753c561f61def3000000.pdf.

Chen, P.P.-S., 1976. The Entity-Relationship Unified View of Data Model-Toward a. *ACM Transactions on Database Systems*, 1(1), pp.9–36. Available at: http://dl.acm.org/citation.cfm?id=320434.320440.

CrossRef, 2016. crossref.org : : info for researchers. Available at: http://www.crossref.org/05researchers/index.html [Accessed June 8, 2016].

CrossRef GitHub, 2016. rest-api-doc/rest_api.md at master · CrossRef/rest-api-doc · GitHub. Available at: https://github.com/CrossRef/rest-api-doc/blob/master/rest_api.md [Accessed June 17, 2016].

Cruz Regalado, D., 2009. Tecnologías de la información y el conocimiento TIC • GestioPolis. Available at: http://www.gestiopolis.com/tecnologias-informacion-conocimiento-tic/ [Accessed June 21, 2016].

Deacon, J., 2009. Model-view-controller (mvc) architecture. *Computer Systems Development*, pp.1–6. Available at: https://techsimplified2.com/Uploads/Agendas/October28,2011.pdf.

Departamento Biblioteca de la Universidad de Cornell, 2003. Metadatos. Available at: https://www.library.cornell.edu/preservation/tutorial-spanish/metadata/table5-1.html [Accessed June 17, 2016].

Dspace, 2015. DSpace 5.x Documentation. *DuraSpace Wiki*, (February), p.795. Available at: https://wiki.duraspace.org/display/DSDOC5x.

DURASPACE, 2016. DSpace | DSpace is a turnkey institutional repository application. Available at: http://www.dspace.org/ [Accessed June 8, 2016].

Eprints, 2016. EPrints Documentation. Available at: http://wiki.eprints.org/w/Main_Page [Accessed June 8, 2016].

FILExt, 2016. FILExt - The File Extension Source. Available at: http://filext.com/ [Accessed June 9, 2016].

Fowler, M. et al., 2003. *Patterns of Enterprise Application Architecture*, Addison Wesley. Available at: http://books.google.de/books?id=FyWZt5DdvFkC.

Fox, E.A., Marchionini, G. & Digitales, B., 2001. Bibliotecas digitales: Greenstone 10. *Association for Computing Machinery. Communications of the ACM*, 44(5), p.30. Available at: http://search.proquest.com/docview/237047266?accountid=14795\nhttp://yh2ww8ft4r.search.serialssolutions.com/directLink?&atitle=Digital+libraries&author=Fox,+Edwa

rd+A;Marchionini,+Gary&issn=00010782&title=Association+for+Computing+Machinery.+Communications.

Gamma, E. et al., 1993. Design Patterns: Abstraction and Reuse of Object-Oriented Design. *Lecture Notes in Computer Science*, 707(Mvc), pp.406–431. Available at: http://citeseer.ist.psu.edu/gamma93design.html.

GNU.ORG, 2016. El sistema operativo GNU. Available at: http://www.gnu.org/philosophy/free-sw.es.html [Accessed June 17, 2016].

Gosling, J. et al., 2005. *The Java® Language Specification - jls8.pdf* Third Edit., Addison-Wesley. Available at: https://docs.oracle.com/javase/specs/jls/se8/jls8.pdf.

Grobid, 2016. Introduction - GROBID Documentation. Available at: https://grobid.readthedocs.io/en/latest/Introduction/ [Accessed June 8, 2016].

H2 Developers, 2016. H2 Database Engine. Available at: http://www.h2database.com/html/main.html [Accessed June 9, 2016].

Hillmann, D. & Bruce, T., 2004. The Continuum of Metadata Quality: Defining, Expressing, Exploiting. Available at: http://ecommons.cornell.edu/bitstream/handle/1813/7895/Bruce_Hillmann_corr_final.doc?sequence=1 [Accessed June 8, 2016].

Hípola, P., Vargas-Quesada, B. & Senso, J.A., 2000. Bibliotecas digitales: situación actual y problemas. *El Profesional de la Informacion*, 9(4), pp.4–13. Available at: http://tinyurl.com/d3d5yrf.

Huespe, A., 2012. TICs en la Biblioteca del siglo XXI: dispositivos móviles y redes sociales. Available at: https://prezi.com/vqt19g81qvu2/tics-en-la-biblioteca-del-siglo-xxi-dispositivos-moviles-y-redes-sociales/ [Accessed June 21, 2016].

JavaLite, 2016. JavaLite Documentation. Available at: http://javalite.io/activejdbc [Accessed June 9, 2016].

Marcum, D., 2005. La biblioteca digital: requisitos. *Boletín de la Asociación Andaluza de Bibliotecarios*, 20(79), pp.57–68. Available at:

http://dialnet.unirioja.es/servlet/articulo?codigo=1846637\nfile:///C:/Users/Martta mlb/Downloads/Dialnet-LaBibliotecaDigital-1846637 (1).pdf\nhttp://dialnet.unirioja.es/descarga/articulo/1846637.pdf.

Mattmann, C. a & Zitting, J.L., 2011. *Tika In Action* C. Kane, ed., NY: Manning Publications Co.

Medrano, J.F., Figuerola, C.G. & Alonso, J.L., 2012. Repositorios Digitales en España y calidad de Metadatos. *Scire: representación y organización del conocimiento*, 18(2), pp.109–121. Available at: http://www.ibersid.eu/ojs/index.php/scire/article/view/3977 [Accessed June 8, 2016].

Méndez, E. & Senso, J.A., 2004. 7. Uso del Dublín Core (DCMI). ISO 15836-2003. Available at: http://www.sedic.es/autoformacion/metadatos/tema7.htm [Accessed June 8, 2016].

Microsoft Corporation, 2009. Application Architecture Guide.

Miranda, A., 2010. DOCUMENTALISTA HOY: Bibliotecas virtuales vs tradicionales. Available at: http://documentalistahoy.blogspot.com/2010/08/bibliotecas-virtuales-vs-tradicionales.html [Accessed June 17, 2016].

Ochoa Gutiérrez, J., 2012. Biblioteca y TIC: medios de información y comunicación para la formación de ciudadanía crítica. *IFLA*, pp.1–10.

Ochoa, X., 2008. *Learnometrics : Metrics for Learning Objects*, Available at: https://repository.libis.kuleuven.be/dspace/bitstream/1979/1891/2/ThesisFinal.pdf.

Pinilla, A., Gutiérrez, M. & Ballejos, L., 2014. Extracción Automática de Metadatos a partir de Objetos de Aprendizaje en un Repositorio Institucional : Estado del Arte. *Simposio Argentino de Tecnología y Sociedad*, pp.67–82.

Pressman, R., 2011. *Ingeniería del Software*,

REBELLABS, 2014. JAVA TOOLS AND TECHNOLOGIES LANDSCAPE FOR 2014. In pp. 1–25.

Sanjo, J., 2007. Adoption of Open Source Digital Library Software Packages A Survey. *Convention on Automation of LIBraries in Education and Research Institutions*. Available at: http://arizona.openrepository.com/arizona/handle/10150/105732 [Accessed June 17, 2016].

Senso, J.A. & Rosa Piñero, A. de la, 2003. El concepto de metadato: algo más que descripción de recursos electrónicos. *Ciência da Informação*, 32(2), pp.95–106.

Smith, J. & Schirling, P., 2006. Metadata standards roundup. *IEEE MultiMedia*. Available at: http://www.computer.org/csdl/mags/mu/2006/02/u2084-abs.html [Accessed June 8, 2016].

Sommerville, I., 2013. Ingeniería del software. *Journal of Chemical Information and Modeling*, 53(9), pp.1689–1699.

Stamou, G. et al., 2006. Multimedia annotations on the semantic Web. *IEEE Multimedia*, 13(1), pp.86–90. Available at: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1580438 [Accessed June 8, 2016].

Tramullas, J., 2004. Bibliotecas digitales: una revisión de conceptos y técnicas. *Bibliodocencia*, 1(2), pp.26–31. Available at: http://eprints.rclis.org/19301/.

Tramullas Saz, J., 2002. Propuestas de concepto y definición de la biblioteca digital. *III Jornadas de Bibliotecas Digitales : (JBIDI'02) : El Escorial (Madrid) 18-19 de Noviembre de 2002*, (March), pp.11–20. Available at: http://dialnet.unirioja.es/servlet/articulo?codigo=957732&info=resumen&idioma=SPA\nhttp://infonautica.net/docs/jbidi/jbidi2002/04_2002.pdf.

tutorialspoint.com, 2016. TIKA - Overview. Available at: http://www.tutorialspoint.com/tika/tika_overview.htm [Accessed June 8, 2016].

Universidad Autónoma Metropolitana, 2016. Definición y Objetivo de la Biblioteca Digital UAM. Available at: http://www.bidi.uam.mx/index.php?option=com_content&view=article&id=56:defini

cion-y-objetivo-de-la-biblioteca-digital-uam&catid=37:la-biblioteca-digital-uam&Itemid=37 [Accessed June 21, 2016].

WorldCat Search API, 2016. WorldCat Search API | ProgrammableWeb. Available at: http://www.programmableweb.com/api/worldcat-search [Accessed June 17, 2016].

WorldCat.org, 2016. What is WorldCat? [WorldCat.org]. Available at: https://www.worldcat.org/whatis/default.jsp [Accessed June 8, 2016].